IDA PAPER P-3074

THE ARIES PROJECT FOR RULE-BASED-DESIGN KNOWLEDGE ACQUISITION

Karen J. Richter, *Project Leader*
Ken Blemel
Carl Grewe
Ali Hashmi
Dennis Hoffman
Ray Johnston
Doug Ott
Carl Rust

June 1995

Prepared for
Director, Test, Systems Engineering and Evaluation

DTIC QUALITY INSPECTED 5

INSTITUTE FOR DEFENSE ANALYSES
1801 N. Beauregard Street, Alexandria, Virginia 22311-1772

DEFINITIONS

IDA publishes the following documents to report the results of its work.

Reports

Reports are the most authoritative and most carefully considered products IDA publishes. They normally embody results of major projects which (a) have a direct bearing on decisions affecting major programs, (b) address issues of significant concern to the Executive Branch, the Congress and/or the public, or (c) address issues that have significant economic implications. IDA Reports are reviewed by outside panels of experts to ensure their high quality and relevance to the problems studied, and they are released by the President of IDA.

Group Reports

Group Reports record the findings and results of IDA established working groups and panels composed of senior individuals addressing major issues which otherwise would be the subject of an IDA Report. IDA Group Reports are reviewed by the senior individuals responsible for the project and others as selected by IDA to ensure their high quality and relevance to the problems studied, and are released by the President of IDA.

Papers

Papers, also authoritative and carefully considered products of IDA, address studies that are narrower in scope than those covered in Reports. IDA Papers are reviewed to ensure that they meet the high standards expected of refereed papers in professional journals or formal Agency reports.

Documents

IDA Documents are used for the convenience of the sponsors or the analysts (a) to record substantive work done in quick reaction studies, (b) to record the proceedings of conferences and meetings, (c) to make available preliminary and tentative results of analyses, (d) to record data developed in the course of an investigation, or (e) to forward information that is essentially unanalyzed and unevaluated. The review of IDA Documents is suited to their content and intended use.

The work reported in this document was conducted under contract DASW01 94 C 0054 for the Department of Defense. The publication of this IDA document does not indicate endorsement by the Department of Defense, nor should the contents be construed as reflecting the official position of that Agency.

IDA PAPER P-3074

THE ARIES PROJECT FOR RULE-BASED-DESIGN KNOWLEDGE ACQUISITION

Karen J. Richter, *Project Leader*
Ken Blemel
Carl Grewe
Ali Hashmi
Dennis Hoffman
Ray Johnston
Doug Ott
Carl Rust

June 1995

INSTITUTE FOR DEFENSE ANALYSES
Contract DASW01 94 C 0054
Task T-B6-1216

PREFACE

This paper was prepared by the Institute for Defense Analyses (IDA) for the Office of the Director of Test, Systems Engineering, and Evaluation under the cognizance of Headquarters, Army Materiel Command, Research and Development. The work was performed under task *Feasibility Study for a Rule-Based Concurrent Engineering—Computer-Aided Design Practice.*

The objective of the task reported on in this paper is to determine the feasibility of defining the data capture techniques and capturing rule-based data in a way that is useful to the CAD/CAE industry, thus enabling the development of interactive or background rule-based and expert systems useful to defense, commercial, and dual-use design teams.

This paper was reviewed by Dr. Michael Pecht of the University of Maryland Computer-Aided Life Cycle Engineering (CALCE) Electronic Packaging Center (EPRC); Ms. Barbara Bicksler of IDA; Mr. Matt Tracy, Air Force Armstrong Laboratory Human Resources Division; Mr. Sid Markowitz, Army Armament Research, Development, and Engineering Center (ARDEC), Quality Assurance; and Mr. Ed Smith, Office of the Assistant Secretary of the Navy (Research, Development and Acquisition), Product Integrity.

ACKNOWLEDGMENTS

The authors wish to thank the team members' companies for donating their employee's time to the project, the Reliability and Maintainability Symposium (RAMS) Board of Directors for their support, and the Tri-Service Process Action Team for making this project possible.

The authors also wish to thank Leta Horine, IDA, for all her help with the meetings and in preparation of this paper.

CONTENTS

APPENDICES

FIGURES

TABLES

EXECUTIVE SUMMARY

In the current environment of austere defense budgets and a fiercely competitive global marketplace, U.S. defense and commercial companies must exploit the best practices and technologies available to improve quality and avoid costly design changes and lengthy development cycles. Concurrent Engineering (CE) and Integrated Product and Process Development (IPPD), which fosters an integrated, collaborative team approach to product development, is being used to reduce, and ultimately eliminate, the number of design iterations and engineering change orders within the product development cycle. Computer-aided design (CAD) and computer-aided engineering (CAE) tools have contributed greatly to speeding up the design evaluation process. However, as industry moves toward CE and as design information becomes available in real time, design decisions are moving toward requiring immediate consideration of a complex myriad of design rules.

To enable such consideration, an interactive, integrated, rule-based system capable of producing optimum or near optimum design recommendations is desirable. Today, each company has to develop its own design rules and practices to be entered into the rule-based shell supplied by the CAD/CAE vendors or developed by the company themselves. A generic solution would require industry- and governmentwide consensus across company-specific boundaries to identify and collate the design rules and practices used to develop rule-based technologies. Such a task would involve defining, summarizing, and providing industrywide design rules and best design practices in correlation with design questions and potential answers that design teams might ask to optimize the design process. The interdependencies, applicability, rationale, and weightings for these rules are critical design parameters that would also be key to the formation of a rule-based design tool. The objective of the task reported on in this paper is to determine the feasibility of such a generic solution.

A. TASK SPONSOR

This task was sponsored by the Office of the Director of Test, Systems Engineering, and Evaluation. Project oversight was performed by the Concurrent

Engineering-Computer-Aided Design (CE-CAD) DoD Tri-Service Process Action Team (PAT).

The Institute for Defense Analyses (IDA) was approached to do this task because of previous work it had done in the areas of CAD and CAE for reliability, maintainability, and supportability. In fact, in one previous task, IDA had attempted to do independent research to determine a generic set of rules for Design for Maintainability, but consensus was never reached among industry management and the Services. The present task documented here truly was a feasibility study that sought to answer the question, Can such a consensus-driven goal of developing a rule base be achieved with a cross-industry, multidiscipline, voluntary team?

B. SCOPE

To maintain project focus and manageability, this feasibility project concentrated on PWB assembly design, circuit implementation, and early detailed design, i.e., the capture of the parts list and schematic. The project focused on "good" (consensus among specialty engineers and designers) analog and digital designs using through-hole technology and discrete devices. The intent was to stay with the functional nature of the design and not get into fabrication and assembly of the board. Through-hole technology was selected because of the need to keep the problem simple to demonstrate the feasibility. Moreover, through-hole technology is a stable technology that involves very little industry proprietary data.

The initial phase of the project focused on developing a rule-based-design capture approach using knowledge acquisition techniques. Subsequent phases included soliciting and capturing nonproprietary corporate design rules and design practices to prove out the techniques and then demonstrating a rule-based feasibility prototype.

The project was conducted by a team of volunteer circuit design specialists who came from several industries and disciplines. Team participants had to be supported in this effort by their companies (through commitment of their time and release of the company's design rules and practices). Companies and team members were motivated by the chance to influence the rule base development and receive the combined rule data base for their own use. IDA's task was to concentrate on the process to show feasibility. Team members spent their own time developing the data base.

To succeed, this team of people from different design and engineering disciplines would have to identify, capture, and concur in a consistent, thorough set of design rules and be able to document that process so that it could be repeated. Success would be demonstrated by implementing the rule set in a software demonstration vehicle—a knowledge-based system.

C. RESULTS

The team members identified and captured a large number of design rules and design practices for PWB assembly design. Some examples of the rules are included in Appendix D to show feasibility.

The team demonstrated implementation of the design rules and practices at the Reliability and Maintainability Symposium (RAMS) on 19 January 1995 at the Washington Hilton and Towers, Washington, D.C. The project members had dealt with many CAD/CAE vendors, but they chose to demonstrate using the *Pre*Amp Rules Definition Facility, which is based upon the knowledge-based software Kappa.

Thus, the objective of this feasibility effort—to develop the design rule capture technique, collect sample data to prove out the technique, and demonstrate a feasibility prototype—has been met. The process is documented in this paper.

D. SUMMARY

IDA's task was to examine the feasibility of using a voluntary team to capture, encapsulate, and reach consensus on an industry-based set of practices and rules that could be implemented in a rule-based system. The task demonstrated that this is possible and that there is industry support for these kinds of activities and tools. We found that most "good commercial practices" companies have many practices and rules in common. Their need to share these practices with their subcontractors for improved quality products and reduced cycle times continues to grow, so there appears to be a reason to expand and continue these efforts in the future.

The CAE work being done by suppliers such as Cadence, ICAD, and *Pre*Amp demonstrates that tools that support rule-based, correct-by-design development are not only possible, but also marketable in industries. Feedback from early users of these tools indicates that such tools need not only to provide an initial capability, but also to allow the users to expand the data base on their own.

This task has shown that cross-industry, multidiscipline teams can help to consolidate design practices and rules for CAE vendors and promote the use of those techniques in industries. A voluntary group can accomplish an end goal when composed of interested individuals who have the full support of their home organizations, proper sponsorship, and effective project management. The team members learned, made contacts, and obtained information that will be of benefit for a long time after the IDA task is complete.

I. INTRODUCTION

A. TASK RATIONALE AND OBJECTIVES

In the new environment of austere defense budgets and a fiercely competitive global marketplace, U.S. defense and commercial companies must exploit the best practices and technologies available to improve quality, and avoid costly design changes and lengthy development cycles. Concurrent Engineering (CE) and Integrated Product and Process Development (IPPD) which foster an integrated, collaborative team approach to product development, are being used to reduce, and ultimately eliminate, the number of design iterations and engineering change orders within the product development cycle. Computer-aided design (CAD) and computer-aided engineering (CAE) tools have contributed greatly to speeding up the design process. However, as industry moves toward CE and as design information becomes available in real time, design decisions are moving toward requiring immediate consideration of a complex myriad of design rules.

To enable such consideration, an interactive, integrated, rule-based system capable of producing optimum or near optimum design recommendations is desirable. Today, each company has to develop its own design rules and practices to be entered into the rule-based shell supplied by the CAD/CAE vendors or developed by the company themselves. A generic solution would require industry- and governmentwide consensus across company boundaries to identify and collate the design rules and practices used to develop rule-based technologies. Such a task would involve defining, summarizing, and providing industrywide design rules and best design practices in correlation with design questions and potential answers that design teams might ask to optimize the design process. The interdependencies, applicability, rationale, and weightings for these rules are critical design parameters that would also be key to the formation of a rule-based design tool.

The objective of the task documented in this paper is to determine the feasibility of defining the techniques for capturing rules data and the design rules and practices in a way that is useful to the CAD/CAE industry. If feasible, these steps would lay the foundation for developing interactive or background rule-based and expert systems useful to defense, commercial, and dual-use design teams. The sample data from this project would prove the feasibility of building a generic rule base.

The long-term goal—beyond this project—is for industry to undertake efforts in design areas beyond our scope. Ultimately, it is hoped that industry will have customizable rule-based design tools, not just a shell, where the rules lead the design engineer through the process. Accomplishing this objective will form the basis for future commercially available rule-based-design CAE tools that reduce the design team's information and constraint overload, reduce design cycle time, store generic and company specific knowledge, and improve product performance and quality.

B. TASK SPONSOR AND SCOPE

This task was sponsored by the Office of Industrial Engineering and Quality under the Assistant Secretary of Defense for Economic Security. Project oversight was performed by a Concurrent Engineering-Computer-Aided Design (CE-CAD) DoD Tri-Service Process Action Team (PAT) and by the Reliability and Maintainability Symposium (RAMS) Board of Directors (BoD).

Numerous professional societies sponsor the annual RAMS. The RAMS, realizing their unique relationship to these professional societies, proposed establishing a Council for Reliability, Quality, and Competitiveness to sponsor potential projects (i.e., front-end analysis of something someone else will fund and oversee) that could be of common benefit to all the societies. The Council recommended to the RAMS BoD that the first project focus on developing information needed to produce an interactive, integrated, rule-based system. This task developed out of that recommendation to determine the feasibility of developing such a system.

To give the project definition and to help attract attention and participation, the project was named the *Aries Project.*[1] For this project, design issues were divided into two categories:

- Design *practices*—good, standard practices used in the design process; design guides not always easily measured by machine; basis for rules.

[1] Various acronyms were tried, but this name was chosen because of the project's association with the RAMS. The ram is the symbol of the astrological sign of Aries.

- Design *rules*—those that every design process needs to satisfy, without which it will not give correct results; measurable and true/false conditions that establish that the design meets its constraints.

IDA was approached to do this task because of previous work it had done in the areas of CAD and CAE for reliability, maintainability, and supportability. In fact, in one previous task, IDA had attempted to do independent research to determine a generic set of rules for Design for Maintainability, but consensus was never reached among industry management and the Services. The present task documented here truly was a feasibility study that sought to answer the question, Can such a consensus-driven goal of developing a rule base be achieved with a cross-industry, multidiscipline voluntary team?

C. PROJECT PARTICIPANTS

1. Requirements

Involvement in the Aries Project was initially projected at three levels: Initial Team, Core Team, and Participating Companies. Difficulties encountered in implementing this multitiered approach are discussed in later chapters. However, the original team concept is delineated here for completeness.

An Initial Team was formed to:

- Refine the approach and schedule.
- Define Core Team member backgrounds and expertise needed.
- Identify and target Core Team members or companies to obtain expertise.
- Solicit CAE/CAD industry involvement in establishing need based on market analysis.
- Develop an industry benefit statement.
- Generate an executive overview of project objectives and benefits for targeted mailing for direct participation.
- Identify knowledge acquisition expert and software systems engineer.
- Finalize project ground rules.
- Expand to Core Team.

The Core Team was to comprise experts from several domains: electrical design, specialty engineering, knowledge acquisition, technical and business aspects of electronic CAE/CAD, and software systems engineering. The Core Team was to define the knowledge acquisition approach and collection format; identify and target the participating companies from which the rules, practices, questions, and answers would be solicited; develop a formatted example (test case) to prove out the technique and approach prior to soliciting; solicit Participating Companies for champions and data submittal; review company-submitted information for applicability, follow-up and acceptance; document the obtained sample information into a final report; capture the sample rule base (with associated information) electronically; and develop a generic knowledge acquisition technique for future rule capture projects for other design areas; and devise training material.

The Participating Companies were not to be members of this project or of the Core Team. They were critical to the success of this project, however, because they would provide the majority of the data collected by this project. The knowledge of their experts and their design rules and practices would be obtained and understood by the Core Team through a heavily involved champion from each company. The Participating Companies were to:

- Identify their champions.
- Provide rule-based data as required (rule/rationale/applicability/weighting).
- Transfer expert knowledge to Core Team so data is understood.

It was also important that Participating Companies provided rules that encompassed the perspectives of the total concurrent engineering team: electrical design, parts selection, reliability, maintainability, testability, safety, producibility, supportability, design-to-cost, and so forth.

Participating Companies were to have an impact on the generic rules data base that would be the basis for future commercially available rule-based-design CAE tools to interactively assist the electrical design engineer early in the design process. By minimizing the costly process of having each company establish and capture its own rule data, Participating Companies facilitate and expedite the availability of turn-key knowledge-based tools to a wide range of companies. Through their participation in the Aries Project, the companies would have not only the opportunity to establish contacts with other company representatives, but also early access to the results of the project for their company's internal use.

Appendix A lists the individuals and affiliations of Aries Project participants.

2. Benefits of Project Participation

To participate in the Aries Project, companies had to contribute both their employee's time and their design rules. In return, they would realize four major benefits.

First, the per unit cost of the common system should be considerably less than the cost to develop a company-unique system. If a company develops its own unique rule-based system, it incurs the total cost of developing that system. But, if a system is developed by CAD/CAE vendors based on an industrywide consensus on that system's requirements, then the company incurs only the costs associated with its proprietary design rules and design practices and not the cost of maintaining the capture product. With the latter approach, the CAD/CAE vendors' costs to incorporate these rules into a CAD environment could be amortized across a potentially large market. Companies without the resources to develop their own system may be able to afford to purchase a system from the CAD vendors.

Second, the common approach has the potential to produce a better product. A common architecture of the rule system allows teams (especially large companies with small subcontractors) to share common rules and implement them easily.[2] The information required to develop the rule-based system is technology driven. As technology changes, the information base must be periodically updated to remain current. Experts from across industry should be able to provide a more comprehensive knowledge base than could be provided by a single company.

Third, corrections of defects in the resulting software as well as the maintenance of the software would be the responsibility of the software CAD developers. Individual companies would only be responsible for quality of the data and extension of the rules.

Finally, participating in this project entitles a company to contribute to the rule base and to have early access to the combined data. The long-term benefit is the availability of rule-based design tools as discussed above. A benefit to both industry and government is the potential to significantly shorten the acquisition process by providing the design team with the capability to "design it right the first time."

[2] The need for a common architecture is called out in a National Science Foundation Workshop report by the Institute for Systems Research at the University of Maryland, *Systems Challenges for the Next Decade*, Technical Report TR 95-38, 8 December 1994.

Note that this project was formulated before Secretary of Defense William Perry issued a memorandum on 29 June 1994, eliminating the requirements for military-specific specifications and standards and stating a requirement for the use of commercial standards instead. When we began this task, we thought the resulting system would be propagated with rules from the military specifications and standards, but this was not the case.

An additional benefit to industry with this tool set would be the educational value available to colleges and universities. Students attending these educational institutions will be trained with these design guidance tools. The impact of this training is the development of engineers who are productive to industry much sooner after graduation.

D. JUSTIFICATION FOR RULE-BASED DESIGN TOOLS

The need for rule-based design tools stems from the difficulties encountered when eliminating the four major obstacles to "design it right the first time":

- Incomplete specification of requirements early in the design process
- Lack of integrated toolsets and shared data bases
- Conflicting goals among engineering specialties
- Design information overload

Incomplete Requirements. Initially, the design requirements are based upon preliminary information. As the design matures, increased information becomes available to allow more accurate requirements. Unfortunately, numerous requirement changes generally preclude succeeding with the first design. If the requirements change and the system has to be redesigned, schedules slip, and costs rise. The institutionalization of concurrent engineering, which fosters a unified, collaborative, team approach to satisfying customer requirements up-front, should reduce changes to design requirements.

Lack of Integrated Toolsets and Shared Data Bases. Traditionally, designers create their initial design based upon their internal list of design rules. The initial design is then passed to the engineering specialty functions, where each specialty has its own rules and analyses to evaluate the design. This evaluation can take anywhere from several days to a few weeks. During this delay, designers are continuously refining and improving the initial design. By the time the engineering specialties come back with their recommended design changes, they may have only limited influence on the current design. CAD/CAE analysis tools have contributed to speeding up the evaluation process. Still, the specialty engineers must first study the design and search data bases, often having to translate data and manually enter the data for their specific CAD/CAE analysis tools. The lack of integrated toolsets and shared data bases diminishes the engineering specialties' ability to provide information and analyses in time to influence the design.

A solution to this problem was first addressed about 10 years ago when the U.S. Army and Air Force started a joint program called RAMCAD (Reliability, Availability, and Maintainability in a Computer-Aided-Design environment). The goal of RAMCAD was to directly influence design decisions on a real-time basis. What RAMCAD accomplished was to link current design information and electronic data bases containing component reliability, maintainability, and supportability information automatically with commercially available CAD/CAE analysis tools. As the designer creates the design at the workstation, RAMCAD automatically pulls off needed CAD information, accesses information from other data bases, and then provides this information to the commercially available CAE tools. These tools then rapidly perform their analysis and provide information back to the designer, allowing the designer to rapidly compare competing alternative designs, including a RAM viewpoint. In this way, RAM directly influences the design process. The Institute for Defense Analyses provided technical support to the Army and the Air Force on this RAMCAD project.

Conflicting Goals Among Engineering Specialties. The availability of real-time CAD/CAE information has only partially resolved the problem. Often, analysis tools of one specialty provide discrete and independent information about the design that can potentially conflict with other CAE analyses. For example, there is currently a CAD tool that, given the components to be mounted on a printed wiring board (PWB) and the planned pin connects, will determine a layout of the components to minimize the cumulative distances between pin connects. However, this CAD tool does not consider the heat generating capacity of the various components on the PWB. Grouping high heat generating components together can create a hot spot. One design option is to move the high heat generating components apart. However, if this option is chosen, the minimum cumulative pin connect distances no longer exist. Where should these components be moved so that an optimum layout exists to satisfy both constraints? This is a complex problem with more than one computational solution. Designers need not only real-time information but also the ability to perform an interactive expert analysis based on a dynamic set of design constraints.

Design Information Overload. As discussed above, designers have their internal list of design rules and the specialty engineers in turn have their rules. As industry moves toward concurrent engineering and as design information becomes available in real time, design decisions require immediate consideration of all these rules, including those traditionally the responsibility of the specialty engineers. While the specialty engineers provide analyses as part of the design team, the burden on the designer increases significantly. Without a method to assist the designer in handling increased decisions, the potential to inadvertently overlook essential rules exists. Design teams need a system capable not only of managing a multitude of rules, but also of providing suggested design solutions. Such systems are rule-based or knowledge-based "expert" systems. Obtaining the data base to fill these expert system shells is the focus of this project.

E. SUCCESS CRITERIA

The objective of this feasibility project was for a team of people from different design and engineering disciplines to identify, capture, and reach consensus on a consistent, thorough set of design rules, and to be able to document that process so that it could be repeated. Success would be demonstrated by implementing the rule set in a software demonstration vehicle—a knowledge-based system.

II. PROJECT DESCRIPTION

This chapter describes the process that the project team followed once the project scope was defined. The steps in this process are those that would be repeated for future efforts in other design areas beyond the scope of this project.

A. SCOPE

To focus this feasibility study and give it a manageable scope, the project concentrated on printed wiring board (PWB) assembly circuit design, circuit implementation, and early detailed design, i.e., the capture of the parts list and schematic. The project focused on "good" (consensus among specialty engineers and designers) for analog and digital design, through-hole, discrete devices (packaged, purchased). The intent was to stay with the functional nature of the design and not get into fabrication and assembly of the board. Through-hole technology was selected because it is a stable technology that involves very little industry proprietary data and would therefore be amenable to use in a demonstration of feasibility.

The initial phase of the Aries Project focused on developing an approach for capturing design rule data using knowledge acquisition techniques. Subsequent phases focused on soliciting and capturing nonproprietary corporate design rules and design practices to prove out the techniques and then demonstrating a rule-based feasibility prototype.

For this project, design practices were defined as common electronic design functions that do not lend themselves to measurement. Design rules were defined as those items that can be measured.

B. PHASE I—DEVELOP CAPTURE APPROACH

To kick off the initial phase of the project and acquaint the team with knowledge acquisition techniques, a knowledge acquisition engineer from Texas Instruments, Bruce Hubanks, attended one of the early meetings. His briefing is included in Appendix B. He explained that domain experts internalize so much of their knowledge that it is difficult for them to explain and delineate the rules. He advised the team to put an initial set of rules into the tool and then let designers start working with them. In that way, we could discover what the rules did that the contributors failed to say they did, and note those rules that were not being followed.

Mr. Hubanks discussed the complexity of multiple domains. He emphasized the need to get conflicting design rules out of the system by doing one domain at a time. He also stressed the importance of the user interface and noted that adding rules suggested by designers within each company would require a configuration control board.

This project refined generic knowledge acquisition methods to focus on the capture of design rules and practices. For each rule, the team defined various data elements such as rule interdependencies, rule interrelationships, applicability, rationale, and weighting. These data element types were defined during phase I, and the approach was documented so that it could be applied in the future capture of specific rules for those segments and functions of the design process that were beyond the scope of this project.

The team members charged with this task defined the design capture process and data structure and mapped the design process flow for digital and analog PWB assembly design. They formulated appropriate questions to be asked during each process step and developed categories for the design rules and practices. It was important to have this framework in place before proceeding with the data collection.

1. Design Capture Process

The team brainstormed the processes that engineers follow when initiating a new design or a redesign. The result of this session was the Design Capture Process shown in Figure II-1, which addresses the up-front design process for PWB assemblies. They then identified the components to be used as the basis for developing the Data Structure Concept and Design Process Flow.

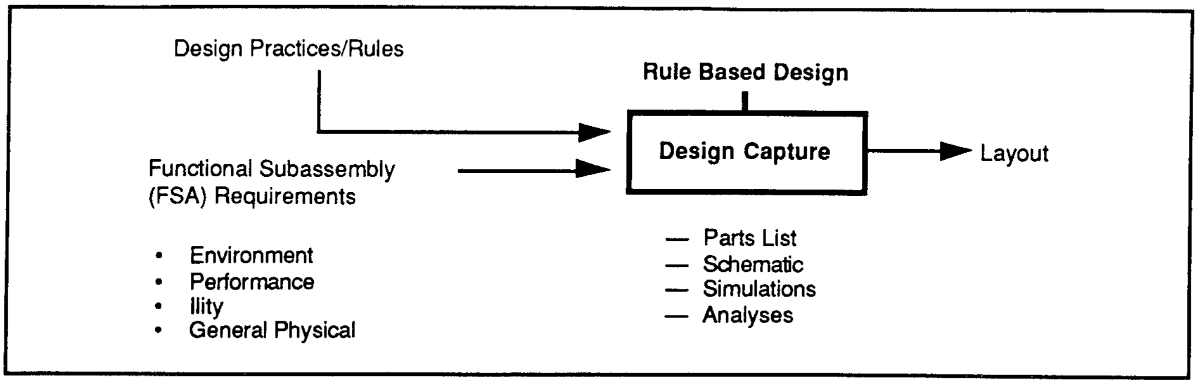


Figure II-1. Design Capture Process for PWB Assemblies

2. Data Structure Process

The team developed the Data Structure Concept (Figure II-2) to subdivide the Design Capture Process into manageable subfunctions. For PWB assemblies, the subfunctions are Requirements Sufficiency, Implementation Planning, Schematic Capture/Simulation/Analyses, and Design Documentation. With these subfunctions of the design process, the team had the foundation on which to build a knowledge base. This knowledge base is queried as the designer proceeds through the up-front design process.

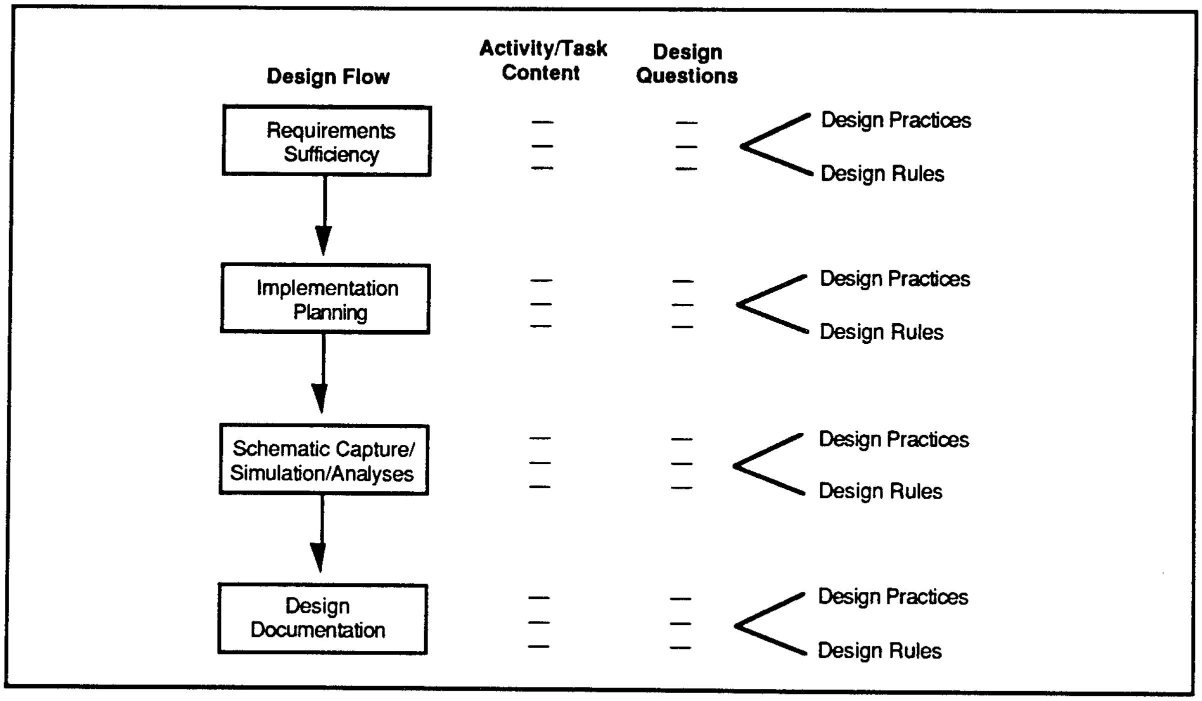


Figure II-2. Data Structure Concept for PWB Assemblies

3. Design Process Flow

Once the Data Structure Concept was determined, the team identified the type of activities a designer would pursue in each subfunction for PWB assemblies (Figure II-3). They then developed a number of questions that addressed these various activities (Table II-1). The questions and associated activities served as the determining factor in organizing the design rules and design practices.

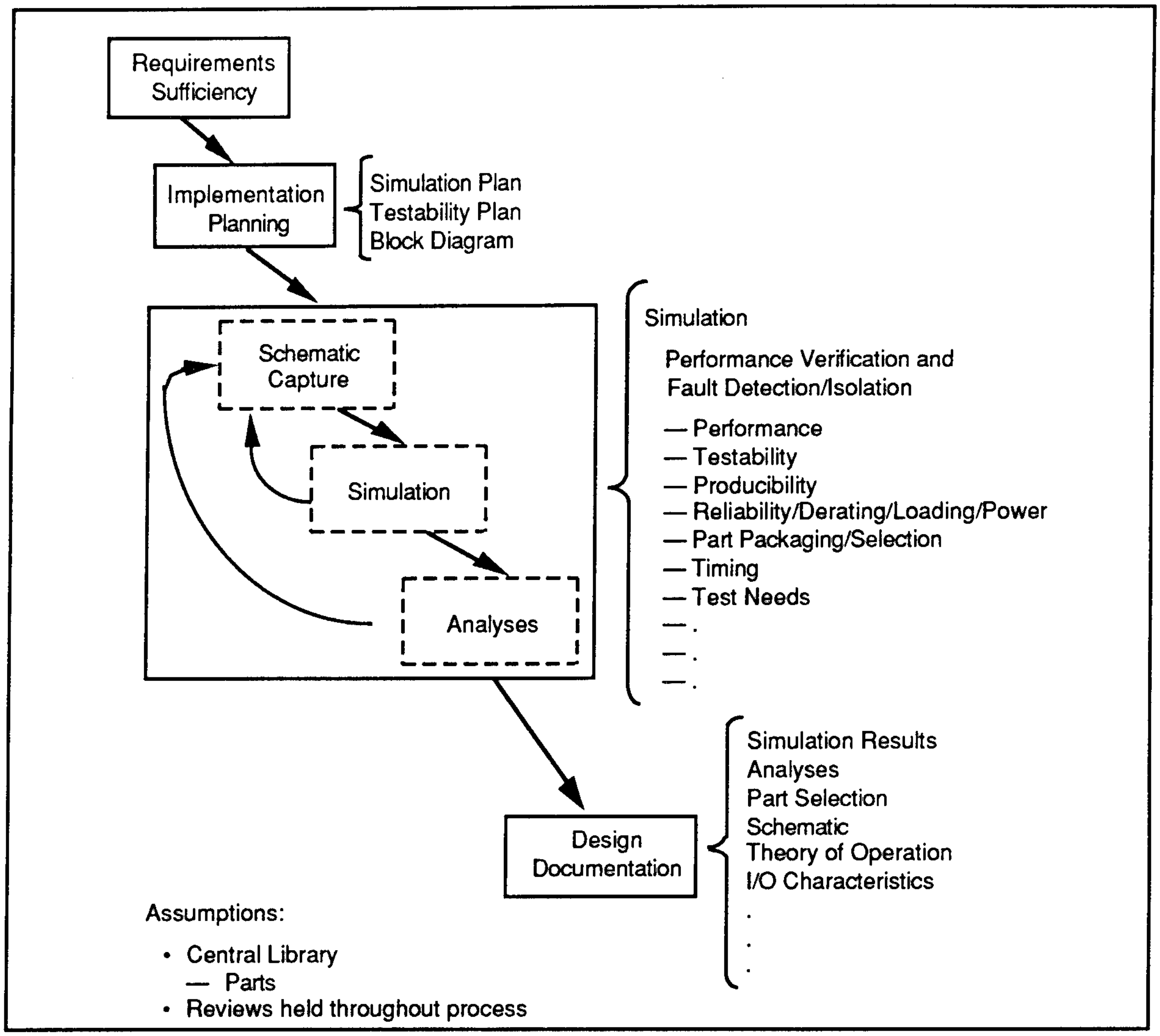


Figure II-3. PWB Assemblies Design Process Flow, Digital and Analog

Table II-1. Questions Regarding Subfunction Activities for PWB Assemblies

Subfunction	Relevant Questions
Requirements Sufficiency	Do you know your "total" requirements? - Specification Requirements? - Internal Constraints? Are there unknowns or ambiguities? Is the technology approach feasible?
Implementation Planning	How are you going to verify your requirements by simulation? By analysis? By test? How are you going to generate your tests for design verification (manufacturing and acceptance operations)? Have you developed a block diagram of the functional requirement? What risk areas have you identified? Have you developed an approach to requirements traceability?
Schematic Capture/ Simulation/Analysis/Part Packaging/Selection	Are needed parts in library? Does schematic conform to drawing requirements? Has schematic been checked for drawing defects? Do simulation and analysis results verify requirements? Were tests simulated or analyzed to verify that design meets testability requirements and test needs? Have you selected the best technology implementation approach for your design? Have you resolved all risk items identified? Have you developed a parts list that meets part selection requirements?
Design Documentation	Have you written theory of operations? Have you captured I/O characteristics? Have you finalized (updated) all design documentation (block diagram, schematic parts list, logic files, simulation and analyses results, test patterns) to reflect version of design you are releasing? Is everything under at least Engineering Configuration Management control? Have you verified requirements traceability?

4. Categorization of Rules and Practices

With the framework completed, the identification of categories for design rules and design practices became the next step in the process. It was determined that design rules would be categorized by Material, Analog, Digital, Global, and Part Selection, while design practices would be separated into Analog, Digital, and Global categories. The team decided that additional factors, such as rationale, interdependence, weight, interrelationship, applicability, and conditions, would be considered when determining rule precedence (Figure II-4).

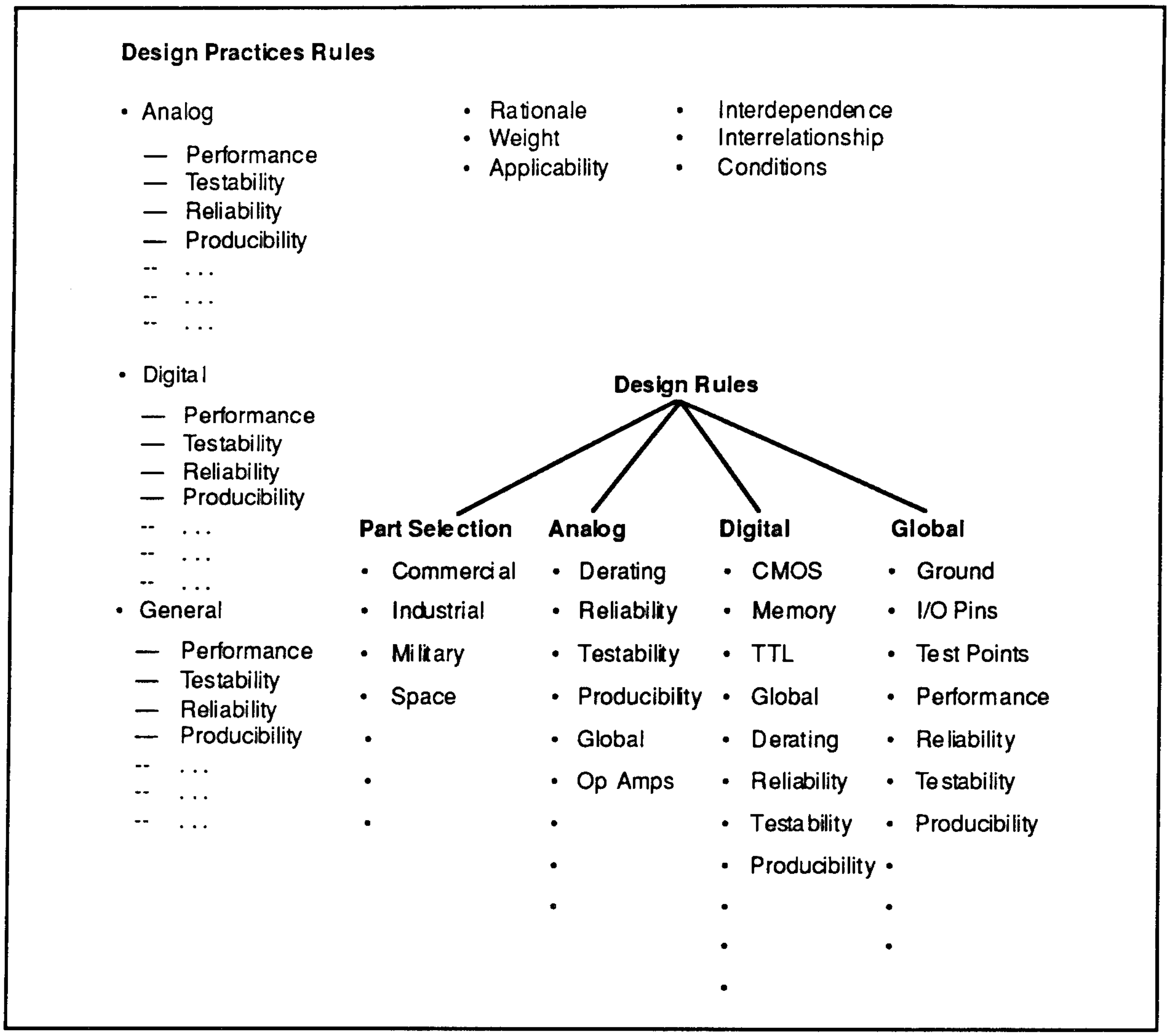


Figure II-4. Design Practices and Rules for PWB Assemblies

C. PHASE II—COLLECT DATA

Phase II of the Aries Project mission was to identify and capture industrywide nonproprietary design rules and best design practices. The identification of these design rules and design practices became the focus of the Aries Project team once the structure of the process was determined. The team reviewed a number of nonproprietary design rule documents and design guides and identified those rules and practices that met the criteria set forth in the structure definitions.

Rule data were collected to populate a sample data base for a knowledge-based tool that would be used to show the feasibility of the longer term vision. As envisioned, embedded rule-based-design tools available from CAD/CAE vendors would be integrated into workstation toolsets, populated with a set of design rules, and capable of providing design decision guidance as the design is developing. For this project, the data captured were from design rules and best design practices associated with electrical PWB assembly design capture—design analyses, board layout impacts, and leaded component placement. Relevant design questions and answers, developed with the design process flow in Phase I, were correlated with specific rules. Interdependencies, interrelationships, applicability, rationale, and weightings for the sample rules were also collected.

Data sources came from Honeywell, NASA, EDS, Texas Instruments, the University of Maryland, Rome Laboratories, Military Standards, the Navy NAVAIR RFP, and Digital and Analog circuit design guide manuals.[3] The Core Team reviewed these data sources, which provided a range of design rules and design practices from which to select those that address the early detailed design level for inclusion in the Aries Project knowledge base.

D. PHASE III—DEMONSTRATE FEASIBILITY PROTOTYPE

Identifying design rules and design practices applicable to the early detailed design level proved to be only a small portion of the project task. Capture of this information in the knowledge base became a long and arduous task. Several capture approaches were discussed. Since funds available for software acquisition were limited, the initial approach

[3] Following are some illustrative titles of documents from which rules were taken: *Circuit Card Assembly (CCA) Design for Testability Rules and Guidelines; CFS/MO Component Derating Standard; Reliability Preferred Practices for Design and Test, Product Design, Manufacturing, and Life Cycle Rules; Digital Design Guide; Analog Circuit Design Guide; Design for Producibility Guide System Interconnect*, MIL-STD-454L.

focused on shareware applications. The team determined, however, that any shareware application would only be a nucleus for the capture process and would require additional software development to achieve a usable product. Since the project timeline didn't allow for such software development, alternatives were sought.

One alternative was to use the *Pre*Amp software.[4] At the April 1994 project meeting, Greg Smith of Boeing Defense & Space Group gave a demonstration of *Pre*Amp and its Rules Definition Facility (RDF). Greg is the architect of the RDF software application, which is based on the knowledge acquisition application, Kappa, from IntelliCorp, Inc. After much discussion about the interaction between the Aries and *Pre*Amp projects, a Memorandum of Agreement was signed in September 1994. For the Aries Project, the RDF would provide the functionality required to capture design rules and design practices. The *Pre*Amp program had developed this tool but did not have a structure in place to populate the data base with design rules. Thus, each organization needed the other's capability and would clearly benefit from working closely together. *Pre*Amp will use the knowledge base generated by the Aries Project during the demonstration phase of the *Pre*Amp program and will serve as the demonstration vehicle for the Aries Project.

1. Technique

The RDF prototype software is a data base editor used to capture knowledge in the form of rules. The RDF provides a user-friendly capability for CAD/CAM-oriented individuals to define and collect rules to be used for PWB engineering design and manufacture. Information defined in the RDF can be stored in the *Pre*Amp data base and later executed based on changes to the data base or at user request.

To simplify the task of defining and executing rules, the RDF organizes rules in rule sets. A rule set is a group of rules that share a common subject. The rule sets and their hierarchy are defined by the user. As an example, a user may wish to define a set of design rules for PWB assemblies. Subsets of rules under the design set could be analog, digital, and mixed. A rule can belong to any number of rule sets. When rules are executed in the *Pre*Amp system they are selected by rule set. Appendix D includes the Aries Project rule sets and a rule set hierarchy.

[4] Appendix A discusses the *Pre*Amp project and other alternatives.

The software is designed to prompt the user with questions and then give the required information. There are multiple ways of accessing a particular rule in the data base:

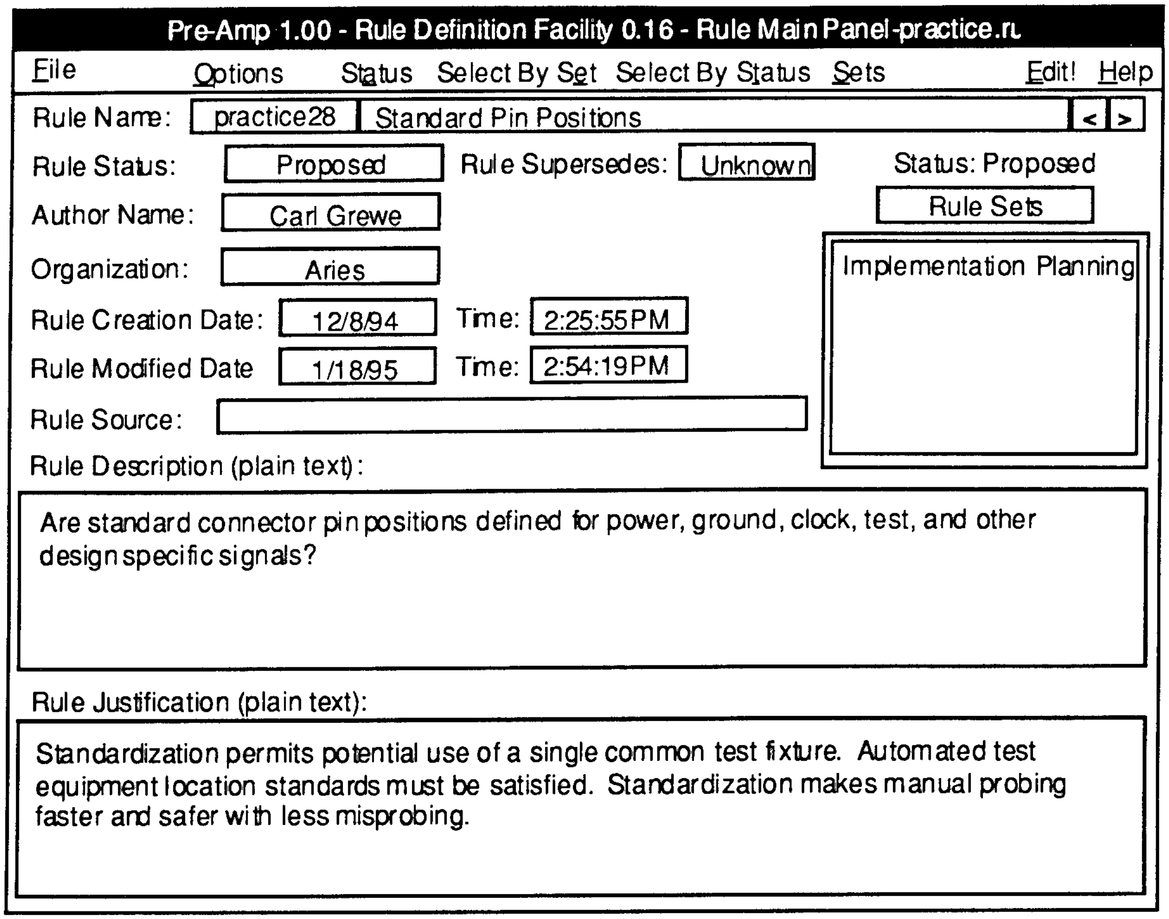
- *Select Rule by Status*: This will give the list of all the rules stored in the particular file. The user can then select the rule of interest and view the information regarding it.
- *Select Rule by Set*: In this option the user can select a rule set and view all the rules under it. This will provide more rules on a particular topic, thus giving more options about the topic.

Invoking the RDF displays the Rule Main Panel. This panel allows for the creation of new rules and the modification of existing rules. The definition and modification of rule management information are the primary purposes of the Rule Main Panel. Information captured on the Rule Main Panel is identified in Table II-2.

Table II-2. Rule Meta-Data

Rule Name	A short textual identification for a rule.
Rule Sets	A rule set is a group of rules that share common subject or theme.
Rule Author	The name of the individual entering the rule.
Rule Organization	The organization of the individual entering the rule.
Rule Source	Textual identification of the source of the rule.
Rule Description	A textual description of the rule.
Rule Justification	A textual justification of the rule.

The Rule Main Panel provides the user an effective knowledge capture tool while precluding the need for knowledge of rule component primitives or data base objects. This capability proved to be an asset for the Aries Project. Figure II-5 depicts the Rule Main Panel.


Pre-Amp 1.00 - Rule Definition Facility 0.16 - Rule Main Panel-practice.rL
File
Options
Status
Select By Set
Select By Status
Sets
Edit!
Help
<
>
Rule Sets

Rule Name: practice28 Standard Pin Positions

Rule Status: Proposed

Rule Supersedes: Unknown

Status: Proposed

Author Name: Carl Grewe

Organization: Aries

Implementation Planning

Rule Creation Date: 12/8/94 Time: 2:25:55PM

Rule Modified Date 1/18/95 Time: 2:54:19PM

Rule Source:

Rule Description (plain text):

Are standard connector pin positions defined for power, ground, clock, test, and other design specific signals?

Rule Justification (plain text):

Standardization permits potential use of a single common test fixture. Automated test equipment location standards must be satisfied. Standardization makes manual probing faster and safer with less misprobing.

Figure II-5. RDF Rule Main Panel

2. Implementation

With the Rules Definition Facility, implementation of the data capture process was straightforward. The team selected separate design rules and design practices pertinent to up-front digital and analog design. It was decided that the University of Maryland CALCE EPRC[5] would perform the design rule data input while EDS would handle the design practices. As the CALCE EPRC team members entered the rules into the RDF, Greg Smith provided training and consulting services on it. Greg also maintained the RDF software and implemented modifications to the software as necessary.

[5] The Computer-Aided Life-Cycle Engineering (CALCE) Electronic Packaging Research Center (EPRC), at the University of Maryland, is a State/Industry/University Cooperative Research Center (SIUCRC) sponsored by the National Science Foundation, the State of Maryland, and 30 industry and government members.

Carl Grewe, EDS, developed the data base for the design practices using Microsoft's Access data base. Entry of the design practices was straightforward. The design practices were identified as those electronic design methods that guide the electronic designer but are unable to be measured with rules. A description for each design practice was formulated. There was not always a justification for the practice, however, because of the nature of practices. For each practice, the data base contains the Description, Rationale, Phase of Design, and Source. The data base was set up to sort on different criteria, and reports for Global, Digital, and Analog could be printed from the data base. The design practice data base was later moved to the *Pre*Amp RDF, allowing the Project to have a single format.

3. Demonstration

The implementation of the design rules and practices was demonstrated in the *Pre*Amp RDF at the Reliability and Maintainability Symposium (RAMS) on 19 January 1995 at the Washington Hilton and Towers, Washington, D.C. *Pre*Amp RDF was the demonstration vehicle to show feasibility of captured design rules being executed in a CAE/CAD environment. This demonstration satisfied the requirement to show feasibility. The *Pre*Amp RDF Users Guide is provided in Appendix E.

E. RESULTS

The Aries Project has identified and captured a great number of design rules and design practices for PWB assembly design. Some examples of the rules are included in Appendix D. With the completion of the demonstration, the objectives of IDA's feasibility effort—to define the data capture techniques, to collect sample data, to prove out the techniques, and to demonstrate a feasibility prototype—has been met.

III. LESSONS LEARNED

Throughout the course of the Aries Project, a number of lessons were learned about how to conduct a project of this type. Future efforts will benefit from a study of these lessons learned. In general, they fell under the categories of managing resources, getting the right participants, handling proprietary information, and distributing and maintaining the data base.

A. RESOURCE MANAGEMENT

Several management issues arose during the project. First is the need for a new sponsor to continue this project after the initial feasibility demonstration. One proposal, to establish a RAMS Council, was withdrawn after lengthy debate by the RAMS Board of Directors (BoD). Although the RAMS BoD approved the Aries Project, it later voted not to endorse the Council on Reliability, Quality, and Competitiveness because of concerns about legal liability stemming from projects that it may sponsor and because the proposal was not endorsed by the sponsoring societies: Institute of Electrical and Electronics Engineers, Institute of Environmental Sciences, Society of Automotive Engineers, Society of Logistics Engineers, American Society for Aeronautics and Astronautics, Institute of Industrial Engineers, American Society of Quality Control, Society of Reliability Engineers, and the System Safety Society. Since the RAMS BoD approved the Aries Project separately, its sponsorship continued.

Suggested possible sponsors of future efforts include the Institute for Electrical and Electronic Engineers (IEEE), the Institute of Environmental Sciences (IES), and the Society of Automotive Engineers (SAE) G-11 Committee. To find a continuing sponsor, the IEEE was approached and asked to attend one of the Aries meetings. Dr. Irving Engelson, former director of IEEE Technical Activities and now director of Corporate Activities, gave the background and organization of the IEEE at the December 1994 meeting. The IEEE has Technical Councils for cross-disciplinary fields across several societies. He said the Aries Project looked like an appropriate candidate for a Technical Council. To make an IEEE Council, 35 societies must approve the proposal, a process that takes about one year. A follow-up meeting took place at the IEEE headquarters in May 1995.

A second management issue is the need to generate revenue to fund continuing efforts. Society sponsorship is a critical component. The funds are needed to help defray travel and other costs if required by the industry participants. Once the benefits of these projects are well known, industry may not require substantial reimbursement. Expenses could be further reduced by holding project meetings in tandem with regular meetings of the society sponsor, e.g., SAE G-11 Committee meetings.

B. INDUSTRY COMMITMENT

Projects such as the Aries Project rely on industry volunteers to provide expertise and data. The Aries Project petitioned companies from the defense, commercial, and CAE/CAD industries for volunteer representatives to staff the Core Team. Filling the positions on the Core Team proved to be more difficult than expected. Under current economic conditions, many companies are cutting the amount of money they will spend on an employee's other activities. In addition, many projects and meetings are now vying for the time of industry members. Many companies have people who are sort of the official meeting goers, but these people are not necessarily the right people for a particular team.

For the Aries Project, it took several meetings before a committed group of volunteers could be assembled. Once this group came together, the project made great strides. Getting people to participate and having their company donate their time was one of the toughest issues. We tried to recruit people in the Washington, DC, area to cut down on travel expenses, but we needed a good cross section of the industry. Approximately 30 companies were contacted beyond those that eventually participated. IDA provided the project management for this feasibility study and kept the documentation. IDA also reimbursed the travel expenses of the industry participants so that the meetings would be held at IDA for tracking the feasibility process. As in the case of other IDA tasks we needed to afford companies the opportunity to see the value in the project since they were paying for the time spent by their representatives.

C. GETTING TO RIGHT CONTACT

The success of this feasibility project was directly dependent on getting the right people involved—not only those with the required expertise, but also those with an interest and desire for success. This project required work on everyone's part. Core Team meetings were held quarterly, beginning in October 1993 and continuing through May 1995, and much work was required in between. The participants needed personal drive and a willingness to work, as well as their company's support for their involvement. Moreover, the companies had to allow data and expertise to be released for general knowledge.

Even when a company showed interest in becoming a Participating Company, it was important to get to the right person within that company. This person not only had to champion the project within their company to attain the design rules, but also had to have the expertise necessary to communicate the knowledge about the rules to the Core Team.

Many mechanisms were used to recruit participants. Initially, an Aries Project information briefing was developed to be given as a call for Participating Companies at the RAMS in January 1994. A flyer announcing the project was also distributed at that meeting. Letters were sent to a multitude of companies—both for Core Team members and for Participating Companies once they showed interest. An article appealing for Participating Companies was published in the *Reliability Analysis Center (RAC) Journal.* And the CALCE EPRC also had an article on the Aries Project in its newsletter.

The Aries Project abandoned the original idea of the Core Team and other Participating Companies in favor of a single, strong, cohesive team. The Aries Project team ultimately comprised industry representatives from Texas Instruments, EDS, Honeywell Inc., ITT Avionics, and AlliedSignal Aerospace; knowledge acquisition expertise and software systems engineering from the University of Maryland CALCE EPRC; and CAE vendor representation.

D. CAE VENDOR INVOLVEMENT

The Aries Project felt that CAE vendor involvement was critical. We wanted the data base to be in a usable format for the CAE vendors and asked them to give comments on the initial data base structure. We wanted the CAE vendors to see a market for this type of product.

Getting CAE vendors involved was more difficult than anticipated, although representatives did attend some meetings and demonstrate software. There were various reasons for this lack of involvement. The vendor industry, like the electronics industry, has too much required activity in the face of declining budgets. The people who are available and whose companies would send them to the meetings are often from Marketing. The expertise required is really the domain of their technical employees, who have too much work to deal with already. Software developers within the company also would not represent the right level of involvement. Also, many of the vendors are technically still at the level of developing schematic capture software rather than rule-based design checkers that lead the designer through a design. Lastly, what is needed is sanity checks of the project for its business case, and this wasn't forthcoming.

E. PROPRIETARY INFORMATION

Guides and guidelines are published internally by many companies. These are the things designers should know before starting the design process. After all, these are the mechanisms that guide them into and through the process. Design rules are those things that should not be violated; consequently, they limit the choices available to the design decision-making process. Many companies do not necessarily have documented design rules. Those that do have documentation usually have it in the form of guides or practices.

When a company does have published design rules, more often than not these are treated as strictly proprietary. After expending the effort to develop the rules, companies don't want to share them. Even guidelines are often considered proprietary. Many potential team members could not participate because they were not allowed to bring design guidance documents outside their company.

The rules were limited to nonproprietary information to enhance company participation, but this still was an issue with many companies and we still had difficulty obtaining the rule documents. Because many companies do not have documented rules and guides, companies that do have them don't want to give that effort away.

F. DISTRIBUTION

As the feasibility project ended, questions about the distribution of the rules data base arose. The proof-of-concept data base was generated in the RDF of *Pre*Amp using the Kappa AI program. If team members were to have the data base as is, their companies would need to buy Kappa. Site licenses were discussed, as were price negotiations and shareware options. There were fees and licensing issues associated with distribution of the Kappa software. *Pre*Amp had bought floating licenses for its team members, but this was not an option for the way the project was structured. To avoid these issues, the Aries Project decided to provide the data base in ASCII format and not be tied to a particular AI shell.

G. MAINTENANCE

Maintenance of any data base is a timing issue. If the data base is picked up immediately by the CAE vendors and used in their tools, industry users can then maintain their own data bases in the purchased tools. If the captured data sits on the shelf for some time, then maintenance and updating become major issues—one that goes back to the issues of the sponsorship and continuing future support.

IV. FUTURE EFFORTS AND RECOMMENDED NEXT STEPS

The need for automated on-line design rules and practices for a wide range of engineering disciplines was recognized early in the Aries Project activity. Limitations on the available funding led to the decision to focus the effort on the subset of digital and analog design rules, which resulted in capturing a sample of those rules in a knowledge-based data base system to test the feasibility of the approach. As the Aries activity continues in the future, the scope of the engineering design practices addressed should be expanded to include a wide range of technologies in the electrical and mechanical domains as well as best practices in design methodologies.

A. SELECTING TECHNOLOGY AREAS

It is recommended that design rules and practices be captured for the following industry segments:

- Telecommunications
- Wireless Data Transfer
- Computers
- Automotive

Communications and computing are the fastest growing market segments. The need for design rules is becoming more apparent as digital technology advances at an exponential rate. Applicable technologies and processes are described below, and projects to be undertaken are prioritized.

1. Remaining Electrical

Radio Frequency (RF) Design. These technologies can be separated into several categories which tend to have different sets of design practices. The first subset includes designs in the frequency band ranging from hundreds of MHz to low GHz. A second subset would include the higher microwave frequencies where the effects of the physical layout and physics are as important as the device and circuit design. Typical technology needs also include wave guide design, couplers, and filters, which combine the electrical and mechanical domains in the design practices. RF integrated circuits and millimeter and microwave integrated circuit (MMIC) devices also have their own set of design rules. These devices are being used extensively in telecommunication, aircraft, and military systems. This is probably the most important and needed area for future efforts.

Power Supplies. Power supply design for electronic systems can be separated by both technology issues and design practices into low voltage and high voltage systems. Similar to the microwave applications, many power supply design practices include mechanical and thermal considerations along with the electrical design rules, an approach that affects the system reliability. This area is also important and needed.

Digital Application Specific Integrated Circuits (ASICs). This technology includes gate arrays, standard cells, and full custom devices. The need for good design practices is especially critical in this technology because of the relatively large cost and schedule impacts involved—conditions that bear out the Aries Project's first-pass success philosophy.

Analog ASICs. The need here is similar to the digital ASIC problem, with the added complexity of large circuit analysis and component variations which need to be resolved. Both digital and analog ASICs are component problems that are being addressed by many companies.

2. Layout

The mechanical printed circuit board layout process uses many design rules and practices, depending on the technology and materials used, and the type of electrical circuits involved. As alluded to in the preceding sections, the rules for layout of digital, analog, RF, microwave, and power supply designs vary significantly and are critical to the overall successful fabrication of the designed system. Other related technologies that have their own unique design practices include multichip modules (MCMs), thick film and thin film hybrid packages, and different board materials such as ceramic, fiberglass, and composites.

3. Design Processes

Best industry practices need to be identified in the area of design tools and methodologies. Most electronic and mechanical systems today are designed using a wide range of sophisticated computerized design tools for simulation, analysis, and physical design. Very little exists, however, in best practices for successful use of these tools. As a specific example, logic simulation tools can be used to simulate digital circuits and systems, but issues such as how thorough the simulations are, how and when they should be done, and under what conditions are left to be decided based on the experience of a design organization or engineer. Similar problems exist with analog and RF simulation, finite element analysis, board layout, and thermal analysis. Design rules and practices for the optimal use of tools and processes would be a benefit to the overall design community.

Further extensions to industry-accepted design rules could include narrower disciplines such as antenna design, AC power systems, and distribution.

4. Mechanical and Structural Design

The areas of mechanical and structural design involve a multitude of design rules and practices that affect system design from its earliest tasks to the most detailed design of specific structural members. As the initial design aspects of an airplane, tank, or automobile are defined, the space available for fuel, people, electronics, and power plants are determined. These early design decisions are based on the performance and supportability envelopes for the final system and therefore constrain the design rules and practices that affect the mechanical, structural, and electrical elements of the system. Underlying the system-level design rules is a fairly unlimited scope of potential technology areas that could be addressed in rule-based projects such as the Aries Project. For example, the materials and shapes used to build a bulkhead are based on known practices and rules that define the strength and weight of the member. The weight and strength, combined with the shape and performance requirements of the system, define the space available to route plumbing and wiring. All of these issues affect the overall performance and supportability aspects of the system and subsystems. These issues are affected by and constrain the specific design rules and practices used in other mechanical, structural, and electrical designs.

Work must be started in the mechanical and structural areas in parallel with electrical projects if a single integrated rule-based architecture is going to be built. Ignoring the mechanical and structural aspects could easily lead to a system that only supports part of the design process. What is needed is a system that allows intelligent use of all design data to properly determine the rules and practices that affect the task at hand.

5. Recommended Projects

Given that many areas are in need of rule development, we felt that we could best serve the industry by describing a list of design types and prioritizing them. Designers in the electronics industry would benefit most from rule sets in these areas:

- High speed digital—The need for high computing power has pushed microprocessor bus speeds to exceed 66Mhz. Higher board speeds now run into signal noise and timing issues.
- PWB Layout with Surface Mount Devices (SMD)—Smaller and lower cost packaging and the need for hand-held, light-weight products promotes SMD designs. Those designs require specific mounting procedures.
- RF Design—Telecommunication and wireless devices are growing in popularity, speed and features. Federal Communications commission (FCC) regulations must be a part of the design process.
- Hardware Description Language (HDL)-Based Design (Very High Speed Integrated Circuit (VHSIC) HDL, Verilog)—Top-down design methodologies including synthesis are mandatory for designing 50K gate (or larger) ASICs. These are needed to maintain Intellectual Properties in sophisticated designs.
- Mixed-Signal Design (Analog and Digital)—The fastest growing areas include satellite communications and automotive, where sensors and actuators interface to digital processing circuits.
- Virtual Co-Design Re-Use Libraries—The need to use known-good components at the requirements and design stages to reduce cycle time is growing. Practices and rules for the correct use of those components will also be needed. Support for embedded software components, and for components composed of hardware and software (co-design), will also be needed.

B. APPLYING LESSONS LEARNED

This final report demonstrates the feasibility of a rule-based-design data base and includes sample information captured. It also includes the generic knowledge acquisition technique for future capture of design rule information. Chapter III and the following lessons learned can provide training material for future efforts. It is important for future efforts to take advantage of the lessons learned by the Aries Project.

1. Common Focus

It is essential to scope the project carefully and set the boundaries of what the new effort is specifically going to tackle. Project participants need to have common understanding before proceeding. The obstacles must be known in advance. The following requirements must be addressed:

- Definition of data to be captured
- A rationale
- A plan for capturing data in electronic format

At one point in the Aries Project, we considered whether we should set a specific example and then ask for the rules for it. For example, we might have given a basic circuit, and asked for judgment on good, better, and best rules against which to judge the circuit. We settled instead on the process described in Chapter II. It is important to establish the data structure before trying to capture the rules. This will probably mean segregating the data into sub-design elements.

2. Team Consistency and Consensus

It is essential that the team consist of permanent members for a defined scope (area of expertise). Introducing someone new midstream in the process is not easy. By April 1994, we knew that regular, consistent participation and attendance by all team members was critical from that point forward. If people wanted access to the results, they had to contribute and attend project meetings. In addition, we found that meetings needed to last at least 2 days to allow for start-up problems in getting down to work.

Once the Aries Project had enough information to sort through, we made viewgraphs of pages out of documents so that the whole team could view them together and reach consensus. Reaching consensus on the verification of the rules themselves was an iterative process—once the rules were captured, printouts of the data base were distributed at subsequent meetings and much discussion still occurred.

3. Rule Selection Considerations

In selecting the rules for the knowledge-based system, several questions arose. They are generic in the sense that any future effort must also consider them.

- How do we get large compilation inputs?
- How do we get a representative set?

- How do we get the rules evaluated?
- How do we separate rules from good design practice?
- How do we compile them to make them usable to CAD vendors?
- How do we make sure that the rules we are entering are robust enough to check different types of circuit design?
- Are the data always complete and correct?
- What kind of information should be extracted from the rule to formulate its premise and conclusion?
- How do we capture the syntax of the premise and conclusion of a particular rule?

4. Rule Structure

In general, a rule is composed of a premise and a conclusion. The premise component (or IF condition) defines what actions are required before a rule can succeed. The conclusion component (or THEN condition) consists of actions to take upon achieving the actions called for in the premise.

The rule premise is a logical combination (ANDing or ORing) of premise functions. Premise functions are logical (e.g., greater than, less than), mathematical (e.g., sum, power, division), or data base (retrieve object attributes or facts). These functions include storing temporal information into the data base (facts), storing permanent information into the data base (object attributes), and providing responses to the user in the form of messages and issues.

The rule conclusion is a logical combination (ANDing) of conclusion functions. These functions include storing temporal information into the data base (facts), storing permanent information to the data base (derived knowledge), and providing feedback to the user in the form of messages and issues.

Note that conclusion functions include the storing of facts and/or object attributes. A rule premise may include functions that test facts or object attributes. Using the two preceding capabilities allows one rule to call another rule. This process is known as *chaining*. This ability can be used to capture the intent of a multistage rule with several smaller rules.

5. Use of Shall, Will, Should, and May in the Rules

Use *shall* whenever a specification expresses a provision that is binding. Use *should* and *may* whenever it is necessary to express nonmandatory provisions. *Will* may be used to express a declaration of purpose on the part of the contracting agency. It may be necessary to use *will* in cases where the simple future tense is required.

IEEE Standards use *shall*, Recommended Practices use *should*, and a Guide uses *may*.

C. SECURING FUNDING TO SUPPORT PROJECT

1. Commercialization

The government funding for this feasibility project was intended to be the seed money that would enable the Aries Project to be self-sufficient. The Perry memo eliminating the use of defense-specific standards and specifications enables a wider market base for the rule bases because of the emphasis on commercial standards. Indeed, toward the end of the Aries Project, a major automobile corporation was showing great interest. The corporation balked, however, at the requirement for entrance to the project—release of its design rules and practices. It has a wide variety of suppliers that all use their own rules and practices. The corporation intends to standardize the rules and practices—and could benefit substantially from a data base such as the one developed under the Aries Project (see Section I.C.2, Benefits of Project Participation).

2. Society Funding and Sponsorship

Beyond this feasibility project and sample data, a distribution mechanism for a populated data base is needed. This distribution mechanism must provide sponsorship and credible recognition. Preferably it would be an established business entity that has marketing and distribution channels with contractual vehicles. This mechanism would take control of future Aries Project "products"—the design rule data base—and market it to provide a revenue stream for expansion of the Aries Project into the recommended areas.

a. IEEE

In May 1995, a meeting was held at the IEEE Headquarters in New Jersey so that Aries Project team members could meet in depth with IEEE's Technical Documentation and Standards personnel. We needed information on the revenue generated from the sales of

IEEE technical reports, standards, and best practices. The IEEE Rules and Practices fall under the administration of the Standards Boards of IEEE. Consensus for adoption of a true standard requires the approval of 75 percent of the voters present, and the voters present must represent 75 percent of the Board membership. IEEE as the sponsor is then responsible for the maintenance after publication. The future Aries Project products will be aimed at the best practices type of document. The IEEE is interested in this type of activity, but much coordination must be accomplished before a relationship can be established.

b. SAE

A meeting was held with the chairman and other ranking members of the SAE G-11 Reliability, Maintainability, and Supportability (RMS) Committee in March 1995. The purpose was to determine the potential for performing an effort similar to the Aries Project in the mechanical or structural area. Although the interest was high, the SAE does not make sufficient profit on the sale of standards to provide any funding for the effort. Thus, any work performed under SAE sponsorship must be done completely, including travel, from individual company funds. We are continuing to pursue this function as a special project under G-11 sponsorship. The final decision on the project is up to the G-11 members based on whether they can and will volunteer their time, rules, and practices to the project. A briefing is being developed for the October 1995 G-11 meeting.

3. Additional Government Funding

As we finish this feasibility task, we see the need for activities such as the Aries Project to be ever more important as commercial and defense design rules begin to merge. Rule-based design has been driven by the companies' needs to improve quality, time, and cost, and now, also by the elimination of MIL-SPECS and MIL-STDS. A mechanism for funding must be found or expansion of the design rule and practice data base into other areas will not be possible. Team members are highly skeptical that this type of activity alone would generate enough money to pay the expenses of the volunteers and continue the project. It is hoped, as a result of the Perry Initiative, that additional government funding is possible to advance the dual-use rule-based design concepts into reality.

The best approach for project funding would be as part of a winning technology proposal that includes rule-based design concepts. Ken Blemel, Management Sciences, Incorporated (MSI), successfully requested that the Aries Project be included in a

RASSP-related proposal. Some type of government technology program funding is probably the best avenue for keeping the design rule knowledge acquisition project active.

Investigation should be made into possible funding under the Technology Reinvestment Project (TRP) within DoD or the Advanced Technology Program (ATP) under Department of Commerce, or whatever the latest direct-funding proposal mechanism happens to be. Alternatively, some industry consortium could fund such an activity, but we know of no consortium for Design.

V. SUMMARY

The goal of the IDA task was to prove that a volunteer team, with members from across industry and multiple disciplines, could identify, define, and demonstrate a design rule capture process that could be codified for encapsulating any industry's set of practices and rules into a rule-based system. We have demonstrated that this is possible and that there is industry support for these kinds of activities and tools. We found that most "good commercial practices" companies have many practices and rules in common. Their need to share these practices with their subcontractors for improved quality products and reduced cycle times continues to grow, so there appears to be a reason to expand and continue these efforts in the future.

The CAE work being done by suppliers like Cadence, ICAD, and *Pre*Amp all demonstrate that tools that support rule-based, correct-by-design development are not only possible, but also marketable in our industries. Feedback from early users of these tools indicates that these tools not only need to provide an initial capability, but also need to allow the users to expand the system on their own. Focus groups like the Aries Project can help to consolidate design practices and rules for CAE suppliers and promote the use of these techniques in our industries.

Both DoD and commercial industry stand to benefit from consistency of the design rules across a company's supplier base. Design variability would be reduced. If properly done, we would have more robust designs with reduced cycle time and first-pass success. If rule-based shells are populated with approved design rules, each user company would save time and effort and industry as a whole would benefit.

A set of rules and guides for electronic design could be used with university or technical school CAD curriculum to introduce students to current design practices. It would support trial-and-error analysis initially, and correct-by-design developments later as their skills improve. The initial capability would allow the students to determine (on their own) the solution to a problem, after which the result could be analyzed according to best industry practices. The second capability would introduce the students to the design environment they would likely encounter in industry. Having had this exposure, graduates will be more productive sooner in their chosen fields. The concept is extensible to all engineering disciplines, although the Aries Project addressed only a small portion of the Electrical Engineering discipline.

The Aries Project team has shown that a voluntary group can accomplish an end goal when populated by interested individuals who are supported by their home organization and have proper sponsorship and project management. The team members learned, made contacts, and obtained information that will be of benefit for a long time after this phase of the Aries Project is complete.

Appendix A

ARIES PROJECT PARTICIPANTS

Appendix A
ARIES PROJECT PARTICIPANTS

Team Members

Dennis Hoffman, Chairman, Texas Instruments
Kenneth Blemel, Management Sciences Inc
John Fink, Honeywell Inc
Carl G. Grewe, EDS
Ali Hashmi, University of Maryland, CALCE EPRC
Ray Johnston, Texas Instruments
Pradeep Lall, University of Maryland, CALCE EPRC
Guoqing Li, University of Maryland, CALCE EPRC
Douglas E. Ott, ITT Avionics
Marty Rosman, AlliedSignal Aerospace
Karen J. Richter, Institute for Defense Analyses
Carl Rust, University of Maryland, CALCE EPRC

*Pre*Amp Representatives

Gerry Graves, SCRA
Greg Smith, Boeing/*Pre*Amp

Knowledge-Base Systems

Phil George, Cadence
Rich Kilgore, ICAD
Warren Mudd, Cognition
Scott Smith, ICAD

Tri-Service CE-CAD Process Action Team (PAT)

Sid Markowitz, U.S. Army, ARDEC
Doug Patterson, OASN(R, D&A) Product Integrity, DoD
Ed Smith, OASN(R, D&A) Product Integrity, DoD
Matt Tracy, U.S. Air Force, AL/HRGA
Bob Whiteley, U.S. Army Materiel Command (AMC)

RAMS Board

Tom Fagan, ITT Defense & Electronics
Richard E. Sackett, Hernandez Engineering Inc

Appendix B

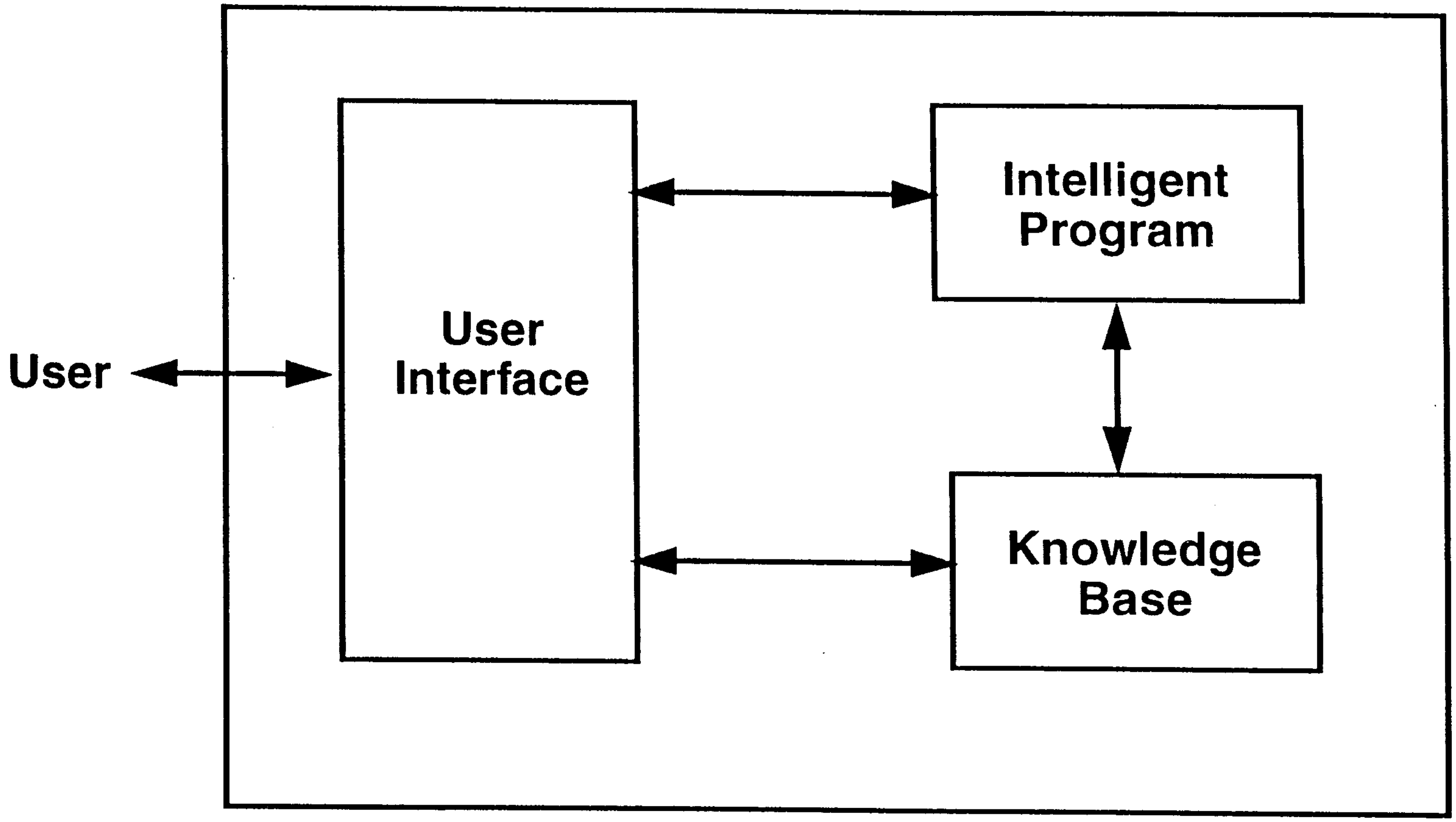
KNOWLEDGE-BASED SYSTEMS AND KNOWLEDGE ACQUISITION

Bruce Hubanks
Texas Instruments

Definition of Knowledge-Based System

- A computerized system that uses knowledge about some domain to arrive at a solution to a problem from that domain. This solution is essentially the same as that concluded by a person knowledgeable about the domain when confronted with the same problem.

Knowledge-Based System



Knowledge-Based System

- Domain specific knowledge base
  - Contains the knowledge of the system
- Intelligent program
  - Process techniques to organize and access the knowledge base.
- User interface
  - Provide communications between the user and the intelligent program.

KBS Advantages

- Wide Distribution of Scarce Expertise
- Ease of Modification
- Consistency of Answers
- Perpetual Accessibility
- Preservation of Expertise
- Solutions involving incomplete or uncertain data
- Explanation of Solution

KBS Disadvantages

- Answers not Always Correct
- Knowledge Limited to Particular Domain
- Lack of Common Sense

Signal and Information Processing

Current and Future Knowledge-Based Technologies

Rule-Based Systems

- Most Popular and Common (Expert Systems)
- Rules and Inference Method (Forward / Backward)
- Advantages
  - Modularity
  - Uniformity
  - Naturalness
- Disadvantages
  - Infinite Chaining
  - Addition of new contradictory knowledge
  - Modification of existing rules
  - Inefficiency
  - Opacity
  - Coverage of Domain

Model-Based Systems

- Represents Physical System as a Model
- Advantages
  - Reduce knowledge acquisition
  - Robust problem solving
- Disadvantages
  - Manipulation of heuristic knowledge
  - Uncertain data
  - Single feature analysis
  - Limited problem domain

Qualitative Reasoning

- Predicts Behavior of Modeled System
- Advantages
  - General Inferencing
- Disadvantages
  - Accuracy

Case-Based Reasoning

- Represents Knowledge as Explicit Cases
- Retrieves Case that is most similar to the current problem
- Advantages
  - Patterned after human reasoning
  - Analogical reasoning
  - Simplified knowledge acquisition
  - Broad base of knowledge
- Disadvantages
  - Computation Cost

Temporal Reasoning

- Reasons about time sequence of events

Neural Networks

- Interconnection of simple processing units (Neurons)
- Advantages
  - Learning
- Disadvantages
  - Limited domain
  - Training time

Related Topics

Blackboards

- Contains three parts
  - The blackboard
  - The knowledge sources
  - The Control
- Advantages
  - Suitable to diversity of problems
- Disadvantages
  - Expensive
  - Difficult to partition knowledge sources

Uncertainty Management

- Four Popular Methods
  - Bayesian Probability
  - Certainty Factors
  - Dempster-Schafer
  - Fuzzy Sets

Truth Maintenance

- Retraction of reasoning based on a fact that is no longer valid.

Knowledge Acquisition

- Elicitation of knowledge from experts
  - Interviews
  - Incremental Development
  - Prototypes

Verification and Validation

- Difficult for knowledge-based systems.
- KBS are not objective
- Contain some degree of uncertainty
- Require expert to aid in verification
- Correctness by consensus
- No universal accepted method

Current State-of-the-Art

- Development Tools Exist that implement Individual and Combinations of Reasoning Techniques
  - Appendix I of Technology Survey

Feasibility Analysis

- Does a Problem Really Exist?
- Is a Knowledge-Based Technique Suited?
  - Is human knowledge being replicated
  - Is the problem-solving heuristic or algorithmic
  - Does the knowledge change periodically
  - Is the expertise fairly well understood
  - Are the data always complete and correct?
  - Can the problem be solved through other means
  - Does it pass the telephone test?
- Is a Knowledge-Based Approach Really Justified
  - Cost
  - Complexity

TI Capability

- Developed and Demonstrated a Real-Time Distributed Cognitive Decision Aiding System
- Covers mission, tactical and situational awareness for operators of military vehicles
- Flight Tested in Helicopter Flight Test Vehicle

Knowledge Acquisition

Knowledge Acquisition

- Primary method is face-to-face discussions between expert and knowledge engineer.
- Represents a significant portion of the KBS development effort.
- Knowledge Engineer must establish and maintain a good working relationship with the domain experts.

Knowledge Acquisition Techniques

- One-on-One Interview
  - Kickoff Interview
  - Knowledge Elicitation Sessions
  - General knowledge-gathering sessions
  - Specific problem-solving, knowledge-gathering sessions
  - Knowledge Organization
  - Knowledge Documentation
- Observation
  - Quiet on-site observation
  - On-site observation with discussion
  - Exercising the Expert
  - Problem description and analysis
- Intuitive Techniques.

Alternative Knowledge Acquisition Methods.

- **Knowledge Engineering Facilitators:**
  - Tools that interact with a user to help structure knowledge to generate a set of domain rules.
- **Inductive Tools:**
  - Create a set of rules from example cases presented by the user.
- **Automated knowledge extraction from databases.**
  - Requires little or no human intervention
  - Machine Learning

Appendix C

ALTERNATIVE APPROACHES

Descriptions of Rule-Based Design Systems

Appendix C
ALTERNATIVE APPROACHES

Software CAD/CAE developers assisted in the information development process to ensure that the information could be used in the CAD/CAE environment and that they understood what the information means. It was felt that if we used a standard format and worked with the CAE vendors, the rules would be used by them—sort of a sanity check. The Aries Project also needed a demonstration vehicle for its sample data base to show feasibility. Throughout the Project, demonstrations were given by various vendors of CAD/CAE and knowledge-based tools.

A. *PRE*AMP AND THE PDES, INC. ELECTRICAL/ELECTRONICS (EE) PROJECT

Gerry Graves of the South Carolina Research Activity (SCRA) gave the *Pre*Amp briefing at the April 1994 meeting. *Pre*Amp is the name for the Pre-Competitive Advanced Manufacturing Processes project whose goal is enabling concurrent engineering for Printed Circuit Assemblies (PCAs) for the electronics industry using the Standard for the Exchange of Product model data (STEP) with intelligent information sharing, automated rule specialization based on manufacturing process capabilities, and automated process planning.

*Pre*Amp is funded by the NIST Advanced Technology Program (ATP), run by the South Carolina Research Authority (SCRA), which also contains the Rapid Access for Manufactured Parts (RAMP) Program and the Joint Center for Flexible Computer-Aided Manufacturing (FCIM). Member companies include: Boeing, Digital Equipment Corporation, Mentor Graphics, Rockwell International, Hewlett Packard, Martin Marietta, Versant, Rockwell Collins, and Hughes. The program grew out of the PDES, Inc., Consortium (Product Data Exchange Standard using STEP). They are also coordinating with the Rensselaer Polytechnic Institute (RPI) Agile Manufacturing Research Institute (AMRI) for Electronics, which is funded by the National Science Foundation (NSF).

The primary objectives of the *Pre*Amp Program are to define and demonstrate the technology that enables:

- Intelligent information-sharing for concurrent engineering automation in Product Design, Product Manufacturing, and Manufacturing Process Design.
- Automated manufacturing process rule specialization from manufacturing capabilities.
- Shared data base access.

The *Pre*Amp program will enable shared information environments across organizational boundaries. Sharing information electronically supports just-in-time production and the early discovery and resolution of design and manufacturing issues. It reduces non-value-added activities, and it provides timely information on product or process changes.

The strategy is to use existing commercial-off-the-shelf (COTS) tools where possible in the areas of Product Design, Manufacturing Process Development, and Manufacturing Process Planning. It uses an Object-Oriented Data Base Management System (DBMS) and is developing STEP Application Protocols (AP 210 and 220) in the areas of product and process design and planning for printed circuit assemblies (PCA). It uses an EXPRESS-driven translation and the Standard Data Access Interface (SDAI). Applications include a Manufacturing Resource Editor (MRE), a Producibility Advisor (PA), Computer-Aided Process Planner, a Knowledge Acquisition Facility, and a Knowledge Execution Facility. It uses a three-schema architecture for the knowledge acquisition: between the external schema of the domain expert and the internal schema of the knowledge engineer in the knowledge base there is an intermediate representation or conceptual schema.

The PDES, Inc., Electrical/Electronics (EE) Project mission is to accelerate development and implementation of a standard for computer-interpretable EE product data descriptions. That standard should support all EE life-cycle phases. The project's initial focus is on information models for the design and manufacture of PCAs. The information models will build on the knowledge and expertise of existing EE standards.

PDES, Inc., is a joint industry and government consortium including more than 20 major technology companies. The PDES, Inc., goal is to accelerate the development and implementation of the ISO (International Organization for Standardization) STEP standard. STEP defines a standard for computer-interpretable product data. Industry has identified computer-interpretable product data as a crucial technology in the worldwide competition to improve new product development. That is why industry has recognized STEP's competitive significance. The U.S. government, recognizing the same benefit, is planning on including STEP in future acquisition requirements.

PDES, Inc., launched the EE project in August 1991. The PDES, Inc., EE project is building on the progress PDES, Inc., has already made in accelerating STEP by developing and testing STEP models for the EE product life-cycle. The EE project team members are experts in EE design, manufacture, and Design Automation. The member companies (Boeing, Digital Equipment Corporation, Hughes, Hewlett-Packard, Martin Marietta, Rockwell, NIST, U.S. Navy, and U.S. Army) are jointly focusing a dedicated team on the development of EE STEP and exerting combined influence on the EE application software vendor community.

B. CAD/CAE INDUSTRY

At the January 1994 meeting, Warren Mudd from Cognition demonstrated their rule-based system for Component Cost. The Cognition system interfaces with its own CAD system and those of two other vendors.

ICAD was demonstrated at the April 1994 meeting. The ICAD system is neither a mechanical design package nor a parametric modeling tool, but knowledge-based engineering software that allows companies to shorten the product development cycle for critical components and large assemblies. In the ICAD system, the collection of rules—design rules, standard engineering rules, physical product attributes, or manufacturing process information—is the Smart Model^{TM} or product model. The model takes input specifications, applies the relevant rules, and generates a product design automatically. The model also contains information for outputs such as reports, data for analysis, 3-D geometric models, bills of material, cost reports, and manufacturing instructions. When the design changes, so does the output.

Major system components include the ICAD Design Language^{TM} (IDL^{TM}) for defining the rules in a Smart Model, geometry and drawing tools for creating rule-based definitions of complex surfaces, solids and drawings, user interfaces for developing and interacting with a Smart Model, and data integration tools for linking to other software products. Advantages of the system include:

- Object-oriented, declarative, with structured query language
- Applies best practice consistently
- Supports continuous improvement (successive refinement) and concurrent engineering.

- Graphical debugging tools (ICAD Browser), see geometry as developed
- Integratable with other software programs; interface with PASTRAN, NASTRAN
- Close ties with major CAD systems—especially in aerospace and automotive industries.

Rick Kilgore of ICAD gave a presentation at the August 1994 meeting on how ICAD chooses and develops projects. ICAD has an Enterprise Strategy Development Process used to facilitate and exploit Integrated Product Development (IPD) through knowledge-based engineering (KBE). This includes: Education and Awareness, Business Case Development, Proof of Concept, and Pilot Projects within both small scale and large scale implementation. Rick sees the design practices as the driving force behind the design rules. He suggested that the Aries Project supply the design process flow to the vendors so they know what the designer does first, second—What's most important?

Cadence has recently released a knowledge-based tool and a rule checker called CheckPlus™. Phil George from Cadence attended the December 1994 meeting and gave the team handouts on the new tools and expressed a willingness to team with Aries. This product, called CheckPlus, has the following key features:

1. The Cadence solution, from design to layout, can be rules driven. Users add their constraints up-front in the process and downstream tools follow those constraints.
2. CheckPlus is an advanced rules-checking and advisement system used up-front in the design process. It features:
    a. Advanced Rules Language (ARL®) which simplifies rules writing.
    b. An intuitive user interface
    c. An evaluation engine
    d. A CAE/CAD capture tool interface enhancement (called "markers") to highlight errors found in the design.

Half a dozen other CAE vendors were invited to participate and received early information packets about the Project. They did not participate, although some expressed interest.

Appendix D

DATA STRUCTURE AND EXAMPLES OF RULES

Appendix D
DATA STRUCTURE AND EXAMPLES OF RULES

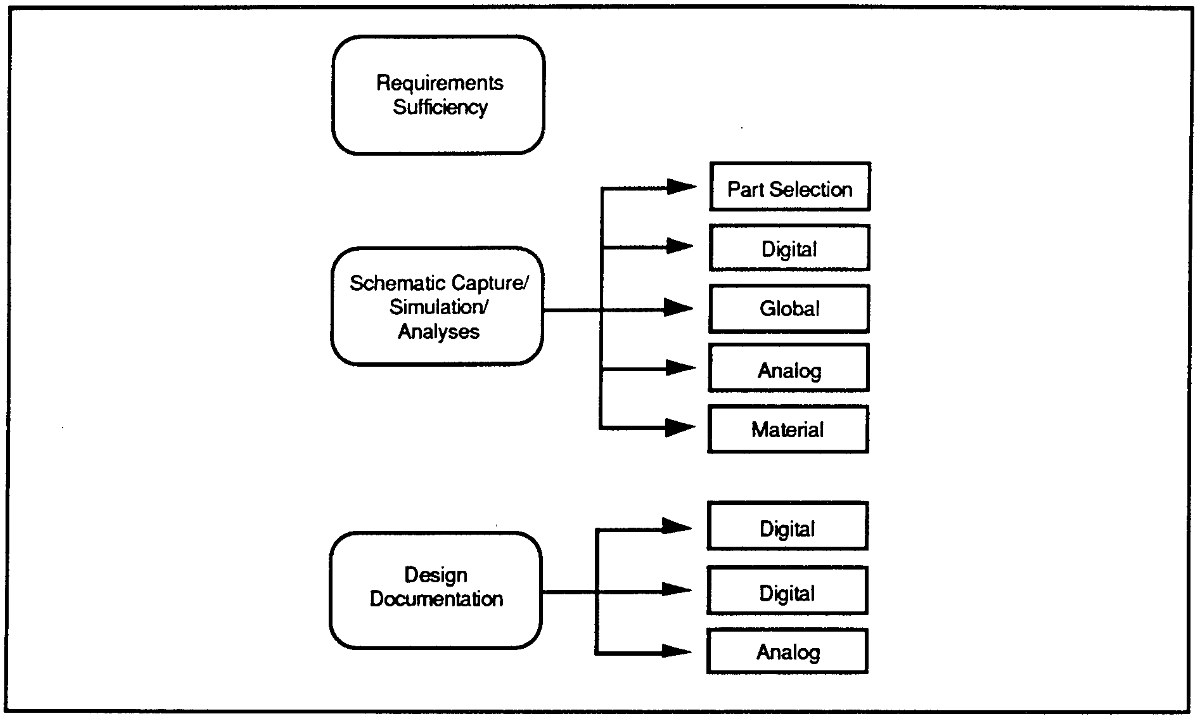


Figure D-1. Rule Sets Schematic

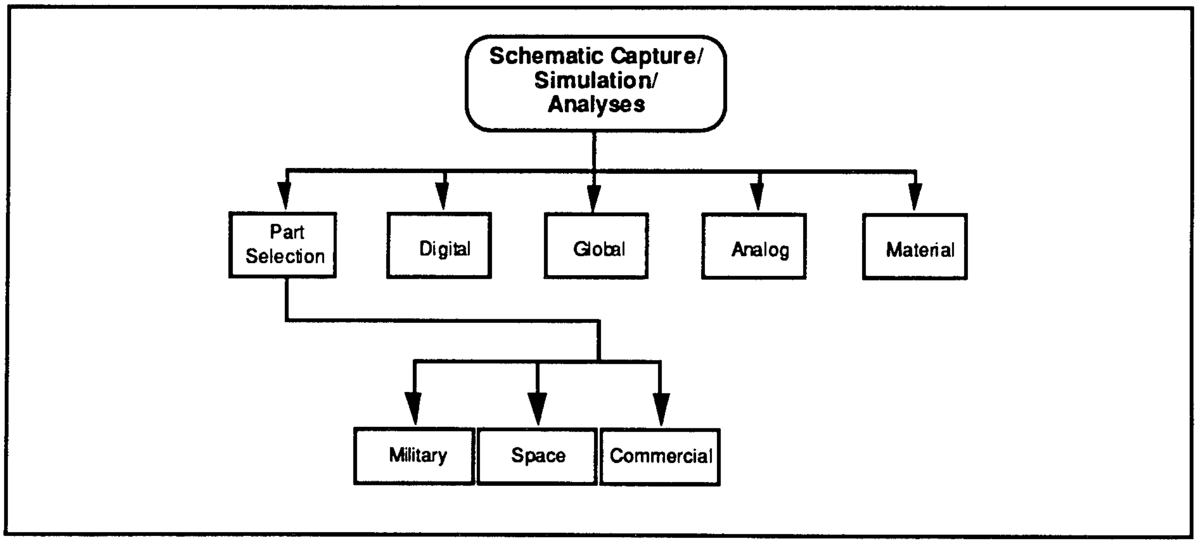


Figure D-2. Schematic Capture/Simulation/Analyses—Part Selection

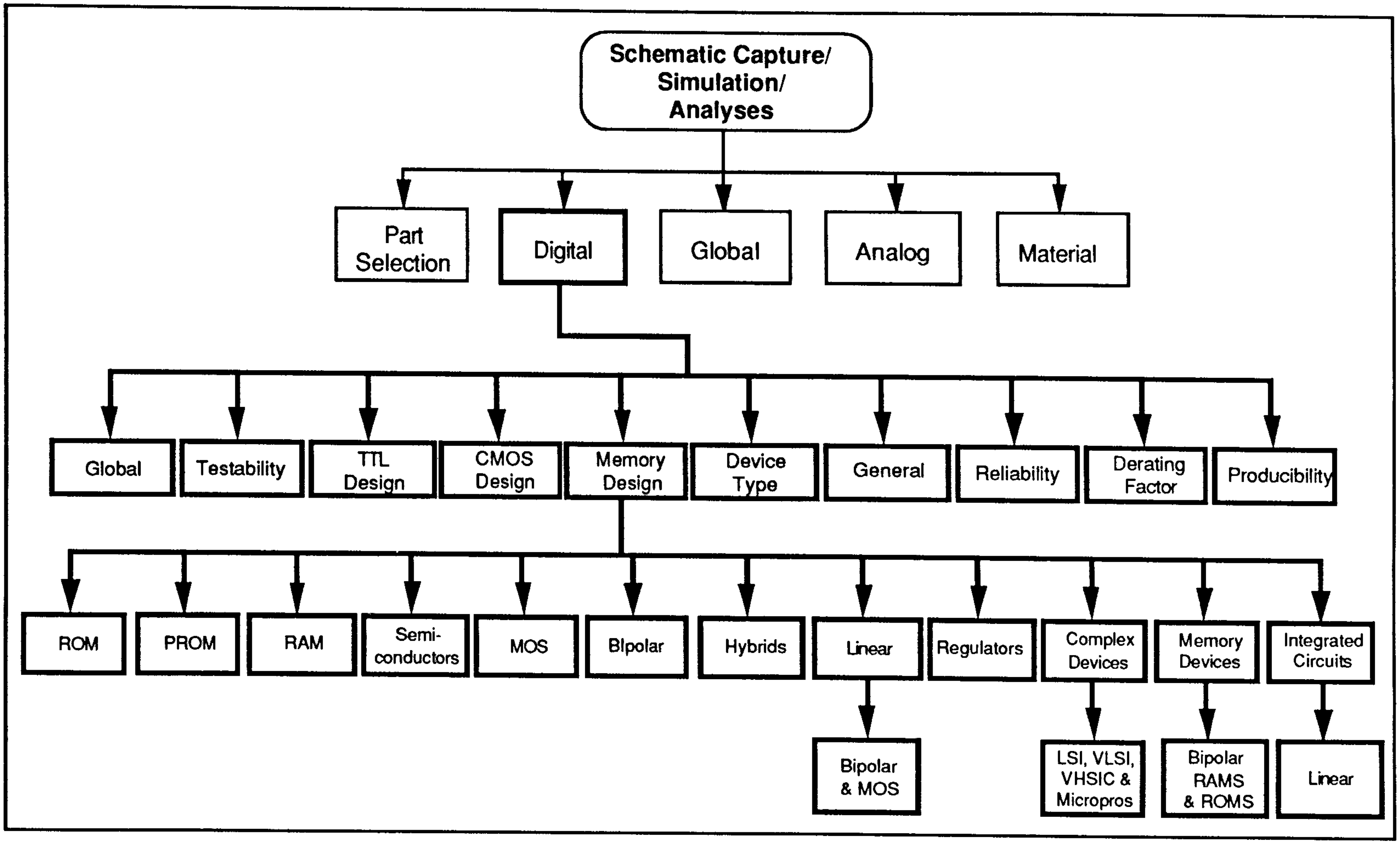


Figure A-3. Schematic Capture/Simulation/Analyses—Digital

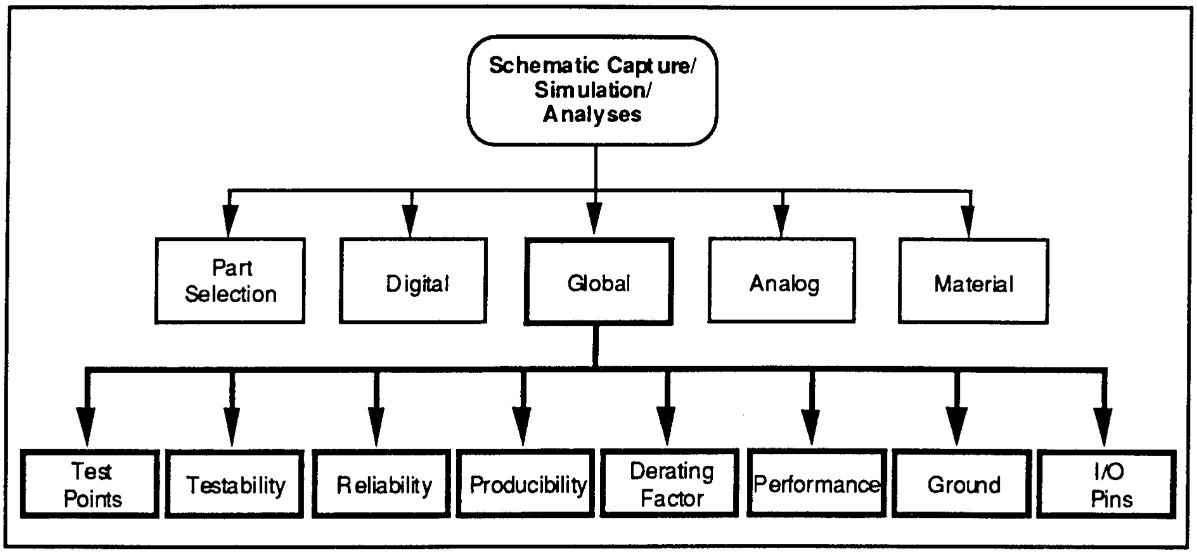


Figure D-4. Schematic Capture/Simulation/Analyses—Global

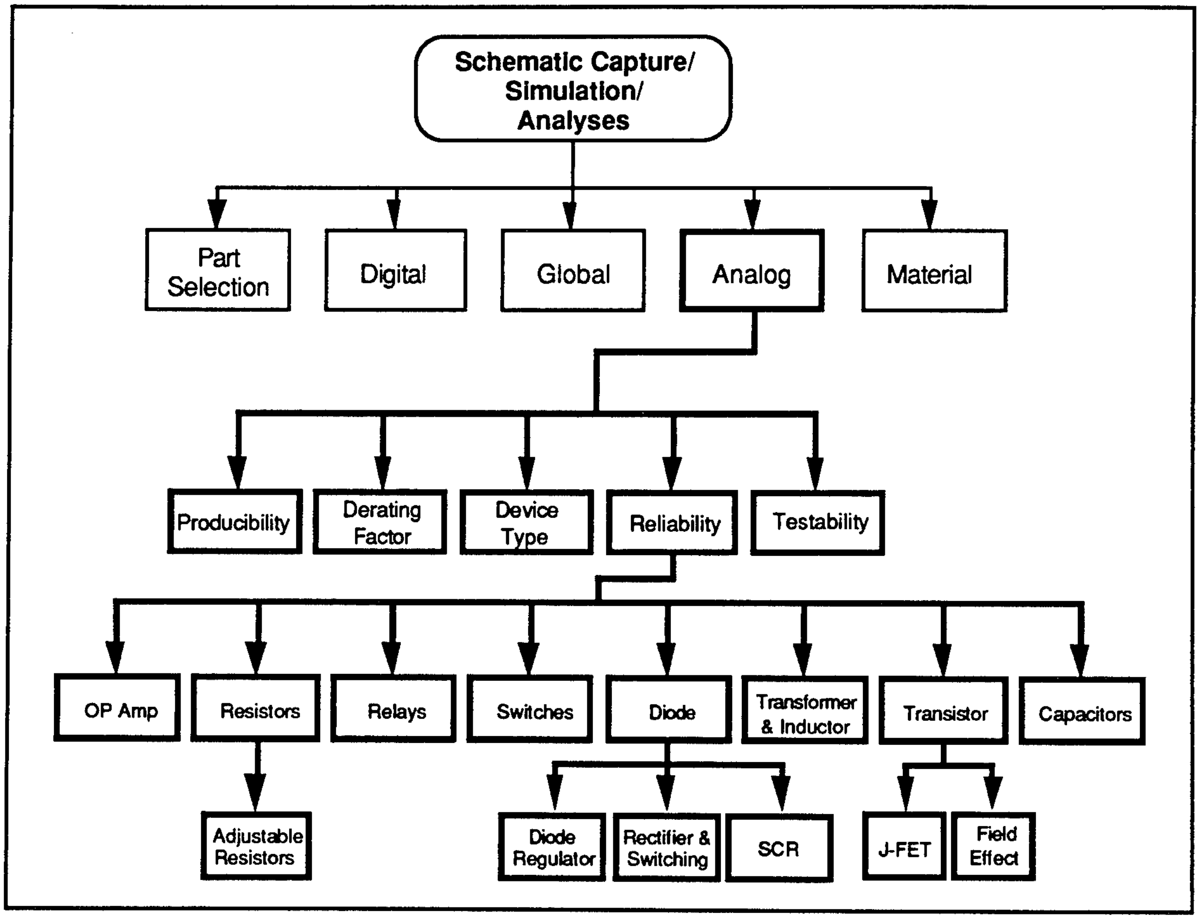


Figure D-5. Schematic Capture/Simulation/Analyses—Analog

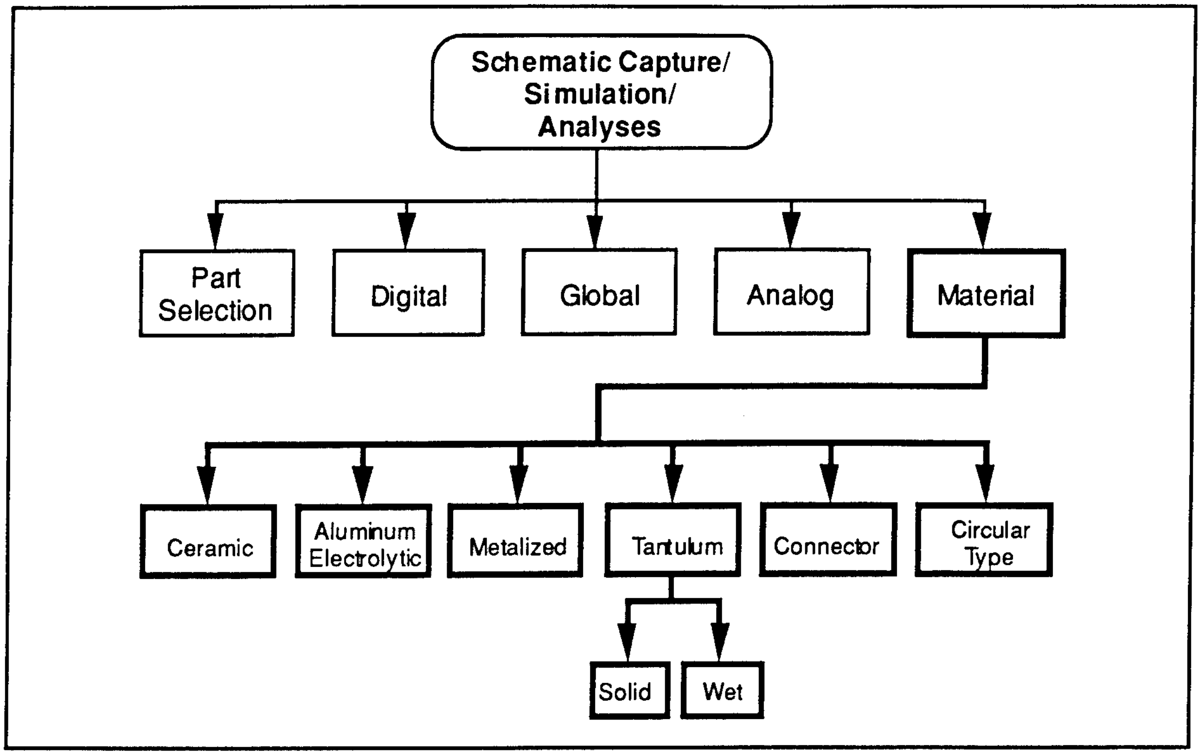
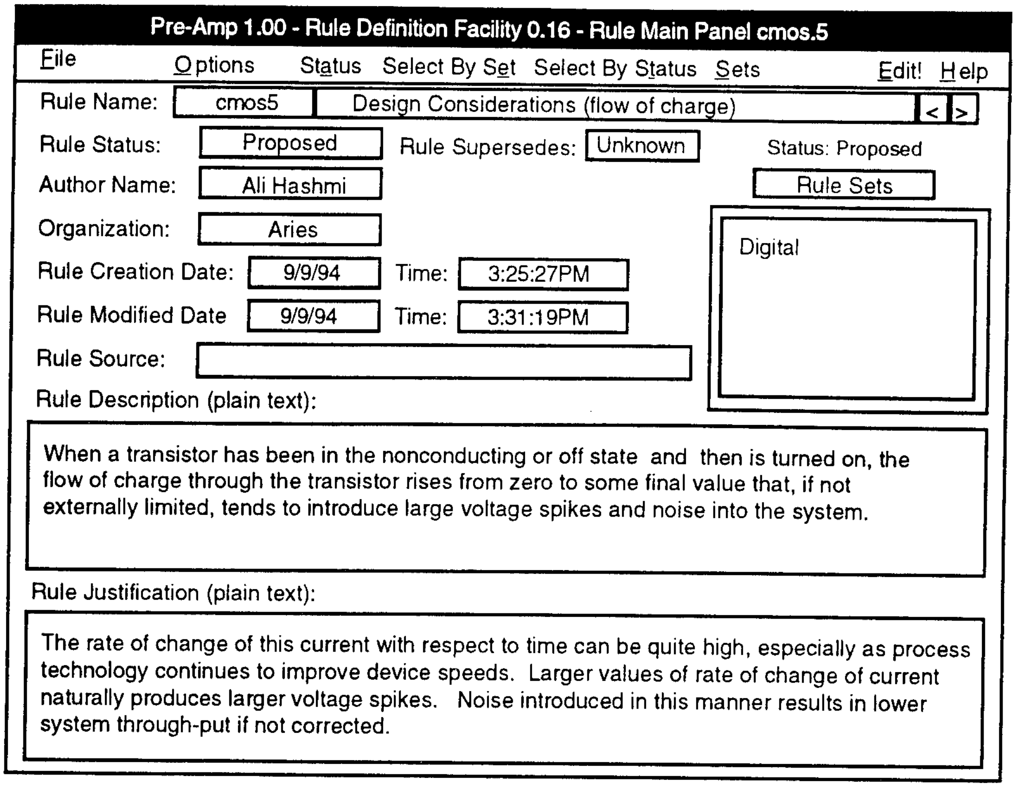


Figure D-6. Schematic Capture/Simulation/Analyses—Material

The following figures of *Pre*Amp RDF computer screen views provide examples of the rules collected by the Aries Project.



Pre-Amp 1.00 - Rule Definition Facility 0.16 - Rule Main Panel cmos.5

Rule Name: cmos5 Design Considerations (flow of charge)

Rule Status: Proposed Rule Supersedes: Unknown Status: Proposed

Author Name: Ali Hashmi

Organization: Aries

Rule Creation Date: 9/9/94 Time: 3:25:27PM

Rule Modified Date 9/9/94 Time: 3:31:19PM

Rule Source:

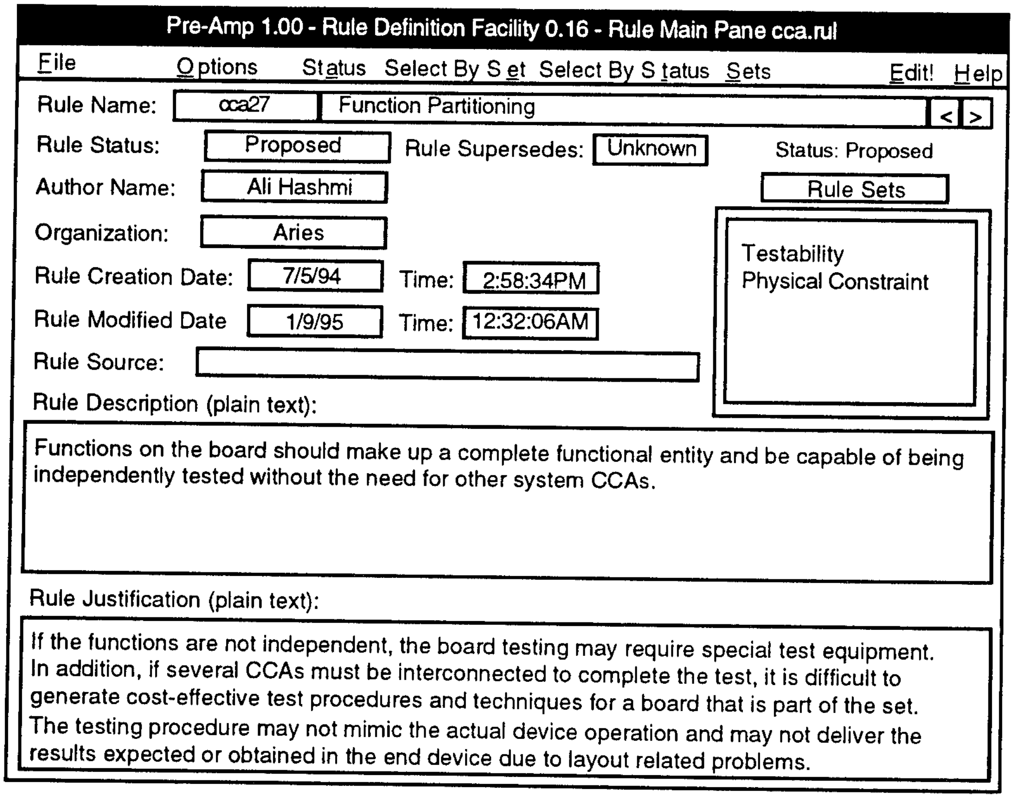
Digital

Rule Description (plain text):

When a transistor has been in the nonconducting or off state and then is turned on, the flow of charge through the transistor rises from zero to some final value that, if not externally limited, tends to introduce large voltage spikes and noise into the system.

Rule Justification (plain text):

The rate of change of this current with respect to time can be quite high, especially as process technology continues to improve device speeds. Larger values of rate of change of current naturally produces larger voltage spikes. Noise introduced in this manner results in lower system through-put if not corrected.



Pre-Amp 1.00 - Rule Definition Facility 0.16 - Rule Main Pane cca.rul

Rule Name: cca27 Function Partitioning

Rule Status: Proposed Rule Supersedes: Unknown Status: Proposed

Author Name: Ali Hashmi

Organization: Aries

Rule Creation Date: 7/5/94 Time: 2:58:34PM

Rule Modified Date 1/9/95 Time: 12:32:06AM

Rule Source:

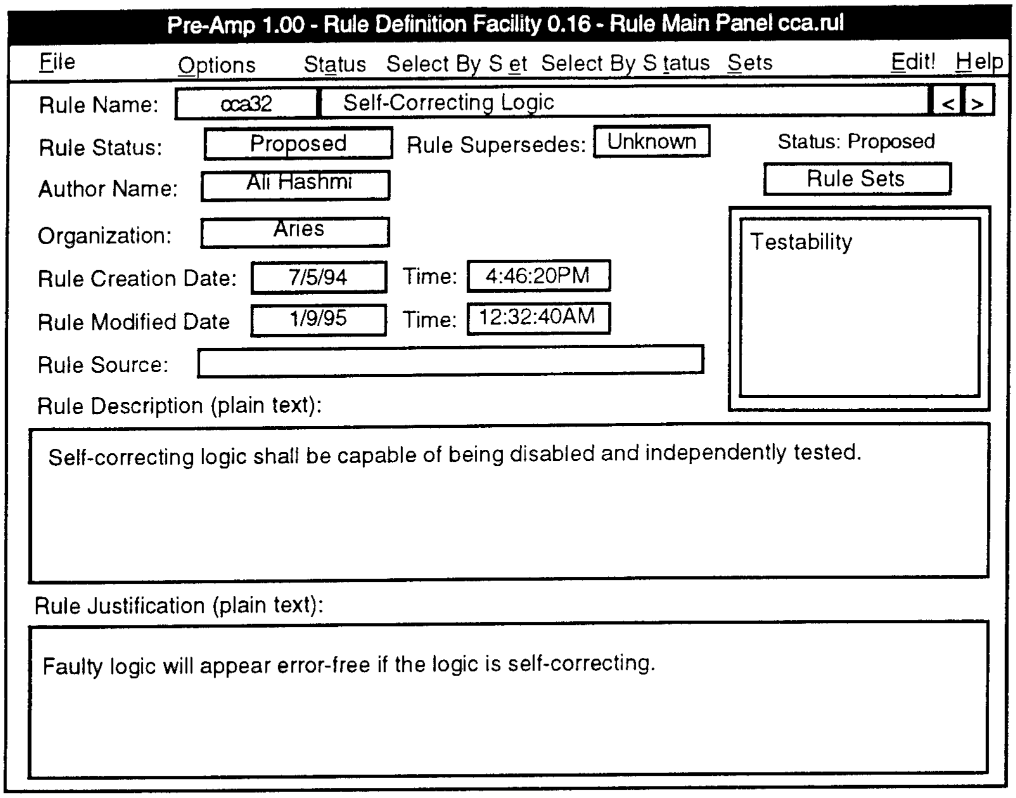
Testability
Physical Constraint

Rule Description (plain text):

Functions on the board should make up a complete functional entity and be capable of being independently tested without the need for other system CCAs.

Rule Justification (plain text):

If the functions are not independent, the board testing may require special test equipment. In addition, if several CCAs must be interconnected to complete the test, it is difficult to generate cost-effective test procedures and techniques for a board that is part of the set. The testing procedure may not mimic the actual device operation and may not deliver the results expected or obtained in the end device due to layout related problems.



Pre-Amp 1.00 - Rule Definition Facility 0.16 - Rule Main Panel cca.rul

Rule Name: cca32 Self-Correcting Logic

Rule Status: Proposed

Rule Supersedes: Unknown

Status: Proposed

Author Name: Ali Hashmi

Organization: Aries

Testability

Rule Creation Date: 7/5/94 Time: 4:46:20PM

Rule Modified Date 1/9/95 Time: 12:32:40AM

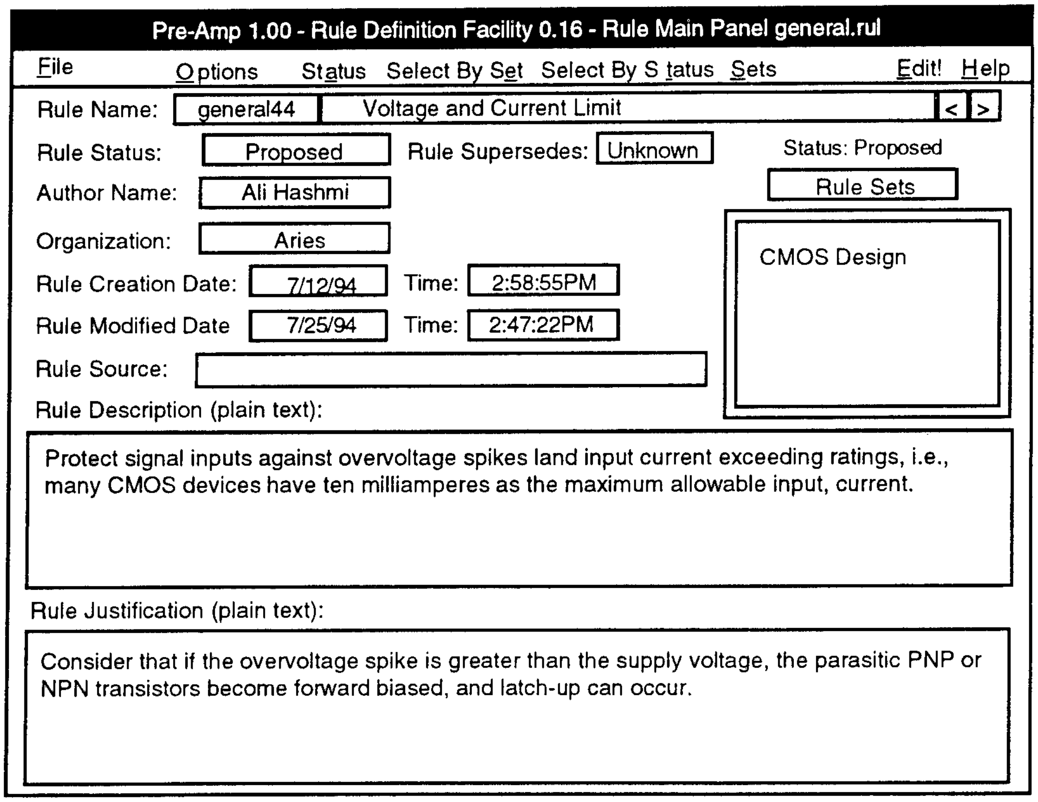
Rule Source:

Rule Description (plain text):

Self-correcting logic shall be capable of being disabled and independently tested.

Rule Justification (plain text):

Faulty logic will appear error-free if the logic is self-correcting.



Pre-Amp 1.00 - Rule Definition Facility 0.16 - Rule Main Panel general.rul

Rule Name: general44 Voltage and Current Limit

Rule Status: Proposed

Rule Supersedes: Unknown

Status: Proposed

Author Name: Ali Hashmi

Organization: Aries

CMOS Design

Rule Creation Date: 7/12/94 Time: 2:58:55PM

Rule Modified Date 7/25/94 Time: 2:47:22PM

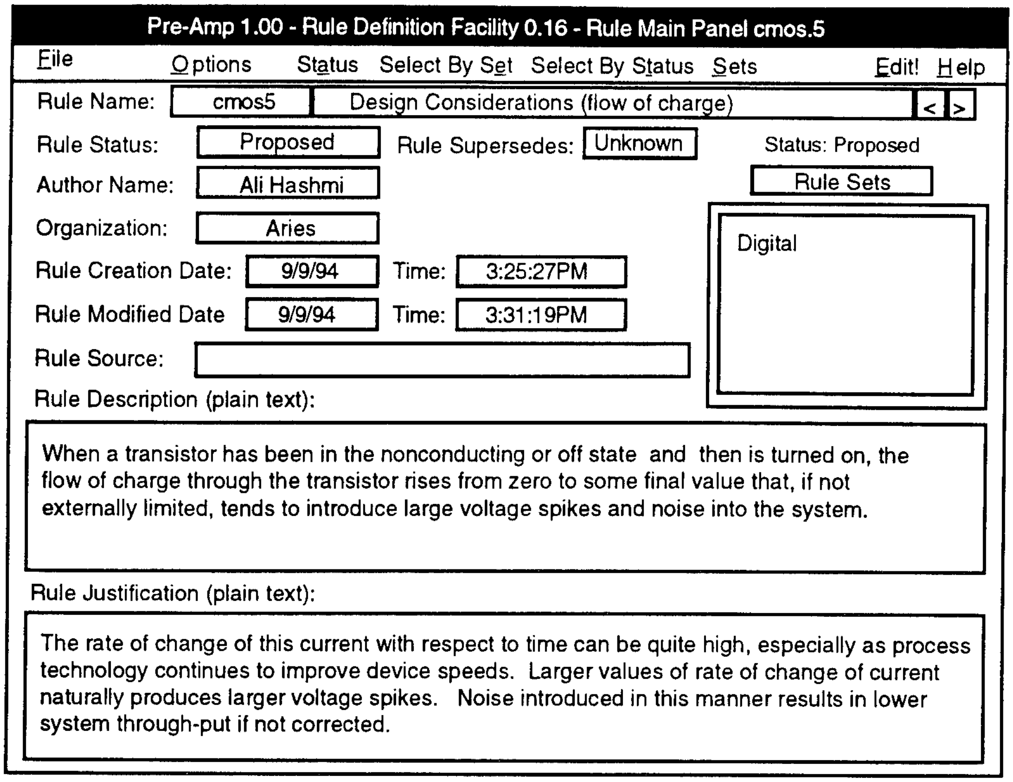
Rule Source:

Rule Description (plain text):

Protect signal inputs against overvoltage spikes land input current exceeding ratings, i.e., many CMOS devices have ten milliamperes as the maximum allowable input, current.

Rule Justification (plain text):

Consider that if the overvoltage spike is greater than the supply voltage, the parasitic PNP or NPN transistors become forward biased, and latch-up can occur.



Rule Name: cmos5 Design Considerations (flow of charge)

Rule Status: Proposed

Rule Supersedes: Unknown

Status: Proposed

Author Name: Ali Hashmi

Organization: Aries

Rule Sets: Digital

Rule Creation Date: 9/9/94 Time: 3:25:27PM

Rule Modified Date 9/9/94 Time: 3:31:19PM

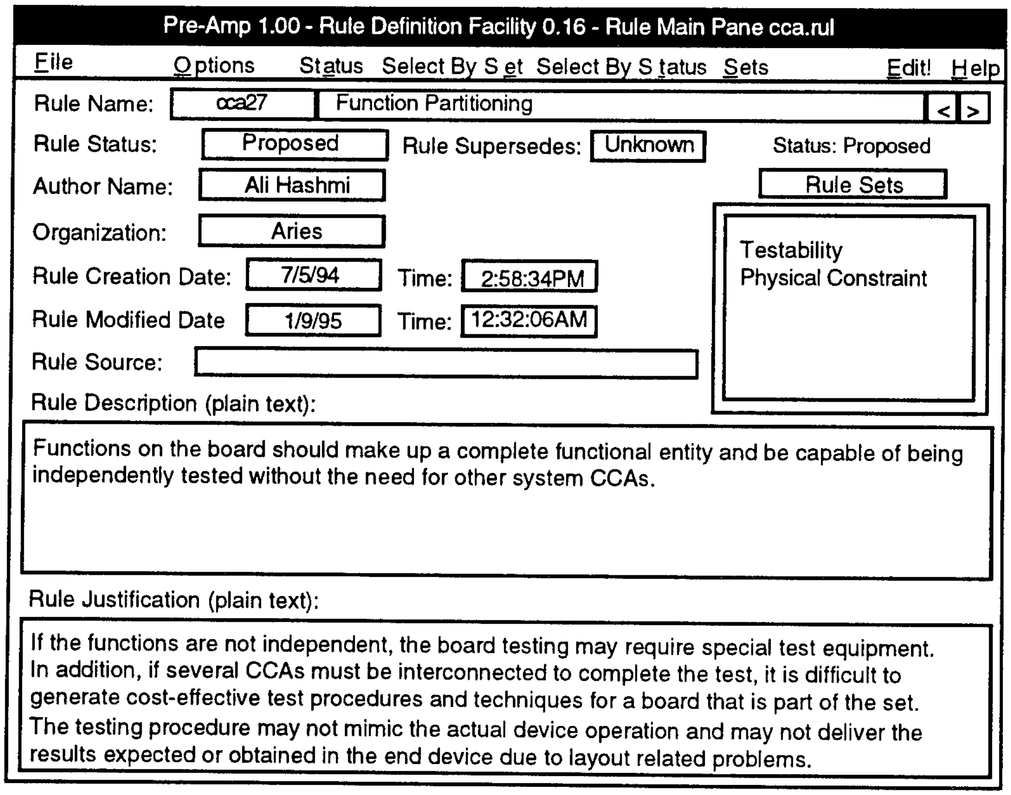
Rule Source:

Rule Description (plain text):

When a transistor has been in the nonconducting or off state and then is turned on, the flow of charge through the transistor rises from zero to some final value that, if not externally limited, tends to introduce large voltage spikes and noise into the system.

Rule Justification (plain text):

The rate of change of this current with respect to time can be quite high, especially as process technology continues to improve device speeds. Larger values of rate of change of current naturally produces larger voltage spikes. Noise introduced in this manner results in lower system through-put if not corrected.



Rule Name: cca27 Function Partitioning

Rule Status: Proposed

Rule Supersedes: Unknown

Status: Proposed

Author Name: Ali Hashmi

Organization: Aries

Rule Sets: Testability
Physical Constraint

Rule Creation Date: 7/5/94 Time: 2:58:34PM

Rule Modified Date 1/9/95 Time: 12:32:06AM

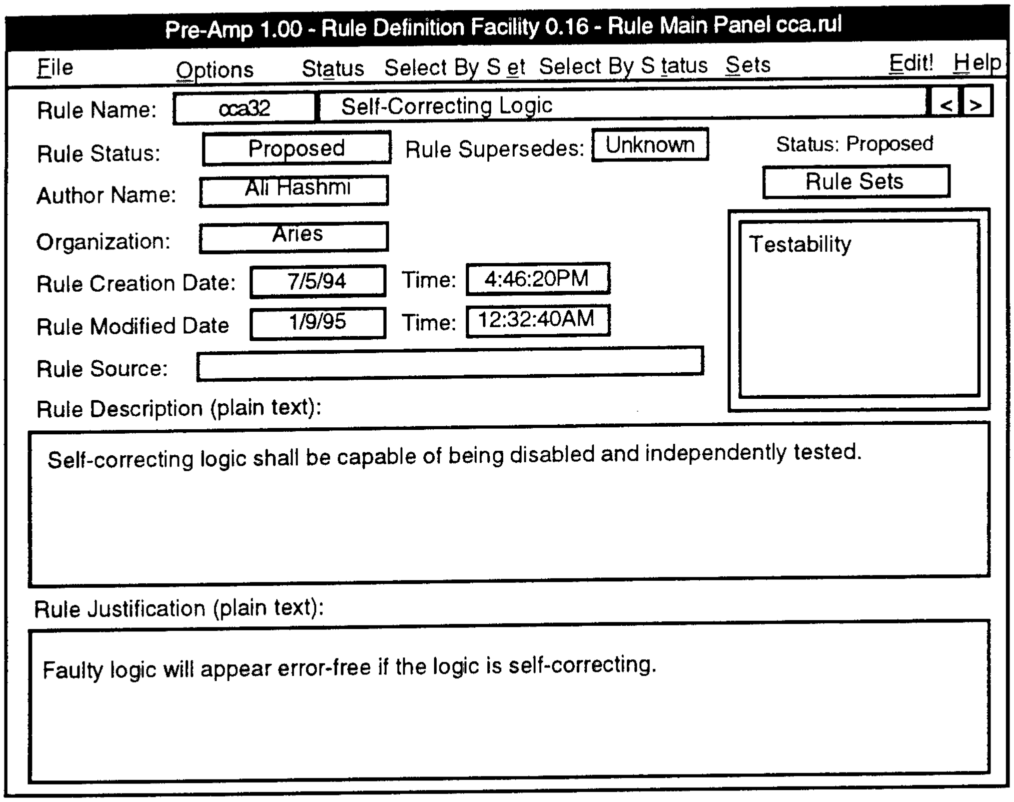
Rule Source:

Rule Description (plain text):

Functions on the board should make up a complete functional entity and be capable of being independently tested without the need for other system CCAs.

Rule Justification (plain text):

If the functions are not independent, the board testing may require special test equipment. In addition, if several CCAs must be interconnected to complete the test, it is difficult to generate cost-effective test procedures and techniques for a board that is part of the set. The testing procedure may not mimic the actual device operation and may not deliver the results expected or obtained in the end device due to layout related problems.



Rule Name: cca32 Self-Correcting Logic

Rule Status: Proposed

Rule Supersedes: Unknown

Status: Proposed

Author Name: Ali Hashmi

Organization: Aries

Rule Creation Date: 7/5/94 Time: 4:46:20PM

Rule Modified Date 1/9/95 Time: 12:32:40AM

Rule Source:

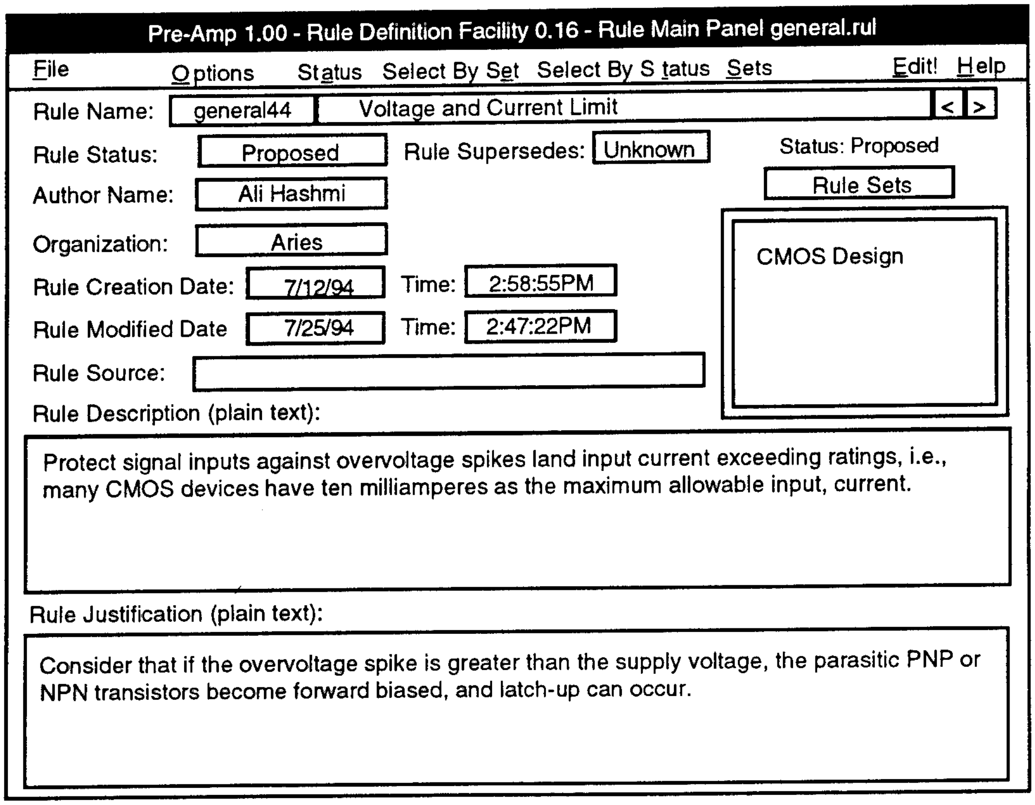
Testability

Rule Description (plain text):

Self-correcting logic shall be capable of being disabled and independently tested.

Rule Justification (plain text):

Faulty logic will appear error-free if the logic is self-correcting.


Pre-Amp 1.00 - Rule Definition Facility 0.16 - Rule Main Panel general.rul
File
Options
Status
Select By Set
Select By S tatus
Sets
Edit!
Help
<
>
Rule Sets

Rule Name: general44 Voltage and Current Limit

Rule Status: Proposed

Rule Supersedes: Unknown

Status: Proposed

Author Name: Ali Hashmi

Organization: Aries

Rule Creation Date: 7/12/94 Time: 2:58:55PM

Rule Modified Date 7/25/94 Time: 2:47:22PM

Rule Source:

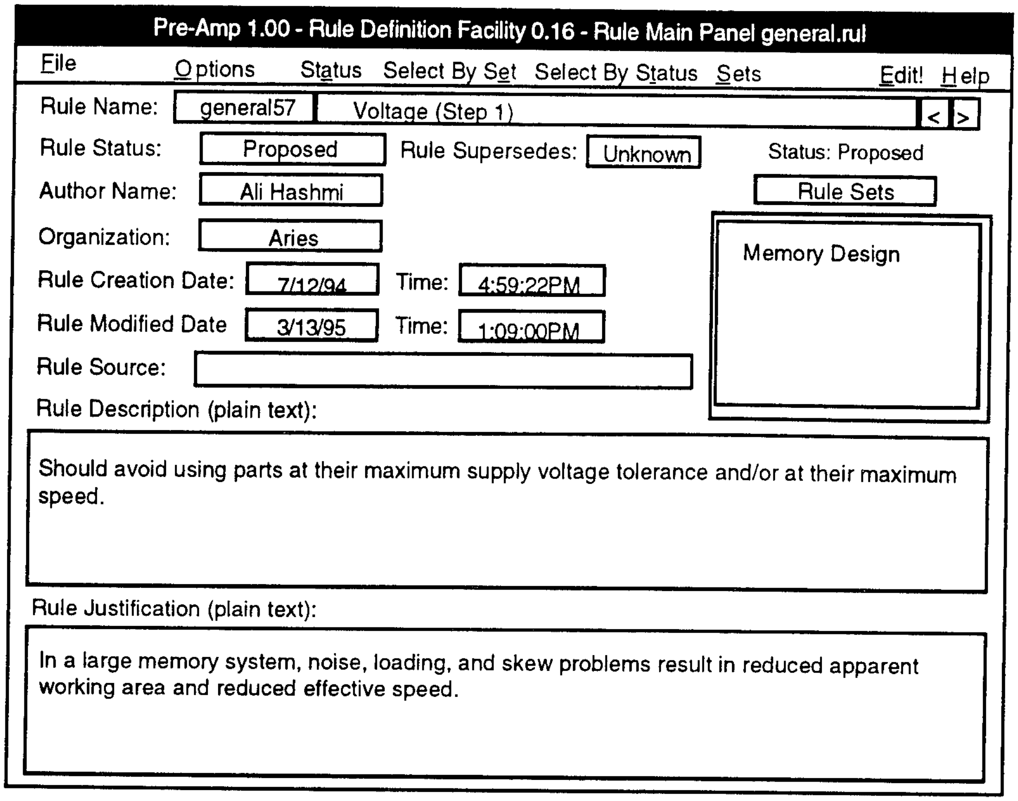
CMOS Design

Rule Description (plain text):

Protect signal inputs against overvoltage spikes land input current exceeding ratings, i.e., many CMOS devices have ten milliamperes as the maximum allowable input, current.

Rule Justification (plain text):

Consider that if the overvoltage spike is greater than the supply voltage, the parasitic PNP or NPN transistors become forward biased, and latch-up can occur.



Pre-Amp 1.00 - Rule Definition Facility 0.16 - Rule Main Panel general.rul
File
Options
Status
Select By Set
Select By Status
Sets
Edit!
Help
<
>
Rule Sets

Rule Name: general57 Voltage (Step 1)

Rule Status: Proposed

Rule Supersedes: Unknown

Status: Proposed

Author Name: Ali Hashmi

Organization: Aries

Memory Design

Rule Creation Date: 7/12/94 Time: 4:59:22PM

Rule Modified Date 3/13/95 Time: 1:09:00PM

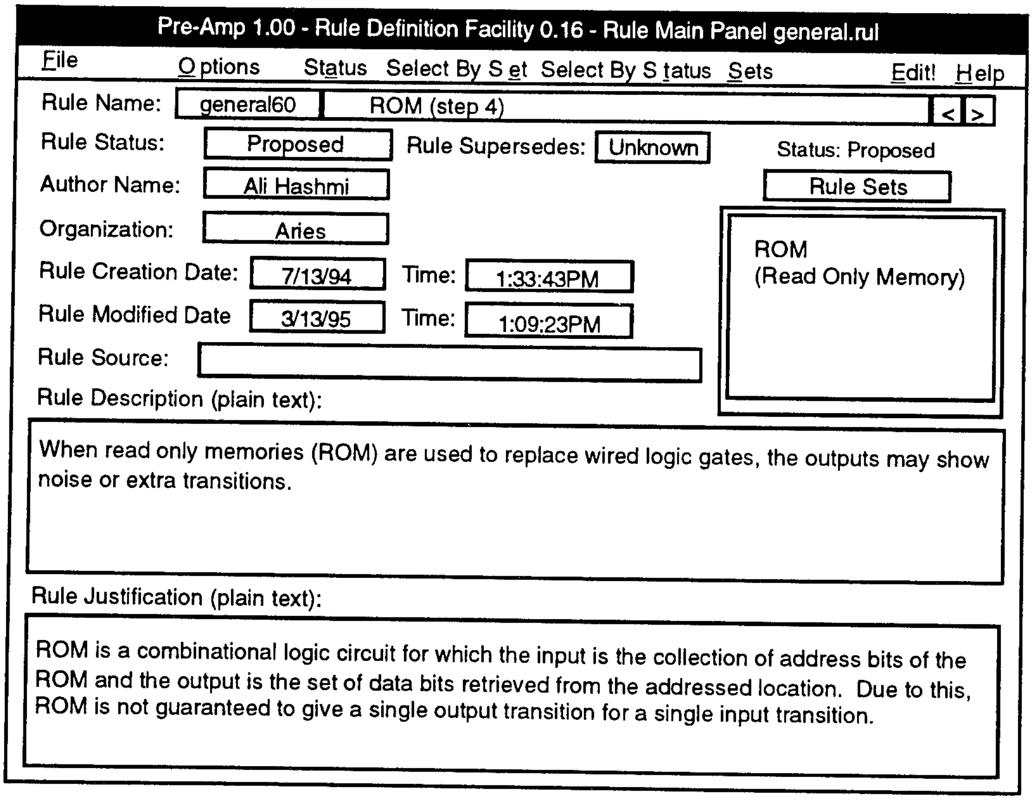
Rule Source:

Rule Description (plain text):

Should avoid using parts at their maximum supply voltage tolerance and/or at their maximum speed.

Rule Justification (plain text):

In a large memory system, noise, loading, and skew problems result in reduced apparent working area and reduced effective speed.



Pre-Amp 1.00 - Rule Definition Facility 0.16 - Rule Main Panel general.rul
File
Options
Status
Select By Set
Select By Status
Sets
Edit!
Help
<
>
Rule Sets

Rule Name: general60 ROM (step 4)

Rule Status: Proposed

Rule Supersedes: Unknown

Status: Proposed

Author Name: Ali Hashmi

Organization: Aries

ROM
(Read Only Memory)

Rule Creation Date: 7/13/94 Time: 1:33:43PM

Rule Modified Date 3/13/95 Time: 1:09:23PM

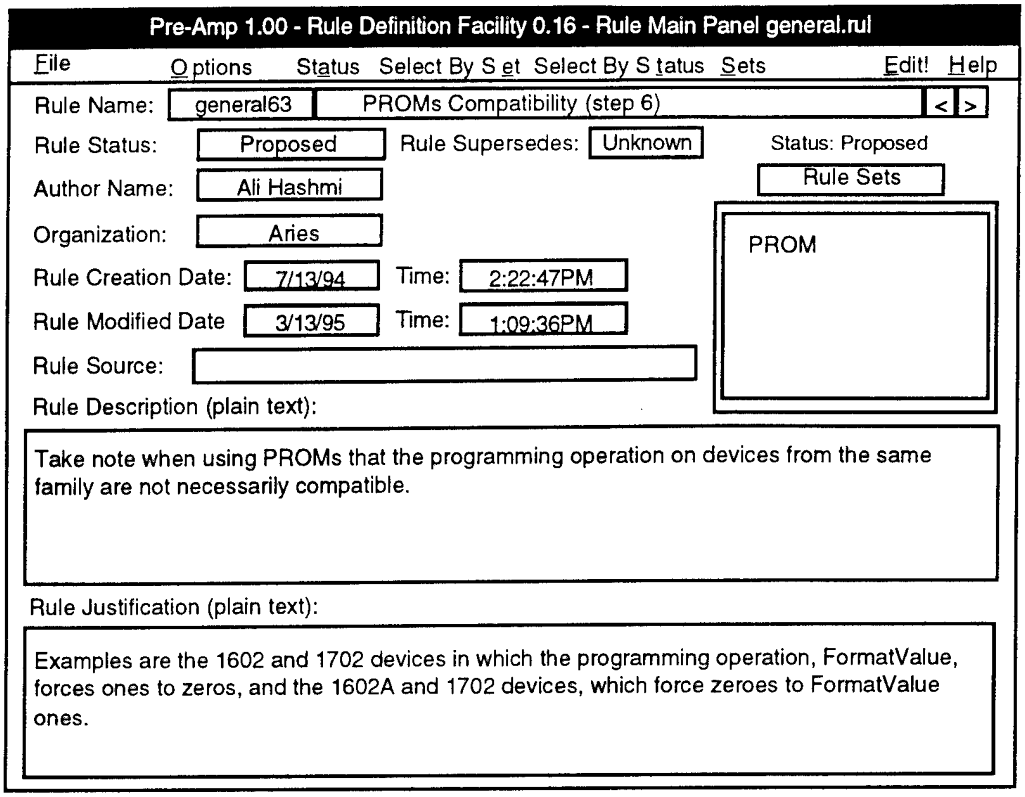
Rule Source:

Rule Description (plain text):

When read only memories (ROM) are used to replace wired logic gates, the outputs may show noise or extra transitions.

Rule Justification (plain text):

ROM is a combinational logic circuit for which the input is the collection of address bits of the ROM and the output is the set of data bits retrieved from the addressed location. Due to this, ROM is not guaranteed to give a single output transition for a single input transition.



Pre-Amp 1.00 - Rule Definition Facility 0.16 - Rule Main Panel general.rul
File
Options
Status
Select By Set
Select By Status
Sets
Edit!
Help
<
>
Rule Sets

Rule Name: general63 PROMs Compatibility (step 6)

Rule Status: Proposed Rule Supersedes: Unknown Status: Proposed

Author Name: Ali Hashmi

Organization: Aries

Rule Creation Date: 7/13/94 Time: 2:22:47PM

Rule Modified Date 3/13/95 Time: 1:09:36PM

Rule Source:

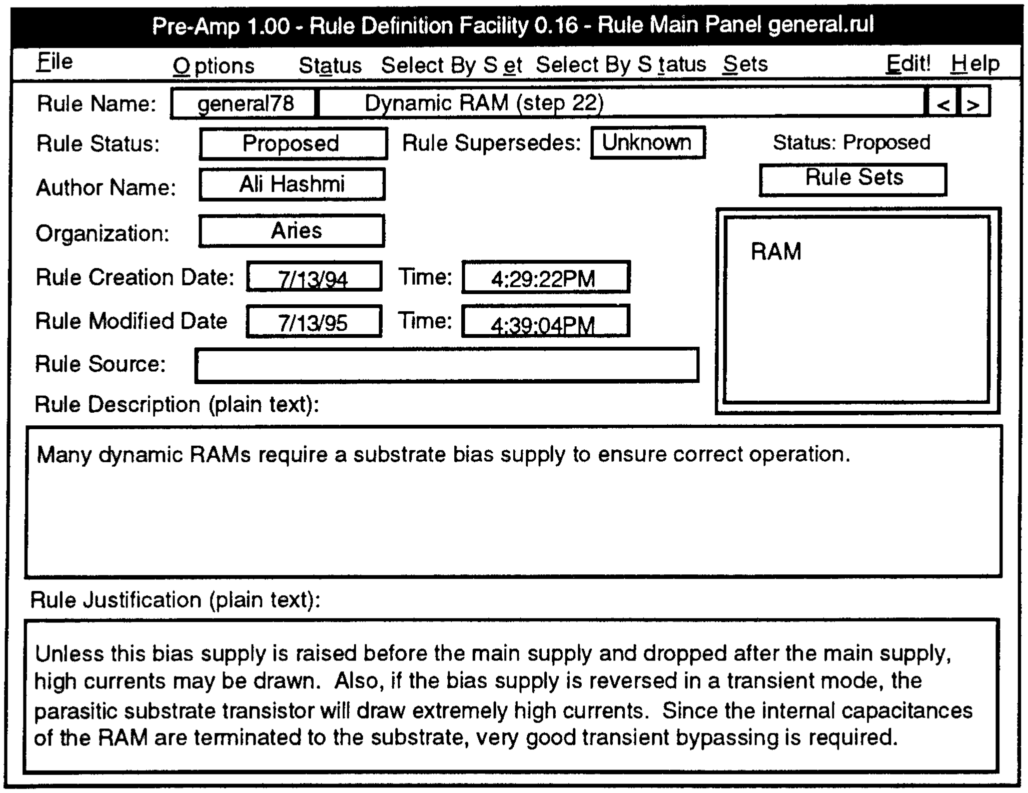
Rule Sets: PROM

Rule Description (plain text):

Take note when using PROMs that the programming operation on devices from the same family are not necessarily compatible.

Rule Justification (plain text):

Examples are the 1602 and 1702 devices in which the programming operation, FormatValue, forces ones to zeros, and the 1602A and 1702 devices, which force zeroes to FormatValue ones.



Pre-Amp 1.00 - Rule Definition Facility 0.16 - Rule Main Panel general.rul
File
Options
Status
Select By Set
Select By Status
Sets
Edit!
Help
<
>
Rule Sets

Rule Name: general78 Dynamic RAM (step 22)

Rule Status: Proposed Rule Supersedes: Unknown Status: Proposed

Author Name: Ali Hashmi

Organization: Aries

Rule Creation Date: 7/13/94 Time: 4:29:22PM

Rule Modified Date 7/13/95 Time: 4:39:04PM

Rule Source:

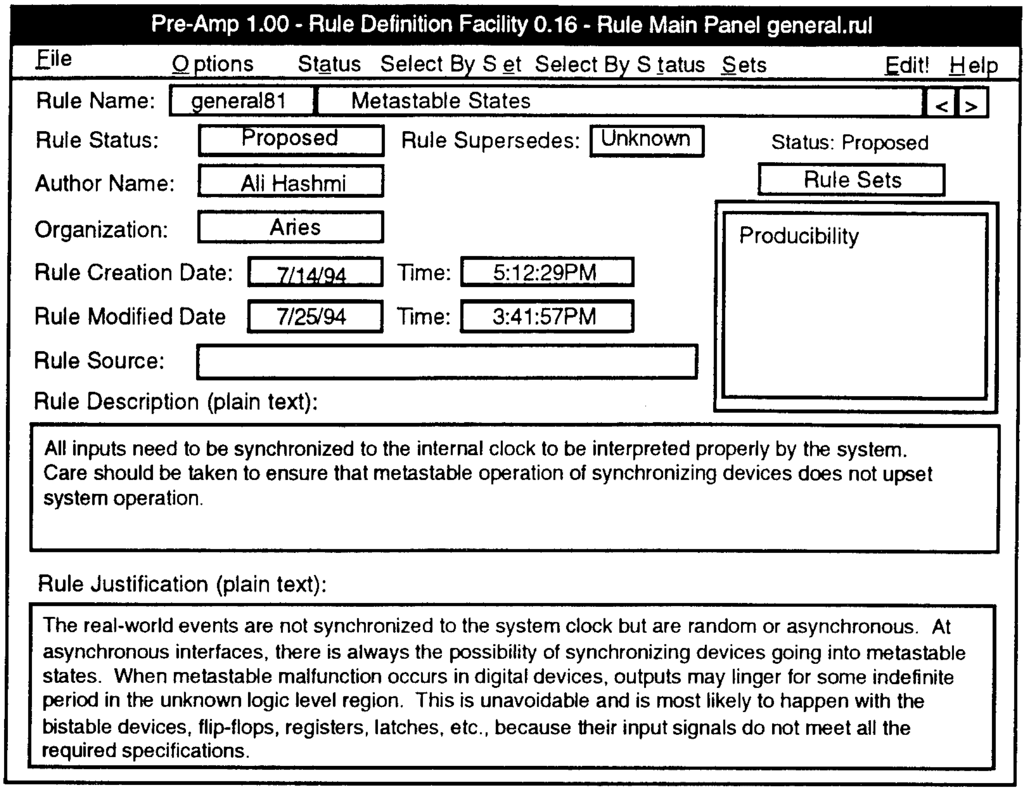
Rule Sets: RAM

Rule Description (plain text):

Many dynamic RAMs require a substrate bias supply to ensure correct operation.

Rule Justification (plain text):

Unless this bias supply is raised before the main supply and dropped after the main supply, high currents may be drawn. Also, if the bias supply is reversed in a transient mode, the parasitic substrate transistor will draw extremely high currents. Since the internal capacitances of the RAM are terminated to the substrate, very good transient bypassing is required.


Pre-Amp 1.00 - Rule Definition Facility 0.16 - Rule Main Panel general.rul
File
Options
Status
Select By Set
Select By Status
Sets
Edit!
Help
<
>
Rule Sets

Rule Name: general81 Metastable States

Rule Status: Proposed

Rule Supersedes: Unknown

Status: Proposed

Author Name: Ali Hashmi

Organization: Aries

Producibility

Rule Creation Date: 7/14/94 Time: 5:12:29PM

Rule Modified Date 7/25/94 Time: 3:41:57PM

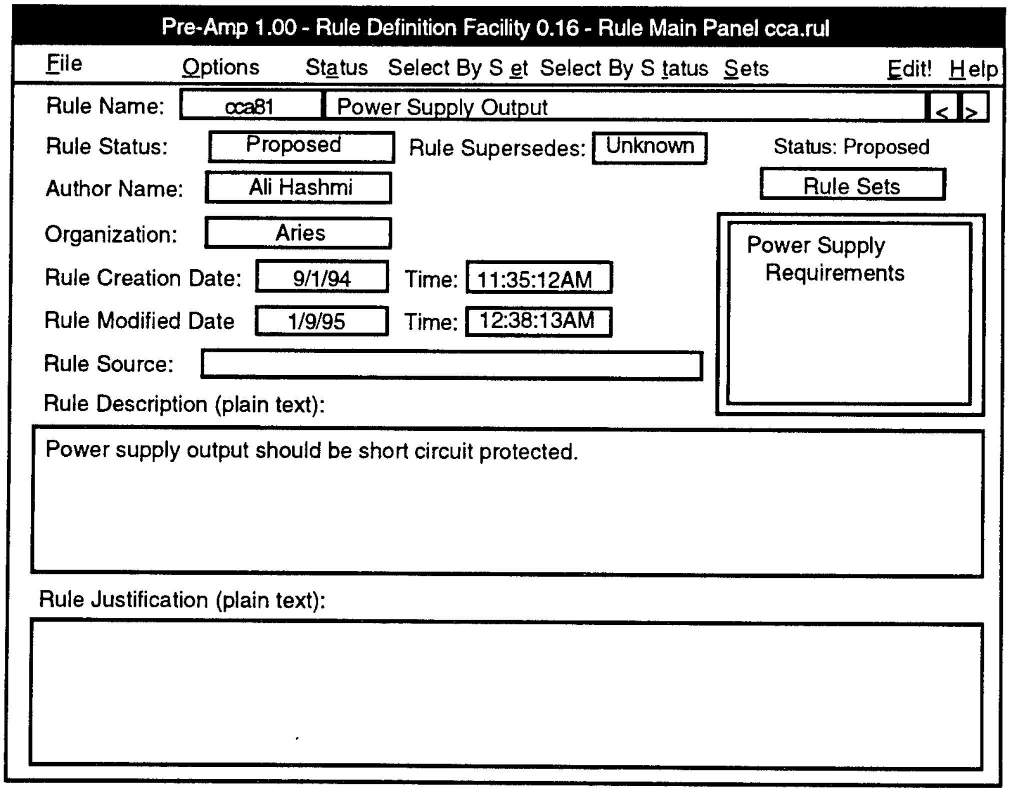
Rule Source:

Rule Description (plain text):

All inputs need to be synchronized to the internal clock to be interpreted properly by the system. Care should be taken to ensure that metastable operation of synchronizing devices does not upset system operation.

Rule Justification (plain text):

The real-world events are not synchronized to the system clock but are random or asynchronous. At asynchronous interfaces, there is always the possibility of synchronizing devices going into metastable states. When metastable malfunction occurs in digital devices, outputs may linger for some indefinite period in the unknown logic level region. This is unavoidable and is most likely to happen with the bistable devices, flip-flops, registers, latches, etc., because their input signals do not meet all the required specifications.


Pre-Amp 1.00 - Rule Definition Facility 0.16 - Rule Main Panel cca.rul
File
Options
Status
Select By Set
Select By Status
Sets
Edit!
Help
<
>
Rule Sets

Rule Name: cca81 Power Supply Output

Rule Status: Proposed

Rule Supersedes: Unknown

Status: Proposed

Author Name: Ali Hashmi

Organization: Aries

Power Supply Requirements

Rule Creation Date: 9/1/94 Time: 11:35:12AM

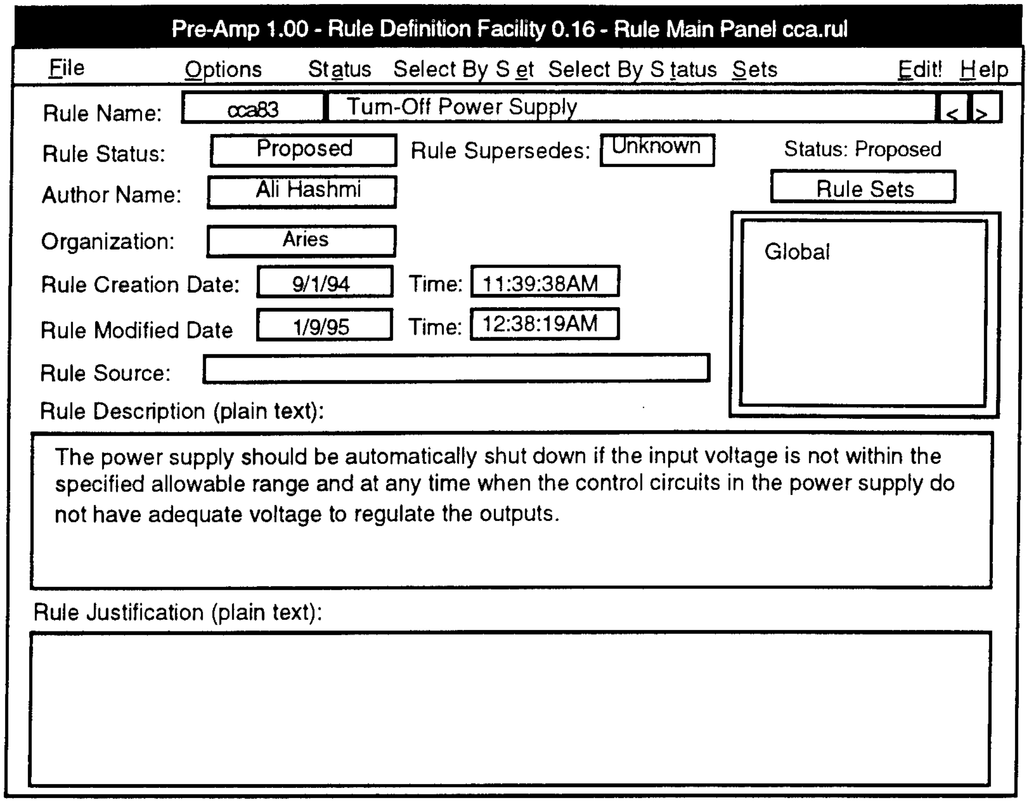
Rule Modified Date 1/9/95 Time: 12:38:13AM

Rule Source:

Rule Description (plain text):

Power supply output should be short circuit protected.

Rule Justification (plain text):



Pre-Amp 1.00 - Rule Definition Facility 0.16 - Rule Main Panel cca.rul

Rule Name: cca83 Turn-Off Power Supply

Rule Status: Proposed Rule Supersedes: Unknown Status: Proposed

Author Name: Ali Hashmi

Organization: Aries

Rule Creation Date: 9/1/94 Time: 11:39:38AM

Rule Modified Date 1/9/95 Time: 12:38:19AM

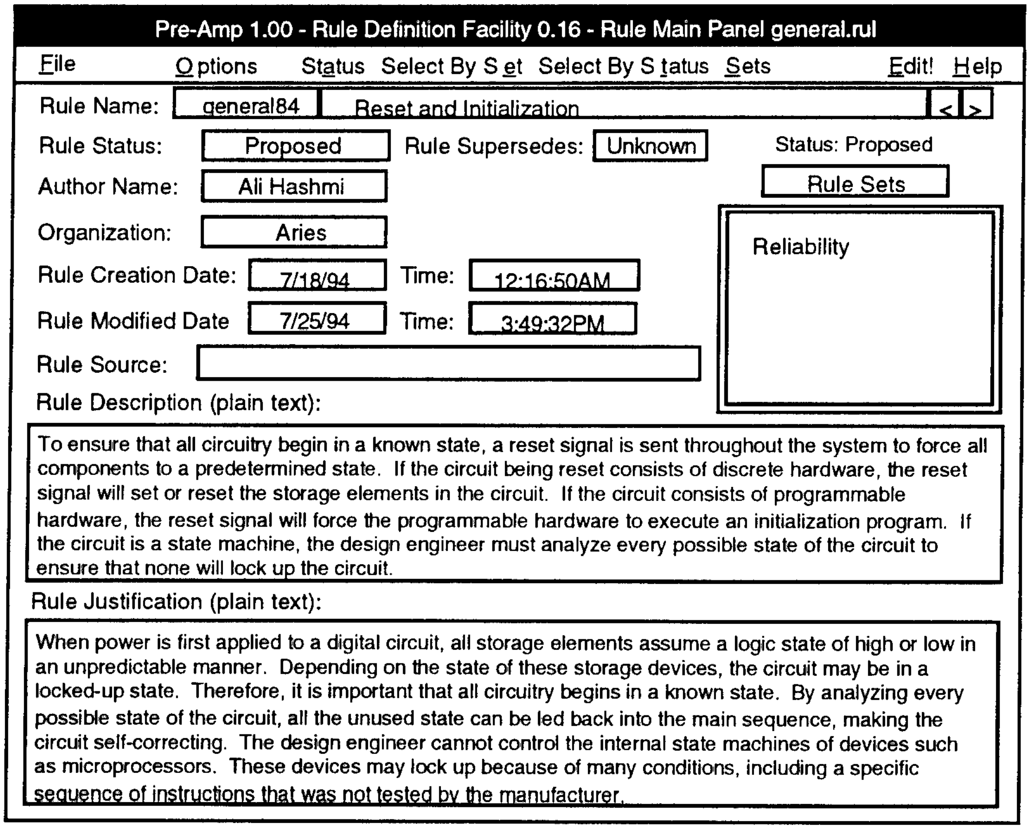
Rule Source:

Rule Sets: Global

Rule Description (plain text):

The power supply should be automatically shut down if the input voltage is not within the specified allowable range and at any time when the control circuits in the power supply do not have adequate voltage to regulate the outputs.

Rule Justification (plain text):



Pre-Amp 1.00 - Rule Definition Facility 0.16 - Rule Main Panel general.rul

Rule Name: general84 Reset and Initialization

Rule Status: Proposed Rule Supersedes: Unknown Status: Proposed

Author Name: Ali Hashmi

Organization: Aries

Rule Creation Date: 7/18/94 Time: 12:16:50AM

Rule Modified Date 7/25/94 Time: 3:49:32PM

Rule Source:

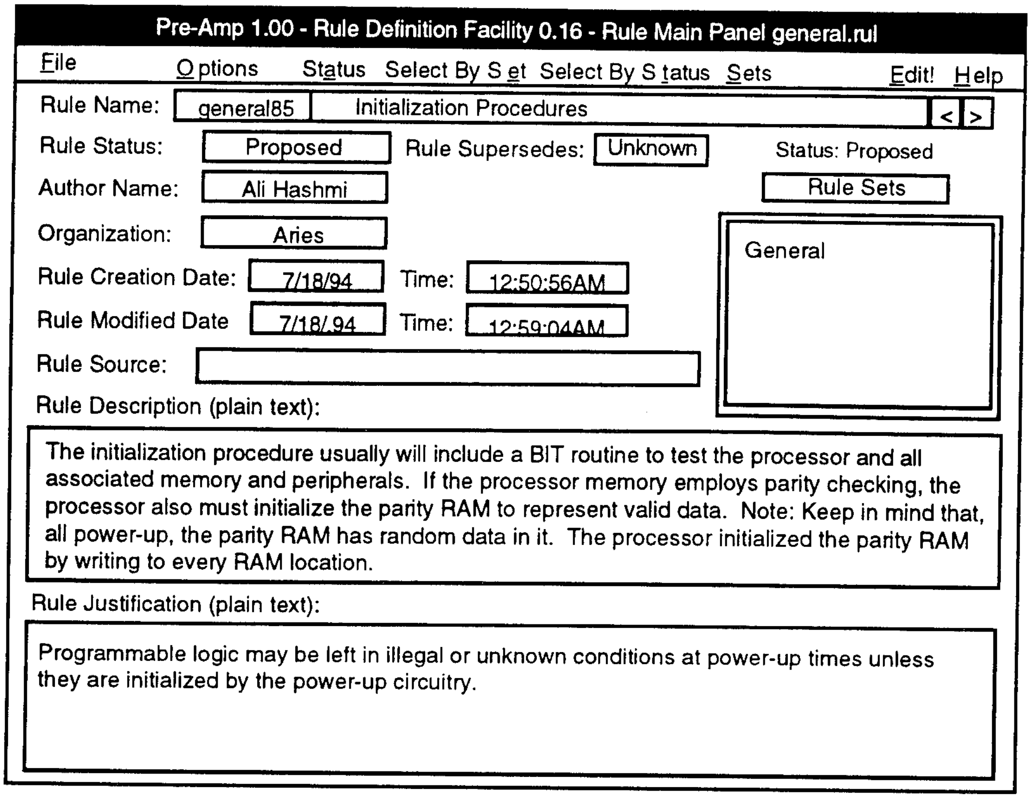
Rule Sets: Reliability

Rule Description (plain text):

To ensure that all circuitry begin in a known state, a reset signal is sent throughout the system to force all components to a predetermined state. If the circuit being reset consists of discrete hardware, the reset signal will set or reset the storage elements in the circuit. If the circuit consists of programmable hardware, the reset signal will force the programmable hardware to execute an initialization program. If the circuit is a state machine, the design engineer must analyze every possible state of the circuit to ensure that none will lock up the circuit.

Rule Justification (plain text):

When power is first applied to a digital circuit, all storage elements assume a logic state of high or low in an unpredictable manner. Depending on the state of these storage devices, the circuit may be in a locked-up state. Therefore, it is important that all circuitry begins in a known state. By analyzing every possible state of the circuit, all the unused state can be led back into the main sequence, making the circuit self-correcting. The design engineer cannot control the internal state machines of devices such as microprocessors. These devices may lock up because of many conditions, including a specific sequence of instructions that was not tested by the manufacturer.


Pre-Amp 1.00 - Rule Definition Facility 0.16 - Rule Main Panel general.rul
File
Options
Status
Select By S et
Select By S tatus
Sets
Edit!
Help
<
>
Rule Sets

Rule Name: general85 Initialization Procedures

Rule Status: Proposed

Rule Supersedes: Unknown

Status: Proposed

Author Name: Ali Hashmi

Organization: Aries

General

Rule Creation Date: 7/18/94 Time: 12:50:56AM

Rule Modified Date 7/18/94 Time: 12:59:04AM

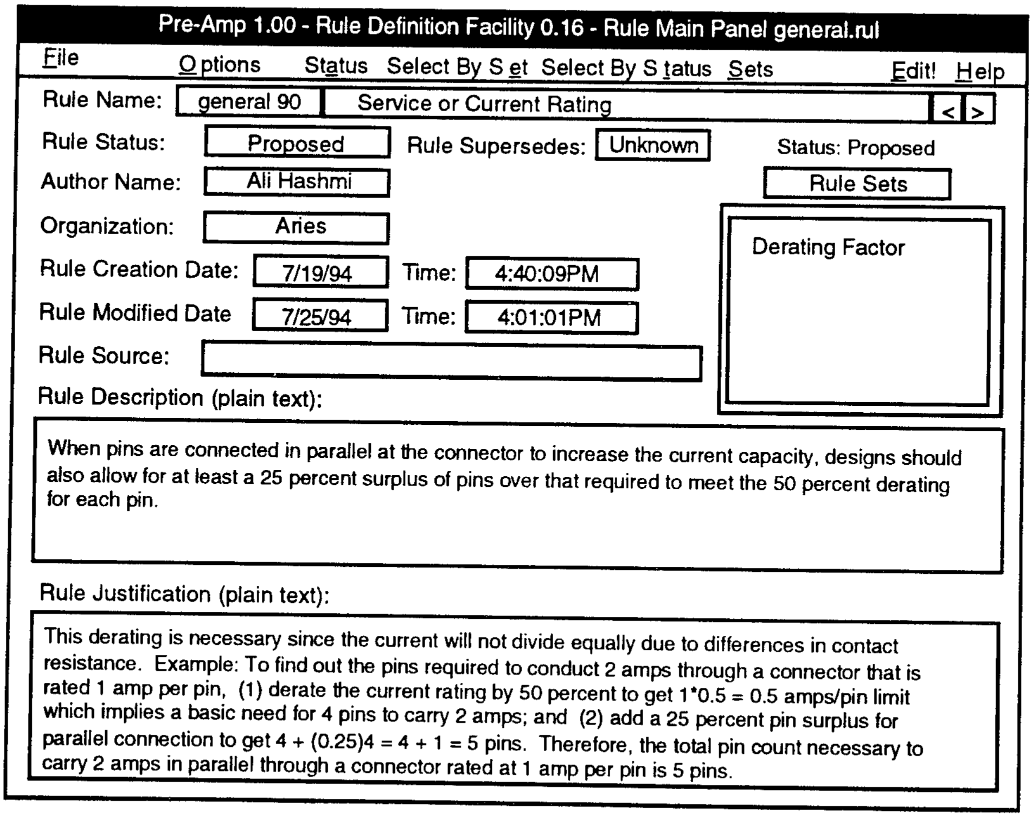
Rule Source:

Rule Description (plain text):

The initialization procedure usually will include a BIT routine to test the processor and all associated memory and peripherals. If the processor memory employs parity checking, the processor also must initialize the parity RAM to represent valid data. Note: Keep in mind that, all power-up, the parity RAM has random data in it. The processor initialized the parity RAM by writing to every RAM location.

Rule Justification (plain text):

Programmable logic may be left in illegal or unknown conditions at power-up times unless they are initialized by the power-up circuitry.


Pre-Amp 1.00 - Rule Definition Facility 0.16 - Rule Main Panel general.rul
File
Options
Status
Select By S et
Select By S tatus
Sets
Edit!
Help
<
>
Rule Sets

Rule Name: general 90 Service or Current Rating

Rule Status: Proposed

Rule Supersedes: Unknown

Status: Proposed

Author Name: Ali Hashmi

Organization: Aries

Derating Factor

Rule Creation Date: 7/19/94 Time: 4:40:09PM

Rule Modified Date 7/25/94 Time: 4:01:01PM

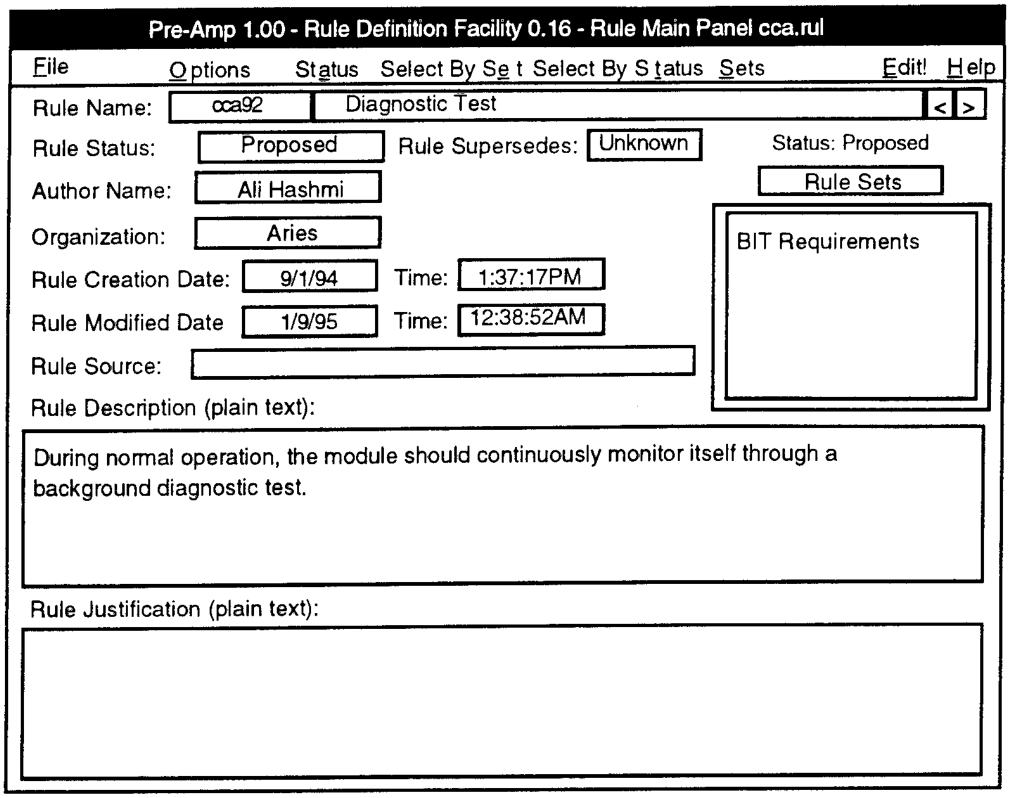
Rule Source:

Rule Description (plain text):

When pins are connected in parallel at the connector to increase the current capacity, designs should also allow for at least a 25 percent surplus of pins over that required to meet the 50 percent derating for each pin.

Rule Justification (plain text):

This derating is necessary since the current will not divide equally due to differences in contact resistance. Example: To find out the pins required to conduct 2 amps through a connector that is rated 1 amp per pin, (1) derate the current rating by 50 percent to get 1*0.5 = 0.5 amps/pin limit which implies a basic need for 4 pins to carry 2 amps; and (2) add a 25 percent pin surplus for parallel connection to get 4 + (0.25)4 = 4 + 1 = 5 pins. Therefore, the total pin count necessary to carry 2 amps in parallel through a connector rated at 1 amp per pin is 5 pins.



Pre-Amp 1.00 - Rule Definition Facility 0.16 - Rule Main Panel cca.rul

Rule Name: cca92 Diagnostic Test

Rule Status: Proposed Rule Supersedes: Unknown Status: Proposed

Author Name: Ali Hashmi

Organization: Aries

BIT Requirements

Rule Creation Date: 9/1/94 Time: 1:37:17PM

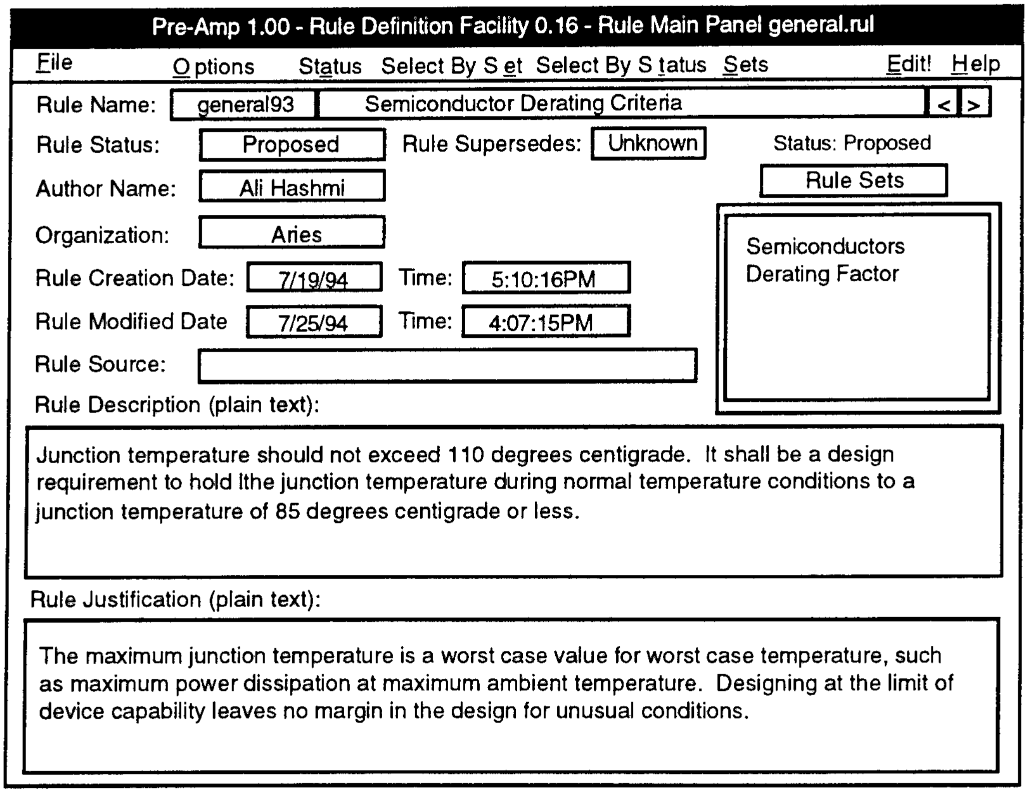
Rule Modified Date 1/9/95 Time: 12:38:52AM

Rule Source:

Rule Description (plain text):

During normal operation, the module should continuously monitor itself through a background diagnostic test.

Rule Justification (plain text):



Pre-Amp 1.00 - Rule Definition Facility 0.16 - Rule Main Panel general.rul

Rule Name: general93 Semiconductor Derating Criteria

Rule Status: Proposed Rule Supersedes: Unknown Status: Proposed

Author Name: Ali Hashmi

Organization: Aries

Semiconductors Derating Factor

Rule Creation Date: 7/19/94 Time: 5:10:16PM

Rule Modified Date 7/25/94 Time: 4:07:15PM

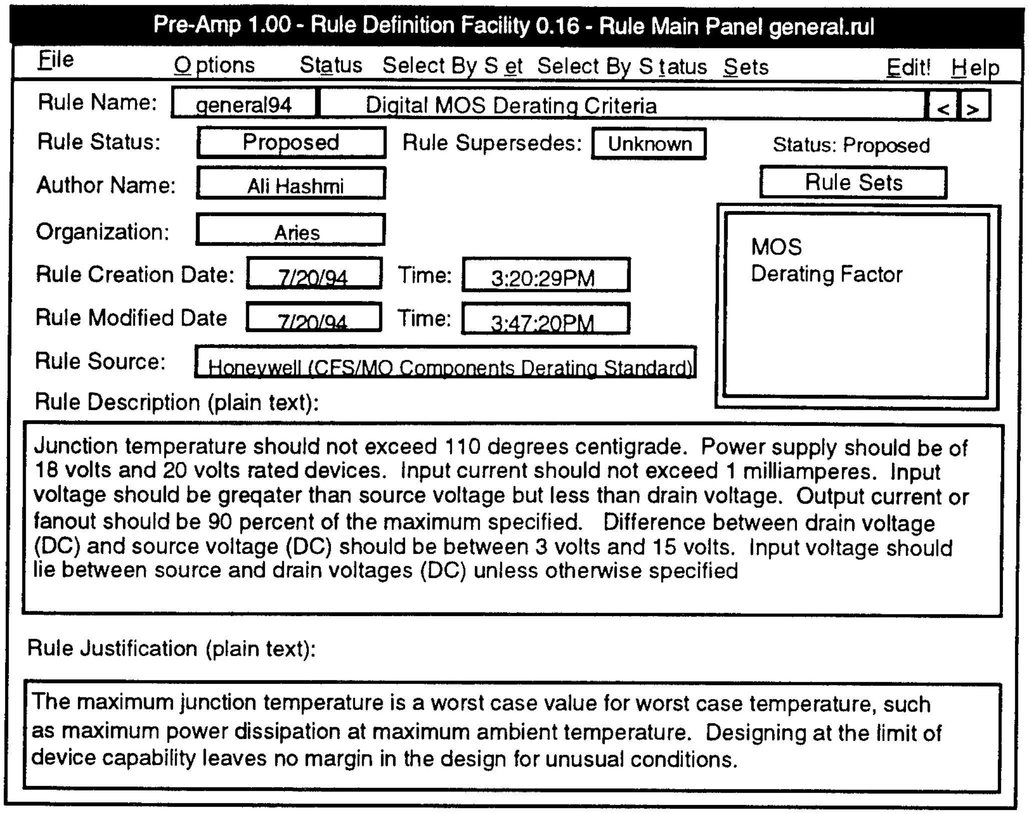
Rule Source:

Rule Description (plain text):

Junction temperature should not exceed 110 degrees centigrade. It shall be a design requirement to hold lthe junction temperature during normal temperature conditions to a junction temperature of 85 degrees centigrade or less.

Rule Justification (plain text):

The maximum junction temperature is a worst case value for worst case temperature, such as maximum power dissipation at maximum ambient temperature. Designing at the limit of device capability leaves no margin in the design for unusual conditions.



Pre-Amp 1.00 - Rule Definition Facility 0.16 - Rule Main Panel general.rul

Rule Name: general94 Digital MOS Derating Criteria

Rule Status: Proposed

Rule Supersedes: Unknown

Status: Proposed

Author Name: Ali Hashmi

Organization: Aries

MOS
Derating Factor

Rule Creation Date: 7/20/94 Time: 3:20:29PM

Rule Modified Date 7/20/94 Time: 3:47:20PM

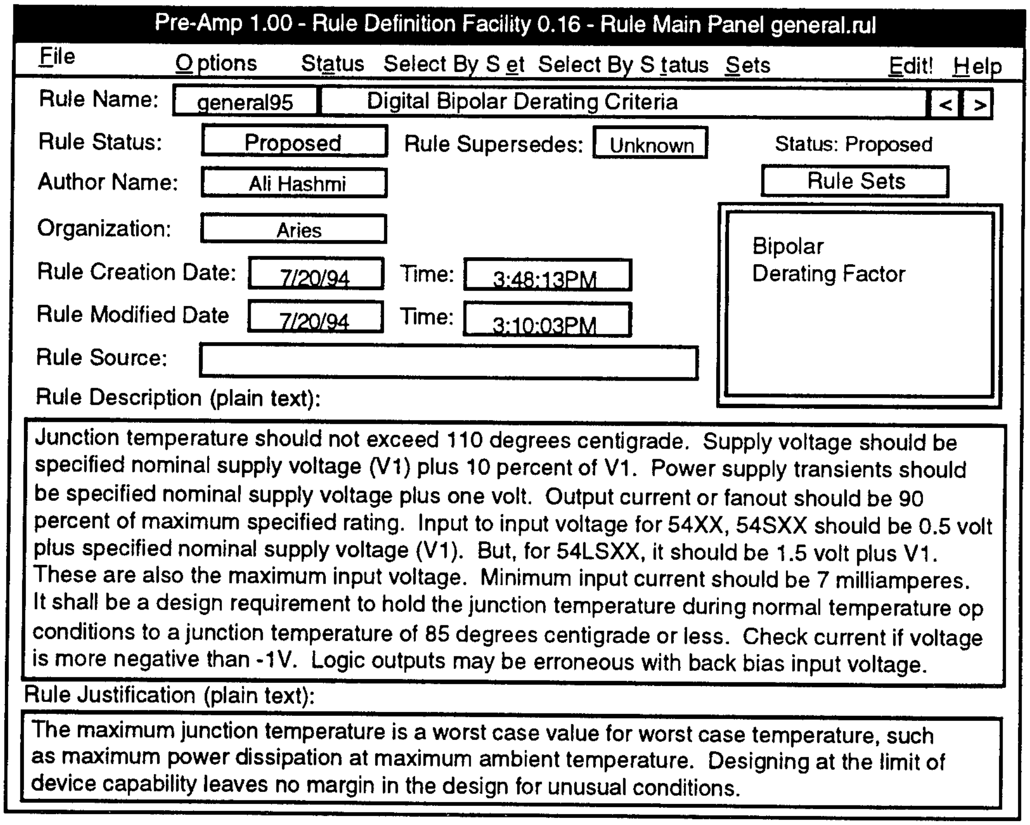
Rule Source: Honeywell (CFS/MO Components Derating Standard)

Rule Description (plain text):

Junction temperature should not exceed 110 degrees centigrade. Power supply should be of 18 volts and 20 volts rated devices. Input current should not exceed 1 milliamperes. Input voltage should be greqater than source voltage but less than drain voltage. Output current or fanout should be 90 percent of the maximum specified. Difference between drain voltage (DC) and source voltage (DC) should be between 3 volts and 15 volts. Input voltage should lie between source and drain voltages (DC) unless otherwise specified

Rule Justification (plain text):

The maximum junction temperature is a worst case value for worst case temperature, such as maximum power dissipation at maximum ambient temperature. Designing at the limit of device capability leaves no margin in the design for unusual conditions.



Pre-Amp 1.00 - Rule Definition Facility 0.16 - Rule Main Panel general.rul

Rule Name: general95 Digital Bipolar Derating Criteria

Rule Status: Proposed

Rule Supersedes: Unknown

Status: Proposed

Author Name: Ali Hashmi

Organization: Aries

Bipolar
Derating Factor

Rule Creation Date: 7/20/94 Time: 3:48:13PM

Rule Modified Date 7/20/94 Time: 3:10:03PM

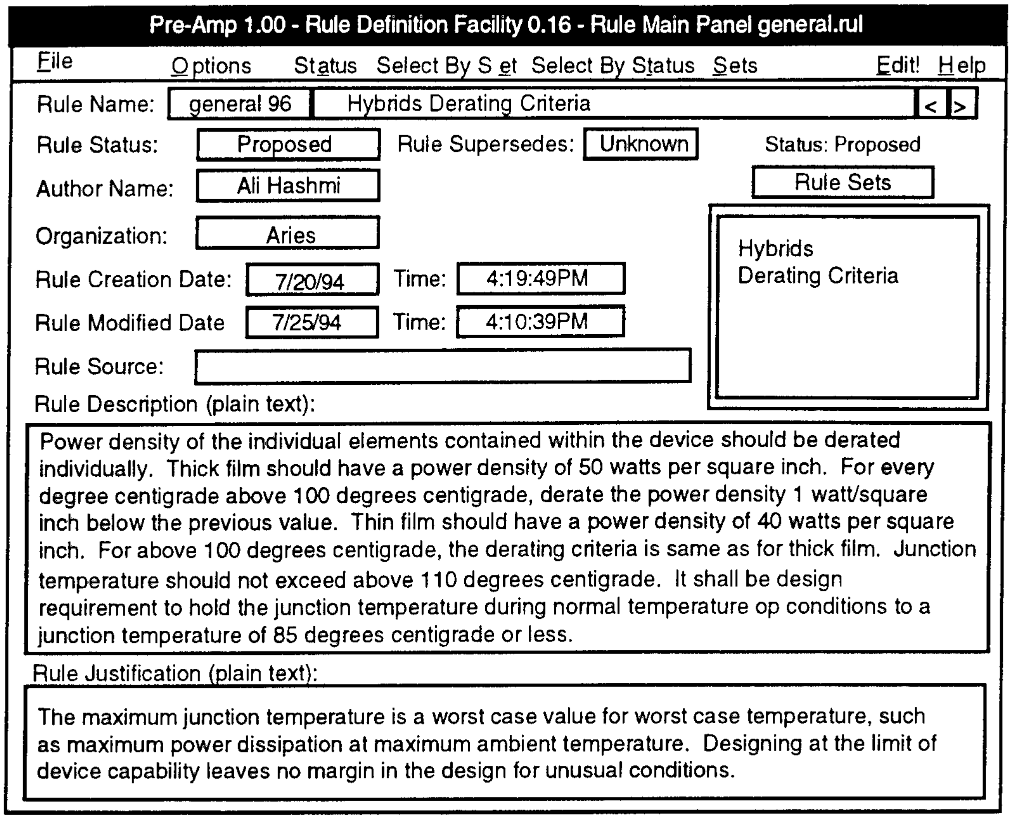
Rule Source:

Rule Description (plain text):

Junction temperature should not exceed 110 degrees centigrade. Supply voltage should be specified nominal supply voltage (V1) plus 10 percent of V1. Power supply transients should be specified nominal supply voltage plus one volt. Output current or fanout should be 90 percent of maximum specified rating. Input to input voltage for 54XX, 54SXX should be 0.5 volt plus specified nominal supply voltage (V1). But, for 54LSXX, it should be 1.5 volt plus V1. These are also the maximum input voltage. Minimum input current should be 7 milliamperes. It shall be a design requirement to hold the junction temperature during normal temperature op conditions to a junction temperature of 85 degrees centigrade or less. Check current if voltage is more negative than -1V. Logic outputs may be erroneous with back bias input voltage.

Rule Justification (plain text):

The maximum junction temperature is a worst case value for worst case temperature, such as maximum power dissipation at maximum ambient temperature. Designing at the limit of device capability leaves no margin in the design for unusual conditions.


Pre-Amp 1.00 - Rule Definition Facility 0.16 - Rule Main Panel general.rul
File
Options
Status
Select By S et
Select By Status
Sets
Edit!
Help
<
>
Rule Sets

Rule Name: general 96 Hybrids Derating Criteria

Rule Status: Proposed Rule Supersedes: Unknown Status: Proposed

Author Name: Ali Hashmi

Organization: Aries

Rule Creation Date: 7/20/94 Time: 4:19:49PM

Rule Modified Date 7/25/94 Time: 4:10:39PM

Rule Source:

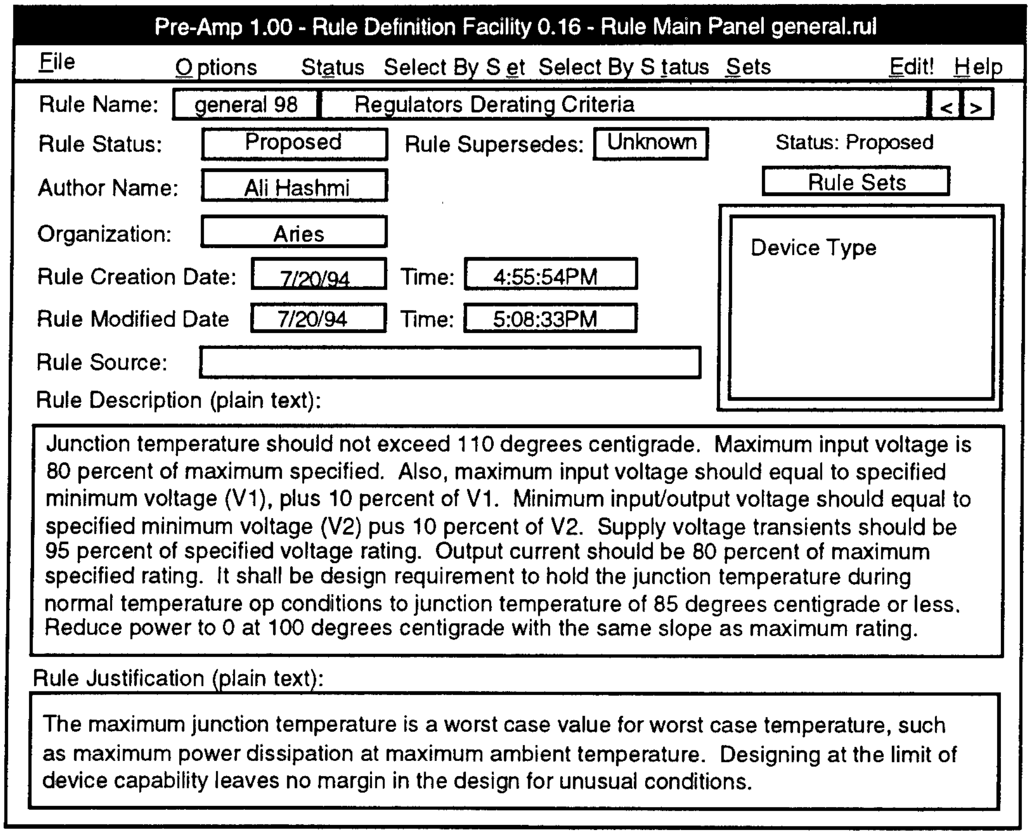
Hybrids
Derating Criteria

Rule Description (plain text):

Power density of the individual elements contained within the device should be derated individually. Thick film should have a power density of 50 watts per square inch. For every degree centigrade above 100 degrees centigrade, derate the power density 1 watt/square inch below the previous value. Thin film should have a power density of 40 watts per square inch. For above 100 degrees centigrade, the derating criteria is same as for thick film. Junction temperature should not exceed above 110 degrees centigrade. It shall be design requirement to hold the junction temperature during normal temperature op conditions to a junction temperature of 85 degrees centigrade or less.

Rule Justification (plain text):

The maximum junction temperature is a worst case value for worst case temperature, such as maximum power dissipation at maximum ambient temperature. Designing at the limit of device capability leaves no margin in the design for unusual conditions.


Pre-Amp 1.00 - Rule Definition Facility 0.16 - Rule Main Panel general.rul
File
Options
Status
Select By S et
Select By S tatus
Sets
Edit!
Help
<
>
Rule Sets

Rule Name: general 98 Regulators Derating Criteria

Rule Status: Proposed Rule Supersedes: Unknown Status: Proposed

Author Name: Ali Hashmi

Organization: Aries

Rule Creation Date: 7/20/94 Time: 4:55:54PM

Rule Modified Date 7/20/94 Time: 5:08:33PM

Rule Source:

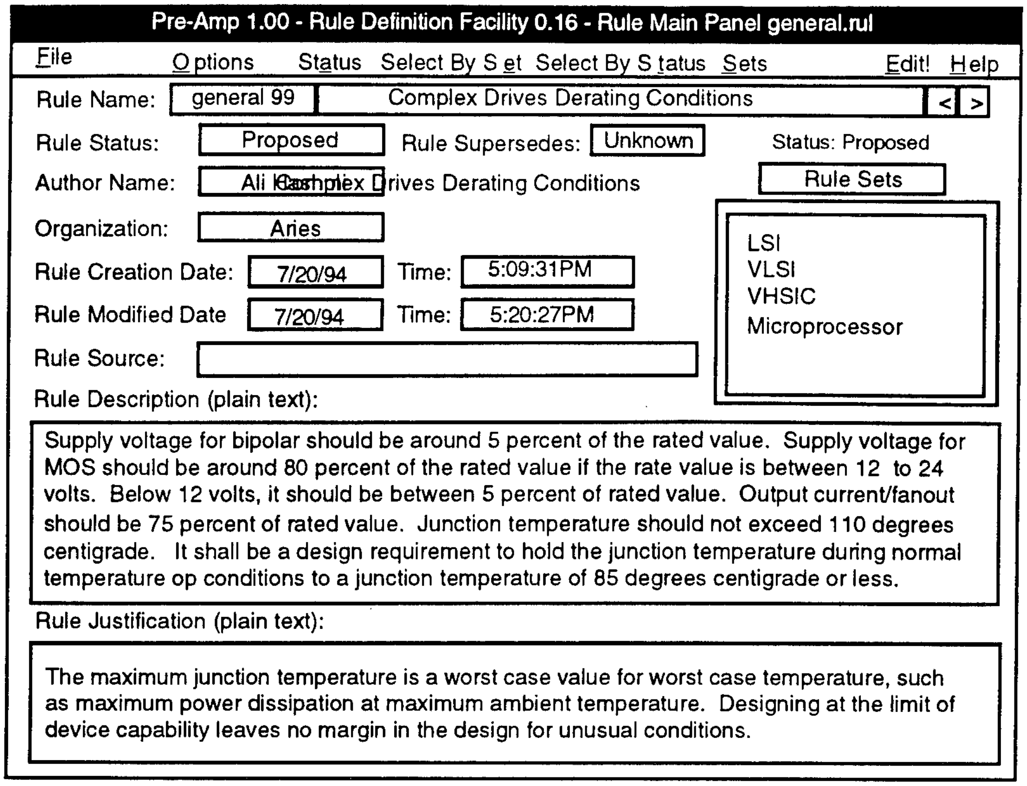
Device Type

Rule Description (plain text):

Junction temperature should not exceed 110 degrees centigrade. Maximum input voltage is 80 percent of maximum specified. Also, maximum input voltage should equal to specified minimum voltage (V1), plus 10 percent of V1. Minimum input/output voltage should equal to specified minimum voltage (V2) pus 10 percent of V2. Supply voltage transients should be 95 percent of specified voltage rating. Output current should be 80 percent of maximum specified rating. It shall be design requirement to hold the junction temperature during normal temperature op conditions to junction temperature of 85 degrees centigrade or less. Reduce power to 0 at 100 degrees centigrade with the same slope as maximum rating.

Rule Justification (plain text):

The maximum junction temperature is a worst case value for worst case temperature, such as maximum power dissipation at maximum ambient temperature. Designing at the limit of device capability leaves no margin in the design for unusual conditions.



Pre-Amp 1.00 - Rule Definition Facility 0.16 - Rule Main Panel general.rul

Rule Name: general 99 Complex Drives Derating Conditions

Rule Status: Proposed Rule Supersedes: Unknown Status: Proposed

Author Name: Ali Complex Drives Derating Conditions

Organization: Aries

Rule Creation Date: 7/20/94 Time: 5:09:31PM

Rule Modified Date 7/20/94 Time: 5:20:27PM

Rule Source:

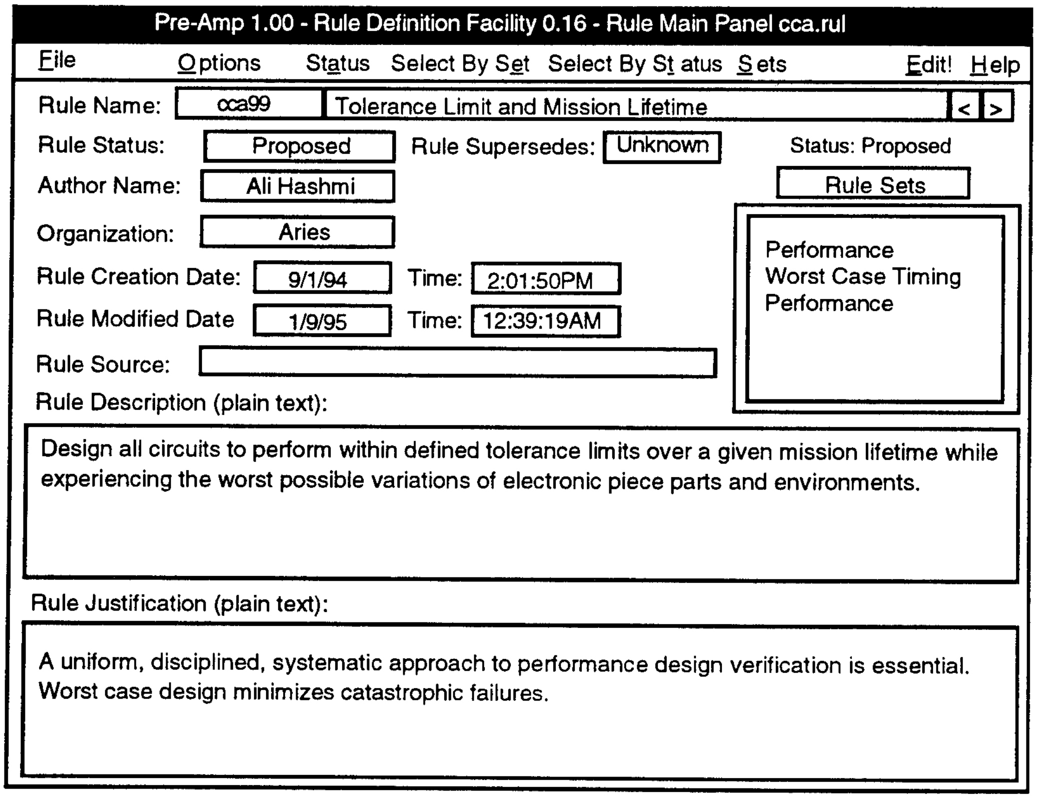
LSI
VLSI
VHSIC
Microprocessor

Rule Description (plain text):

Supply voltage for bipolar should be around 5 percent of the rated value. Supply voltage for MOS should be around 80 percent of the rated value if the rate value is between 12 to 24 volts. Below 12 volts, it should be between 5 percent of rated value. Output current/fanout should be 75 percent of rated value. Junction temperature should not exceed 110 degrees centigrade. It shall be a design requirement to hold the junction temperature during normal temperature op conditions to a junction temperature of 85 degrees centigrade or less.

Rule Justification (plain text):

The maximum junction temperature is a worst case value for worst case temperature, such as maximum power dissipation at maximum ambient temperature. Designing at the limit of device capability leaves no margin in the design for unusual conditions.



Pre-Amp 1.00 - Rule Definition Facility 0.16 - Rule Main Panel cca.rul

Rule Name: cca99 Tolerance Limit and Mission Lifetime

Rule Status: Proposed Rule Supersedes: Unknown Status: Proposed

Author Name: Ali Hashmi

Organization: Aries

Rule Creation Date: 9/1/94 Time: 2:01:50PM

Rule Modified Date 1/9/95 Time: 12:39:19AM

Rule Source:

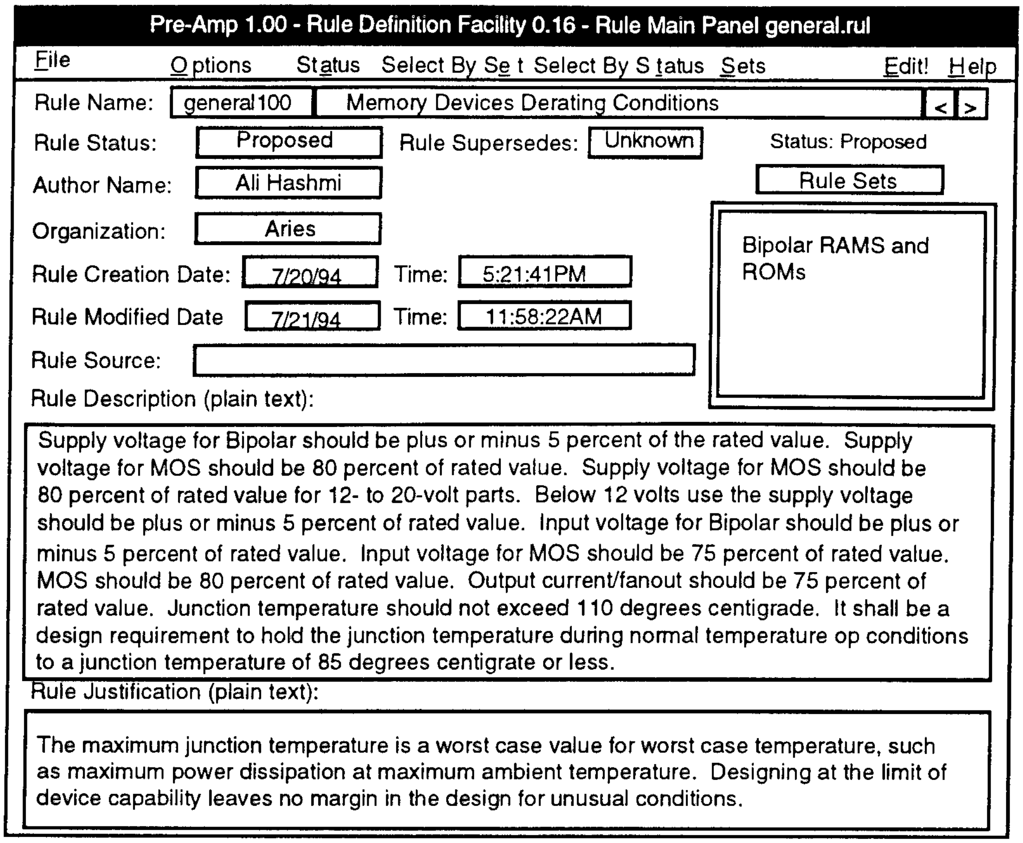
Performance
Worst Case Timing
Performance

Rule Description (plain text):

Design all circuits to perform within defined tolerance limits over a given mission lifetime while experiencing the worst possible variations of electronic piece parts and environments.

Rule Justification (plain text):

A uniform, disciplined, systematic approach to performance design verification is essential. Worst case design minimizes catastrophic failures.


Pre-Amp 1.00 - Rule Definition Facility 0.16 - Rule Main Panel general.rul
File
Options
Status
Select By Se t
Select By S tatus
Sets
Edit!
Help
<
>
Rule Sets

Rule Name: general100 Memory Devices Derating Conditions

Rule Status: Proposed Rule Supersedes: Unknown Status: Proposed

Author Name: Ali Hashmi

Organization: Aries

Rule Creation Date: 7/20/94 Time: 5:21:41PM

Rule Modified Date 7/21/94 Time: 11:58:22AM

Rule Source:

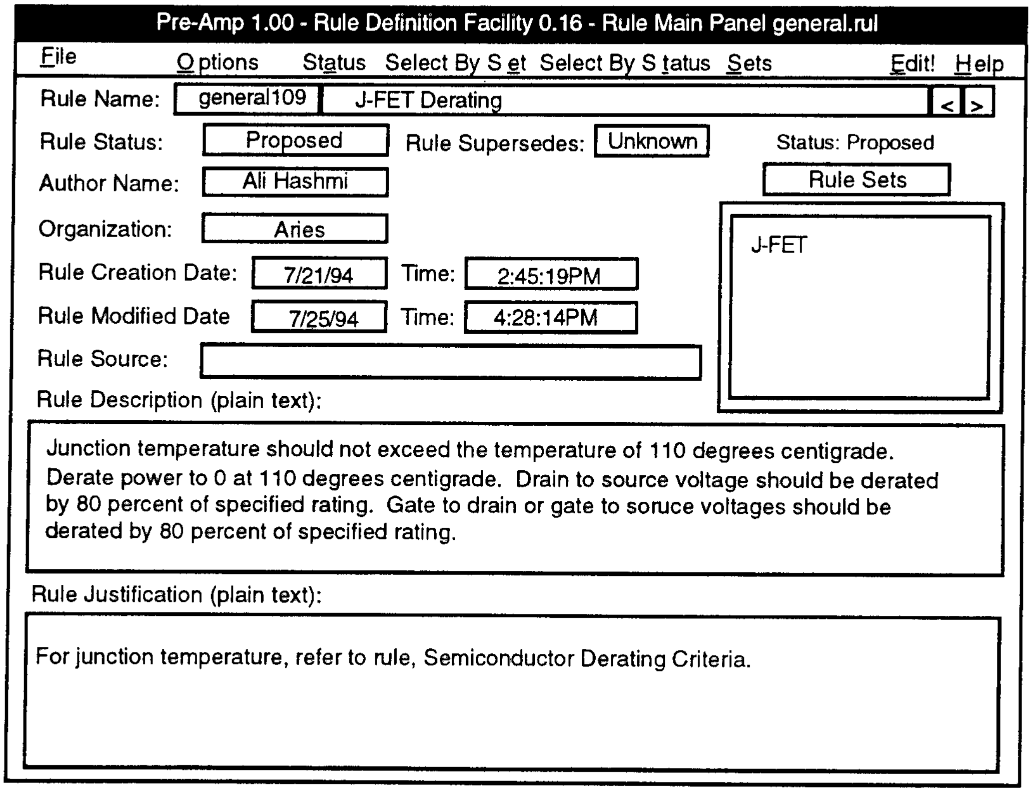
Bipolar RAMS and ROMs

Rule Description (plain text):

Supply voltage for Bipolar should be plus or minus 5 percent of the rated value. Supply voltage for MOS should be 80 percent of rated value. Supply voltage for MOS should be 80 percent of rated value for 12- to 20-volt parts. Below 12 volts use the supply voltage should be plus or minus 5 percent of rated value. Input voltage for Bipolar should be plus or minus 5 percent of rated value. Input voltage for MOS should be 75 percent of rated value. MOS should be 80 percent of rated value. Output current/fanout should be 75 percent of rated value. Junction temperature should not exceed 110 degrees centigrade. It shall be a design requirement to hold the junction temperature during normal temperature op conditions to a junction temperature of 85 degrees centigrate or less.

Rule Justification (plain text):

The maximum junction temperature is a worst case value for worst case temperature, such as maximum power dissipation at maximum ambient temperature. Designing at the limit of device capability leaves no margin in the design for unusual conditions.


Pre-Amp 1.00 - Rule Definition Facility 0.16 - Rule Main Panel general.rul
File
Options
Status
Select By S et
Select By S tatus
Sets
Edit!
Help
<
>
Rule Sets

Rule Name: general109 J-FET Derating

Rule Status: Proposed Rule Supersedes: Unknown Status: Proposed

Author Name: Ali Hashmi

Organization: Aries

Rule Creation Date: 7/21/94 Time: 2:45:19PM

Rule Modified Date 7/25/94 Time: 4:28:14PM

Rule Source:

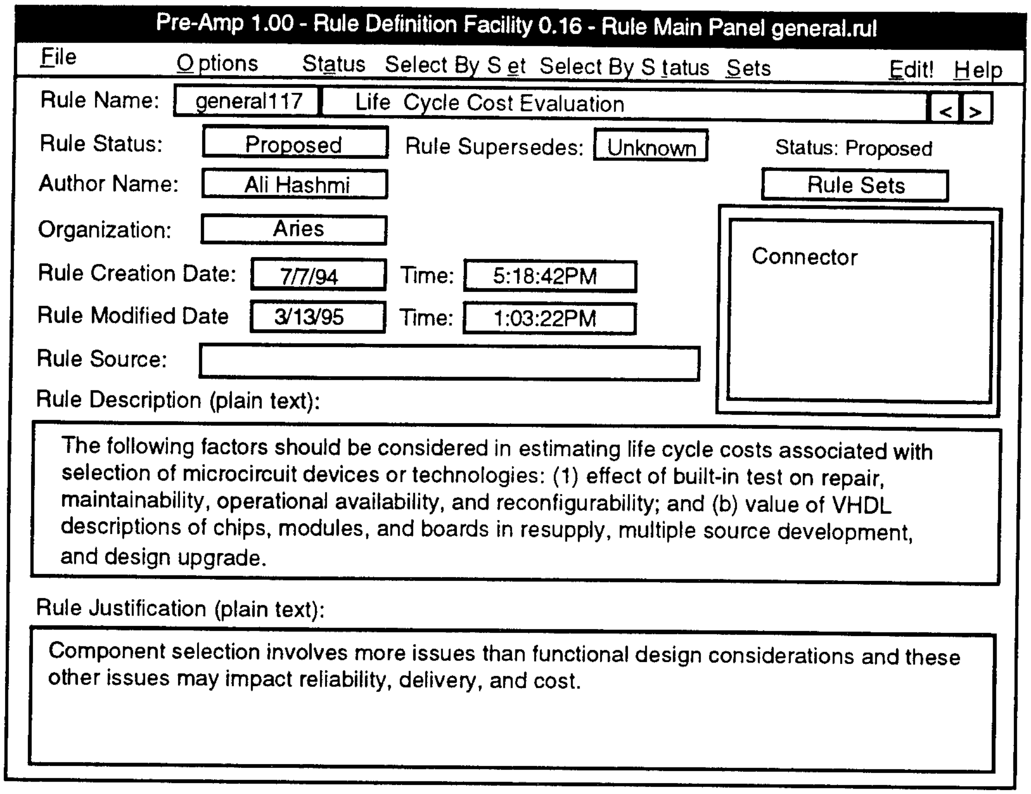
J-FET

Rule Description (plain text):

Junction temperature should not exceed the temperature of 110 degrees centigrade. Derate power to 0 at 110 degrees centigrade. Drain to source voltage should be derated by 80 percent of specified rating. Gate to drain or gate to soruce voltages should be derated by 80 percent of specified rating.

Rule Justification (plain text):

For junction temperature, refer to rule, Semiconductor Derating Criteria.



Pre-Amp 1.00 - Rule Definition Facility 0.16 - Rule Main Panel general.rul

Rule Name: general117 Life Cycle Cost Evaluation

Rule Status: Proposed Rule Supersedes: Unknown Status: Proposed

Author Name: Ali Hashmi

Organization: Aries

Rule Creation Date: 7/7/94 Time: 5:18:42PM

Rule Modified Date 3/13/95 Time: 1:03:22PM

Rule Source:

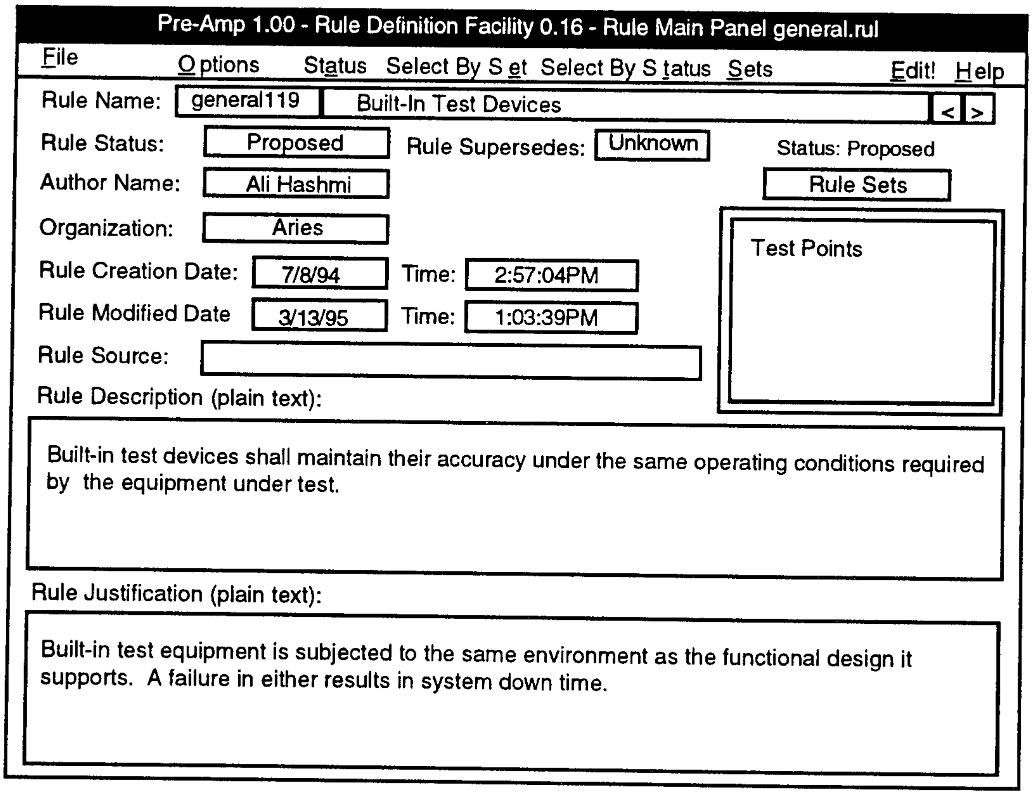
Connector

Rule Description (plain text):

The following factors should be considered in estimating life cycle costs associated with selection of microcircuit devices or technologies: (1) effect of built-in test on repair, maintainability, operational availability, and reconfigurability; and (b) value of VHDL descriptions of chips, modules, and boards in resupply, multiple source development, and design upgrade.

Rule Justification (plain text):

Component selection involves more issues than functional design considerations and these other issues may impact reliability, delivery, and cost.



Pre-Amp 1.00 - Rule Definition Facility 0.16 - Rule Main Panel general.rul

Rule Name: general119 Built-In Test Devices

Rule Status: Proposed Rule Supersedes: Unknown Status: Proposed

Author Name: Ali Hashmi

Organization: Aries

Rule Creation Date: 7/8/94 Time: 2:57:04PM

Rule Modified Date 3/13/95 Time: 1:03:39PM

Rule Source:

Test Points

Rule Description (plain text):

Built-in test devices shall maintain their accuracy under the same operating conditions required by the equipment under test.

Rule Justification (plain text):

Built-in test equipment is subjected to the same environment as the functional design it supports. A failure in either results in system down time.

Appendix E

PROOF OF CONCEPT

*Pre*Amp Rules Definition Facility User's Report

RULE DEFINITION FACILITY USER MANUAL

for the *PreAmp* Program

January 13, 1995

PreAmp

Version 1.01

PREPARED BY

PreAmp Team

5300 International Blvd.

N. Charleston, SC 29418

PreAmp - Rules Definition Facility Users Guide

PC Software Version 0.15
UNIX Software Version 0.15

Preface

The Rules Definition Facility (RDF) Users Guide describes how to install, execute, and utilize the *Pre*A*mp* Rule Definition Facility. This guide is divided into the following parts:

Part 1

- System Installation
- Execution Procedures
- Hardware/Software Requirements
- System Operations Discussion

Part 2

- Function Listing
- Function Reference

Part 3

- Summary

PreAmp - Rules Definition Facility Users Guide - Part 1

PC Software Version 0.15
UNIX Software Version 0.15

Overview

The Rules Definition Facility (RDF) is a database editor used to capture PCA related knowledge in the form of rules. Using the Rules Definition Facility rules can be created, edited, and/or removed. Additionally, rule meta information (concerning rule approval, origination, description, justification, etc.) can be defined. Information defined in the Rules Definition Facility can be stored in the *PreAmp* database and later executed based on changes to the database or at user request.

Note: To get a quick start at loading the software while you are reading the users guide skip ahead to section 1.3. Return and continue reading as the software is loading.

The Rules Definition Facility has been demonstrated as a standalone software component in the DOS/Windows environment, as well as part of the *PreAmp* framework in the UNIX/X windows environment. The RDF provides a user-friendly capability for CAD/CAM oriented individuals to define and collect rules to be used for printed circuit assembly (PCA) engineering design and manufacture. The DOS/Windows version provides a portable, standalone capability for defining rules. The UNIX/X windows version provides a capability for defining and executing rules. The UNIX/X windows version is integrated with the *PreAmp* System User Interface (SUI). The RDF facilitates the acquisition of product knowledge in a generic form that removes syntax considerations from the rule definition. The syntax free rule definition environment provides a simpler and more meaningful methodology for users to capture rule components and their parameters. The generic rule form can be imported into the *PreAmp* system and run against data in the *PreAmp* database.

The initial version of the RDF was created using the DOS/Windows version of Kappa. It included an early version of STEP AP 210 for the KAPPA database. Development resources have been redirected to migrate the RDF to the UNIX/X windows environment in support of the *PreAmp* framework. This conversion from KAPPA PC in Windows to KAPPA UNIX in X windows has been completed. The UNIX/X windows version is accessible from the *PreAmp* System User Interface (SUI) or the UNIX command line. In the DOS/Windows environment the Rules Definition Facility is accessed via a Windows icon. Note that in the DOS/Windows environment the Rules Definition Facility acts as a standalone system where the rules can be defined but not be executed. Rules can however, be transferred to the UNIX/X windows version and executed in the *PreAmp* framework.

Although the latest database entities and glossary definitions have been included in this release, AP 210 developers are still incorporating updates to their schema and glossary.

Database schema and glossary information will be updated as newer data is received from AP 210 developers.

Software Requirements (PC Version only)

This version of the Rules Definition Facility (RDF) has been tested on both Microsoft Windows version 3.1 or Windows for WorkGroup version 3.11. Running on other Window versions is not advised. The RDF software uses the Windows Notepad editor for basic text editing. No other software is used or required.

Hardware Requirements (PC Version only)

This version of the Rules Definition Facility (RDF) has been tested on IBM compatible 386 and 486 machines. Disk space requirements for the program is 2 Mbytes. Disk space requirements for the data is approximately 10 Mbytes depending in the number of rules, their complexity, and size of the database schema. It is suggested that 8 Mbytes of memory be available, however the system will run with 4 Mbytes. The RDF requires a mouse for selection of certain interface options, as well as, simplifying its overall use.

1.0 User Manual Introduction

This manual is designed to assist an engineer in entering PCA related design and manufacturing rules into an object oriented database for future use in an integrated knowledge based framework or system. All system menus and commands will be discussed. Where possible actual system screens will be included. An example rule will be created and edited.

1.1 System Windows

The Rules Definition Facility is a point and click based tool. By clicking on menus and button icons the user can alter attributes, create new rules, define rule approval, and save data to the database. (Unless otherwise stated, all window buttons and objects are activated by a single click of the left mouse button.) The RDF tool has three windows or panels for the capturing of design and manufacturing rules:

1. The Rule Main Panel
2. The Rule Edit Panel
3. The Database Glossary Panel

The rule main panel first appears as the Rules Definition Facility is invoked. This panel is used to open existing rules files as well as create new rule files. New rules can be created and existing rules can be edited. Rule meta-data can be defined and altered. Meta-data includes rule description, justification, categorization, author, organization, and source.

The rule component or rule edit panel provides a menu-based environment to define rule premise and conclusion components. Premise components, or if conditions, define what actions are required before the rule conclusion component is executed. The conclusion component consists of actions to take as a result of all of the premise conditions being met. The rule premise is a logical combination (ANDing or ORing) of premise functions. Premise functions are logical (greater than, less than, etc.), mathematical (sum, power, division, etc.), and/or database (retrieve object attributes or facts) functions. The rule conclusion is a logical combination (ANDing) of conclusion functions. These functions include: storing temporal information into the database (facts), storing permanent information into the database (object attributes), and providing feedback to the user in the form of messages and issues.

Note that conclusion functions include the storing of facts and/or object attributes. A rule premise may include functions that test facts and/or object attributes. Utilizing the two preceding capabilities allows one rule to call another rule. This process is known as 'chaining'. This ability can be used to capture the intent of a multistage rule with several smaller rules. This topic is covered again in a later section.

The database glossary panel provides a methodology for traversing the combined AP 210 (PCA product information), AP 220 (processing planning information), and *PreAmp* (rules, design rule check, factory, and issues information) database schemas. Additionally, functions are provided for examining glossary definitions of objects and attributes for all schemas (where glossary definitions have been defined).

Unless otherwise noted the text in this users guide corresponds to both the DOS/Windows version and the UNIX/X Windows version of the Rules Definition Facility.

1.2 System Concepts

The intent of providing a multi-windowed user interface is to present the end user with only the amount of detail required for his/her specific task. The rule main panel can be utilized to define basic information concerning a rule. Here the rule's source, textual description, and textual justification can be defined. Knowledge of rule component primitives and/or database object attributes is not required. Organizations dealing with the capture of knowledge could deal specifically with the rule selection panel without actually defining the mechanics of the rule using the rule edit panel.

The rule edit panel was designed for individuals that have a basic understanding of rule components and rule chaining. Although the syntax of the rule definition has been eliminated, the user must still understand rule semantics and database schemas. The database glossary panel was designed to aid the individual in defining rule components by providing a 'browser' mechanism for the database schemas.

1.3 Installing the Rules Definition Facility Software (PC version only)

The DOS/Windows version of the RDF is provided on two or three 3 1/2 inch floppy disks. Each disk contains RDF files under a directory called ruledef.15. At this time a separate Rules Definition Facility install program has not been developed. The following procedure will aid in manually installing the RDF software.

Using the Windows File Manager the ruledef.15 directory on an RDF floppy disk must be selected and copied to the C hard disk drive. The copy process must be repeated for each RDF floppy disk. The order in which the floppy disks are copied is not important. Note that a drive other C may be used by replacing all references to drive C in this document to the appropriate target drive.

1.3.1 Installing the Rules Definition Facility Software (Workstation version only)

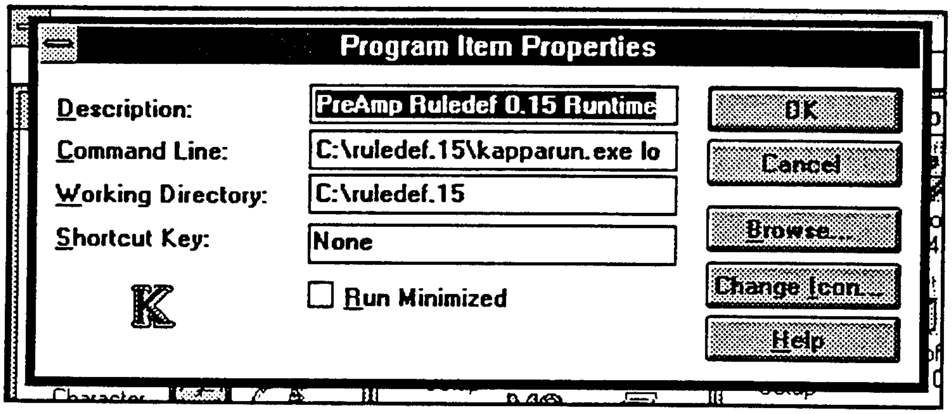
TBA

1.4 Setting up the Rules Definition Software Icon (PC version only)

To execute the Rules Definition Facility in the DOS/Windows environment an execution icon must be defined. From the File Manager program select the File, New menu option. Under the New Program Object select the Program Item option followed by OK. When requested fill in the required entries as defined below:

Description:	***PreAmp* Ruledef 0.15 Runtime**
Command Line:	**c:\ruledef.15\kapparun.exe load.kal**
Working Directory:	**c:\ruledef.15**
Shortcut Key:	**None (or your choice)**

When complete the Program Item Properties display should look as follows:



Program Item Properties

Description: PreAmp Ruledef 0.15 Runtime
Command Line: C:\ruledef.15\kapparun.exe lo
Working Directory: C:\ruledef.15
Shortcut Key: None
□ Run Minimized

Figure 1-1 Program Item Properties (DOS/Windows)

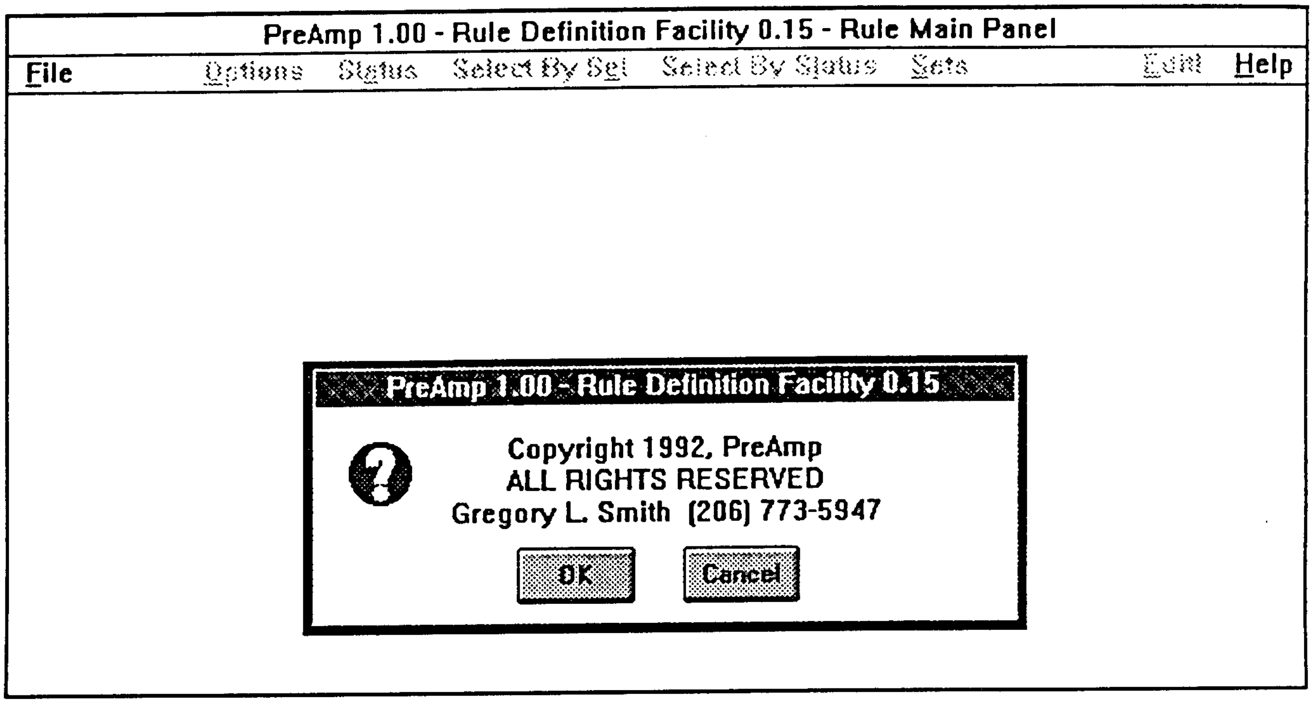
Click OK when all the information is correctly entered. A red KAPPA 'K' should appear in the Program Manager window when processing is complete.

1.4.1 Setting up the Rules Definition Software Icon (Workstation version only)

TBA

2.0 Initiating the System

For the DOS/Windows version, double-clicking on the Rule Definition Facility's red 'K' will initiate the system and the rule selection panel, along with the *PreAmp* copyright panel, will be displayed as depicted by the figure below.


PreAmp 1.00 - Rule Definition Facility 0.15 - Rule Main Panel
File
Options
Status
Select By Set
Select By Status
Sets
Edit
Help
PreAmp 1.00 - Rule Definition Facility 0.15
OK
Cancel

Copyright 1992, PreAmp
ALL RIGHTS RESERVED
Gregory L. Smith (206) 773-5947

Figure 2-1 Initial rule definition screen.

Click OK or Cancel on the *PreAmp* copyright panel to close it.

The majority of functions are not available until an existing rules file is opened or a new rules file is created. To open an existing rules file, click on the **File** menu and select the **Open...** command. To create a new rules file click on the **File** menu and select the New command.

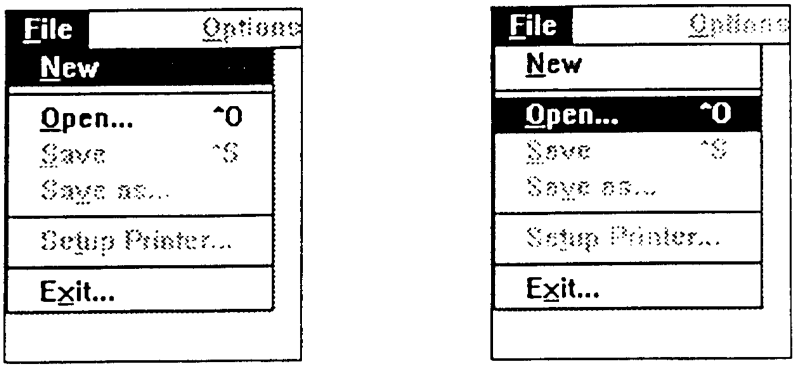
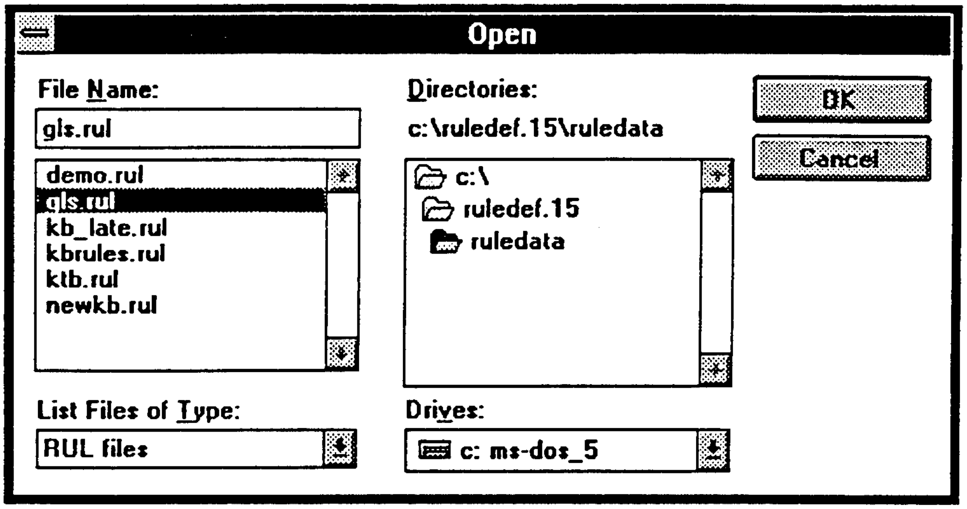


Figure 2-2 File menu for open and new commands.

When the File Open command is selected, the Open menu is provided to the user.



File Name: gls.rul

Directories: c:\ruledef.15\ruledata

List Files of Type: RUL files

Drives: c: ms-dos_5

Figure 2-3 File Open menu.

Files ending with the **.rul** extension are known to the system as rules files. They contain rules previously defined and stored. Selecting any of the existing **.rul** files and selecting OK, or typing in a .rul file into the File Name window component and selecting OK will open that particular rules file. The examples in this guide will use the **ruledata\gls.rul** file.

After the rules file is selected the system will attempt to open two files. The **gls.rul** file will be opened first. This file contains rule instances and rule sets or categories. A second file, **gls.db**, is also opened. This file contains database objects, attributes, and glossary entries. If no **gls.db** file exists (i.e. in the case of creating a new rules file), the **default.db** file is loaded. Note that loading the database file and subsequently creating database menus can take **several minutes** to complete. After the rules and database files have been successfully loaded, the following screen displaying the rule selection panel will be shown.

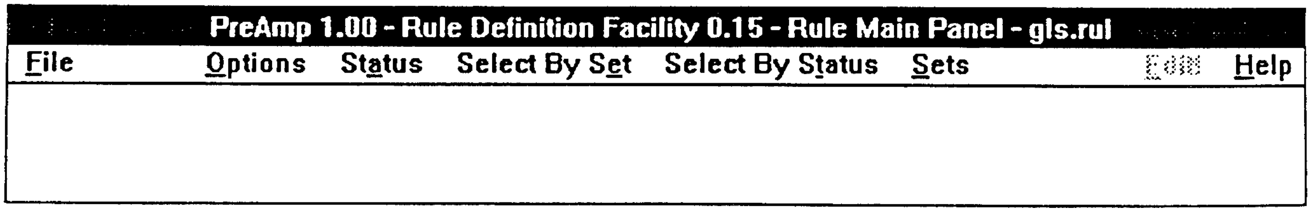

PreAmp 1.00 - Rule Definition Facility 0.15 - Rule Main Panel - gls.rul
File
Options
Status
Select By Set
Select By Status
Sets
Edit
Help

Figure 2-4 Rule Selection Panel after rules file is opened.

Once a valid rules file is opened or created the options in the File menu are altered accordingly.

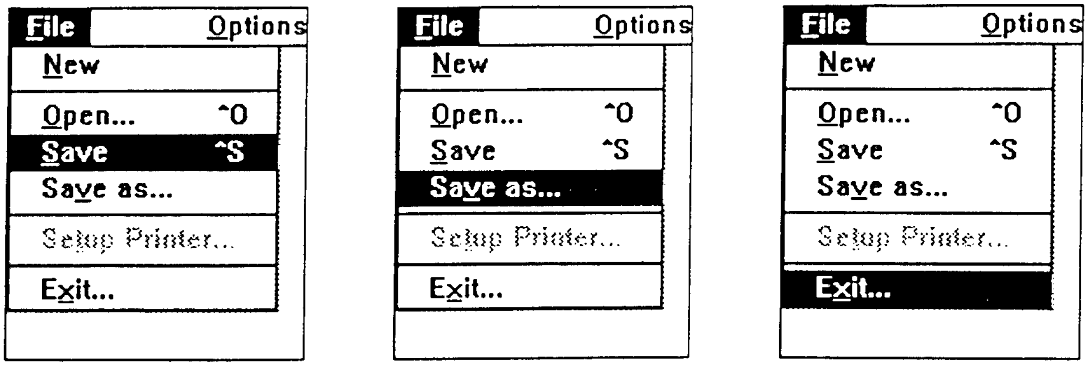
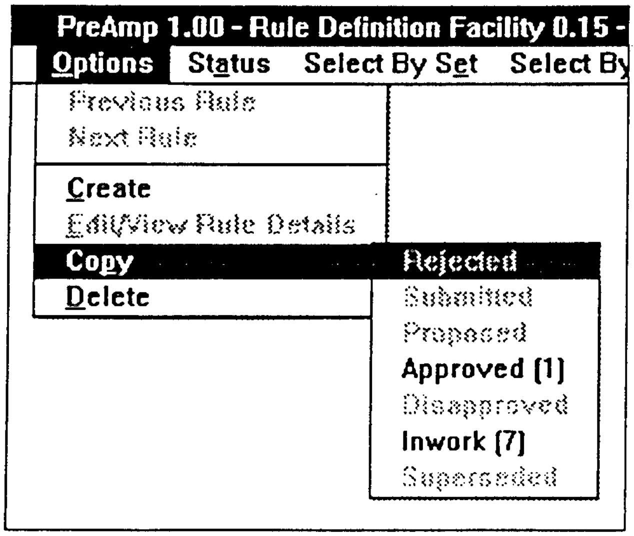
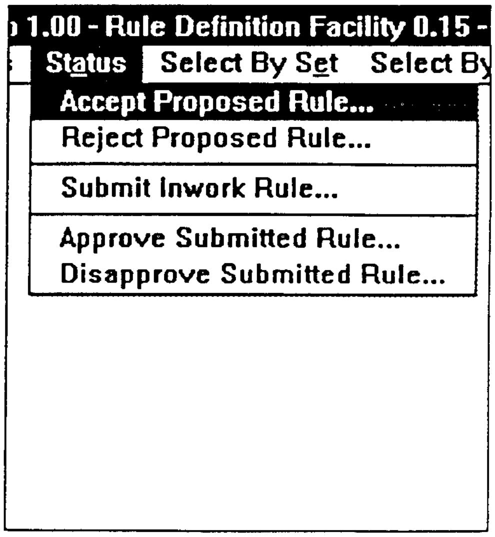


Figure 2-5 File menus after a rules file is available.

At any time, changes to the rules and/or database files can be saved either to themselves by using the **Save** option or to another location using the **Save as...** option. When the save as command is selected a file selection menu will be provided for entering the name of the new rules/database files. Note that in the Rules Definition Facility when a command is selected from any menu whose name is suffixed by '...', a subsequent menu will be provided to the user for additional information and/or command confirmation. When the **Exit...** command is selected the user will be prompted to save the rules and the database (whether or not changes were made), followed by a prompt to confirm the users intent to exit the system.

The following figure shows available options under **Options** (with Copy menu expanded), **Status, Select By Set, Select By Status, Sets,** and **Help** menu options after a rules file has been created or opened.





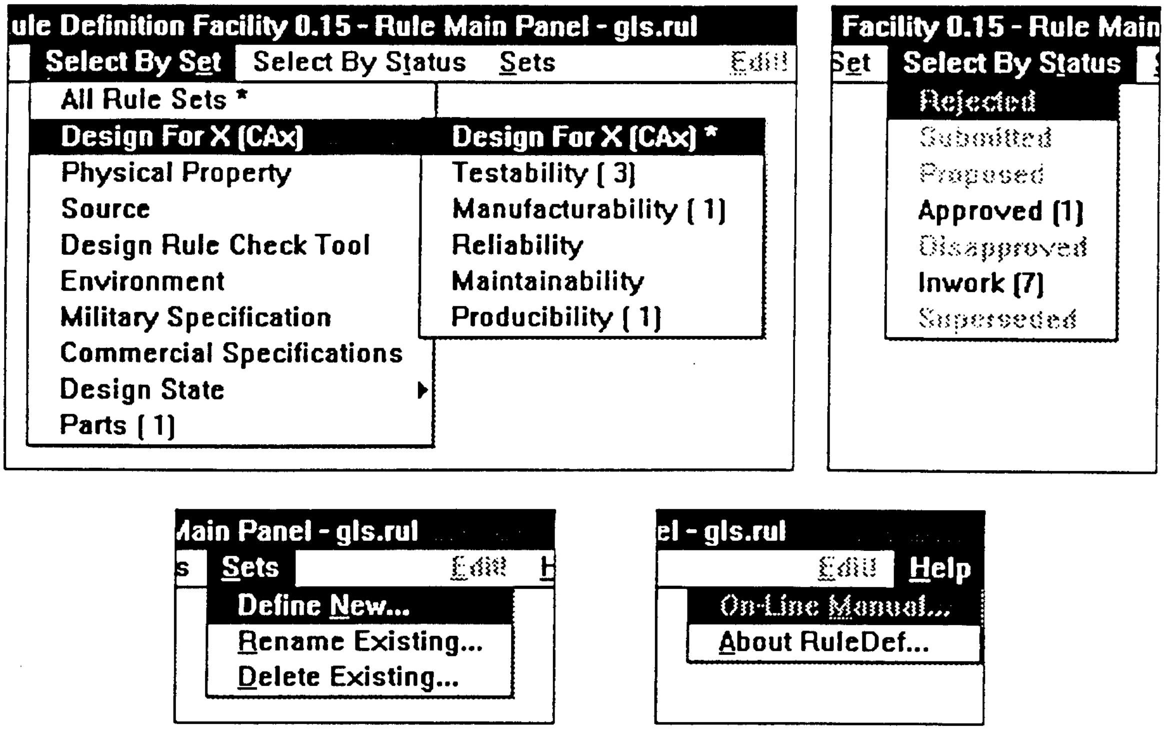


Figure 2-6 Menubar options after rules file is opened.

Under the options menu, commands are provided to create, copy, and delete rules. Options for selecting the next and the previous rule, and editing a rule, are not available until a rule is selected, created or copied. Rules are selected either by their current status or by the rule set that they belong to. The RDF organizes rules by status and set. Rule status defines the progress of a rule in the process of defining, submitting, and accepting rules. Rule sets are user defined categories that aid in organizing rules in small meaningful groups.

2.1 Rule Status

In both the UNIX/X windows environment and the DOS/Windows environment complete functionality is provided for changing the status of a rule. When rules are first created (either by create rule or copy rule) their status is **proposed**. A **proposed** rule contains only meta-data about the rule (its description, justification, etc.). Note the Rule Definition Facility does not disallow the user from inputting rule details via the Rule Edit Panel on a **proposed** rule. A **proposed** rule can be either accepted or rejected. If the rule is rejected, rejection information is requested, its status is changed to **rejected** ,and the rule can not be edited. If the rule is accepted its status is changed to **inwork** and the rule can be edited. Once editing is completed on an **inwork** rule it can be submitted for approval or disapproval and its status is changed to **submitted** while the approval process is taking place. If the **submitted** rule is disapproved, disapproval information is requested, its status is changed to **disapproved** and the rule can not be edited. If the **submitted** rule is approved, its status is changed to **approved**. If an **approved** rule has defined a rule that it supersedes then that rule will have its status changed to **superseded**. The following diagram depicts this rule status process.

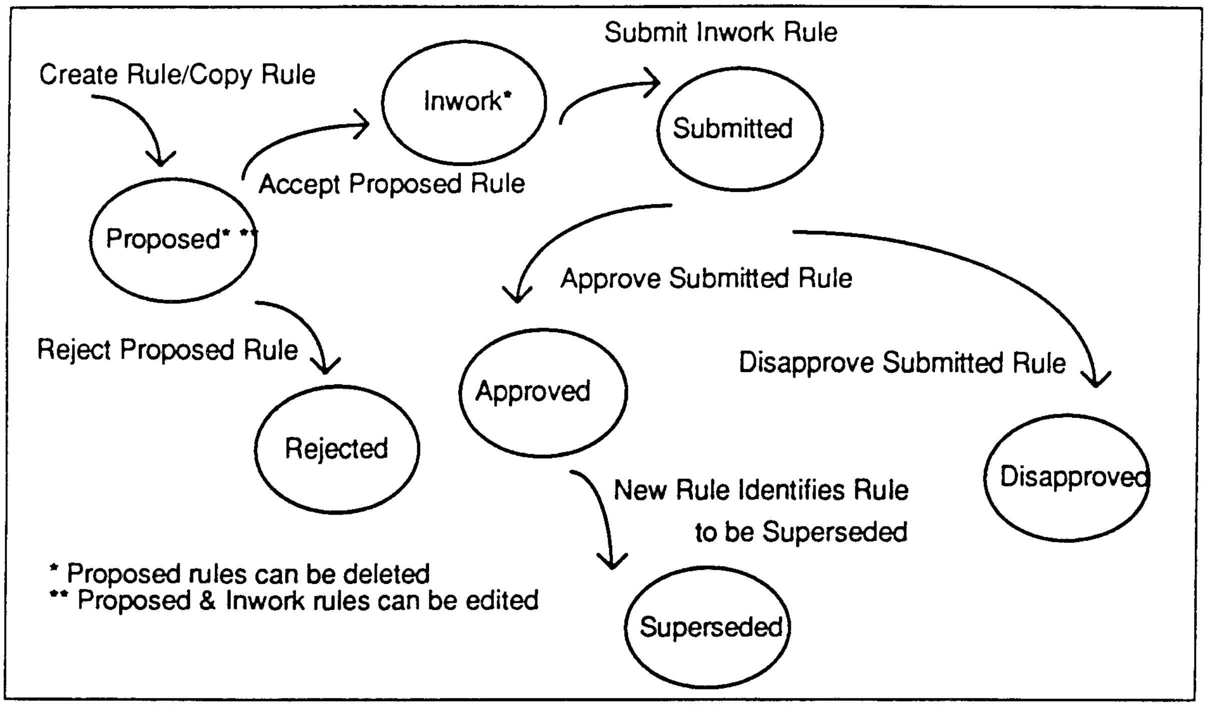


Figure 2-7 Rule status process flow.

2.2 Rule Sets

To simplify the task of defining and executing rules the Rules Definition Facility organizes rules in rule sets. A rule set is a group of rules that share a common subject. The rule sets are arranged in a rule hierarchy. The rule sets and their hierarchy are defined by the user. As an example, a user may wish to define a set of design rules. Subsets of rules under the design set could be analog, digital, and mixed. A rule can belong to any number of rule sets. When rules are executed in the *PreAmp* system they are selected by rule set. The following graphic depicts possible rule sets and a rule set hierarchy.

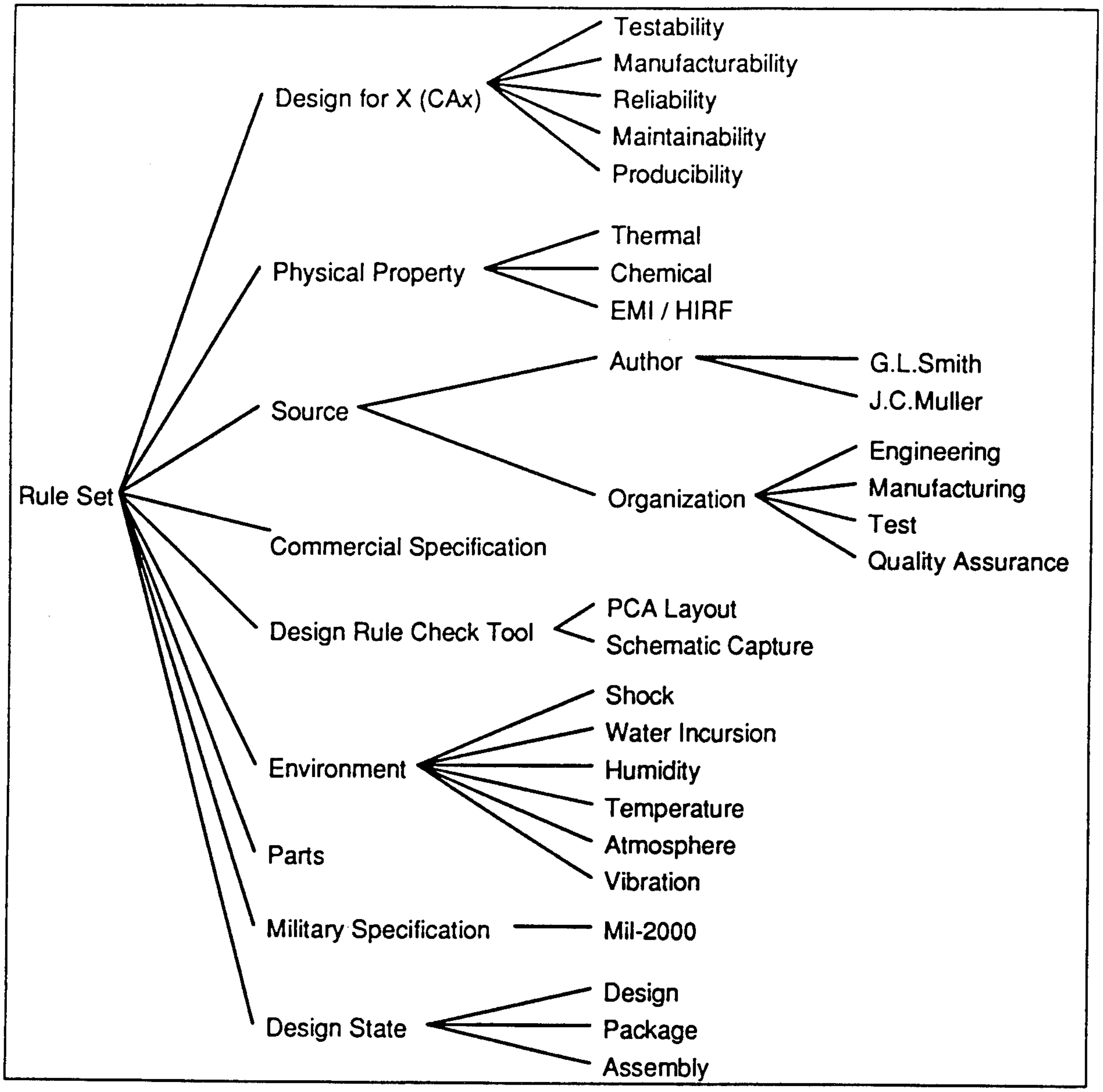
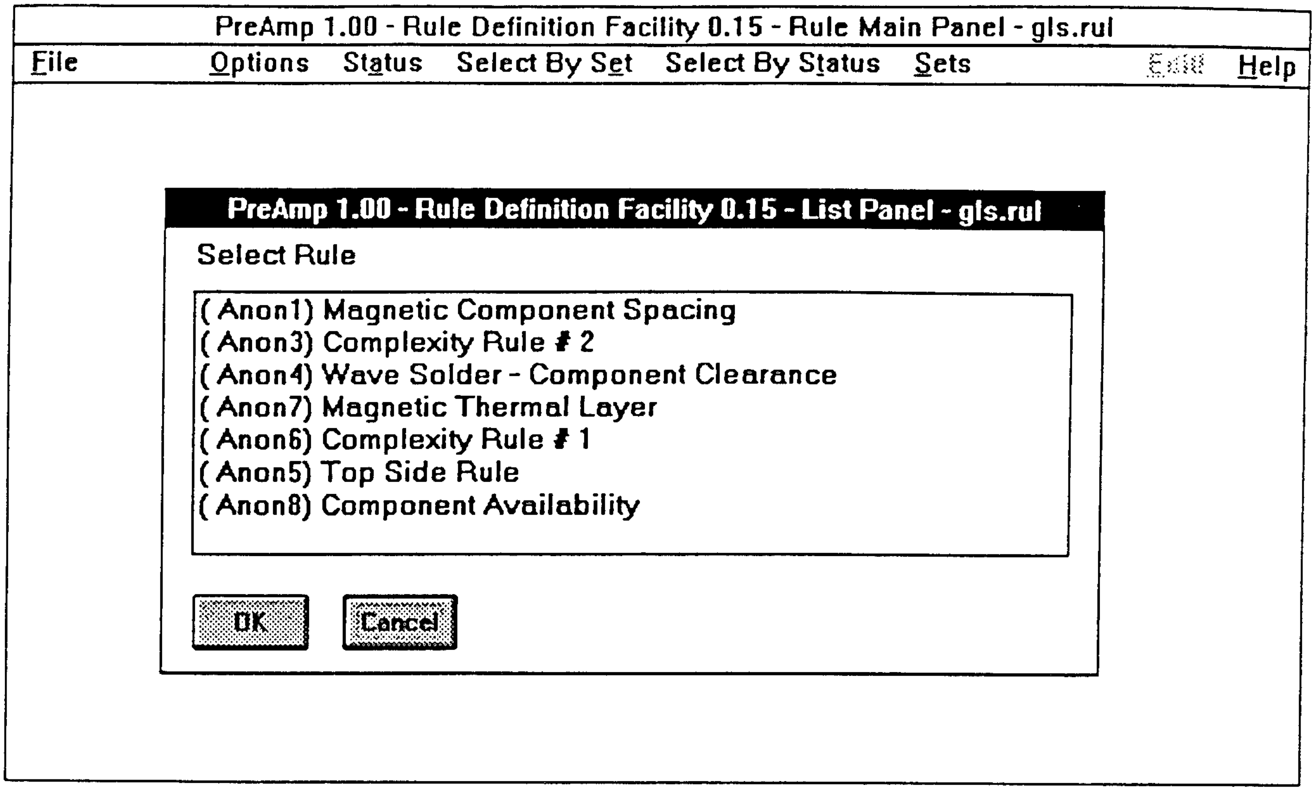


Figure 2-8 Example rule sets and hierarchy.

When selecting an existing rule the **Select By Status** or **Select By Set** menus are used. For this example, an inwork rule will be selected. The following screen is displayed after the inwork rule menu item is selected by using the menu sequence **Select By Status > Inwork(7)**.



PreAmp 1.00 - Rule Definition Facility 0.15 - Rule Main Panel - gls.rul

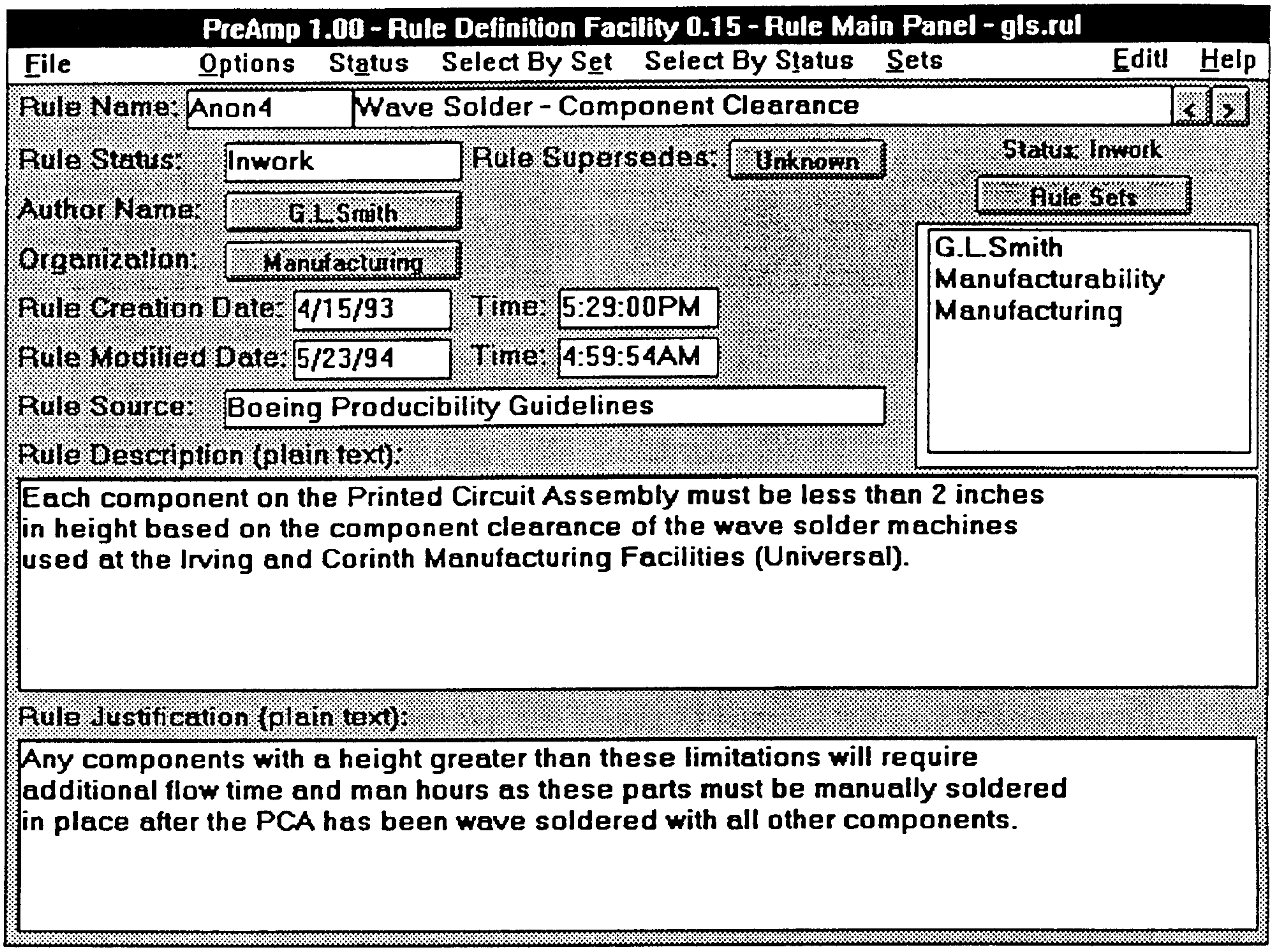
PreAmp 1.00 - Rule Definition Facility 0.15 - List Panel - gls.rul

Select Rule

(Anon1) Magnetic Component Spacing
(Anon3) Complexity Rule # 2
(Anon4) Wave Solder - Component Clearance
(Anon7) Magnetic Thermal Layer
(Anon6) Complexity Rule # 1
(Anon5) Top Side Rule
(Anon8) Component Availability

Figure 2-9 Inwork rule status selection menu.

From the Select Rule menu, the "**(Anon4) Wave Solder - Component Clearance**" rule is selected. Note that in the selection menu each rule name is proceeded by (AnonXX). This information designates the object name of the rule and can be ignored. The rule name does not contain the object name text. When this rule is selected, the rule selection panel will now contain meta information for this rule. The following figure depicts this rule selection.


PreAmp 1.00 - Rule Definition Facility 0.15 - Rule Main Panel - gls.rul
File
Options
Status
Select By Set
Select By Status
Sets
Edit!
Help
<
>
Rule Sets

Rule Name: Anon4 Wave Solder - Component Clearance

Rule Status: Inwork

Rule Supersedes: Unknown

Status: Inwork

Author Name: G.L.Smith

Organization: Manufacturing

G.L.Smith
Manufacturability
Manufacturing

Rule Creation Date: 4/15/93 Time: 5:29:00PM

Rule Modified Date: 5/23/94 Time: 4:59:54AM

Rule Source: Boeing Producibility Guidelines

Rule Description (plain text):

Each component on the Printed Circuit Assembly must be less than 2 inches in height based on the component clearance of the wave solder machines used at the Irving and Corinth Manufacturing Facilities (Universal).

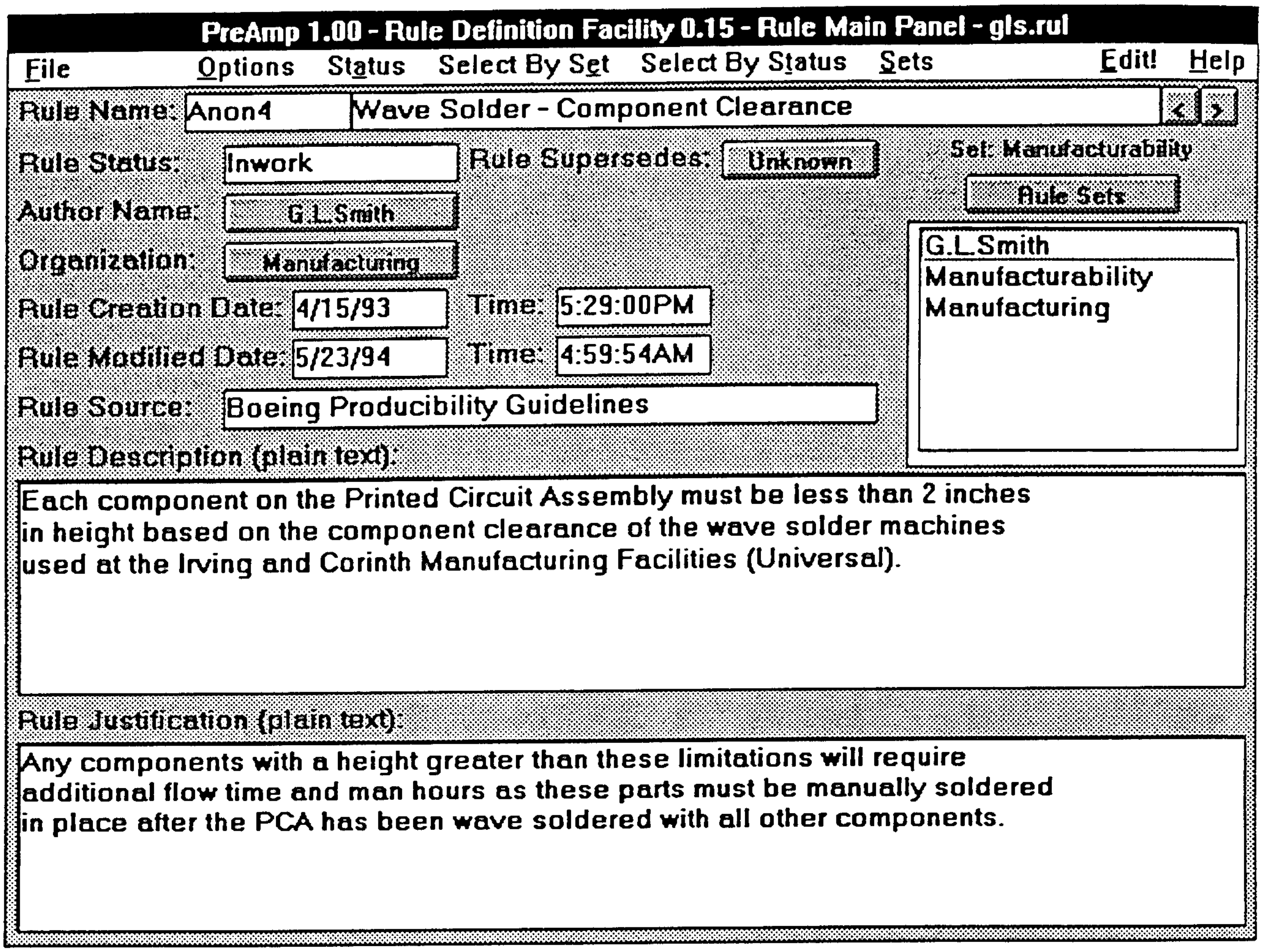
Rule Justification (plain text):

Any components with a height greater than these limitations will require additional flow time and man hours as these parts must be manually soldered in place after the PCA has been wave soldered with all other components.

Figure 2-10 Main window after Inwork rule is selected.

Note in the upper right corner of the panel the displayed text **Status: Inwork**. Since this rule was selected via the menu command **Select By Status**, the displayed text depicts that rule selection is in status mode and the status is **Inwork**. When the **Select Next** or **Select Previous** rule menus commands or button arrows are selected the next/previous rule in the **Inwork** status category will be displayed.

For the next example, the same rule will be selected using menu sequence **Select By Set > Design For X (CAx) > Manufacturability**. The following screen is displayed after the Manufacturability rule menu item is selected.


PreAmp 1.00 - Rule Definition Facility 0.15 - Rule Main Panel - gls.rul
File
Options
Status
Select By Set
Select By Status
Sets
Edit!
Help
<
>
Rule Sets

Rule Name: Anon4 Wave Solder - Component Clearance

Rule Status: Inwork

Rule Supersedes: Unknown

Set: Manufacturability

Author Name: G.L.Smith

Organization: Manufacturing

Rule Creation Date: 4/15/93 Time: 5:29:00PM

Rule Modified Date: 5/23/94 Time: 4:59:54AM

Rule Source: Boeing Producibility Guidelines

G.L.Smith
Manufacturability
Manufacturing

Rule Description (plain text):

Each component on the Printed Circuit Assembly must be less than 2 inches in height based on the component clearance of the wave solder machines used at the Irving and Corinth Manufacturing Facilities (Universal).

Rule Justification (plain text):

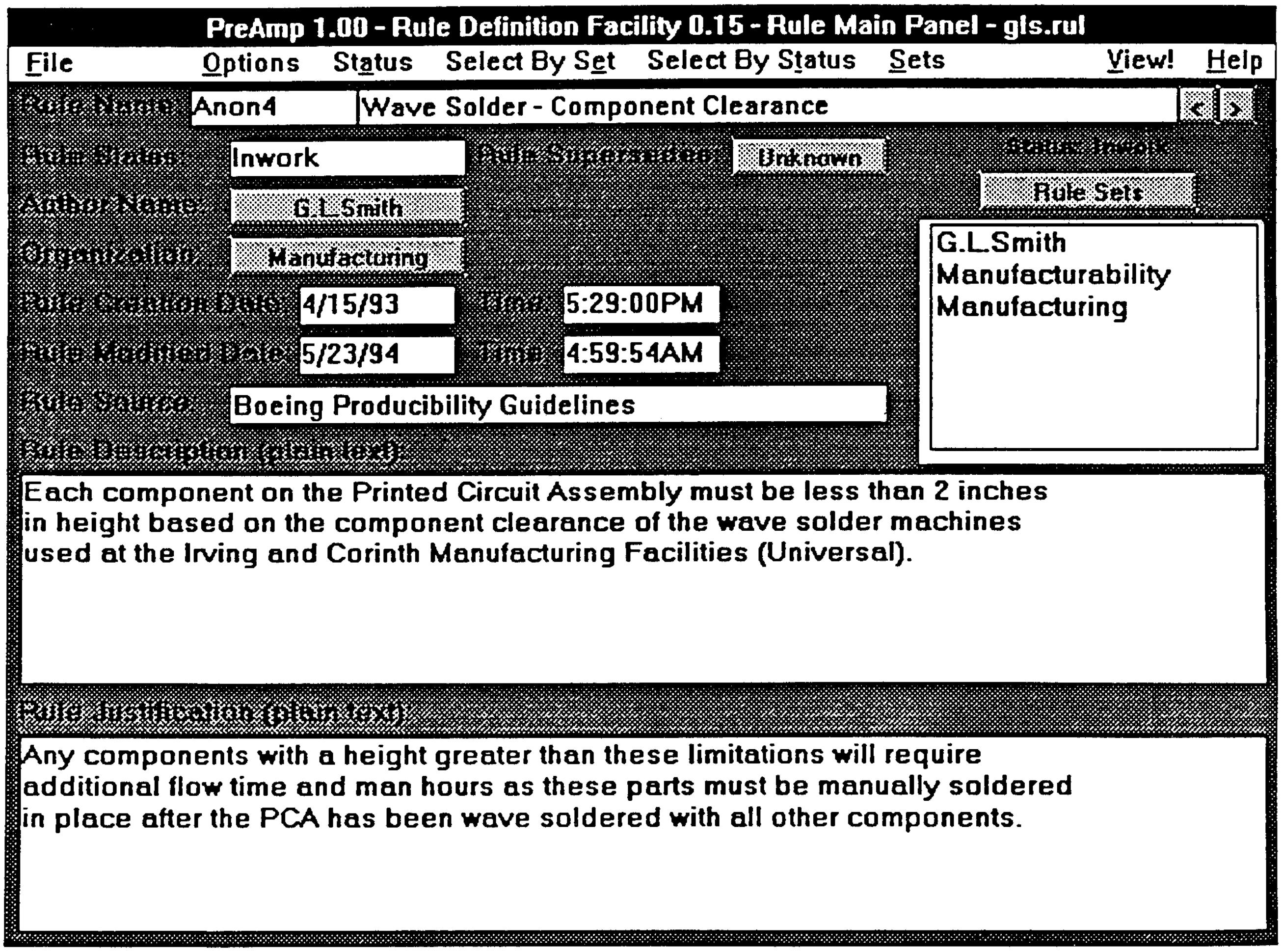
Any components with a height greater than these limitations will require additional flow time and man hours as these parts must be manually soldered in place after the PCA has been wave soldered with all other components.

Figure 2-11 Main window after Manufacturability rule set is selected.

In this example the screen is identical to the last with the exception of the revised displayed text **Set: Manufacturability**. This text depicts that rule selection is in set mode and the set is **Manufacturability**. When the **Select Next** or **Select Previous** rule menu commands or button arrows are selected the next/previous rule in the **Manufacturability** set category will be displayed.

3.0 Defining and Editing Rule Meta-Data

Access to all rule meta-data is provided via the rule main panel now that a rule has been selected. When the main rule panel is in **view** mode (**Edit!** is displayed on the rule main panel menu bar) all panel objects are disabled. Mouse clicking on any of the objects will have no affect. To enter and/or alter any of the data on the rule main panel, the panel must first be switched from **view** mode to **edit** mode. The rule main panel is switched to **edit** mode by selecting **Edit!** on the rule main panel menu bar. Note that when **Edit!** is selected, the rule main panel changes color from grey to red (indicating that changes can be made to the data) and the **Edit!** menu item is replaced with **View!**. All sub windows and buttons are now activated and, unless otherwise stated, are selected by a single left mouse click. The following figure shows the rule main panel in **edit** mode.


PreAmp 1.00 - Rule Definition Facility 0.15 - Rule Main Panel - gls.rul
File
Options
Status
Select By Set
Select By Status
Sets
View!
Help
<
>
Rule Sets

Rule Name: Anon4 Wave Solder - Component Clearance

Rule Status: Inwork

Rule Superseded?: Unknown

Status: Inwork

Author Name: G.L.Smith

Organization: Manufacturing

Rule Creation Date: 4/15/93 Time: 5:29:00PM

Rule Modified Date: 5/23/94 Time: 4:59:54AM

Rule Source: Boeing Producibility Guidelines

G.L.Smith
Manufacturability
Manufacturing

Rule Description (plain text):

Each component on the Printed Circuit Assembly must be less than 2 inches in height based on the component clearance of the wave solder machines used at the Irving and Corinth Manufacturing Facilities (Universal).

Rule Justification (plain text):

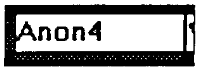
Any components with a height greater than these limitations will require additional flow time and man hours as these parts must be manually soldered in place after the PCA has been wave soldered with all other components.

Figure 3-1 Rule Main Panel in Edit Mode.

3.1 Rule Selection Panel Interface Objects

The rule selection panel has eighteen (18) interface objects that provide information to the user and allow changes to the information. These objects are:

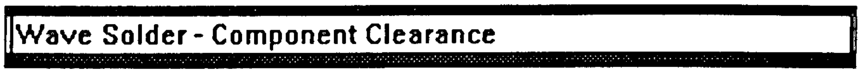
Rule Object Name Display



Anon4

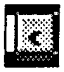
(system controlled - can not be altered directly by user)

Rule Name Edit Box

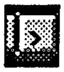


Wave Solder - Component Clearance

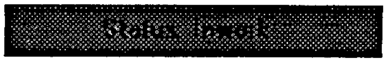
Previous Rule (<) Button



Next Rule (>) Button

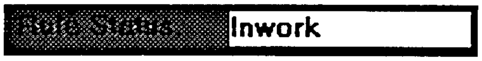


Rule Selection Display Mode and Category



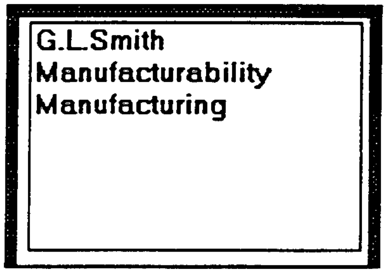
(system controlled - can not be altered directly by user)

Rule Status Display



Inwork

(system controlled - can not be altered directly by user)

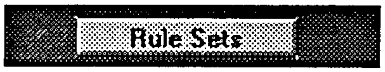


G.L.Smith
Manufacturability
Manufacturing

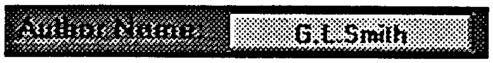
Rule Sets List Box

(system controlled - can not be altered directly by user)

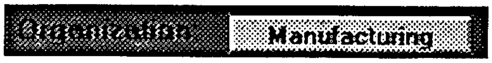
Rule Sets Button



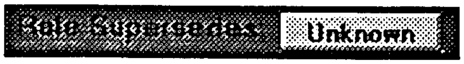
Rule Author Name Button



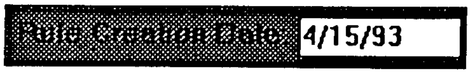
Rule Organization Button



Rule Supersedes Button

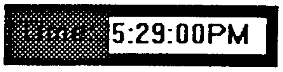


Rule Created Date Display
(system controlled - can not be altered directly by user)



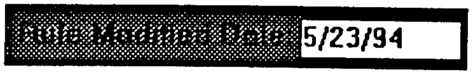
Rule Creation Date: 4/15/93

Rule Created Time Display
(system controlled - can not be altered directly by user)



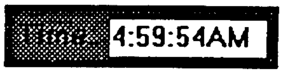
Time: 5:29:00PM

Rule Last Modified Date Display
(system controlled - can not be altered directly by user)



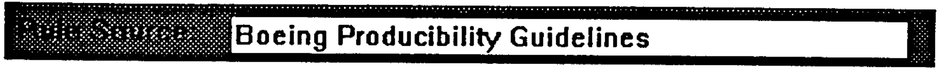
Rule Modified Date: 5/23/94

Rule Last Modified Time Display
(system controlled - can not be altered directly by user)



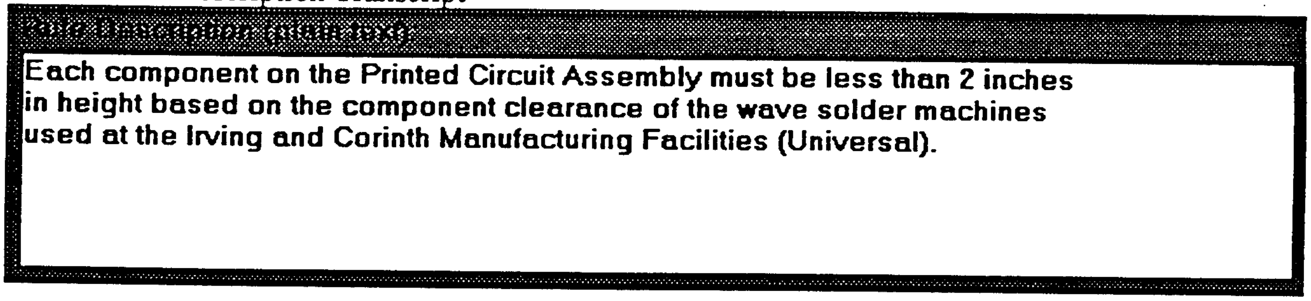
Time: 4:59:54AM

Rule Source Edit Box - Source of rule



Rule Source: Boeing Producibility Guidelines

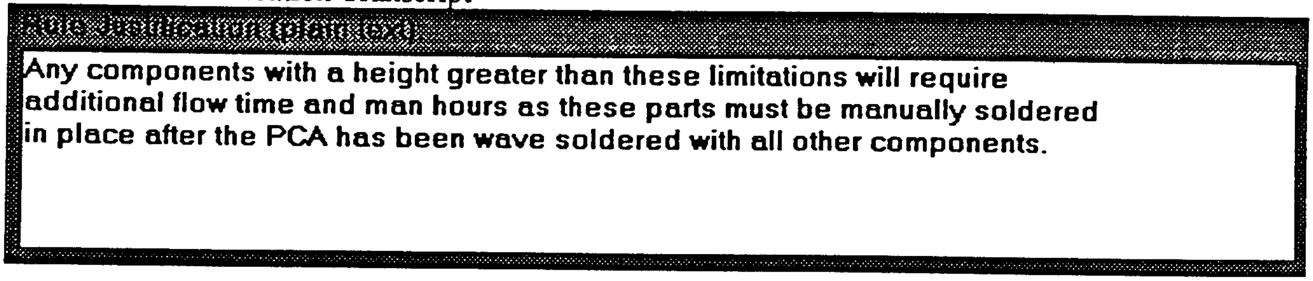
Rule Description Transcript



Rule Description (plain text)

Each component on the Printed Circuit Assembly must be less than 2 inches in height based on the component clearance of the wave solder machines used at the Irving and Corinth Manufacturing Facilities (Universal).

Rule Justification Transcript



Rule Justification (plain text)

Any components with a height greater than these limitations will require additional flow time and man hours as these parts must be manually soldered in place after the PCA has been wave soldered with all other components.

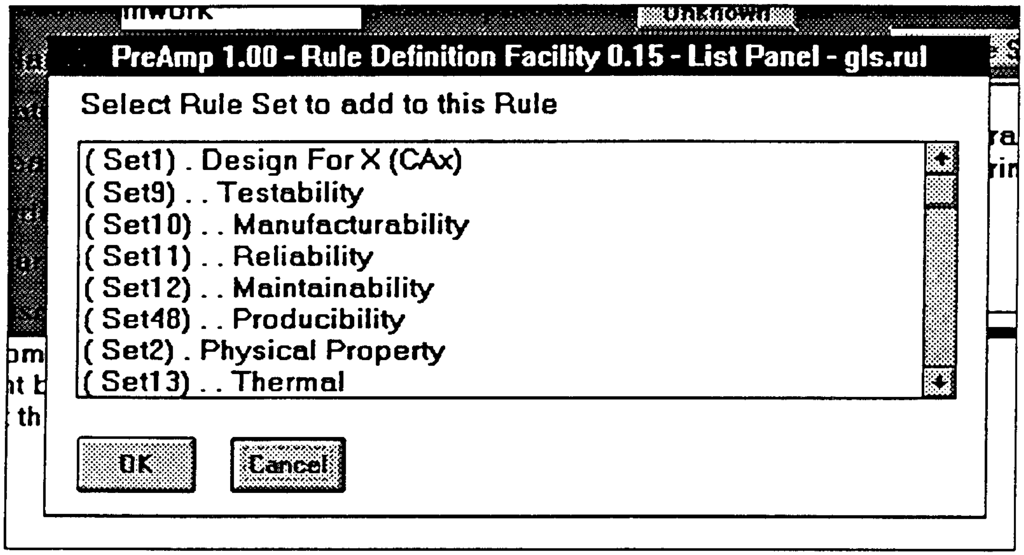
The ten (10) objects that are user selectable are further described below: When any of the user selectable data is altered, the rule modified date and time is updated automatically.

Rule Name Edit Box - A short textual identification for a rule. The rule name can be altered by selecting the current name and typing in a new name.

Previous Rule (<) Button - Selects the previous rule in the current rule mode and category.

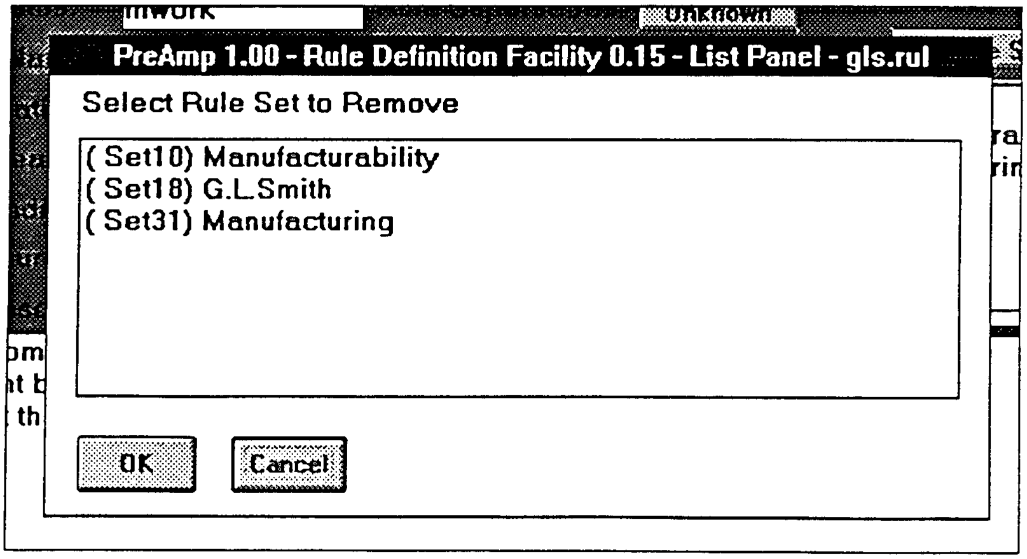
Next Rule (>) Button - Selects the next rule in the current rule mode and category.

Rule Sets Button - A rule set is a group of rules that share common subject or theme. Rule sets can be added by left clicking the rule sets button or removed by right clicking the rule sets button. The figures below depict the rule set add and remove menus. These menus provide for adding and removing rule sets from the selected rule. New rule sets are created, and existing rule sets are renamed and/or deleted using the **Sets** menu selection on the menu bar. When the add rule set command is selected, a menu of currently defined rule sets is provided. The user can select which set the rule is to be added to. When the remove rule set command is selected, a menu of rule sets the rules belongs to is provided. The user can select which set the rule is to be removed from.


PreAmp 1.00 - Rule Definition Facility 0.15 - List Panel - gls.rul
OK
Cancel

Select Rule Set to add to this Rule

(Set1) . Design For X (CAx)
(Set9) . . Testability
(Set10) . . Manufacturability
(Set11) . . Reliability
(Set12) . . Maintainability
(Set48) . . Producibility
(Set2) . Physical Property
(Set13) . . Thermal


PreAmp 1.00 - Rule Definition Facility 0.15 - List Panel - gls.rul
OK
Cancel

Select Rule Set to Remove

(Set10) Manufacturability
(Set18) G.L.Smith
(Set31) Manufacturing

Figure 3-2 Rule set add & remove set command menus.

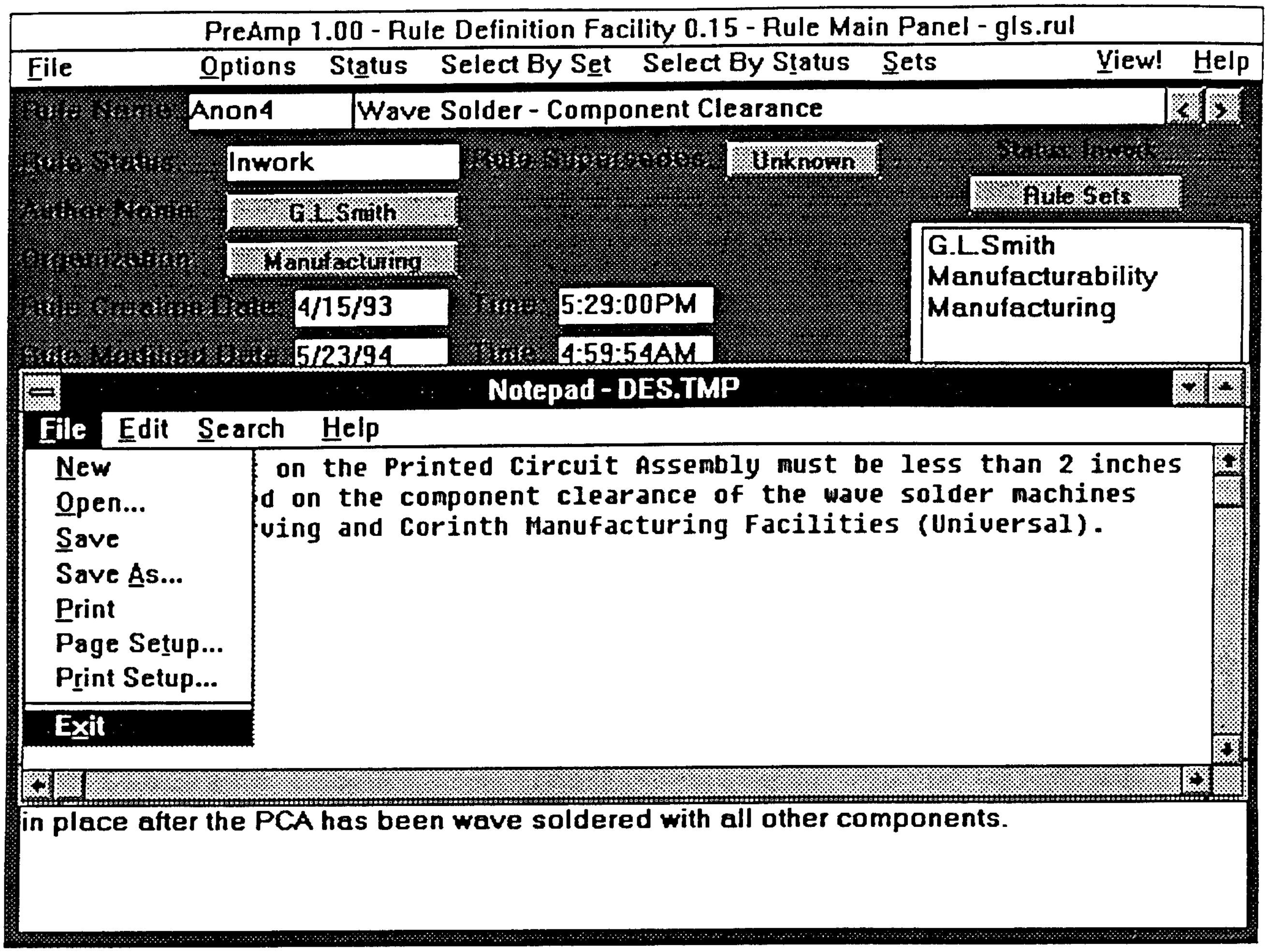
Rule Author Button - The name of the individual entering the rule. The rule author can be altered by selecting the Author Name button and selecting the appropriate name from the author menu. Right clicking on this button removes the current rule author.

Rule Organization Button - The organization of the individual entering the rule. The rule organization can be altered by selecting the Organization button and selecting the appropriate name from the organization menu. Right clicking on this button removes the current organization.

Rule Superseded Button - The rule that this rule supersedes. The superseded rule can be altered by selecting the Supersedes button and selecting the appropriate name from the menu of approved rules. Not only approved rules can be superseded. Right clicking on this button removes the currect superseded rule.

Rule Source Edit Box - Textual identification of the source of the rule (i.e. individual, document, date, etc.). The rule source can be altered by selecting the current source and typing in a new source.

Rule Description Transcript - A textual description of the rule. The rule description can be altered by selecting the current description (by either a left or right click). An edit pad (the Windows Notepad editor) will be provided to aid in the editing of the description text. When editing is complete, select **File > Exit** and confirm that changes are to be saved. (Note: Leaving a description edit pad open when exiting the system or attempting to open more than one description edit pad will result in warning messages exclaiming that fact to be displayed.) The figure below depicts an edit pad opened after the rule description transcript was selected. Note the file menu has been selected and that the Exit option is shown.



PreAmp 1.00 - Rule Definition Facility 0.15 - Rule Main Panel - gls.rul

Rule Name: Anon4 Wave Solder - Component Clearance

Rule Status: Inwork

Rule Supersedes: Unknown

Status: Inwork

Author Name: G.L.Smith

Organization: Manufacturing

Rule Creation Date: 4/15/93 Time: 5:29:00PM

Rule Modified Date: 5/23/94 Time: 4:59:54AM

G.L.Smith
Manufacturability
Manufacturing

```
 on the Printed Circuit Assembly must be less than 2 inches
d on the component clearance of the wave solder machines
ving and Corinth Manufacturing Facilities (Universal).
```

in place after the PCA has been wave soldered with all other components.

Figure 3-3 Edit pad opened with rule description information.

Rule Justification Transcript - A textual justification of the rule. The rule justification can be altered by selecting the current justification (by either a left or right click). An edit pad (The Windows Notepad editor) will be provided to aid in the editing of the justification text. When editing is complete, select **File > Exit** and confirm that changes are to be saved. (Note: Leaving a justification edit pad open when exiting the system or attempting to open more than one justification edit pad will result in warning messages exclaiming that fact to be displayed.)

Note that following the completion meta-data entry the user may select another existing rule by using the '>' button in the upper right and display the next defined rule, or traverse backward with the '<' button to a previous rule. To edit rule components, the **Edit/View Rule Details** command under the **Options** menubar is activated.

3.2 Rule Selection Panel Menubar Commands

The five menu options across the rule selection panel and the command options under them are explained below.

File

New	Create a new rule definition file
Open	Open an existing rule definition file
Save	Save the current rule definition file
Save As	Save the current rule definition file to another file
Setup Printer	N/A
Exit	Exit the Rule Facility

Options

Previous Rule	Select the previous rule
Next Rule	Select the next rule
Create	Create a new rule
Edit/View Rule Details	Edit the selected rule (opens the rule edit panel)
Copy	Make an exact copy of a rule
Delete	Delete a rule

Status

Accept Proposed Rule	Converts a proposed rule to an inwork rule
Reject Proposed Rule	Converts a proposed rule to a rejected rule (rejection criteria is requested) (rejected rules can not be deleted or edited) (rejected rules can not have their status changed)
Submitted Inwork Rule	Converts an inwork rule to a submitted rule (submitted rules can not be deleted or edited)
Approve Submitted Rule	Converts a submitted rule to approved (approved rules can not be deleted or edited) (if the approved rule has defined a rule it supersedes then that rule is converted superseded status)
Disapprove Submitted Rule	Converts a submitted rule to disapproved (disapproved rules can not be deleted or edited) (disapproved rules can not have their status changed)

Select by Set

All Rule Sets *	Displays a menu of set rules

Select by Status

Rejected(x)	Displays a menu of rejected rules
Submitted(x)	Displays a menu of submitted rules
Proposed(x)	Displays a menu of proposed rules
Approved(x)	Displays a menu of approved rules
Disapproved(x)	Displays a menu of disapproved rules
Inwork(x)	Displays a menu of inwork rules
Superseded(x)	Displays a menu of superseded rules

Note: x represents the number of rules in that status.

Sets

Define New	Create a new set
Delete Existing	Remove a rule set from the system
Rename Existing	Rename a rule set

View! / Edit!

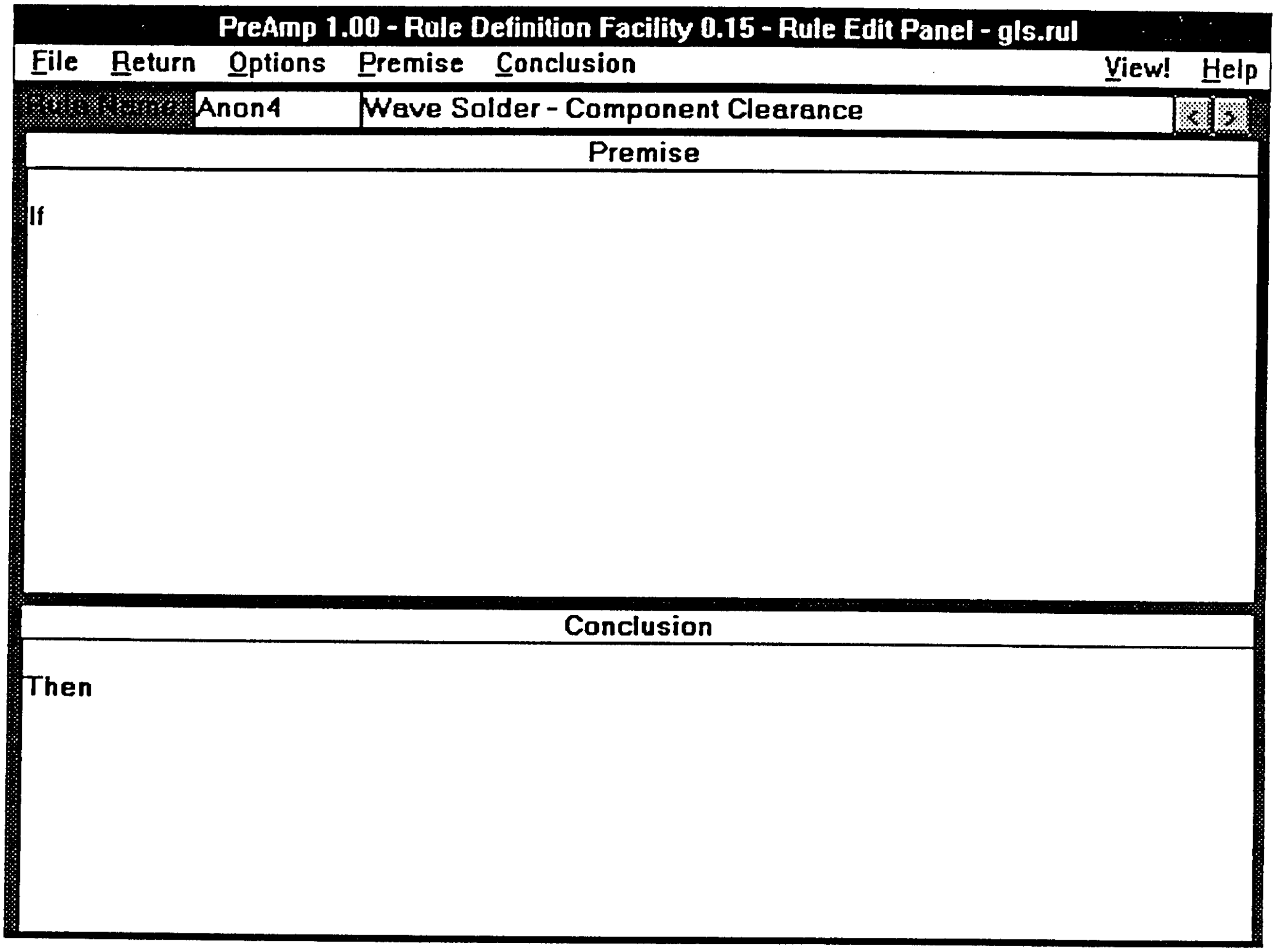
View / Edit Toggle	Toggle panel between view and edit modes

Help

On-Line Manual	N/A
About RuleDef	Copyright and version information.

4.0 Editing Rule Components

Defining the details of a rule to the Rules Definition Facility is decomposed into the definition of a rule premise (rule if conditions) and a rule conclusion (rule consequences that are executed when all if conditions are met). To begin the rule detail or component editing we select the menu sequence **Options > Edit/View Rule Details** from the rule main panel. The following figure shows the Rule Edit Panel that is displayed after the **Edit/ View Rule Details** process is initiated.


PreAmp 1.00 - Rule Definition Facility 0.15 - Rule Edit Panel - gls.rul
File
Return
Options
Premise
Conclusion
View!
Help
<
>
Premise
Conclusion

Rule Name: Anon4 Wave Solder - Component Clearance

If

Then

Figure 4-1 Rule Edit panel when first initiated.

The commands available under the **File** and **Help** menus are identical to those found on the Rule Main Panel. The following figures show commands available options under **File**, **Options, Premise, Conclusion,** and **Edit! / View!** menu options after a rules file has been created or opened.

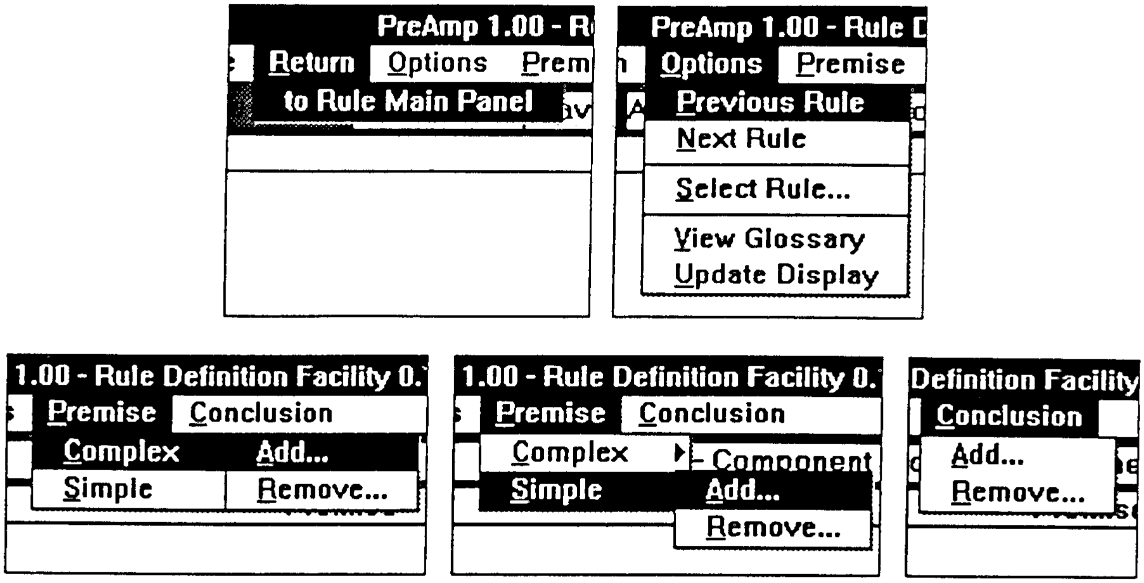
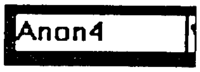


Figure 4-2 Menubar options after rule is selected.

4.1 Rule Edit Panel Interface Objects

The rule edit panel has six (6) interface objects that provide information to the user and allow changes to the information. These objects are:

Rule Object Name Display

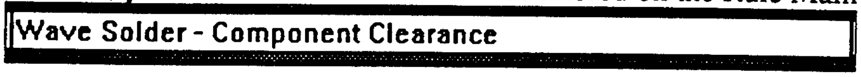


Anon4

(system controlled - can not be altered directly by user)

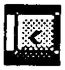
Rule Name Edit Box

(system controlled here - must be altered on the Rule Main Panel)

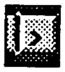


Wave Solder - Component Clearance

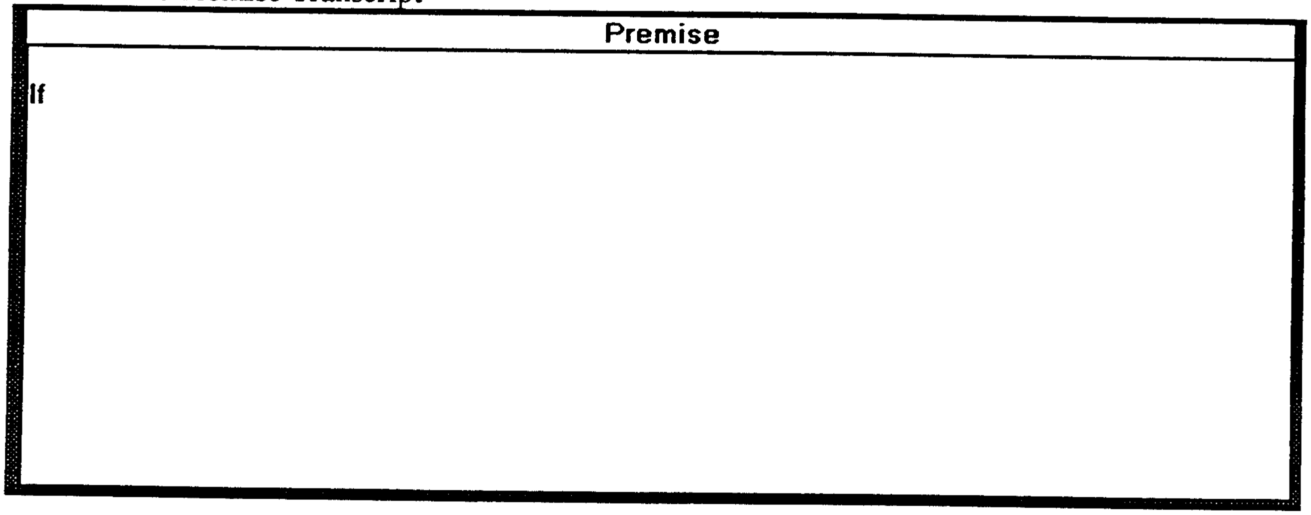
Previous Rule (<) Button



Next Rule (>) Button



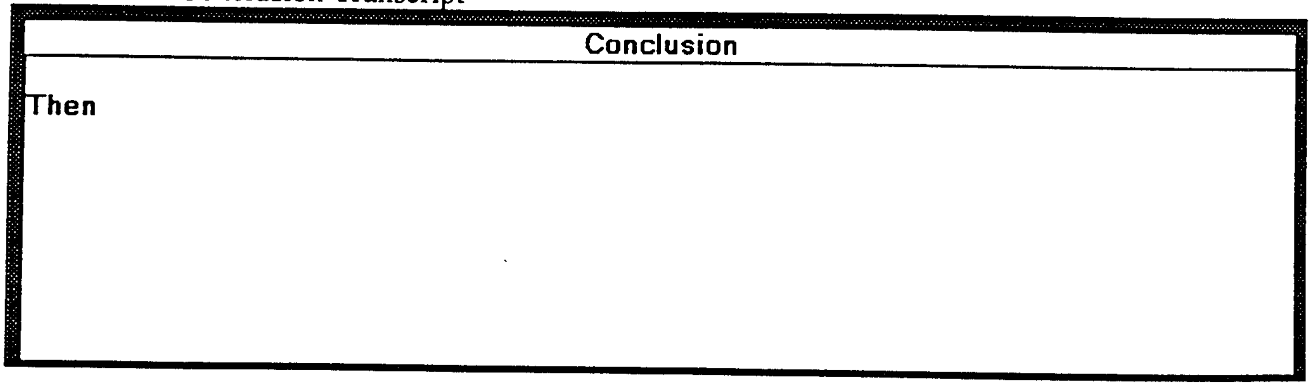
Rule Premise Transcript



Premise

If

Rule Conclusion Transcript



Conclusion

Then

The four (4) objects that are user selectable are further described below:

Previous Rule (<) Button - Selects the previous rule in the current rule mode and category.

Next Rule (>) Button - Selects the next rule in the current rule mode and category.

Rule Premise Transcript - The rule premise is a logical combination of conditions that must be met in order for the rules conclusion to be executed. Left mouse clicking anywhere in this transcript will execute the default premise menu (further discussion to follow).

Rule Conclusion Transcript - The rule conclusion is a logical combination of functions to be executed when the rule premise is found to be true. Left mouse clicking anywhere in this transcript will execute the default conclusion menu (further discussion to follow).

4.2 Rule Edit Panel Menubar Commands

The seven menu options across the rule edit panel and the command options under them are explained below.

File

New	Create a new rule definition file
Open	Open an existing rule definition file
Save	Save the current rule definition file
Save As	Save the current rule definition file to another file
Setup Printer	N/A
Exit	Exit the Rule Facility

Return

to Rule Main Panel	Rule editing is complete, return to the Rule Main Panel

Options

Previous Rule	Select the previous rule
Next Rule	Select the next rule
Select	Select rule by status
View Glossary	Display the database glossary panel
Update Display	N/A

Premise

Complex

Add...	Add a complex premise to this rule
Remove...	Remove a complex premise from this rule

Simple

Add...	Add a simple premise to this rule
Remove...	Remove a simple premise from this rule

Conclusion

Add...	Add a simple conclusion to this rule
Remove...	Remove a simple conclusion from this rule

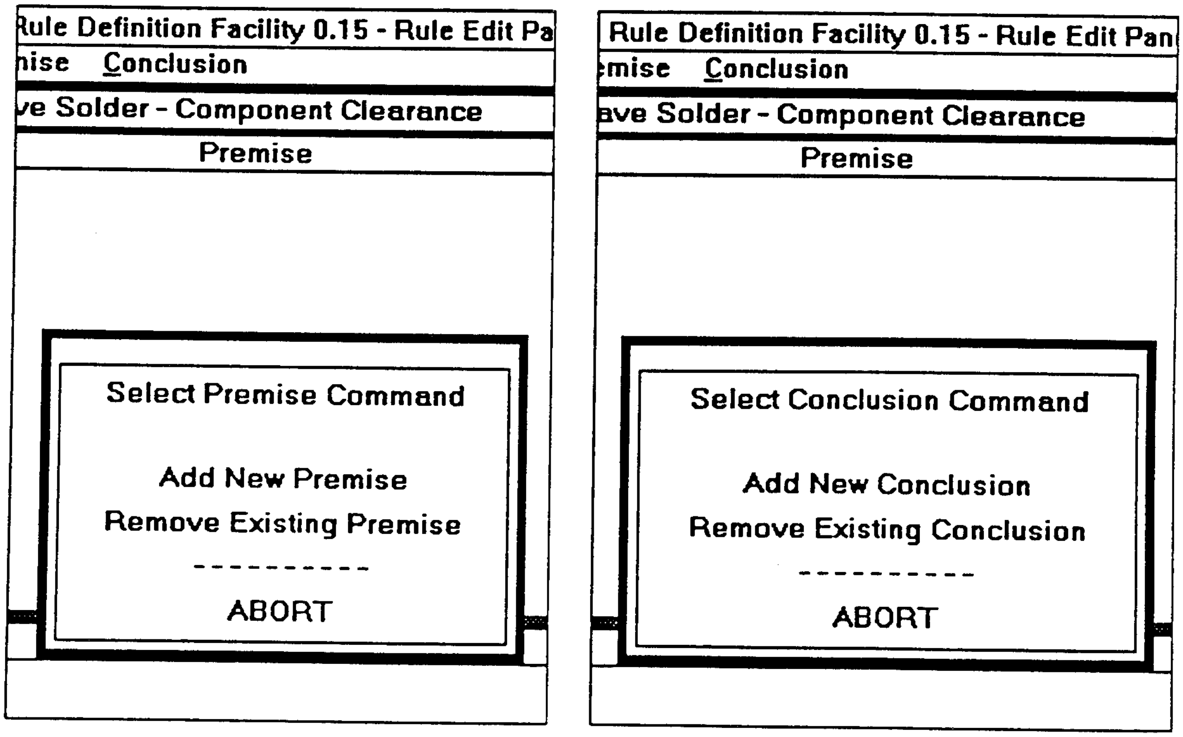
View! / Edit!

View / Edit Toggle	Toggle panel between view and edit modes

Help

On-Line Manual	N/A
About RuleDef	Copyright and version information.

Note that both the premise and conclusion component windows have default menus associated with them. These menus replicate the commands that are available from the menu bar (but may be faster to select). These menus are shown below:



Rule Definition Facility 0.15 - Rule Edit Pa
nise Conclusion
ve Solder - Component Clearance
Premise
Select Premise Command
Add New Premise
Remove Existing Premise

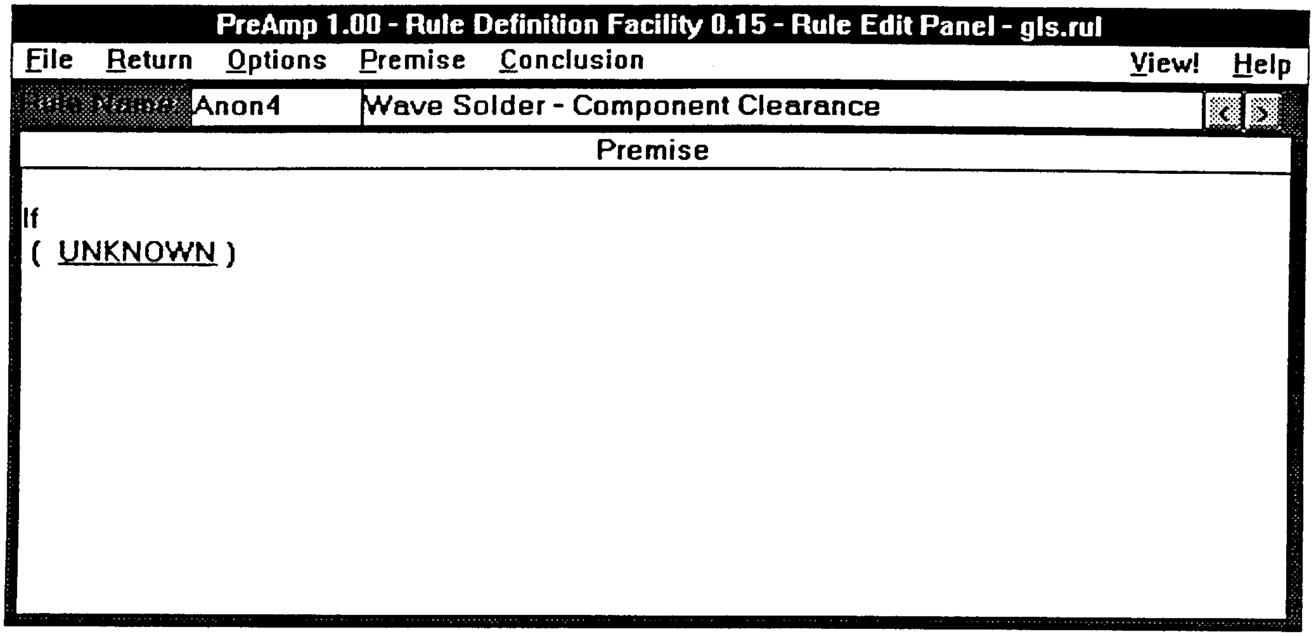
ABORT
Rule Definition Facility 0.15 - Rule Edit Pan
emise Conclusion
ave Solder - Component Clearance
Premise
Select Conclusion Command
Add New Conclusion
Remove Existing Conclusion

ABORT

Figure 4-3 Premise / Conclusion component menus.

4.3 Defining the Rule Premise Component

All rules must have a least one complex premise. If the logic in a rule requires it, a rule may have more than one complex premise. For a rule conclusion component to execute any one of the complex premises must evaluate to true. Select the premise transcript window and click on the menu item **Add New Premise.** A complex premise will be added to the selected rule. The following figure shows the results of this addition:



PreAmp 1.00 - Rule Definition Facility 0.15 - Rule Edit Panel - gls.rul

Rule Name: Anon4 Wave Solder - Component Clearance

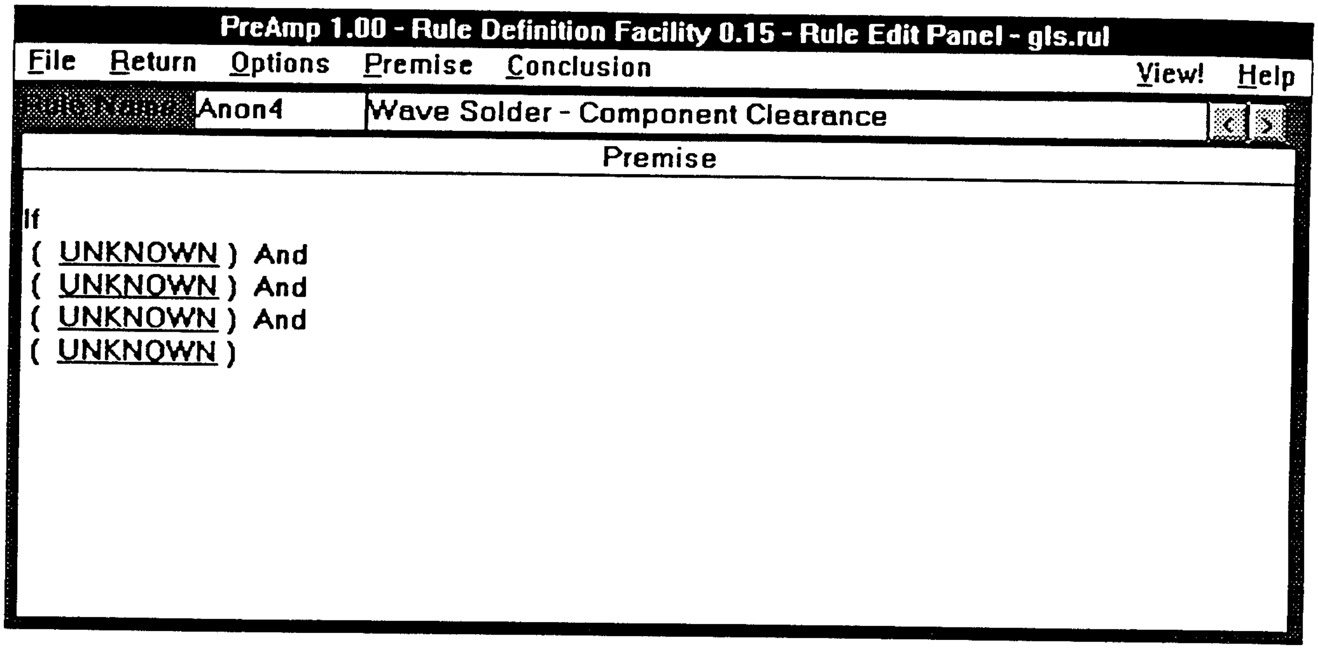
Premise

If
(UNKNOWN)

Figure 4.3-1 Single Complex Premise.

When two or more complex premises are added to a rule, the position of complex premise to be added will be requested. The order of complex premises is important in writing high performance rules. If any one of a rules complex premises are evaluated to true then the rule will be true. Complex premises that are most often evaluated to true should be placed at the beginning of the rule premise. Complex premises that are least often evaluated to true should be placed at the end of the rule premise.

All complex premises must have a least one simple premise. If the logic in a rule requires it, a complex premise may have more than one simple premise. For a rule conclusion component to execute any one of the complex premises must be true. For a complex premise to be true all of its simple premises must be true. Three more simple premises will be added to the selected rule. The following figure shows the results of this addition:


PreAmp 1.00 - Rule Definition Facility 0.15 - Rule Edit Panel - gls.rul
File
Return
Options
Premise
Conclusion
View!
Help
Rule Name
Premise

Anon4

Wave Solder - Component Clearance

If
(UNKNOWN) And
(UNKNOWN) And
(UNKNOWN) And
(UNKNOWN)

Figure 4.3-2 Single Complex Premise with Four (4) Simple Premises.

When two or more simple premises are added a complex premise the position of the simple premise to be added will be requested. The order of simple premises is important in writing high performance rules. The most specific simple premises should be located at the front of a complex premise. The most general simple premises should be located at the back of a complex premise.

Typically, logical functions are initially added to simple premises. As an example, it is common to determine if one value is greater than another value or that the value of an objects attribute is true or false.

Note that any underlined text and/or text displayed in red in a premise transcript is selectable via a left mouse click. Context sensitive menus will be displayed when this hot text is selected. When the UNKNOWN is selected in the first simple premise the following menu is provided:

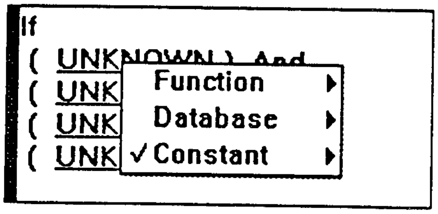


Figure 4.3-3 Premise Function Menu.

This menu provides menu items for adding Premise Functions, Database Functions, and Constants to the premise component.

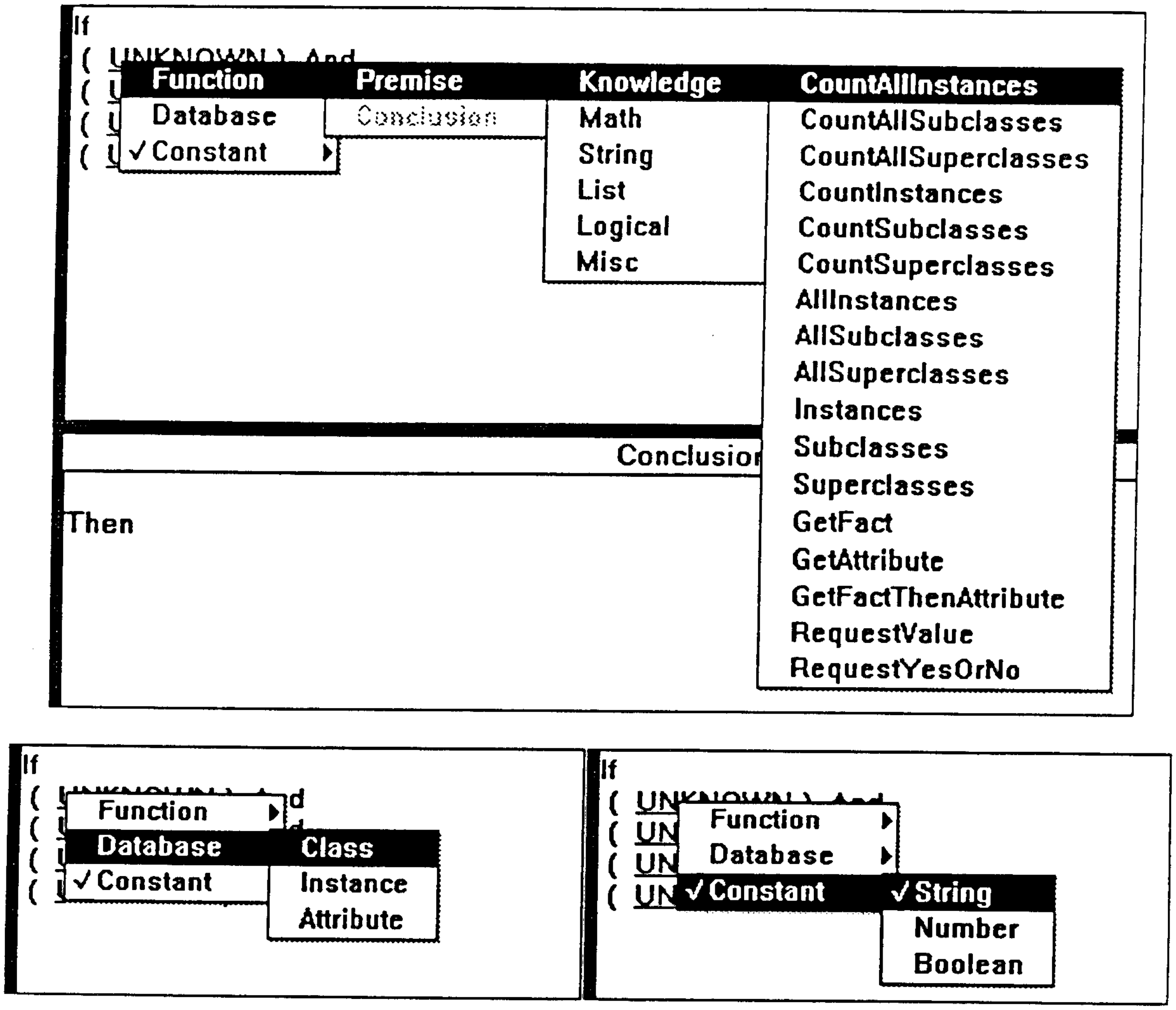
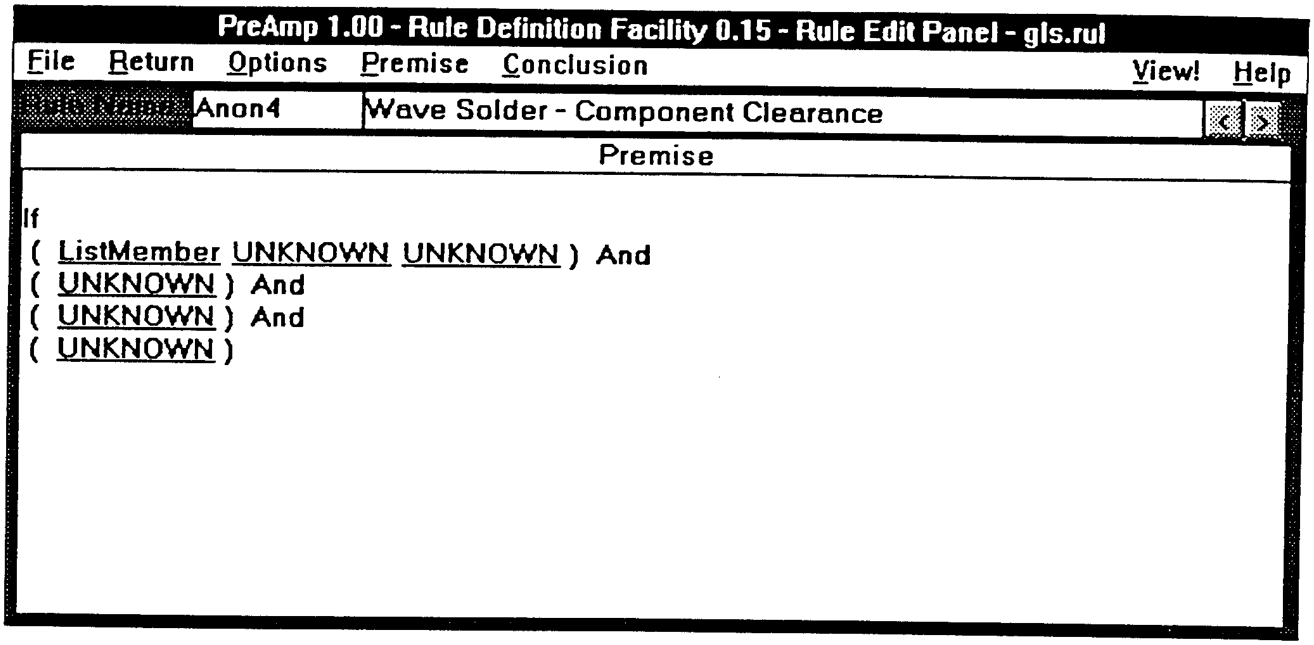


Figure 4.3-4 Premise Function Menu Paths.

For the first simple premise the logical function **ListMember** will be added. The intent of this simple premise will be to determine if a production line (that we are considering) is a member of the list of production lines (that have been selected for consideration). The **ListMember** function will return True of False depending whether or not an item is a member of a list. Using the **Function > Premise > Logical > ListMember** menu path, **ListMember** is selected.

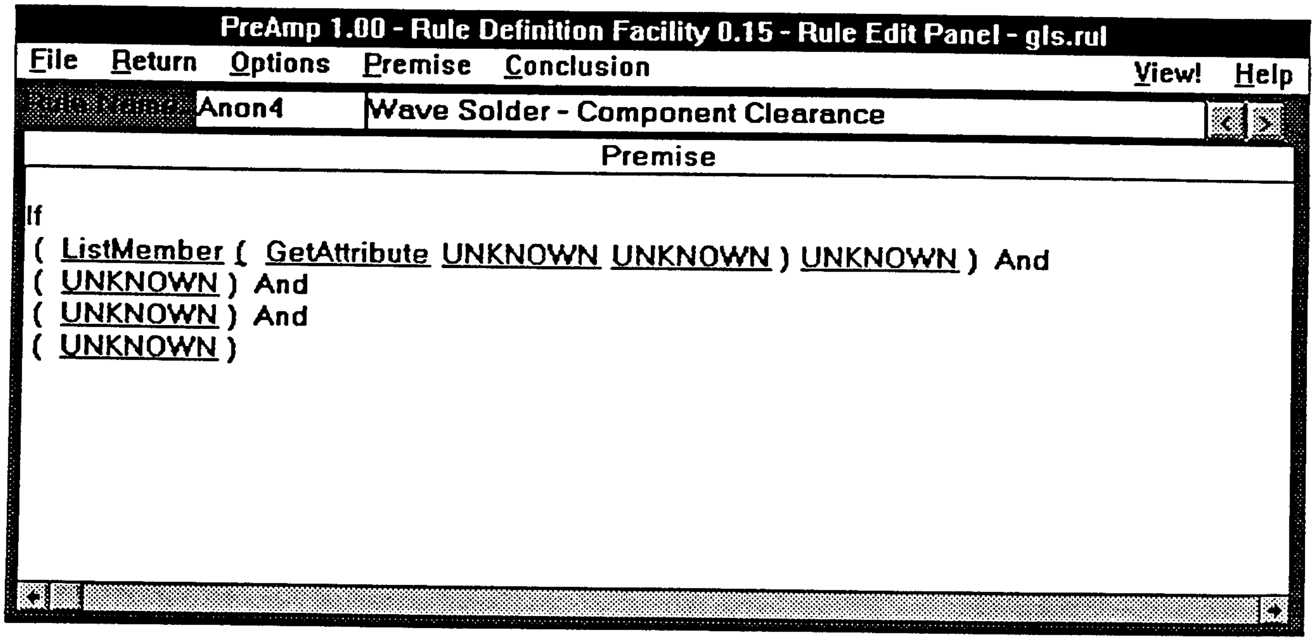

PreAmp 1.00 - Rule Definition Facility 0.15 - Rule Edit Panel - gls.rul
File Return Options Premise Conclusion View! Help
Rule Name:
Premise

Anon4 Wave Solder - Component Clearance

```
If
( ListMember UNKNOWN UNKNOWN ) And
( UNKNOWN ) And
( UNKNOWN ) And
( UNKNOWN )
```

Figure 4.3-5 ListMember Function Added to First Simple Premise.

The first argument to the **ListMember** function is a list of production lines. This information is retrieved from the database by clicking on the first UNKNOWN and following the menu path: **Function > Premise > Knowledge > GetAttribute.**

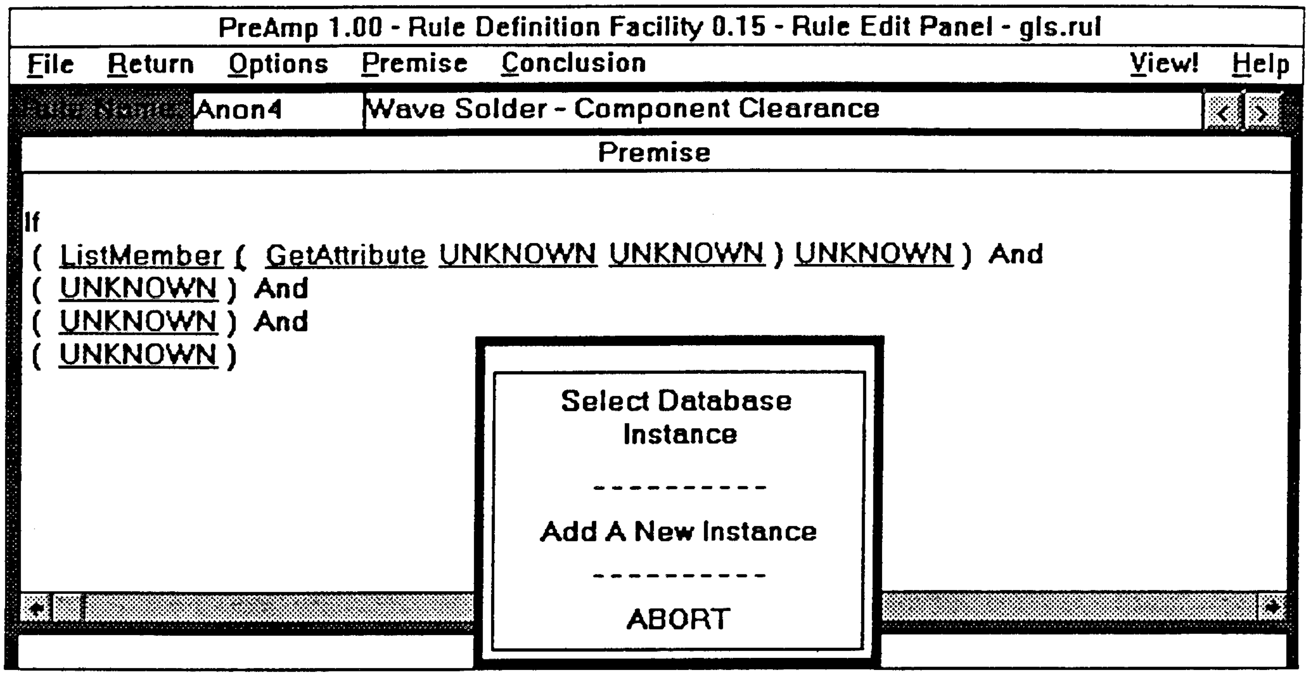

PreAmp 1.00 - Rule Definition Facility 0.15 - Rule Edit Panel - gls.rul
File Return Options Premise Conclusion View! Help
Rule Name:
Premise

Anon4 Wave Solder - Component Clearance

```
If
( ListMember ( GetAttribute UNKNOWN UNKNOWN ) UNKNOWN ) And
( UNKNOWN ) And
( UNKNOWN ) And
( UNKNOWN )
```

Figure 4.3-6 GetAttribute Function Added as First Argument of ListMember.

The first argument to the **GetAttribute** function is an instance of a database class. An instance can be inserted by selecting the first **UNKNOWN** and following the menu path: **Function > Database > Instance.**



PreAmp 1.00 - Rule Definition Facility 0.15 - Rule Edit Panel - gls.rul

Anon4

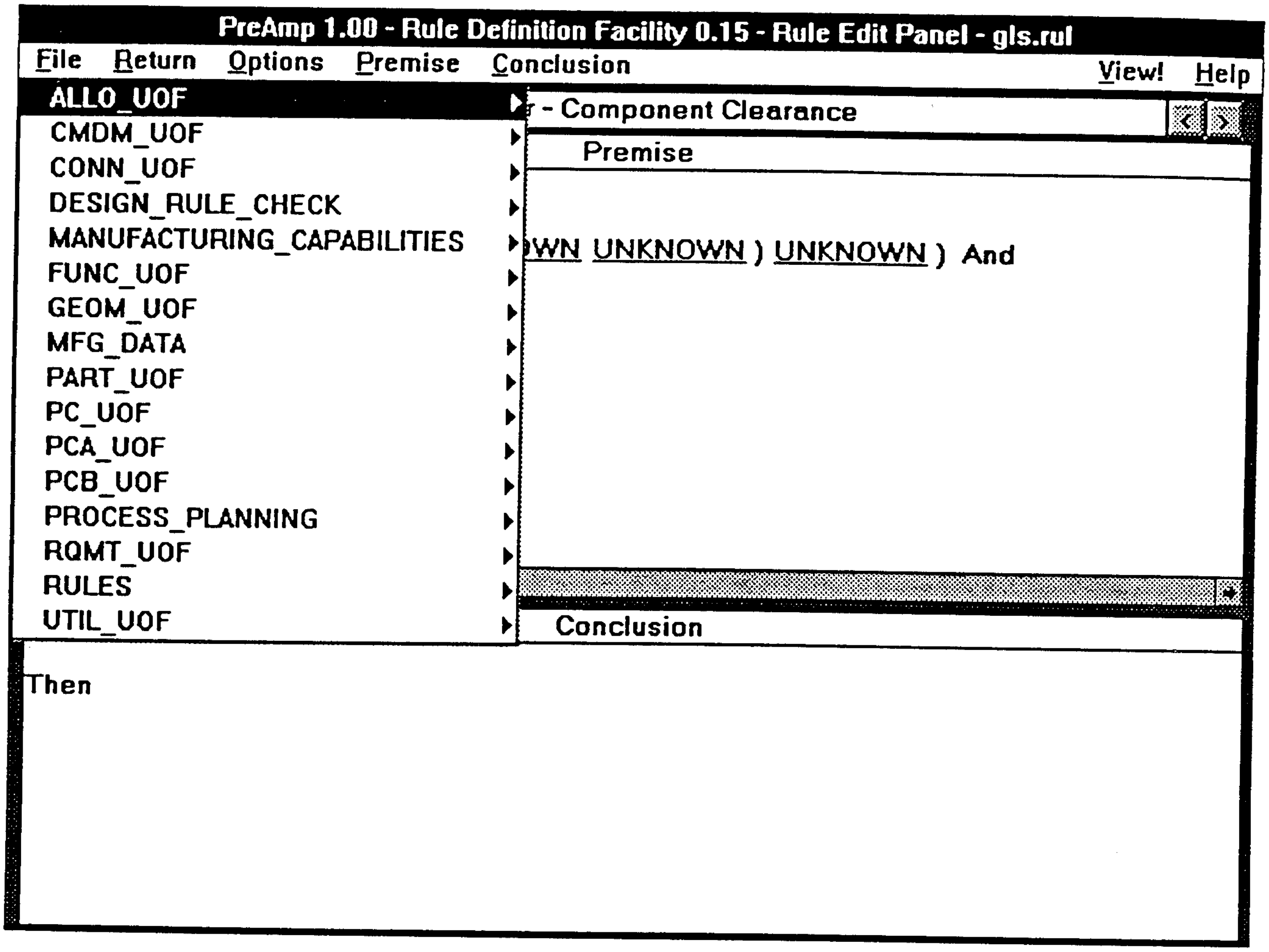
Wave Solder - Component Clearance

Premise

```
If
( ListMember ( GetAttribute UNKNOWN UNKNOWN ) UNKNOWN ) And
( UNKNOWN ) And
( UNKNOWN ) And
( UNKNOWN )
```

Figure 4.3-7 Database Instance Menu.

Select the **Add A New Instance** selection. Following this selection a cascading database schema menu will be displayed.



WN UNKNOWN) UNKNOWN) And

Then

Figure 4.3-8 Database Schema and Object Menu.

Under CMDM_UOF (configuration management unit of functionality) select the product menu item.

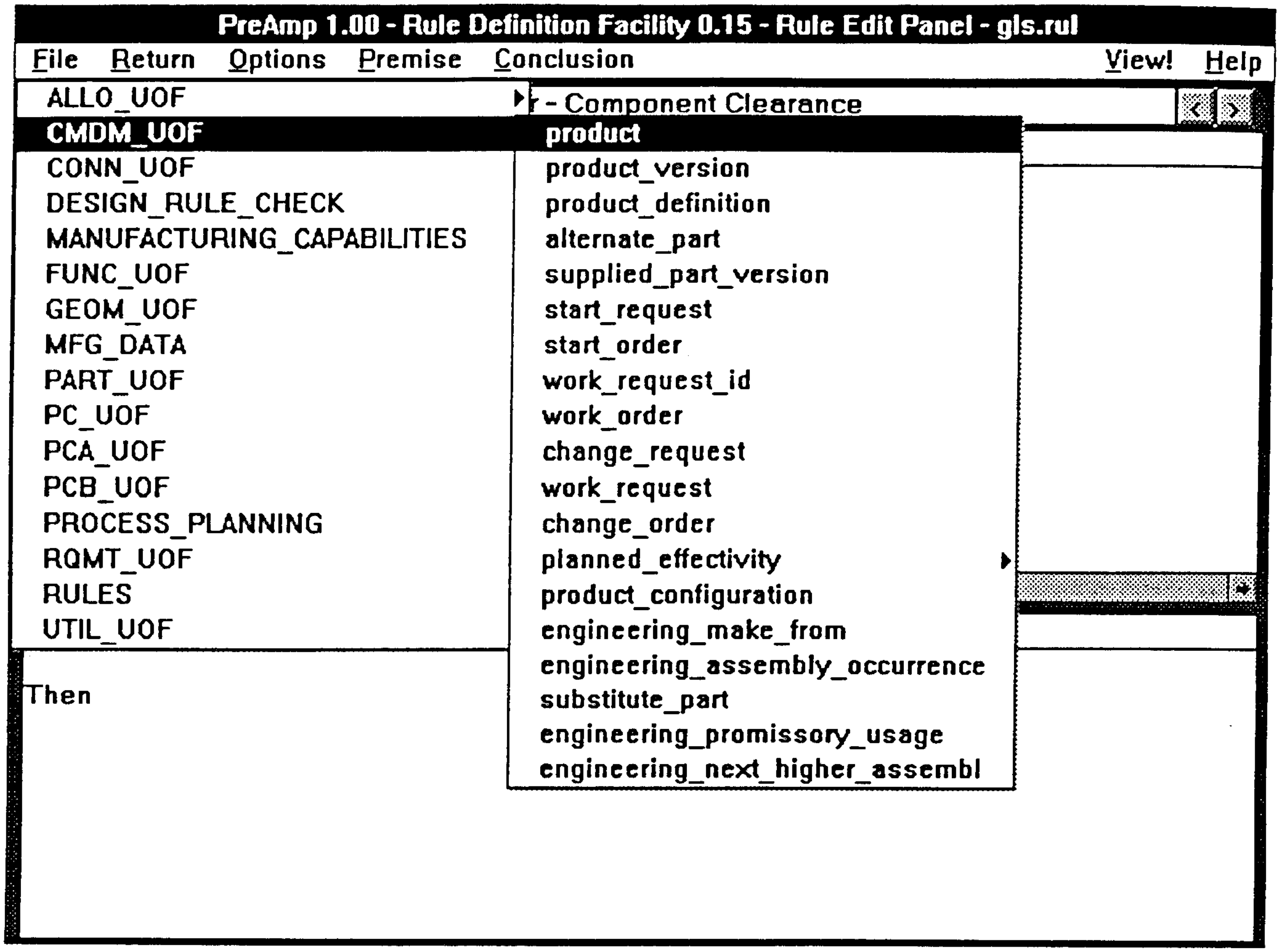
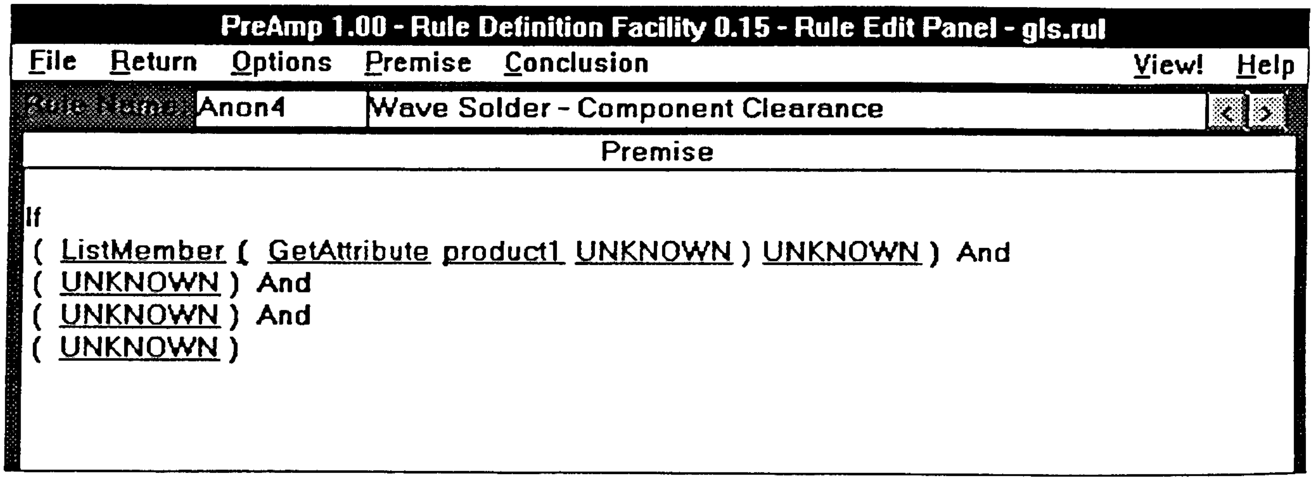


Figure 4.3-9 Expanded Database Schema Menu.

When the **product** menu item is selected the instance menu will reappear with **product1** now added to the list of selections. Select **product1**.


PreAmp 1.00 - Rule Definition Facility 0.15 - Rule Edit Panel - gls.rul
File
Return
Options
Premise
Conclusion
View!
Help

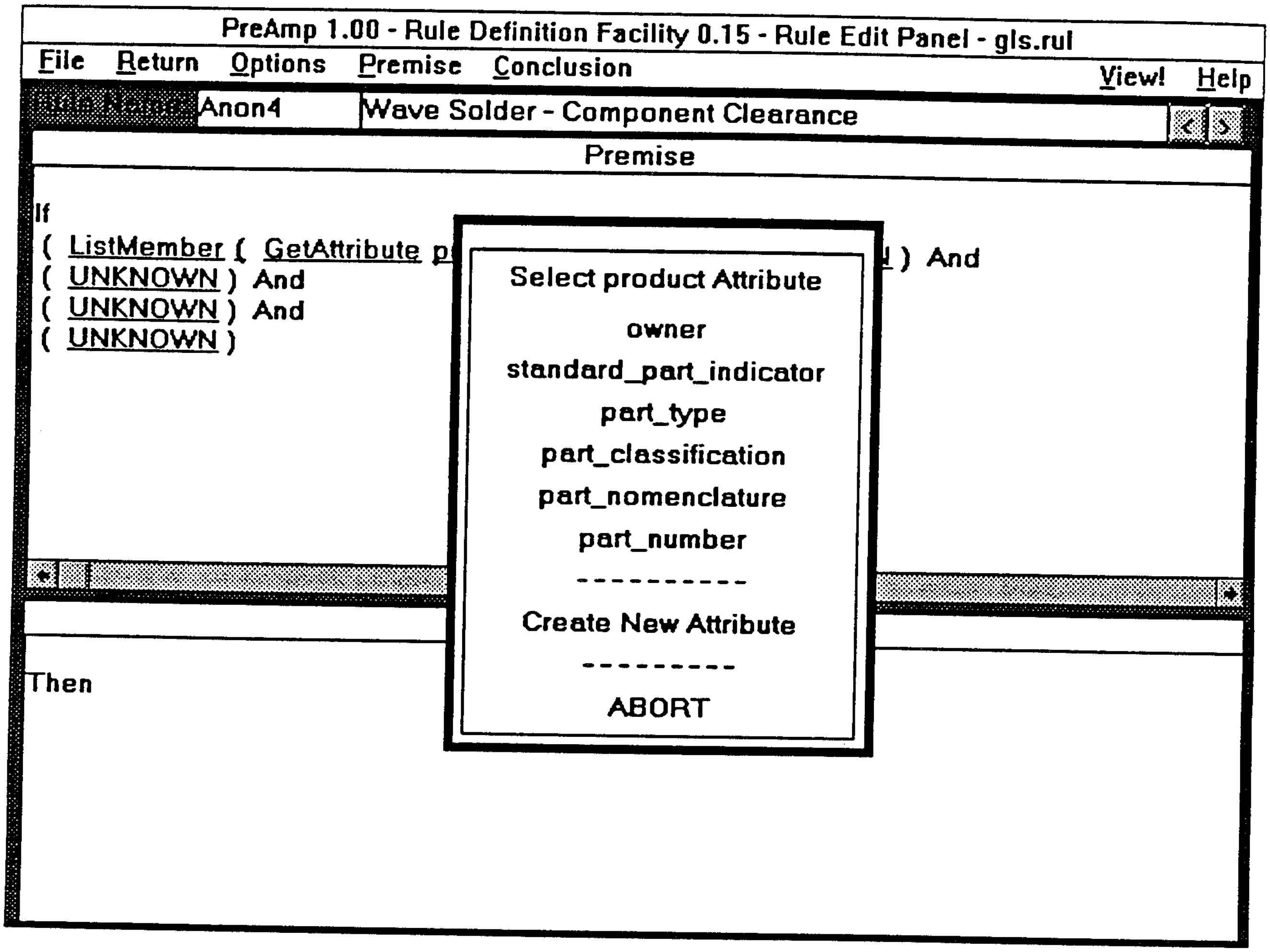
Rule Name: Anon4 Wave Solder - Component Clearance

Premise

```
If
( ListMember ( GetAttribute product1 UNKNOWN ) UNKNOWN ) And
( UNKNOWN ) And
( UNKNOWN ) And
( UNKNOWN )
```

Figure 4.3-10 Database Instance Added to GetAttribute.

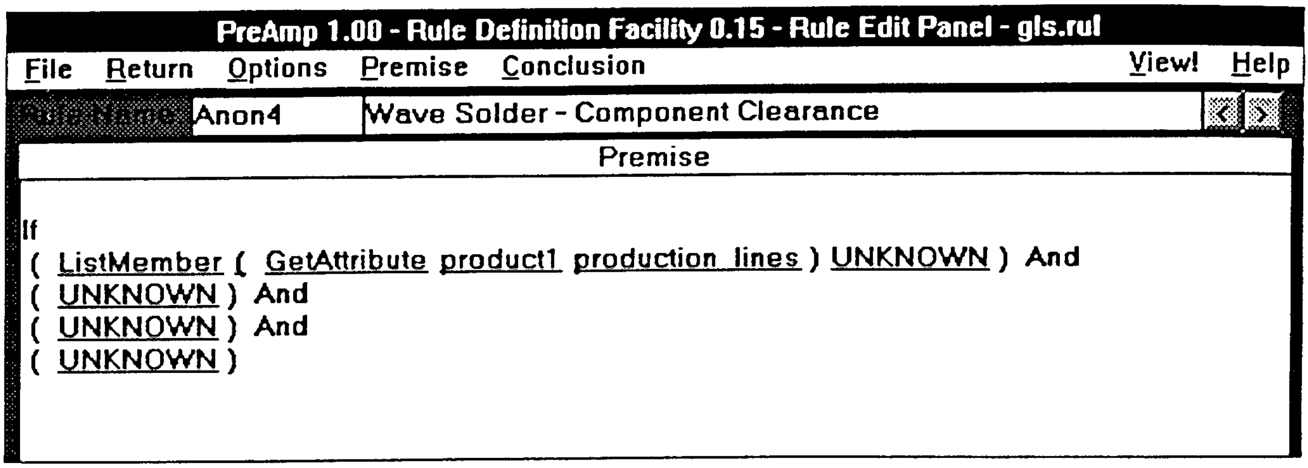
The second argument to the **GetAttribute** function is the attribute name for the instance, object, or class that was defined by the first argument. Select **UNKNOWN** (last argument in **GetAttribute** command) and follow the premise menu path: **Function > Database > Attribute**. The following attribute list will be displayed:



```
If
( ListMember ( GetAttribute p                    ) And
( UNKNOWN ) And
( UNKNOWN ) And
( UNKNOWN )
```

Figure 4.3-11 Attribute List for Database Object product.

Select **production_lines** from the menu if available. Otherwise select **Create New Attribute**, and enter **production_lines** when requested. The simple premise now looks as follows:

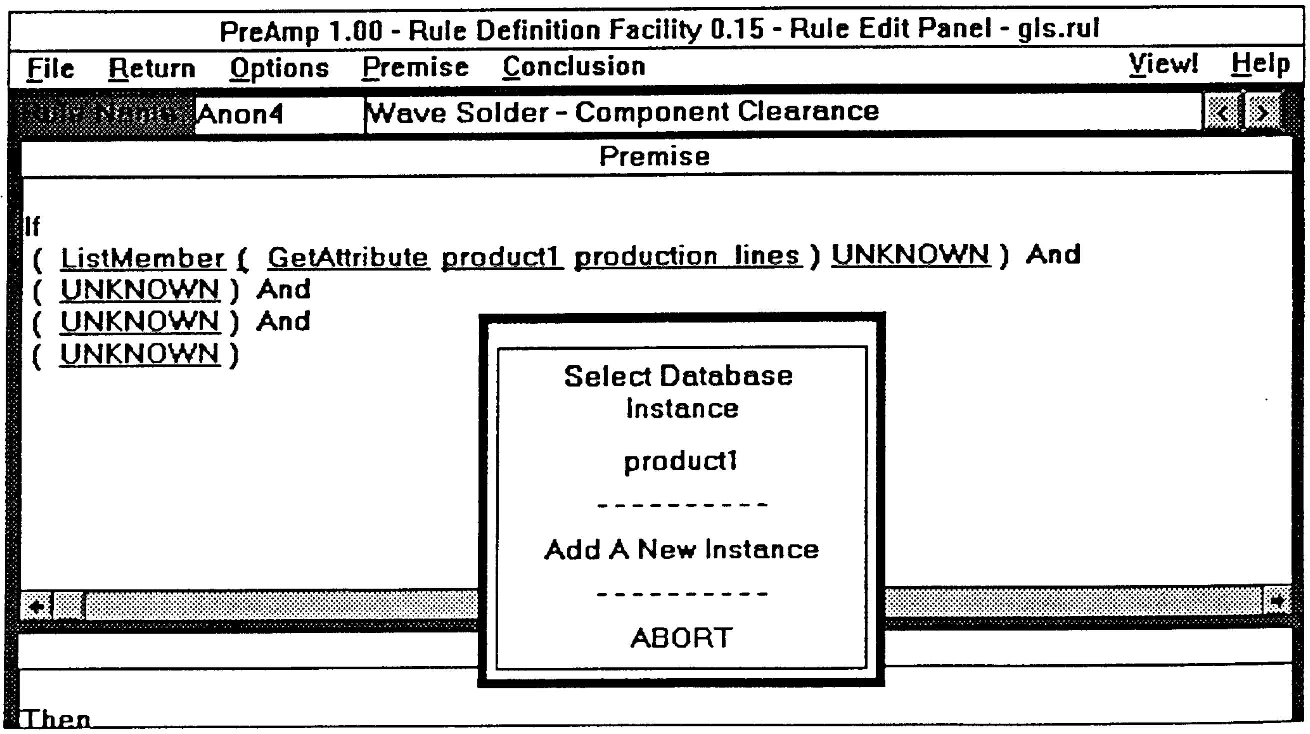

PreAmp 1.00 - Rule Definition Facility 0.15 - Rule Edit Panel - gls.rul
File
Return
Options
Premise
Conclusion
View!
Help
Premise

Rule Name: Anon4 Wave Solder - Component Clearance

```
If
( ListMember ( GetAttribute product1 production_lines ) UNKNOWN ) And
( UNKNOWN ) And
( UNKNOWN ) And
( UNKNOWN )
```

Figure 4.3-11 production_lines Attribute Added to GetAttribute.

The second argument to the **ListMember** function is a list item. An instance from the class of all **production_lines** will used as the list item. This instance is added by selecting **Function > Database > Instance** menu path. When the **Select Database Instance** menu is displayed, select **Add A New Instance.**

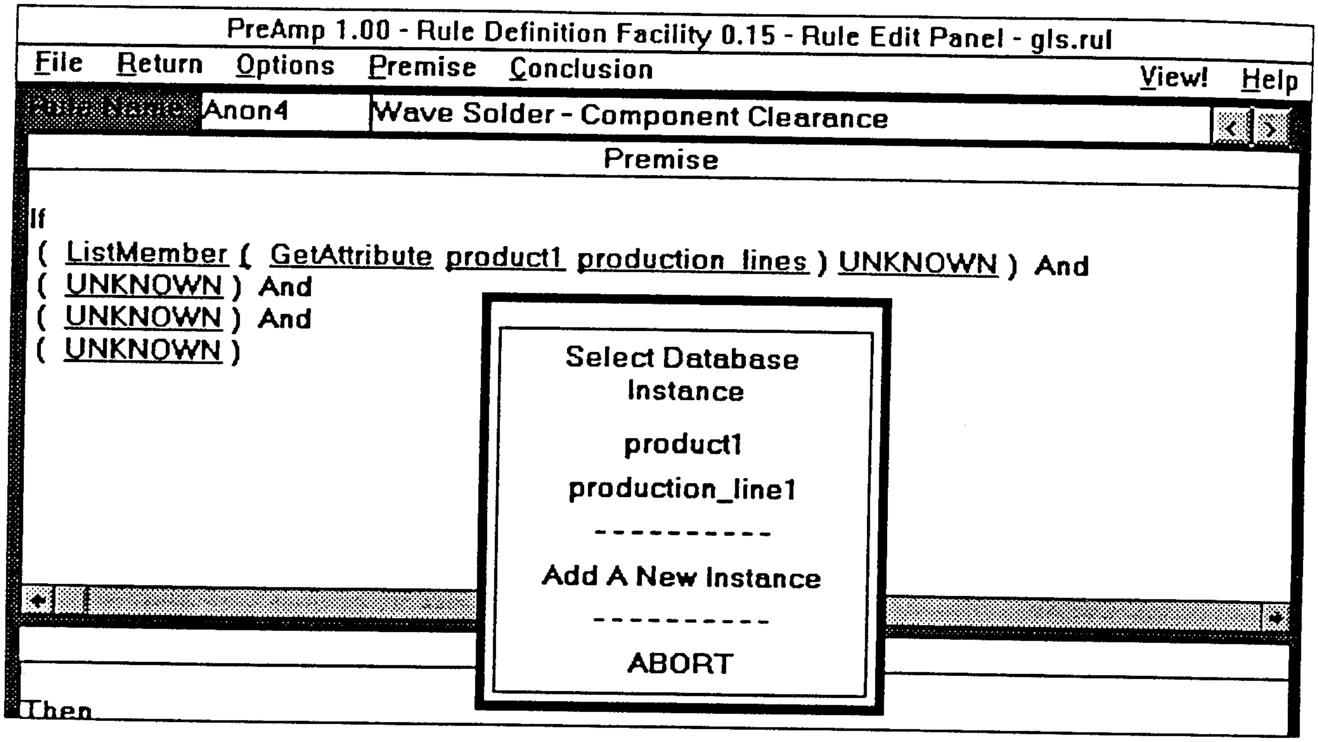


Rule Name: Anon4 Wave Solder - Component Clearance

```
If
( ListMember ( GetAttribute product1 production_lines ) UNKNOWN ) And
( UNKNOWN ) And
( UNKNOWN ) And
( UNKNOWN )
```

Figure 4.3-12 Select Database Instance Menu.

Select the MANUFACTURING_CAPABILITIES > **production_line** items when the cascading database schema menu is displayed.


PreAmp 1.00 - Rule Definition Facility 0.15 - Rule Edit Panel - gls.rul
File Return Options Premise Conclusion View! Help
Rule Name:
Premise
Then
Select Database Instance
Add A New Instance
ABORT

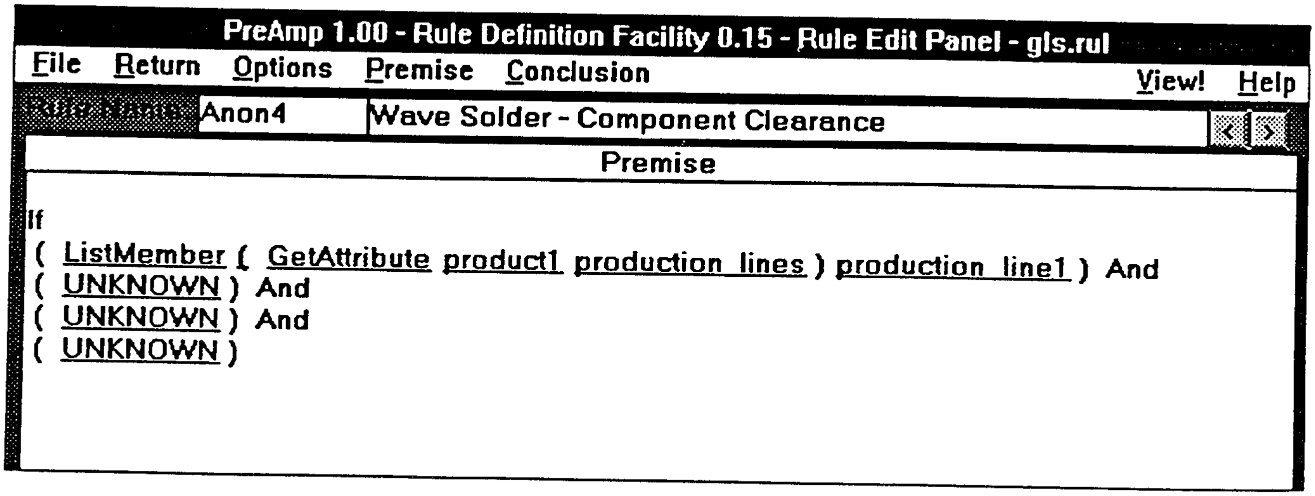
Rule Name: Anon4 Wave Solder - Component Clearance

```
If
( ListMember ( GetAttribute product1 production_lines ) UNKNOWN ) And
( UNKNOWN ) And
( UNKNOWN ) And
( UNKNOWN )
```

Select Database Instance: product1, production_line1

Figure 4.3-13 Select Database Instance Menu with New Instance Added.

Select **production_line1** from the **Select Database Instance** menu. The first simple premise (of the first and only complex premise) is now completed.

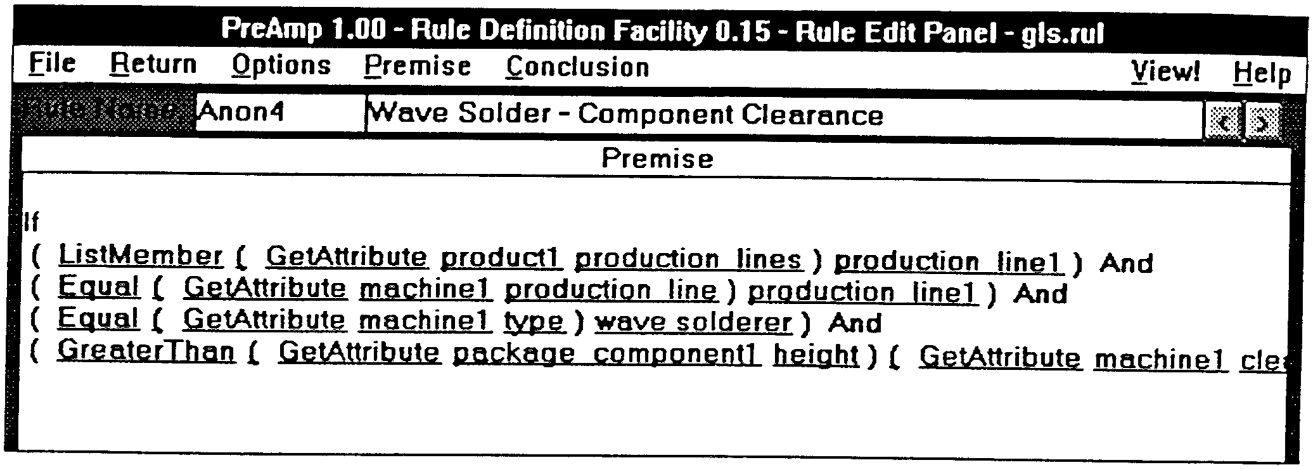

PreAmp 1.00 - Rule Definition Facility 0.15 - Rule Edit Panel - gls.rul
File Return Options Premise Conclusion View! Help
Rule Name:
Premise

Rule Name: Anon4 Wave Solder - Component Clearance

```
If
( ListMember ( GetAttribute product1 production_lines ) production_line1 ) And
( UNKNOWN ) And
( UNKNOWN ) And
( UNKNOWN )
```

Figure 4.3-14 First Simple Premise Completed.

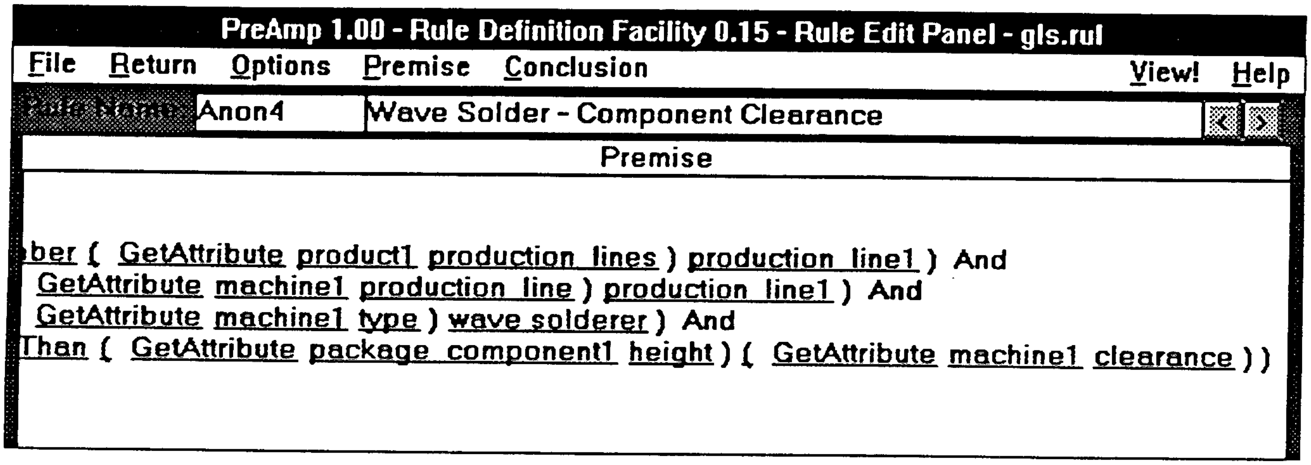
The next three simple premises are left as an exercise for the reader. The completed rule premise is displayed below.


PreAmp 1.00 - Rule Definition Facility 0.15 - Rule Edit Panel - gls.rul
File
Return
Options
Premise
Conclusion
View!
Help
Rule Name
Premise

Anon4 Wave Solder - Component Clearance

```
If
( ListMember ( GetAttribute product1 production lines ) production line1 ) And
( Equal ( GetAttribute machine1 production line ) production line1 ) And
( Equal ( GetAttribute machine1 type ) wave solderer ) And
( GreaterThan ( GetAttribute package component1 height ) ( GetAttribute machine1 cle
```

Figure 4.3-15 Completed Rule Premise (Scrolled Left).


PreAmp 1.00 - Rule Definition Facility 0.15 - Rule Edit Panel - gls.rul
File
Return
Options
Premise
Conclusion
View!
Help
Rule Name
Premise

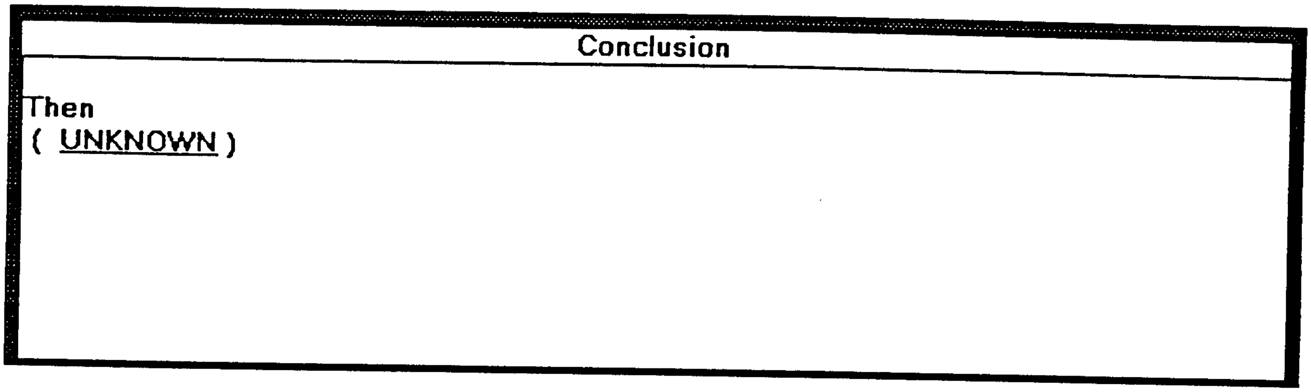
Anon4 Wave Solder - Component Clearance

```
ber ( GetAttribute product1 production lines ) production line1 ) And
 GetAttribute machine1 production line ) production line1 ) And
 GetAttribute machine1 type ) wave solderer ) And
Than ( GetAttribute package component1 height ) ( GetAttribute machine1 clearance ))
```

Figure 4.3-16 Completed Rule Premise (Scrolled Right).

4.4 Defining the Rule Conclusion Component

All rules must have a least one simple conclusion. When all simple premises of any one of a rules complex premises are true then the conclusion will execute. When a conclusion executes, all of the rules simple conclusions are executed. Three simple conclusions will be added to the example rule. Select the conclusion transcript window and click on the menu item **Add New Conclusion**. A simple conclusion will be added to the selected rule.

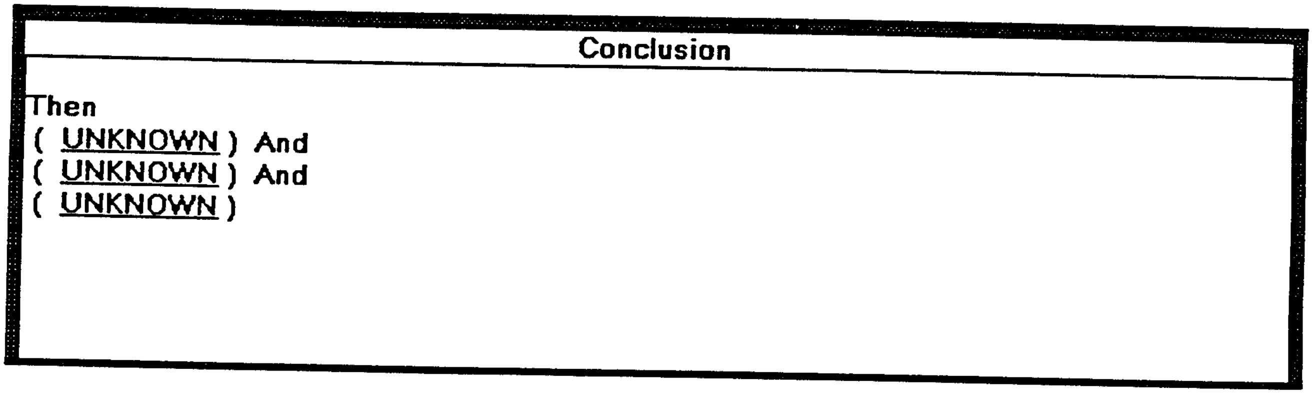


Conclusion

Then
(UNKNOWN)

Figure 4.4-1 Single Simple Conclusion.

Repeat this procedure two more times for a total of three simple conclusions for the example rule. The following figure depicts these additions. Note that when the second and the third simple conclusions were are added the position of the new simple conclusion to be added is requested. The order of insertion of simple conclusions for this example will be ignored.



Conclusion

Then
(UNKNOWN) And
(UNKNOWN) And
(UNKNOWN)

Figure 4.4-2 Three Simple Conclusions.

Note here again, that any underlined text or text in red is selectable via a left mouse click. Context sensitive menus will be displayed when this 'hot' text is selected. When the UNKNOWN is selected in the first simple conclusion the following menu is provided:

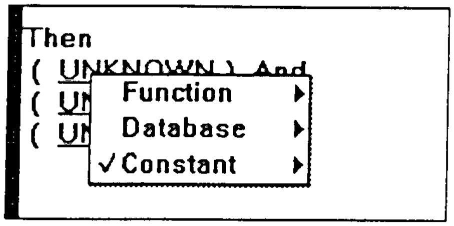


Figure 4.4-3 Conclusion Function Menu.

This menu provides paths for adding **Conclusion Functions**, **Database Functions**, and **Constants** to the conclusion component.

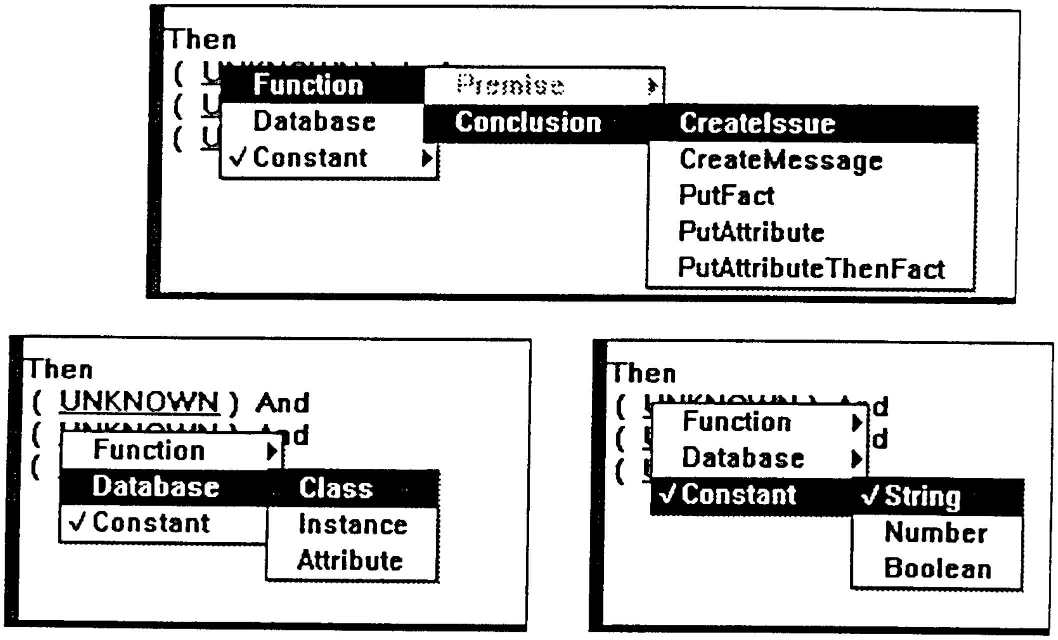
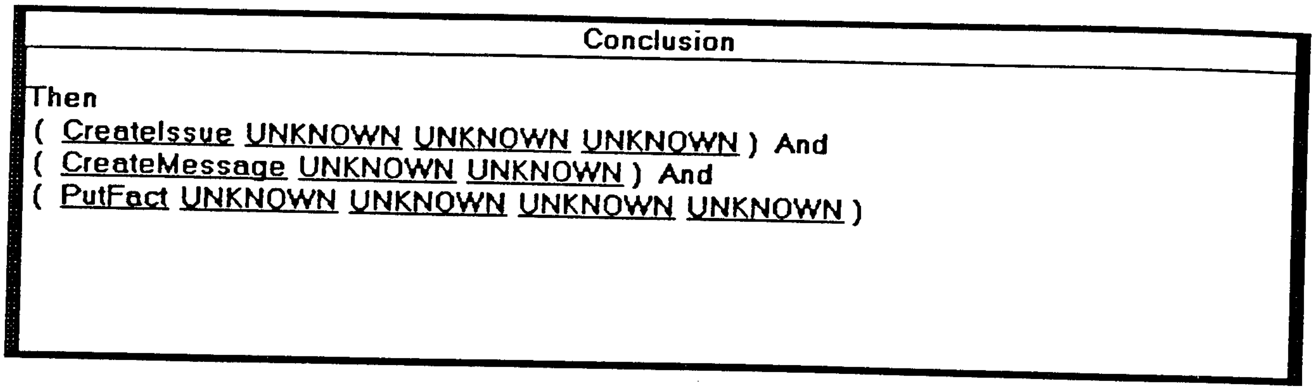


Figure 4.4-4 Conclusion Function Menu Paths.

The menu path **Function > Conclusion > CreateIssue** will be selected to add **CreateIssue** to the first conclusion component. The menu path **Function > Conclusion > CreateMessage** will be selected to add **CreateMessage** to the second simple conclusion component. The menu path **Function > Conclusion > PutFact** will be selected to add **PutFact** to the third simple conclusion component. The overall conclusion component will now appear as displayed below:

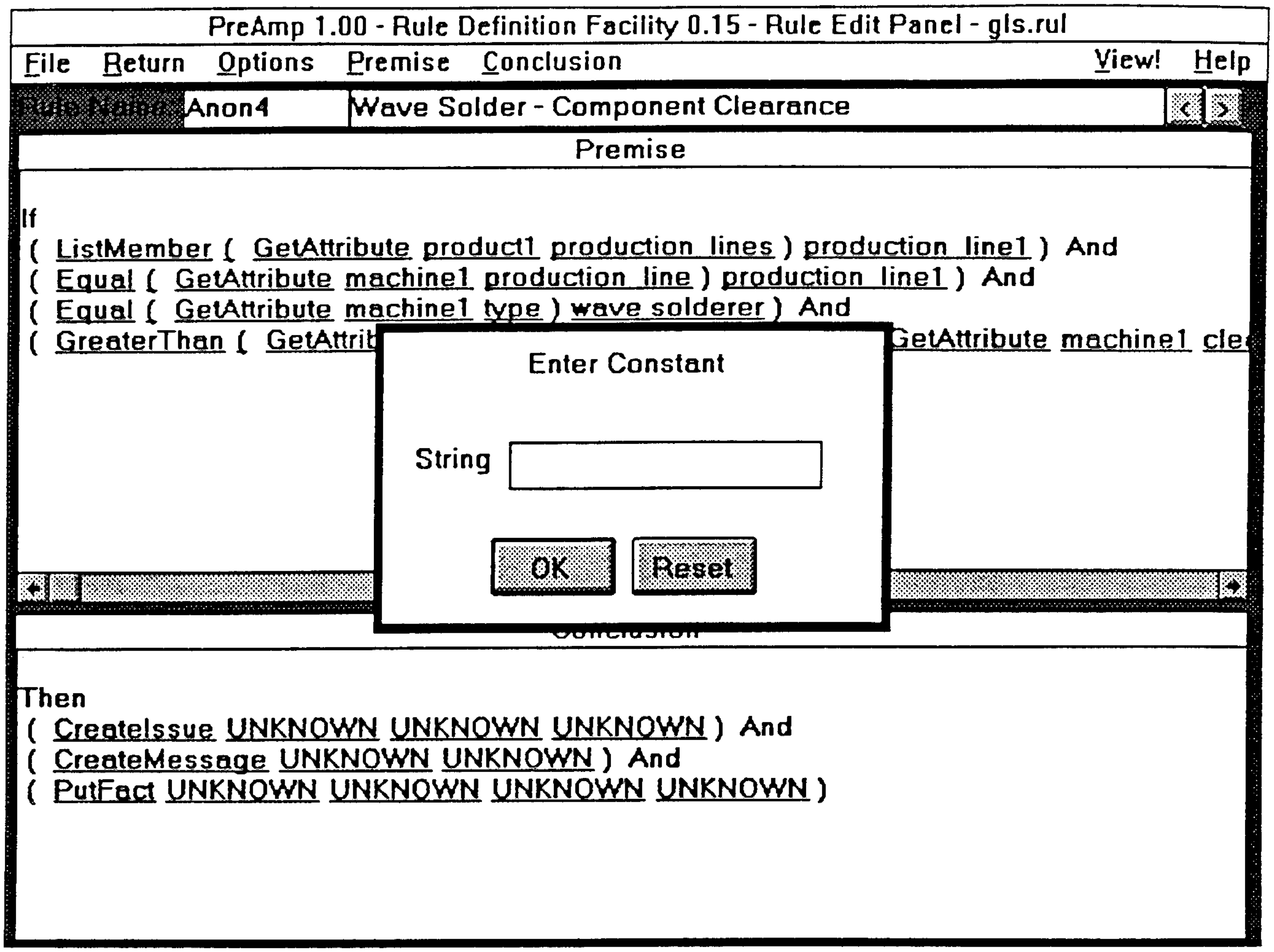


Conclusion

Then
(CreateIssue UNKNOWN UNKNOWN UNKNOWN) And
(CreateMessage UNKNOWN UNKNOWN) And
(PutFact UNKNOWN UNKNOWN UNKNOWN UNKNOWN)

Figure 4.4-5 Conclusion Functions Added.

The first **UNKNOWN** in the **CreateIssue** simple conclusion will be text to send to the author of the rule. The syntax for **CreateIssue** is **CreateIssue** (**issue text, issue source, issue destination**). The first **UNKNOWN** in the **CreateMessage** simple conclusion will be a message to send to the user. The syntax for **CreateMessage** is **CreateMessage** (**message text, additional message pointer**). Using the menu path **Function > Constant > String**, the **String** menu selection will be used to enter text into both of these simple conclusion functions.



Anon4 — Wave Solder - Component Clearance

```
If
 ( ListMember ( GetAttribute product1 production lines ) production line1 ) And
 ( Equal ( GetAttribute machine1 production line ) production line1 ) And
 ( Equal ( GetAttribute machine1 type ) wave solderer ) And
 ( GreaterThan ( GetAttrib                              GetAttribute machine1 cle
```

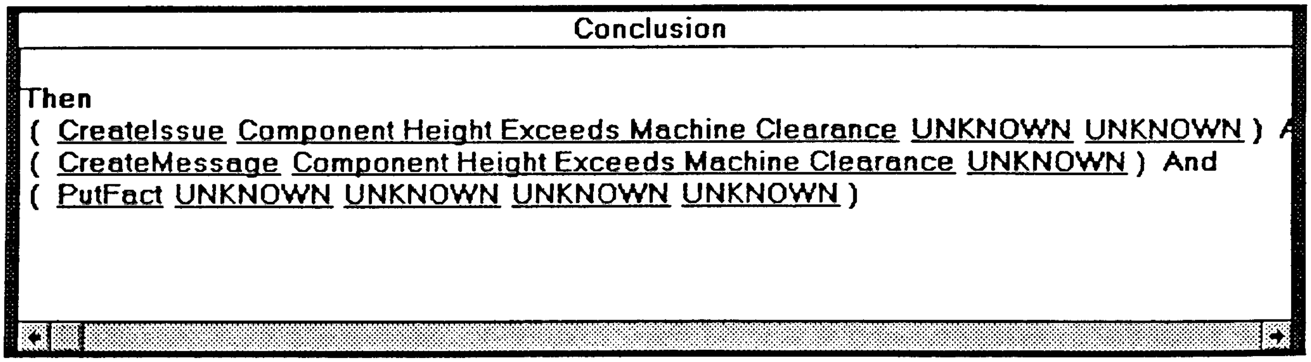
Enter Constant

String

```
Then
 ( CreateIssue UNKNOWN UNKNOWN UNKNOWN ) And
 ( CreateMessage UNKNOWN UNKNOWN ) And
 ( PutFact UNKNOWN UNKNOWN UNKNOWN UNKNOWN )
```

Figure 4.4-6 String Constant Menu.

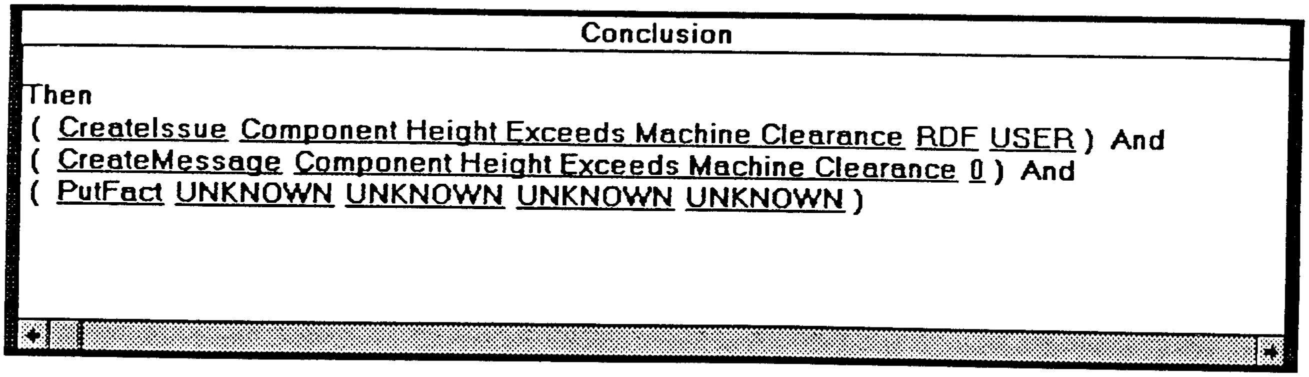
When the string text is entered the conclusion components appear as displayed below:



```
Then
 ( CreateIssue Component Height Exceeds Machine Clearance UNKNOWN UNKNOWN ) A
 ( CreateMessage Component Height Exceeds Machine Clearance UNKNOWN ) And
 ( PutFact UNKNOWN UNKNOWN UNKNOWN UNKNOWN )
```

Figure 4.4-6 Strings Added to CreateIssue & CreateMessage.

The next arguments to CreateIssue are the source and destination for the issue. The second argument to Create Message is a pointer the additional information. With the addition of this information, the conclusion component looks as displayed below.

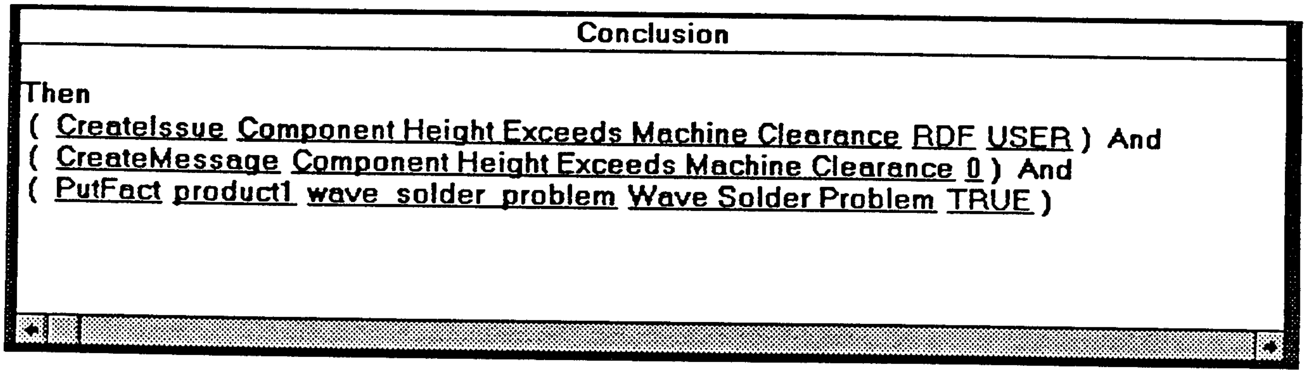


Conclusion

```
Then
( CreateIssue Component Height Exceeds Machine Clearance RDF USER ) And
( CreateMessage Component Height Exceeds Machine Clearance 0 ) And
( PutFact UNKNOWN UNKNOWN UNKNOWN UNKNOWN )
```

Figure 4-4.7 product1 Instances Added.

The arguments for the **PutFact** function consist of an object, an attribute name, a fact name, and the fact value. For the example rule the object will be the instance of **product1** instance. This instance is selected using the **Functions > Database > Instance** menu sequence. The fact name will **"Wave Solder Problem"** and can be entered using the **Functions > Constant > String** menu sequence. The fact value will be **TRUE** and can be entered using the **Function > Constants > Boolean** menu sequence. The completed rule conclusion component is displayed below.



Conclusion

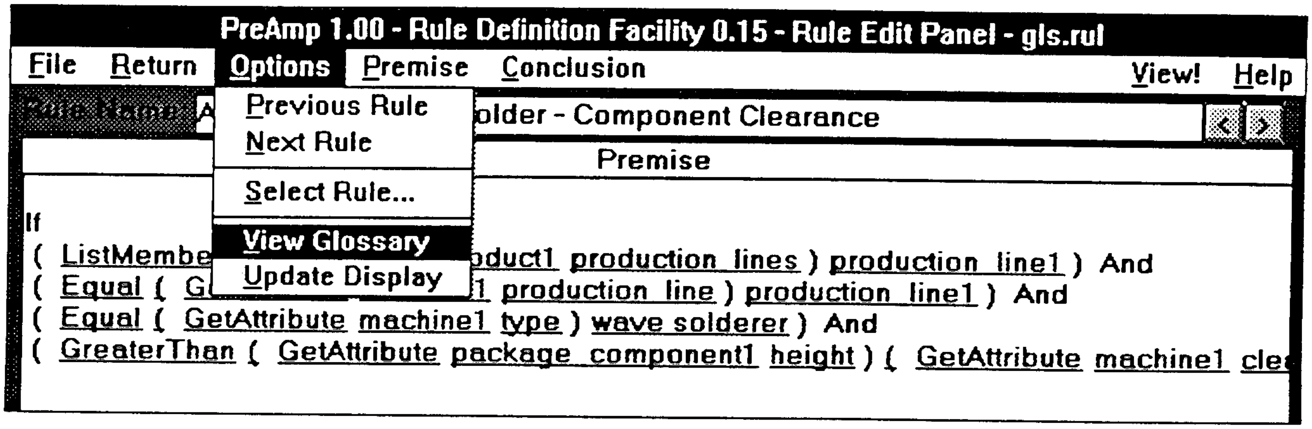
```
Then
( CreateIssue Component Height Exceeds Machine Clearance RDF USER ) And
( CreateMessage Component Height Exceeds Machine Clearance 0 ) And
( PutFact product1 wave solder problem Wave Solder Problem TRUE )
```

Figure 4-4.8 Completed Conclusion Component.

5.0 Using the Database Glossary

Understanding and remembering object and attribute names of database entities is extremely difficult. To aid the user in identifying these object and attribute names a database glossary panel has been added to the Rules Definition Facility. It allows users to browse the database, traverse through its objects, and examine its object and attribute definitions.

With a rule selected, access to the database glossary panel is provided via the Rule Edit Panel. When the menu selection **Options**, **View Glossary** is selected (as displayed below) the database glossary panel is opened.



PreAmp 1.00 - Rule Definition Facility 0.15 - Rule Edit Panel - gls.rul

A ... older - Component Clearance

```
If
( ListMembe          oduct1 production lines ) production line1 ) And
( Equal ( G          1 production line ) production line1 ) And
( Equal ( GetAttribute machine1 type ) wave solderer ) And
( GreaterThan ( GetAttribute package component1 height ) ( GetAttribute machine1 cle
```

Figure 5-1 Accessing the Database Glossary Panel from the Rule Edit Panel.

The initial display of the database glossary panel is shown below. Note that the left list display is a list of database schemas and the right list display is blank.

The upper portion of the screen displays the path through the hierarchy as objects and their sub-objects are selected. As schemas, their objects, and subobjects are selected this display will update to maintain the selected object hierarchy history. (This history is used as the system traverses back up the path when the **Back One Level** button is pushed.) The left list display shows schemas or toplevel objects. The right list display shows subobjects or attributes. The bottom text display shows glossary descriptions of selected schemas, objects, or attributes.

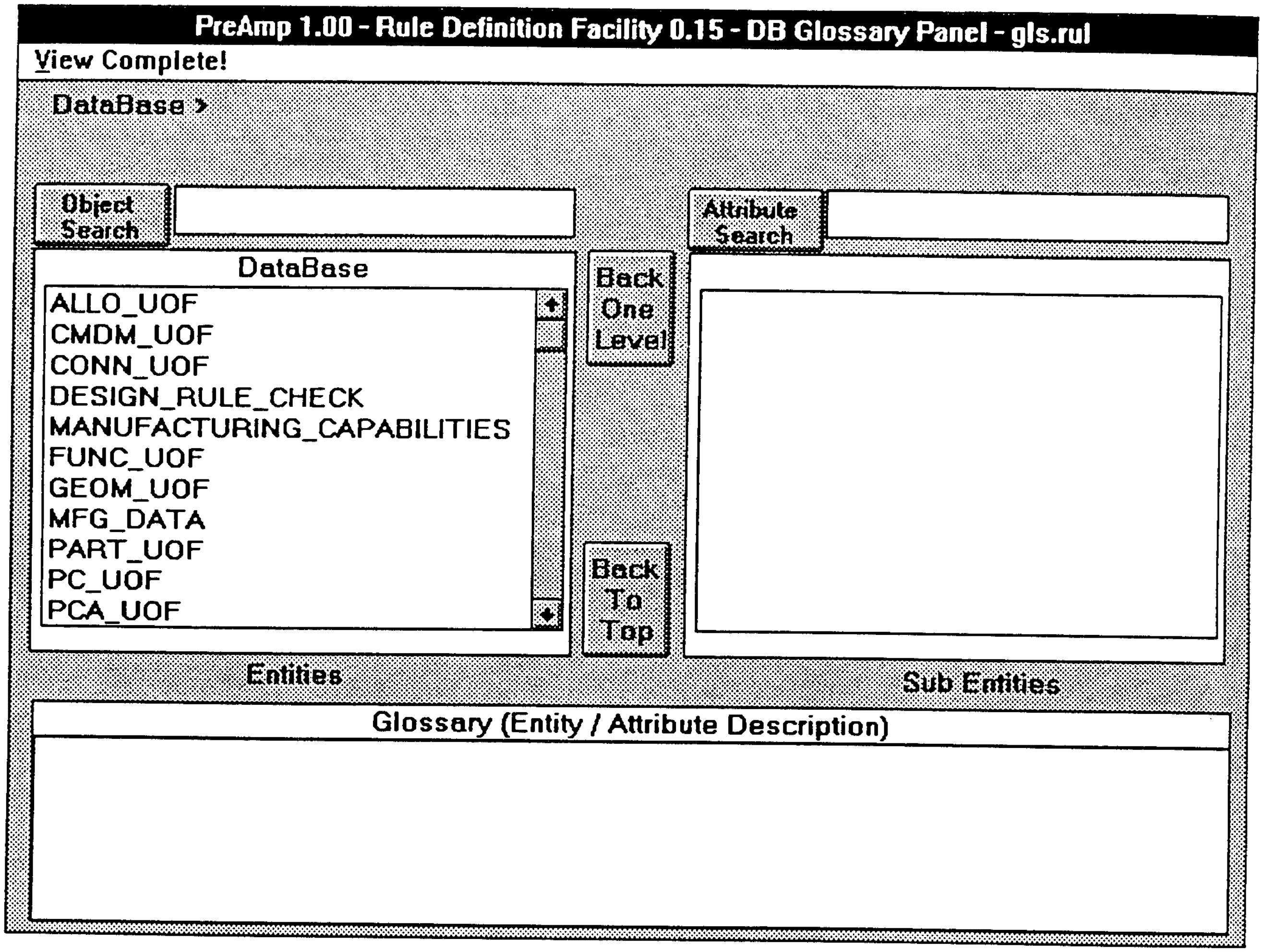


Figure 5-2 Initial DB Glossary Panel display.

The left list display shows top level objects or database schemas. When a schema is selected (by a single left click) its subobjects are displayed in the right list display. If defined, the schemas glossary description will be shown in the text display at the bottom of the panel. The following figure shows the MANUFACTURING_CAPABILITIES schema selected. Note that the database path in the top of the screen has been updated.

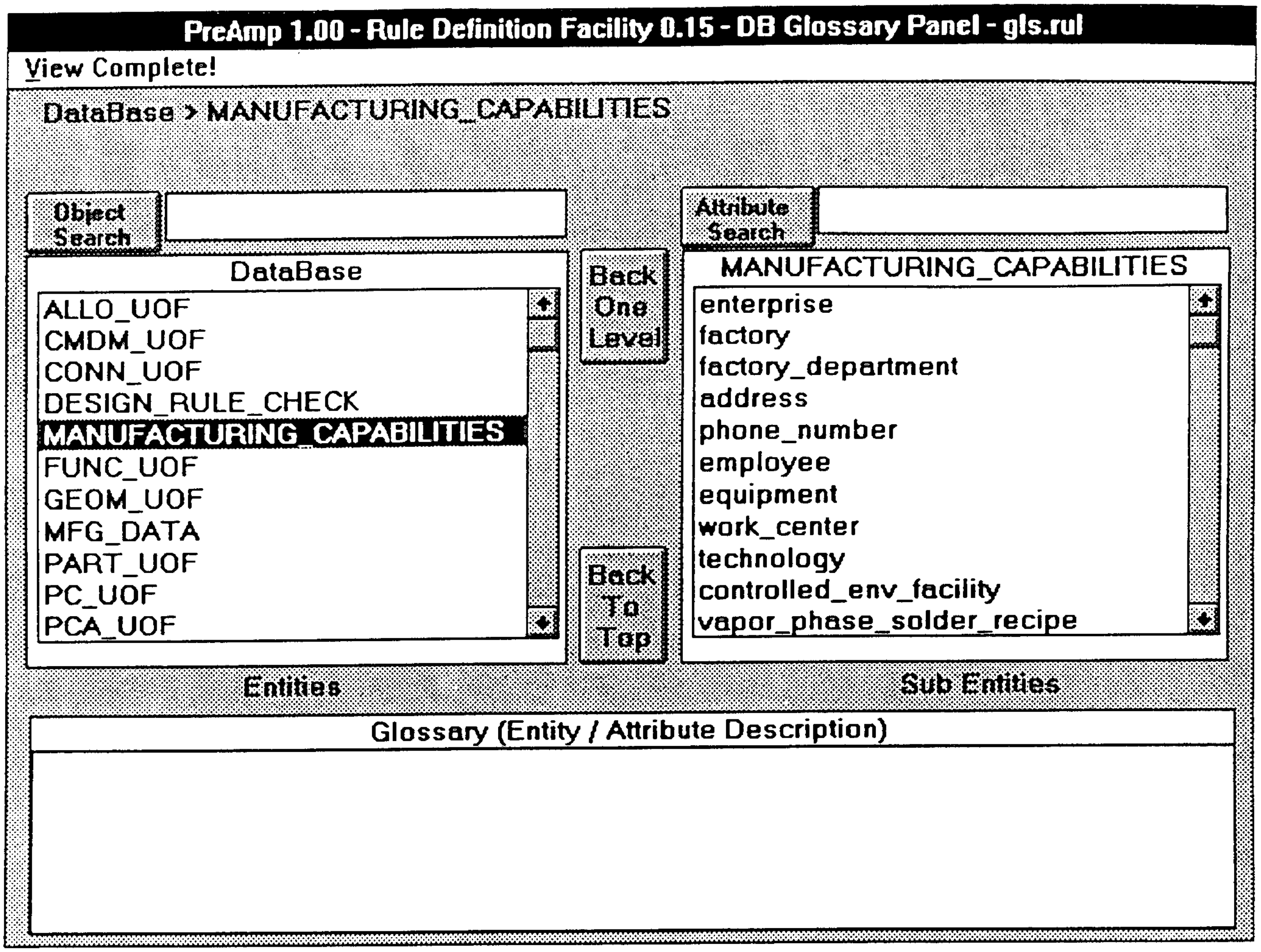
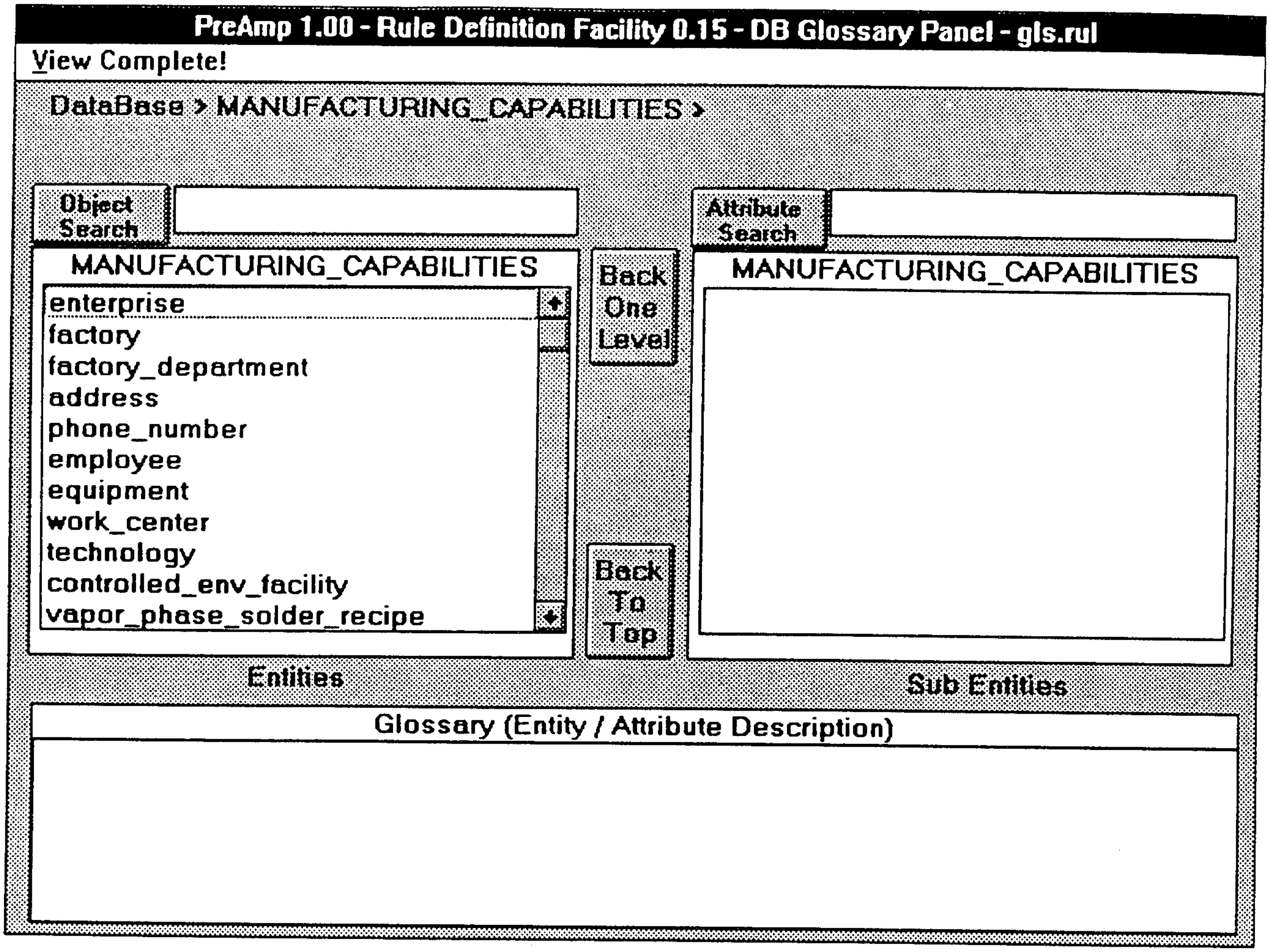


Figure 5-3 Manufacturing Capabilities schema selected.

When a schema is examined (by a double left click) the schema on the left display is replaced with the schemas subobjects. The following figure shows the MANUFACTURING_CAPABILITIES schema examined. Again note that the database path in the top of the screen has been updated.


PreAmp 1.00 - Rule Definition Facility 0.15 - DB Glossary Panel - gls.rul
View Complete!
Object Search
Attribute Search
Back One Level
Back To Top
Entities
Sub Entities
Glossary (Entity / Attribute Description)

DataBase > MANUFACTURING_CAPABILITIES >

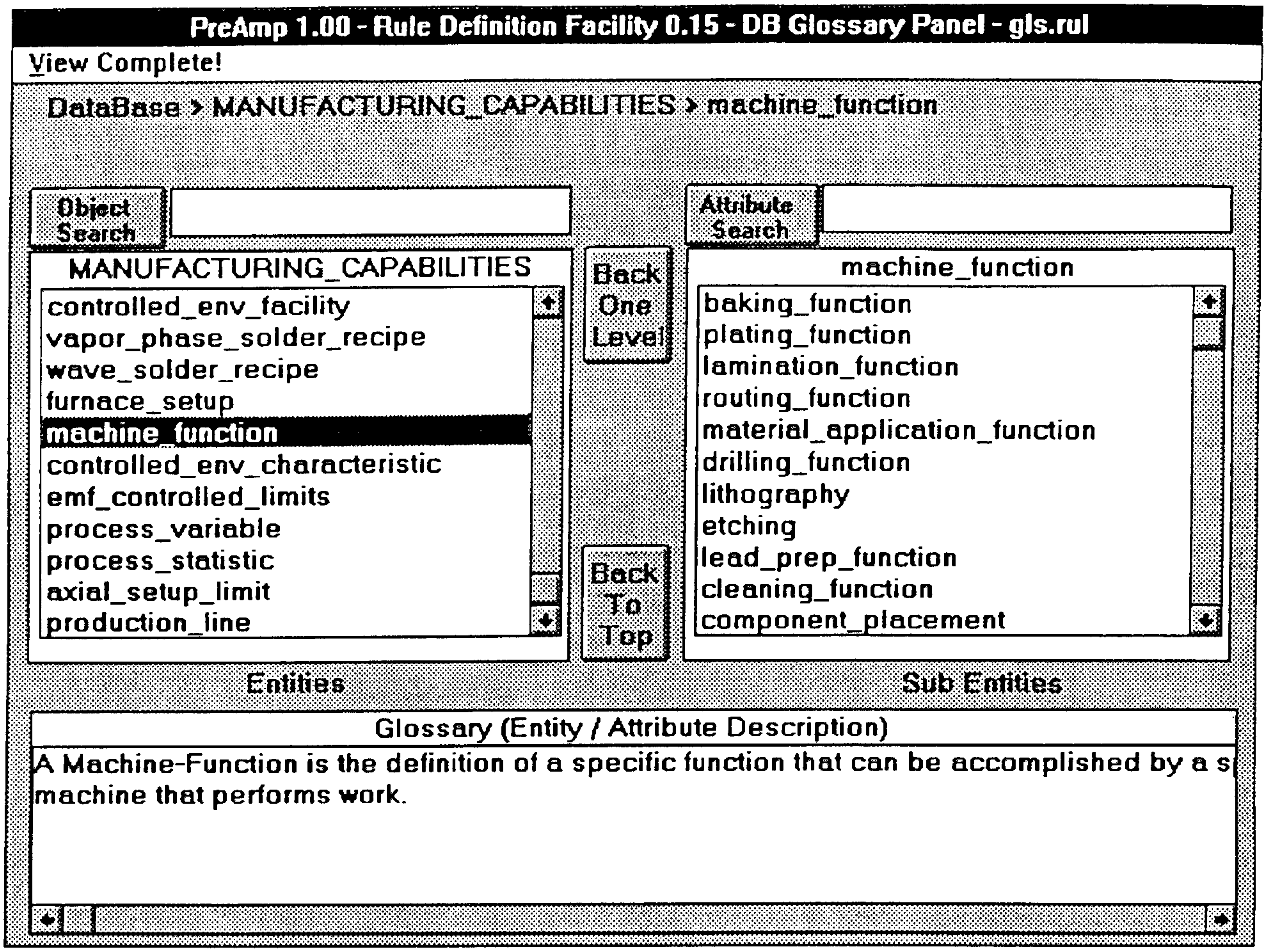
MANUFACTURING_CAPABILITIES

- enterprise
- factory
- factory_department
- address
- phone_number
- employee
- equipment
- work_center
- technology
- controlled_env_facility
- vapor_phase_solder_recipe

MANUFACTURING_CAPABILITIES

Figure 5-4 Manufacturing Capabilities schema examined.

By selecting a subobject (by a single left click) its sub-objects are displayed in the right list display. The following figure shows the **machine_function** object selected. Again note that the database path in the top of the screen has been updated. If defined, the objects glossary description will be shown in the text display at the bottom of the panel.


PreAmp 1.00 - Rule Definition Facility 0.15 - DB Glossary Panel - gls.rul
View Complete!
Object Search
Attribute Search
Back One Level
Back To Top
Entities
Sub Entities
Glossary (Entity / Attribute Description)

DataBase > MANUFACTURING_CAPABILITIES > machine_function

MANUFACTURING_CAPABILITIES

controlled_env_facility
vapor_phase_solder_recipe
wave_solder_recipe
furnace_setup
machine_function
controlled_env_characteristic
emf_controlled_limits
process_variable
process_statistic
axial_setup_limit
production_line

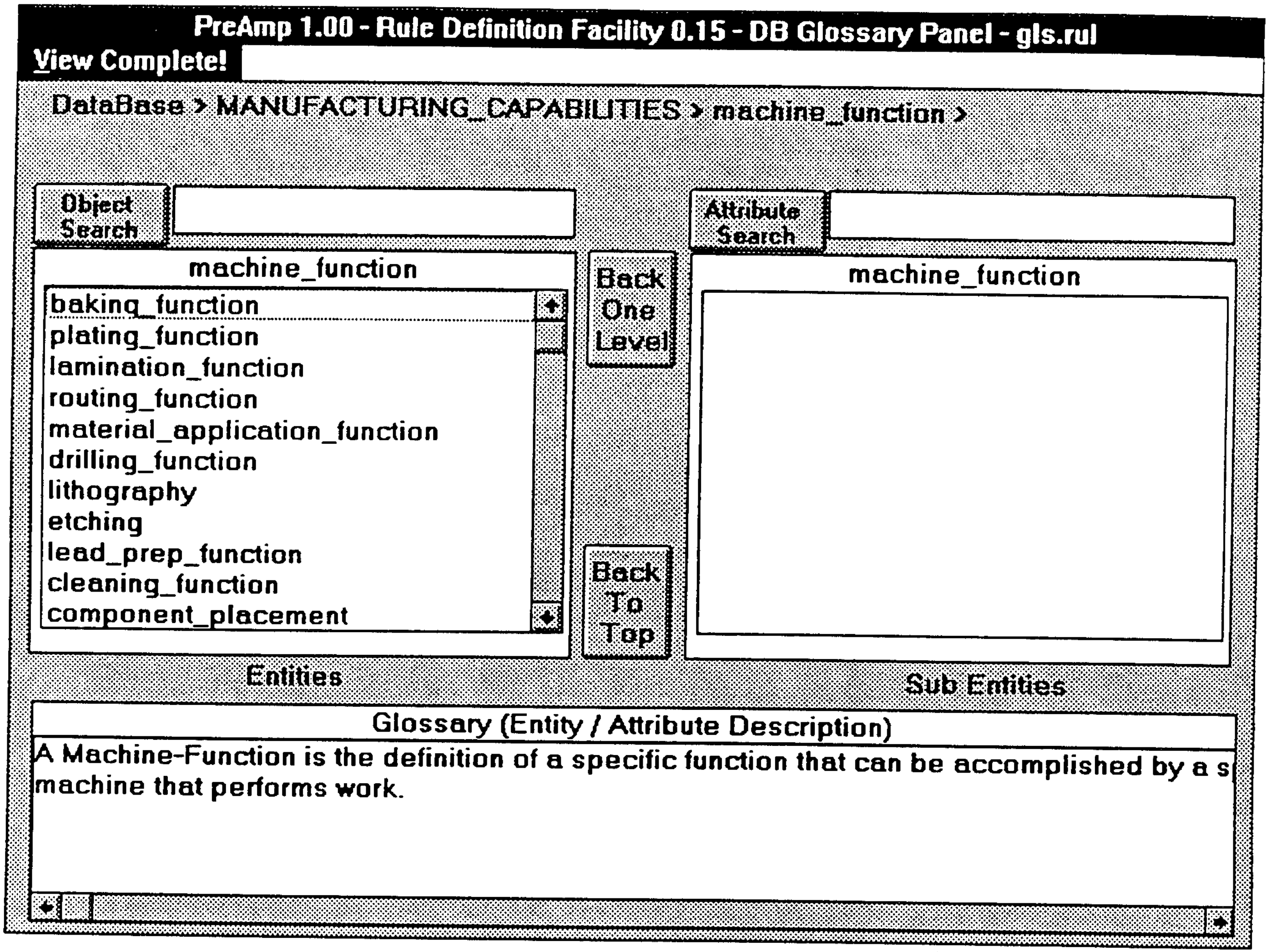
machine_function

baking_function
plating_function
lamination_function
routing_function
material_application_function
drilling_function
lithography
etching
lead_prep_function
cleaning_function
component_placement

A Machine-Function is the definition of a specific function that can be accomplished by a s
machine that performs work.

Figure 5-5 machine_function Subobjects selected.

By examining a subobject (by a double left click) the subobject on the left display is replaced with the subobjects subobjects. The following figure shows the **machine_function** object examined. Again note that the database path in the top of the screen has been updated. If defined, the objects glossary description will be shown in the text display at the bottom of the panel.



PreAmp 1.00 - Rule Definition Facility 0.15 - DB Glossary Panel - gls.rul

DataBase > MANUFACTURING_CAPABILITIES > machine_function >

machine_function

- baking_function
- plating_function
- lamination_function
- routing_function
- material_application_function
- drilling_function
- lithography
- etching
- lead_prep_function
- cleaning_function
- component_placement

Glossary (Entity / Attribute Description)

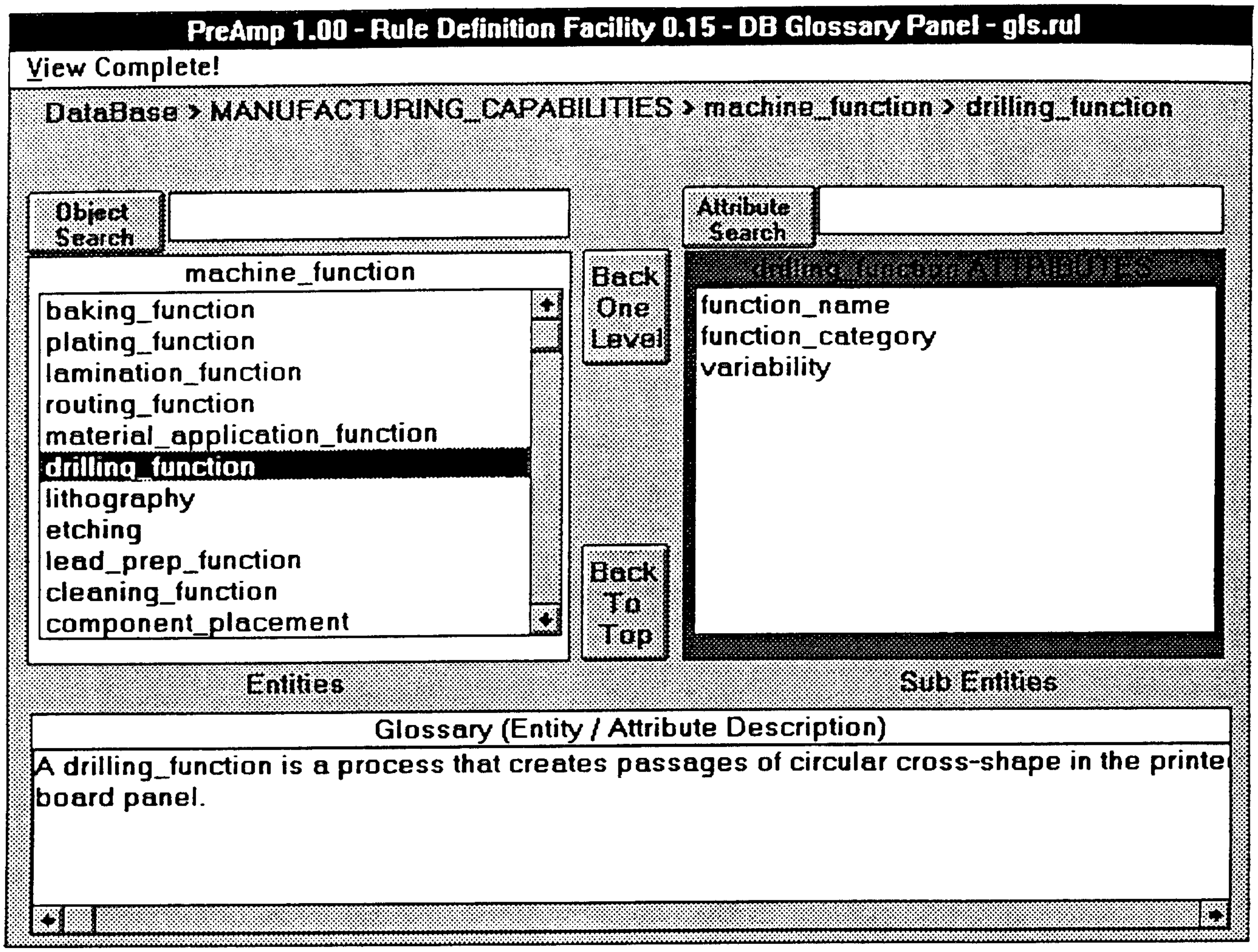
A Machine-Function is the definition of a specific function that can be accomplished by a s
machine that performs work.

Figure 5-6 machine_function Subobjects examined.

This process of stepping down the schema object hierarchy level by level can continue until a leaf object is selected. When a leaf object is selected (by a single left click) its attributes are shown in the right list display.

If we select **drilling_function** (a leaf object) a list of its attributes is shown in the right list display as depicted below.

Note that subobjects and attributes can be selected by using the right list display to show their glossary information.



PreAmp 1.00 - Rule Definition Facility 0.15 - DB Glossary Panel - gls.rul

DataBase > MANUFACTURING_CAPABILITIES > machine_function > drilling_function

machine_function

- baking_function
- plating_function
- lamination_function
- routing_function
- material_application_function
- drilling_function
- lithography
- etching
- lead_prep_function
- cleaning_function
- component_placement

drilling_function ATTRIBUTES

- function_name
- function_category
- variability

Glossary (Entity / Attribute Description)

A drilling_function is a process that creates passages of circular cross-shape in the printe
board panel.

Figure 5-7 drilling_function Attributes selected.

The **Back to Top** button can be selected any time the user wishes to return to the initial schema view of the database. The **Back One Level** button will traverse the displays up the hierarchy based on the path displayed in the top text display. When the user has completed his browsing of the database, the **View Complete!** menu item is selected and the system is returned to the rule detail panel.

5.1 Searching for Objects and/or Attributes

The DB Glossary Panel provides functionality for the user the search for objects and attributes. The search mechanism allows the user to search for substrings in attributes or objects. The object search is performed by entering the desired search string into the input area adjacent to the **Object Search** button. As an example, all objects containing the text string **machine** will be searched for. The Db Glossary Panel will appear as shown below.

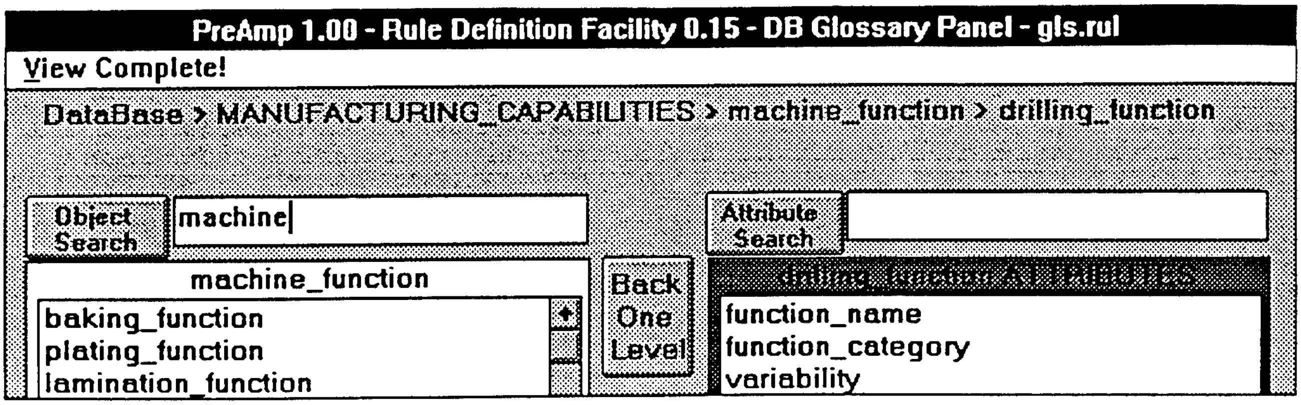

PreAmp 1.00 - Rule Definition Facility 0.15 - DB Glossary Panel - gls.rul
View Complete!
DataBase > MANUFACTURING_CAPABILITIES > machine_function > drilling_function
Object Search
machine
Attribute Search
machine_function
baking_function
plating_function
lamination_function
Back One Level
drilling_function ATTRIBUTES
function_name
function_category
variability

Figure 5-8   Searching for machine objects.

Once the desired search string is entered, the search is initiated by left clicking on **Object Search** button. The system will begin its search and, if successful, will return with a menu of matching objects.

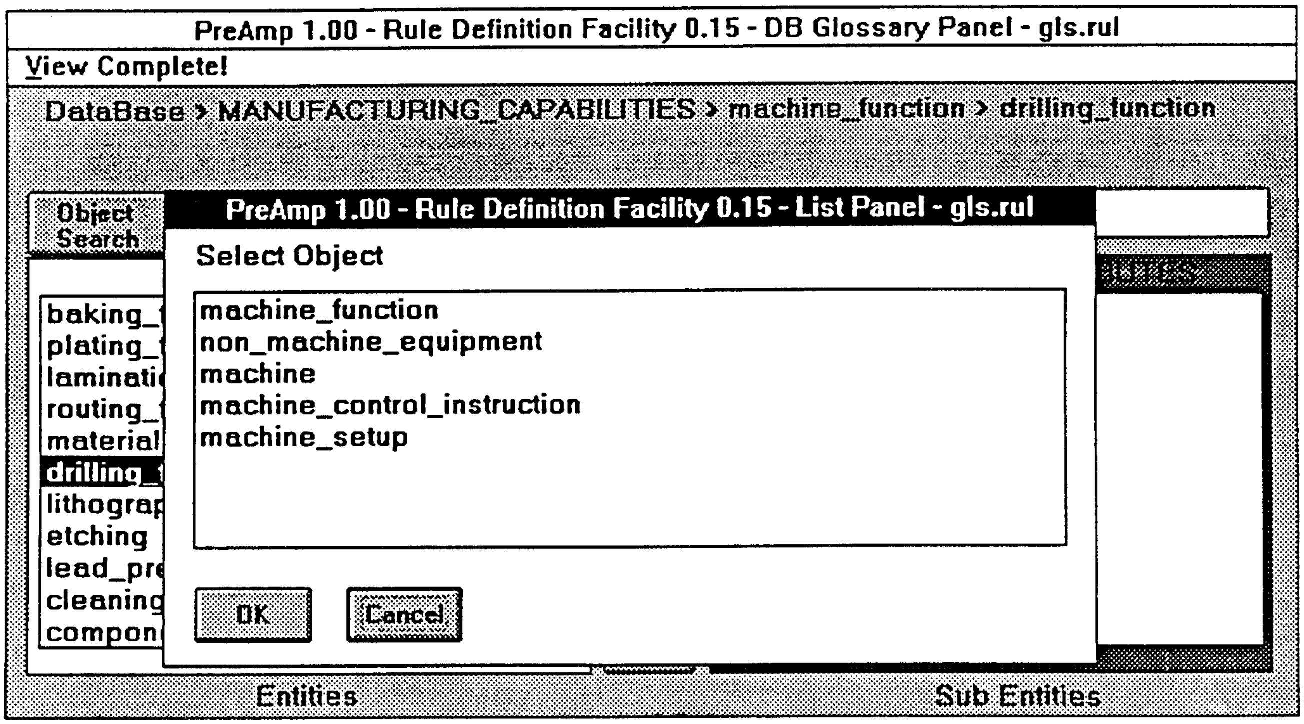
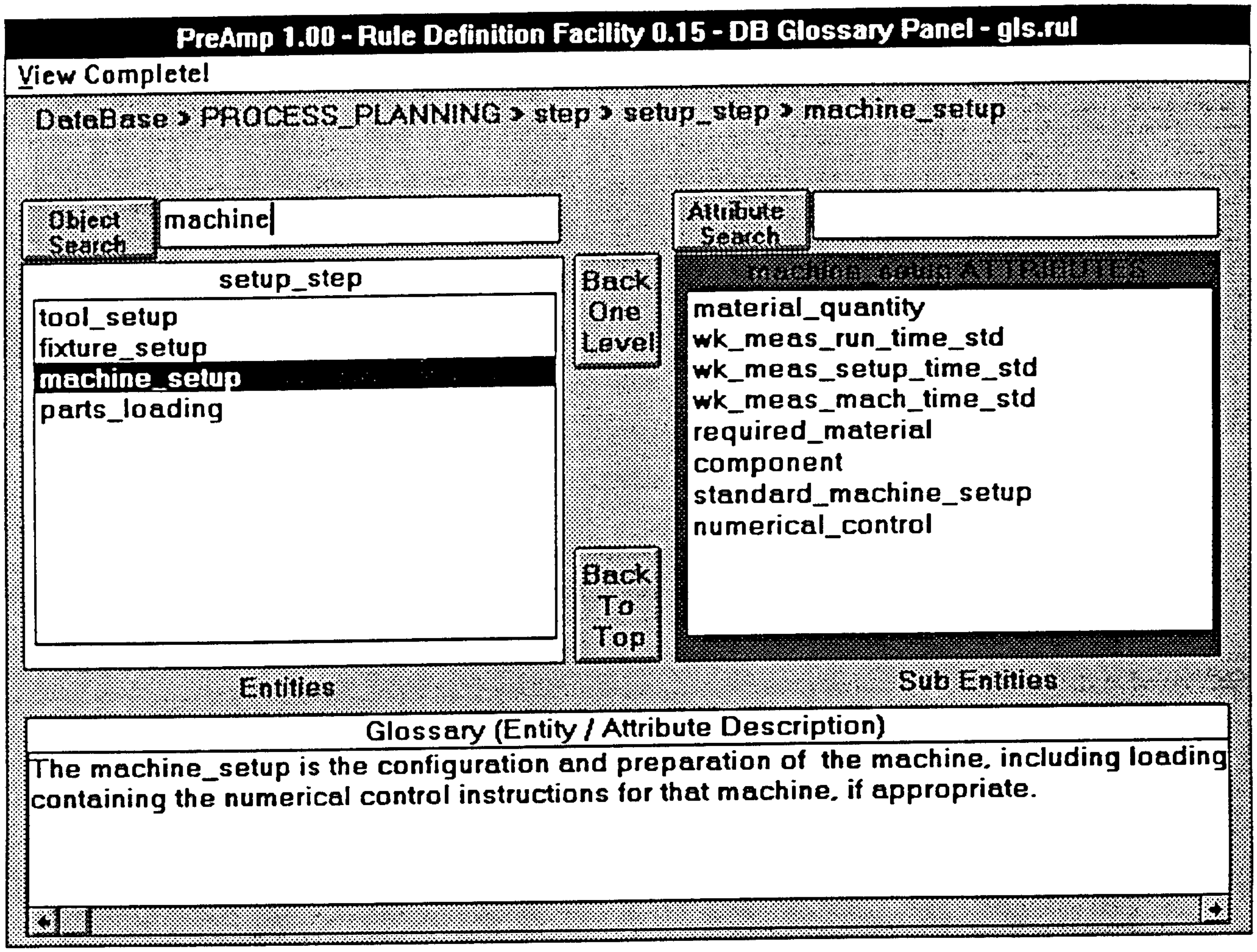

PreAmp 1.00 - Rule Definition Facility 0.15 - DB Glossary Panel - gls.rul
View Complete!
DataBase > MANUFACTURING_CAPABILITIES > machine_function > drilling_function
Object Search
PreAmp 1.00 - Rule Definition Facility 0.15 - List Panel - gls.rul
Select Object
machine_function
non_machine_equipment
machine
machine_control_instruction
machine_setup
OK
Cancel
baking_
plating_
laminati
routing_
material
drilling_
lithogra
etching
lead_pr
cleaning
compon
Entities
Sub Entities

Figure 5-9   List of system identified machine objects.

The user may cancel this menu or select an object from the menu listing. For this example, **machine_setup** will be selected. The DB Glossary Panel will now appear as shown below. Notice that **machine_setup** has been displayed in the left list display. The path to this object is shown, as always, at the top of the panel.



DataBase > PROCESS_PLANNING > step > setup_step > machine_setup

Object Search: machine

setup_step
- tool_setup
- fixture_setup
- machine_setup
- parts_loading

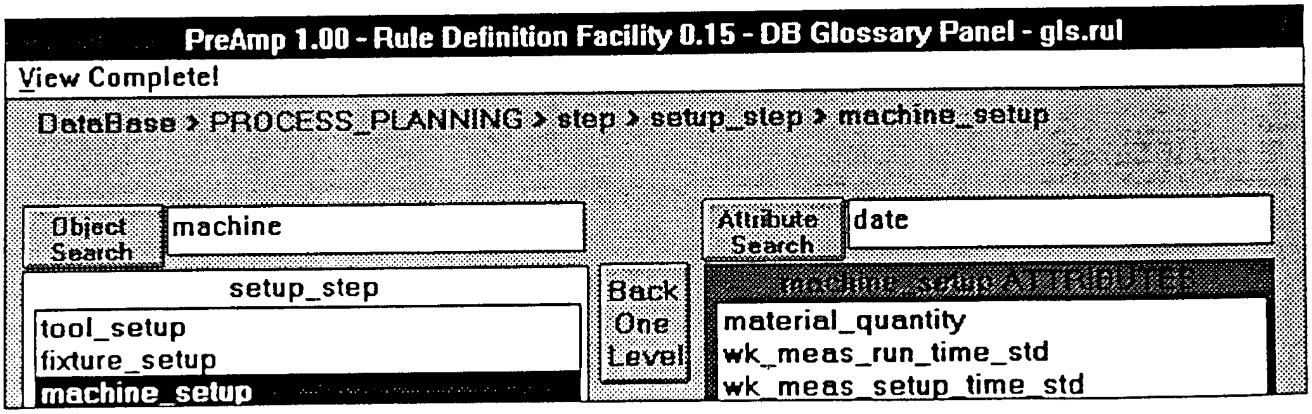
machine_setup ATTRIBUTES
- material_quantity
- wk_meas_run_time_std
- wk_meas_setup_time_std
- wk_meas_mach_time_std
- required_material
- component
- standard_machine_setup
- numerical_control

Glossary (Entity / Attribute Description)

The machine_setup is the configuration and preparation of the machine, including loading containing the numerical control instructions for that machine, if appropriate.

Figure 5-10   machine_setup object displayed.

The object attribute search is performed by entering the desired search string into the input area adjacent to the **Attribute Search** button. As an example, all objects with attributes containing the text string **date** will be searched for. The DB Glossary Panel will appear as shown below.



DataBase > PROCESS_PLANNING > step > setup_step > machine_setup

Object Search: machine

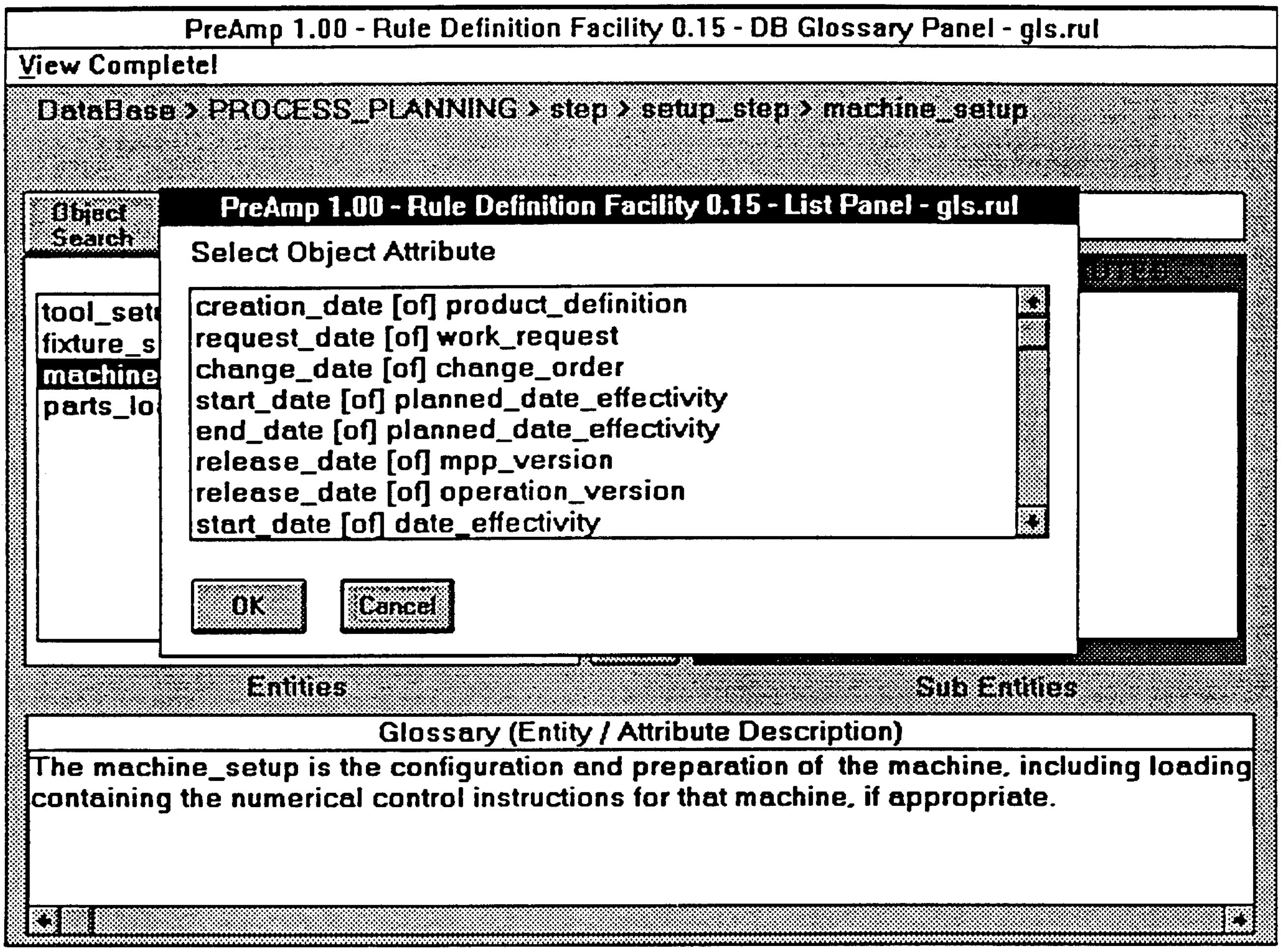
Attribute Search: date

setup_step
- tool_setup
- fixture_setup
- machine_setup

machine_setup ATTRIBUTES
- material_quantity
- wk_meas_run_time_std
- wk_meas_setup_time_std

Figure 5-11   Searching for date object attributes.

Here again, once the desired search string is entered, the search is initiated by left clicking on **Attribute Search** button. The system will begin its search and, if successful, will return with a menu of matching object attributes.



PreAmp 1.00 - Rule Definition Facility 0.15 - DB Glossary Panel - gls.rul

View Complete!

DataBase > PROCESS_PLANNING > step > setup_step > machine_setup

PreAmp 1.00 - Rule Definition Facility 0.15 - List Panel - gls.rul

Select Object Attribute

creation_date [of] product_definition
request_date [of] work_request
change_date [of] change_order
start_date [of] planned_date_effectivity
end_date [of] planned_date_effectivity
release_date [of] mpp_version
release_date [of] operation_version
start_date [of] date_effectivity

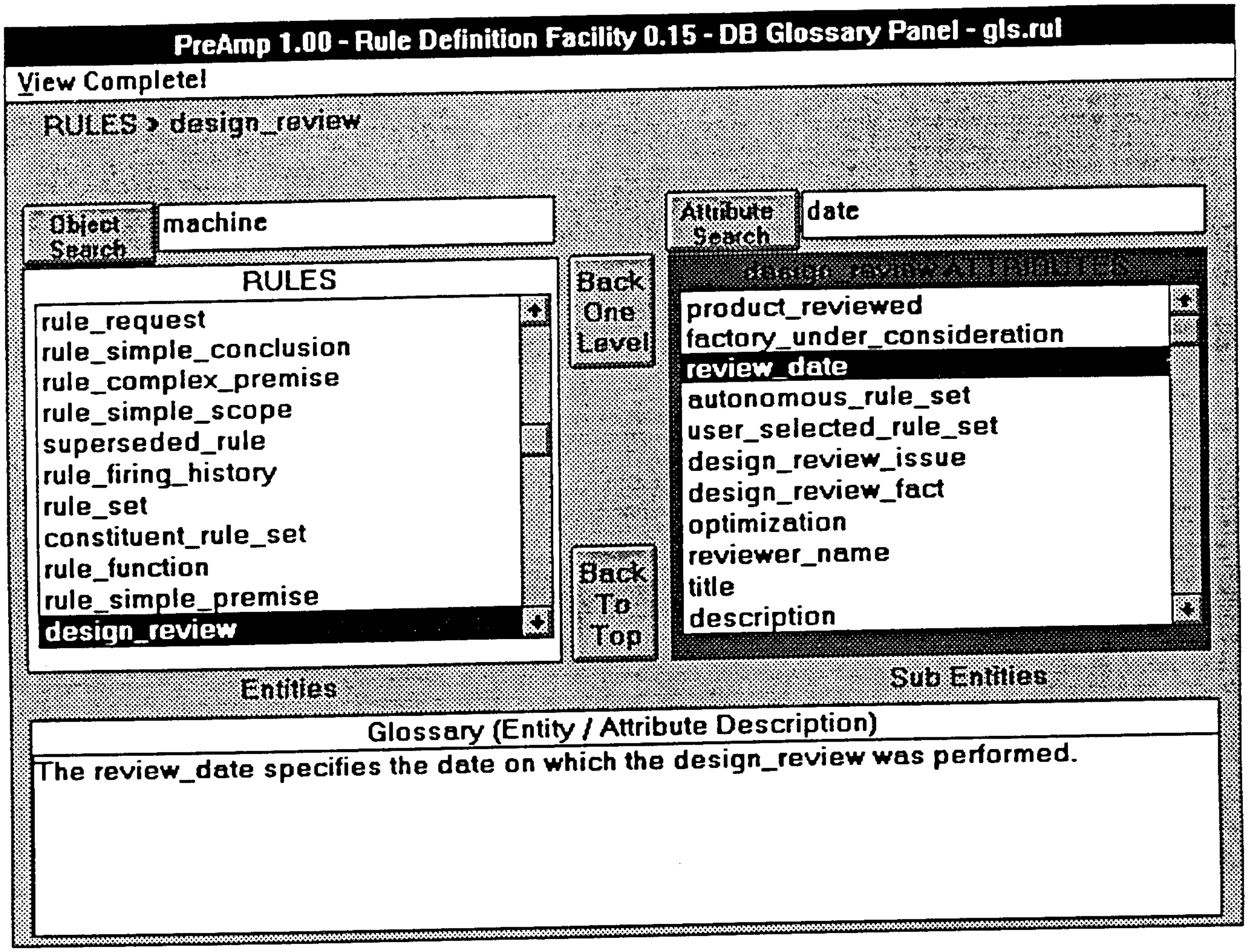
tool_set
fixture_s
machine
parts_lo

Glossary (Entity / Attribute Description)

The machine_setup is the configuration and preparation of the machine, including loading containing the numerical control instructions for that machine, if appropriate.

Figure 5-12 List of system identified date object attributes.

The user may cancel this menu or select an object attribute from the menu listing. For this example, **review_date [of] design_review** will be selected. The DB Glossary Panel will now appear as shown below. Notice that **design_review** has been displayed in the left list display and its attributes in the right display. The **review_date** attribute is hilighted in the right panel with its glossary definition on the bottom of the panel. The path to this object is shown, as always, at the top of the panel.



PreAmp 1.00 - Rule Definition Facility 0.15 - DB Glossary Panel - gls.rul

RULES > design_review

Object Search: machine

Attribute Search: date

RULES

- rule_request
- rule_simple_conclusion
- rule_complex_premise
- rule_simple_scope
- superseded_rule
- rule_firing_history
- rule_set
- constituent_rule_set
- rule_function
- rule_simple_premise
- design_review

design_review ATTRIBUTES

- product_reviewed
- factory_under_consideration
- review_date
- autonomous_rule_set
- user_selected_rule_set
- design_review_issue
- design_review_fact
- optimization
- reviewer_name
- title
- description

Glossary (Entity / Attribute Description)

The review_date specifies the date on which the design_review was performed.

Figure 5-13 design_review object displayed.

5.2 AP 210 AIM versus ARM

The AP 210 development team has released their initial draft for CD. The ARM (application represented model), which holds the objects and attributes as understood by the domain expert has been mapped to an AIM (application interpreted model). The application interpreted model takes advantage of objects and attributes already defined by others APs (application protocols). The AIM, however is not easily understandable by domain experts. To support ease of use by CAD/CAM users and developers the Rules Definition Facility uses the AP210 ARM. When the rules are compiled for execution in the UNIX/X windows version all AP 210 objects and attributes will be, at that time, mapped to the AIM. This functionality will make the use of the AIM transparent to Rules Definition Facility users.

Appendix F

ACRONYMS

Appendix F
ACRONYMS

AI	Artificial Intelligence
AMRI	Agile Manufacturing Research Institute
ARL	Advanced Rules Language
ASCII	American Standard Code for Information Interchange
ASICs	Application Specific Integrated Circuits
ATP	Advanced Technology Program
BoD	Board of Directors
CAD	Computer-aided design
CAE	Computer-aided engineering
CALCE	Computer-Aided-Life-Cycle-Engineering
CE	Concurrent Engineering
CE-CAD	Concurrent Engineering-Computer-Aided Design
COTS	Commercial-off-the-shelf
DBMS	Data Base Management System
EE	Electrical/Electronics
EPRC	Electronic Packaging Research Center
FCIM	Flexible Computer-Aided Manufacturing
GHz	Gigahertz
HDL	Hardware Description Language
I/O	Input/Output
IDA	Institute for Defense Analyses
IEEE	Institute for Electrical and Electronic Engineers
IPPD	Integrated Product and Process Development
KBE	Knowledge-based engineering
MCMs	Multichip modules
MHz	Megahertz
MIL-SPECS	Military Specifications
MIL-STDS	Military Standards
MMIC	Millimeter and Microwave Integrated Circuit

MRE	Manufacturing Resource Editor
MSI	Management Sciences Incorporated
NASA	National Aeronautical and Space Administration
NIST	National Institute of Standards and Technologies
NSF	National Science Foundation
PA	Producibility Advisor
PAT	Process Action Team
PCA	Printed Circuit Assemblies
PWB	Printed wiring board
RAC	Reliability Analysis Center
RAMCAD	Reliability, Availability, and Maintainability Computer-Aided Design Environment
RAMP	Rapid Access for Manufactured Parts
RAMS	Reliability, Availability, and Maintainability Symposium
RASSP	Rapid Acquisition of Signal Processors
RDF	Rules Definition Facility
RF	Radio Frequency
RPI	Rensselaer Polytechnic Institute
SAE	Society of Automotive Engineers
SCRA	South Carolina Research Activity
SDAI	Standard Data Access Interface
SIUCRC	State/Industry/University Cooperative Research Center
SMD	Surface Mount Devices
STEP	Standard for the Exchange of Product
TRP	Technology Reinvestment Project
VHSIC	Very High Speed Integrated Circuit

DISTRIBUTION LIST

IDA Paper P-3074

The Aries Project for Rule-Based-Design Knowledge Acquisition

38 Copies

	Copies
Mr. Frank Doherty ODTSE&E Pentagon Room 3D1075 Washington, DC 20301	1
Mr. Doug Patterson OASNR(R,D&A) Product Integrity 2211 Jefferson-Davis Highway Arlington VA 22244-5104	1
Mr. Ed Smith OASN(R,D&A) Product Integrity 2211 Jefferson Davis Hwy Rm 362 Arlington VA 22244-5104	1
Mr. Bob Whiteley AMC RD-IEC 8538 5001 Eisenhower Avenue Alexandria VA 22333-0001	1
Mr. Mike Zsak 306 Skyline 6 5109 Leesburg Pike Falls Church VA 22041	1
Mr. Ken Blemel Management Sciences, Inc. 6022 Constitution Ave NE Albuquerque NM 87110	1
Mr. Tom Fagan ITT Defense & Electronics 1650 Tysons Blvd #1700 McLean VA 22102-3915	1
Mr. John Fink Honeywell Inc. 2600 Ridgeway Road MM17-1622 Minneapolis MN 55413	1

Mr. Philip George 1
Cadence Design Systems Inc
2655 Seely Road MS 6B1
San Jose CA 95134

Mr. Gerry Graves 1
SCRA
5300 International Blvd
North Charleston SC 29418

Mr. Carl G. Grewe 1
EDS
750 Tower Drive M/S 2234
Troy MI 48098

Mr. Ali Hashmi 1
CALCE/EPRC
University of Maryland
College Park MD 20742

Mr. Dennis Hoffman 2
Texas Instruments
Defense Sys & Electronic Group
8505 Forest Lane MS 8618
PO Box 660246
Dallas TX 75266

Mr. Ray Johnston 1
Texas Instruments
6500 Chase Oaks
MS 8405
Plano TX 75086

Mr. Richard W. Kilgore 1
ICAD
Technical Sales Consultant
201 Broadway
Cambridge MA 02139-1901

Mr. Warren Mudd 1
Cognition
81 Windsor Ave
Acton MA 01720

Mr. Douglas E. Ott 1
ITT Defense
100 Kingsland Road
Clifton NJ 07014-1993

Mr. Marty Rosman 1
AlliedSignal Aerospace
Government Electronic Systems
1300 E. Joppa Road
Towson MD 21286-5999

Mr. Carl Rust 1
CALCE/EPRC
University of Maryland
Building 89 Room 1103
College Park MD 20742

Mr. Richard E. Sackett 1
7701 Greenbelt Road, Suite 204
Greenbelt MD 20770

Mr. Greg Smith 1
Boeing/*Pre*Amp
PO Box 3999
MS 3E73
Seattle WA 98124

Mr. Sid Markowitz 1
US Army ARDEC
Attn: SMCAR-QAH-P
Picatinny Arsenal NJ 07806-5000

Mr. Matt Tracy 1
AL/HRGA
2698 G. Street
WP AFB, OH 45433-7604

Defense Technical Information Center
8725 John J. Kingman Road, STE 0944
Fort Belvoir, VA 22060-6218 2

INSTITUTE FOR DEFENSE ANALYSES 12

General L. Welch	1
Mr. Philip L. Major	1
Ms. Ruth Greenstein	1
Dr. Robert Roberts	1
Mr. Christopher Jehn	1
Dr. Karen Richter	5
C&D	2

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY ***(Leave blank)***

2. REPORT DATE
June 1995

3. REPORT TYPE AND DATES COVERED
Contributing Analysis

4. TITLE AND SUBTITLE
The Aries Project for Rule-Based-Design Knowledge Acquisition

5. FUNDING NUMBERS
C-DASW01-94-C-0054
TA-T-B6-1216

6. AUTHOR(S)
Karen J. Richter, Ken Blemel, John Fink, Carl Grewe, Ali Hashmi, Dennis Hoffman, Ray Johnston, Doug Ott, Carl Rust

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Institute for Defense Analyses
1801 N. Beauregard Street
Alexandria, VA 22311-1772

8. PERFORMING ORGANIZATION REPORT NUMBER
IDA Paper P-3074

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Office of the Director, Test, Systems Engineering and Evaluation
The Pentagon
Washington, DC 20301

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited.

12b. DISTRIBUTION CODE

13. ABSTRACT ***(Maximum 200 words)***

The objective of the study documented in this paper was to demonstrate that a voluntary team of industry participants from different design and engineering disciplines could identify, capture, and reach consensus on a consistent, thorough set of design rules and could document that process so that it could be repeated. Success was demonstrated by implementing a sample of the rule set in a rule-based software demonstration vehicle. The feasibility study concentrated on printed wiring board (PWB) assembly digital and analog circuit design, circuit implementation, and early detailed design, i.e., the capture of the parts list and schematic.

14. SUBJECT TERMS
Rule-based design, knowledge acquisition, digital, analog PWB assembly design rules

15. NUMBER OF PAGES
178

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT
UNCLASSIFIED

18. SECURITY CLASSIFICATION OF THIS PAGE
UNCLASSIFIED

19. SECURITY CLASSIFICATION OF ABSTRACT
UNCLASSIFIED

20. LIMITATION OF ABSTRACT
SAR