Expert System Development Methodolgy and Management

Larry G. Hull
National Aeronautics and Space Administration
Goddard Space Flight Center
Greenbelt, Maryland 20771

Phyllis Kay
Computer Sciences Corporation
Beltsville, Maryland 20705

Abstract

A risk-based expert system development methodology has been developed at NASA Goddard Space Flight Center that provides guidance to project managers and technical personnel and that serves as a standard for developing expert systems. Expert system development differs from conventional software development in that the system requirements are not known at the outset of development and the expert system life cycle more nearly resembles an evolutionary process based upon rapid prototyping. This lack of requirements creates special problems for the expert system project manager. Further, little guidance is available for managing development of a well-engineered product using prototyping, and even less guidance exists for managing deliverables that are evolutionary in nature.

This paper first discusses concerns expressed by managers of expert system development projects. It then describes the expert system life cycle and the risk-based expert system development methodology. The summary reviews features of this methodology that address specific managerial concerns.

1 Introduction

Many concerns have been expressed by managers of expert system development projects, especially by new managers and by managers without previous experience in developing expert systems. We categorize these managerial concerns as:

1. General project management concerns, not specifically related to either software or expert system development
2. General software development concerns, not specifically related to expert systems
3. Expert system development concerns that are related to the project infrastructure and to technical issues
4. Expert system development concerns that are related to the lack of a life-cycle model and development framework

In this paper we do not address either of the first two categories and we address the third only indirectly. Numerous existing papers discuss expert system development project infrastructure and answer technical concerns such as selecting the knowledge engineer, obtaining (and keeping) the services of an expert, building the expert system development team, and selecting expert system tools. In this paper, we specifically address the concern that there is no underlying life-cycle model and that the absence of a framework and lack of system requirements makes expert system development difficult to manage. We provide both a life-cycle model and a development methodology based on the management of risk.

Our risk-based expert system development methodology (ESDM) [1,2,3] is the outgrowth of an effort to merge analysis of the expert system life cycle with experience in developing expert systems at NASA Goddard Space Flight Center. Since the information needed to prepare system requirements is not known at the outset of an expert system project, this methodology concentrates on driving out and validating requirements, using a series of successively more complex prototypes. ESDM employs a highly iterative life-cycle model. Each iteration adds knowledge about what the human expert does and what the requirements should be for the expert system. Each iteration reduces the risks and the uncertainties as to the feasibility and the practicality of using expert system technology for a given system. Our expert system development model borrows and builds on the best attributes of conventional software development methodologies. This includes the risk-driven methodology for conventional system development introduced by Dr. Barry W. Boehm [4]. However the focus in ESDM is on knowledge acquisition whereas Boehm's focus is on product development [5].

2 The Expert System Life Cycle

An expert system is a computer system that emulates the way human experts solve cognitive problems (i.e., problems for which no algorithm is known but which are routinely solved by a human expert). The expert system development task is, therefore, to acquire

CH3024-7/91/0000/0038/$01.00 © 1991 IEEE

information from the human expert and to model how the cognitive task is performed in a form suitable for a computer. The uncertainty associated with this process means that there is always an element of risk associated with expert system development.

Analyses of the development process revealed several differences between the expert system life cycle and either rapid prototyping or conventional software development [6]. While this life cycle closely resembles evolutionary system development based on rapid prototyping [7], two differences appear to be significant. First, the uncertainty regarding feasibility and practicality of an expert system is greater, at least at the onset of a project. Second, the nature of expert system development is highly iterative and that needs to be reflected in the life cycle adopted [8].

The major similarity between expert system development and rapid prototyping is that the espoused development strategy is a prototyping approach in which the system evolves in multiple stages [9]. As discussed in the following section, the ESDM life cycle for expert systems consists of five stages of successively more capable prototypes: feasibility, research, field, production, and operational. Further, each stage in the ESDM life cycle is a five-step process: problem identification, conceptualization, formalization, implementation, and testing.

A major source of frustration for both managers and technicians has been the seeming unmanageability of the expert system development process. There has been little guidance for producing a well-engineered product using prototyping and less for managing deliverables that are evolutionary in nature [10]. ESDM addresses both points.

3 ESDM

Our expert system development methodology presents a life cycle in which a prototype evolves through five stages of development. Each stage contains five steps leading to a prototype. A key point is that the expert system need not proceed through all five stages. ESDM recommends transition to a conventional software development methodology when enough has been learned about the problem to write system requirements. More specifically, ESDM focuses on driving out requirements so that development may be continued with a conventional software methodology.

Although independently generated, ESDM can be represented by a spiral model very similar to Boehm's spiral model for software development. The major difference is that ESDM focuses on knowledge acquisition rather than product development. Figure 1 illustrates the spiral nature of ESDM.

Since all of the expert systems produced are prototypes, ESDM does not recommend any of the standard software development products (e.g., feasibility study and requirements analysis). ESDM does recommend a set of products that can be used to manage the expert system development process prior to transition to a more conventional software development methodology. ESDM also provides guidelines for this transfer from expert system development to a conventional life cycle.

Figure 2 illustrates the ESDM life cycle and shows the five decision points that determine whether work moves to one or another of the five ESDM stages. In this methodology, iteration of a stage or iteration of steps within a stage may also occur. (Iteration of steps is not explicitly shown.)

In the ESDM life cycle, there are three major management decisions:

- Are expert system development techniques suitable for the problem?
- Are the requirements sufficiently well known for transition to a conventional software development methodology? If the answer is no, then the present stage or an earlier stage may be repeated or ESDM may continue to a new stage.
- Are the risks too high to continue expert system development? If the answer is yes and if the problem is not simply a wrong turn taken at an earlier stage (in which case that earlier stage should be repeated), the project should be aborted and the Project Termination Report generated.

As indicated earlier, the project need not proceed through all five stages if these questions can be answered at an earlier point. ESDM recommends a metric, the Test for Application of Risk-Oriented Technology (TAROT), that provides guidance to managers at each decision point.

3.1 The Five Stages

ESDM has five prototype development increments or stages [11]. Each stage contains five steps [12]. The following subsections discuss these stages. The next section discusses the five steps found in each of these stages. At the end of each stage, managers must decide (1) whether the risks are too high to continue and, (2) if the risks are acceptable, whether to iterate or continue to a new stage.

Stage 1: Feasibility Demonstration Prototype

This first stage seeks to determine if at least one key cognitive function of the expert can be

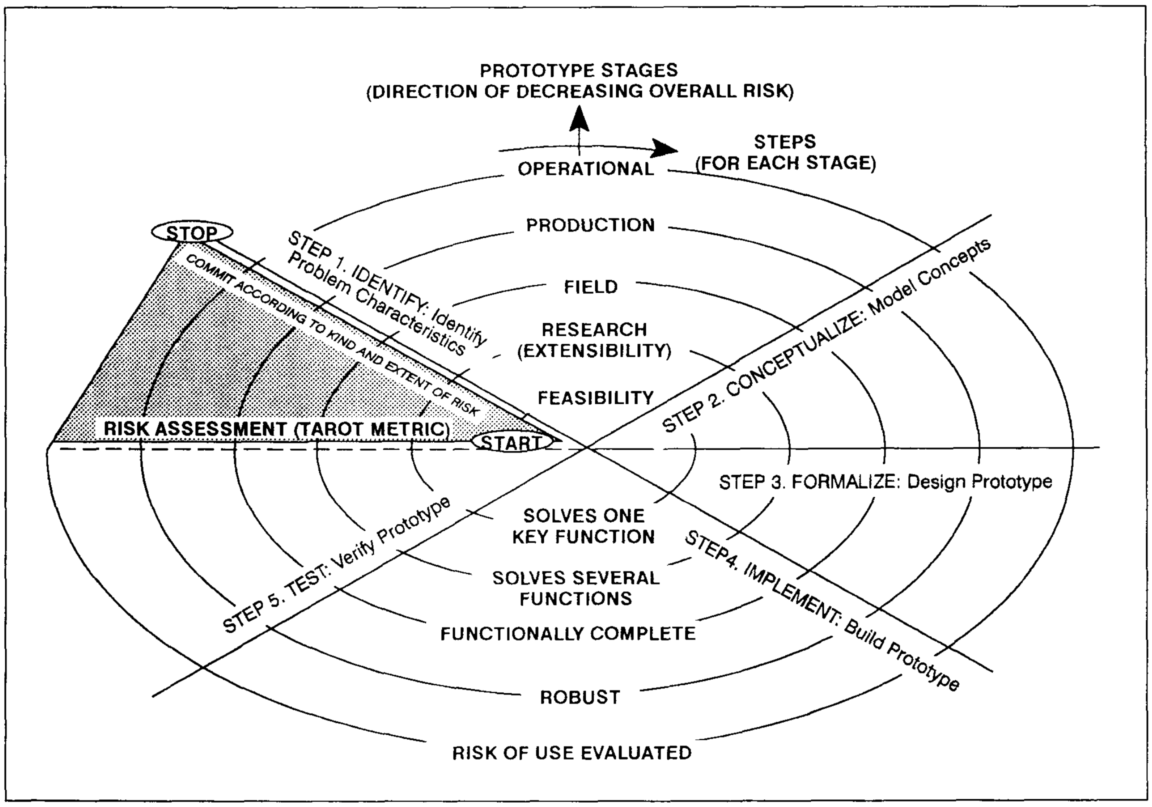

Figure 1. Spiral Model of ESDM

modeled. A key cognitive function is one that is central to the overall problem the expert solves. This stage prototypes at least one, and possibly only one, cognitive function.

Stage 2: Research Prototype

The second stage determines whether additional functions can be modeled. The goal is to determine the extent to which the entire set of cognitive functions of the expert can be modeled. This stage prototypes a sufficient number of the cognitive functions of the expert to answer this question.

Stage 3: Field Prototype

The third stage ascertains the combination of conventional and expert system techniques needed to model all necessary cognitive functions. The goal is a prototype that can be tested in a realistic setting. The prototype for this stage is functionally "complete," solves most of the required problems, and is reasonably robust although possibly lacking in performance. It may not handle all the situations that arise in realistic settings.

Stage 4: Production Prototype

The fourth stage asks whether the manual system can be modeled in a way that is robust and efficient. The goal is to determine if it is possible to achieve the desired scope of expertise and speed of operation. It may be necessary to refine the design, recode, or transport the field prototype to a new host. The production prototype solves nearly all required problems, is very robust, and is easy to use.

Stage 5: Operational Prototype

By the time the fifth stage is reached, the prototype successfully emulates the human expert. The goal is to show by exhaustive analysis and evaluation that the risks and costs of deployment are acceptable. The focus is now

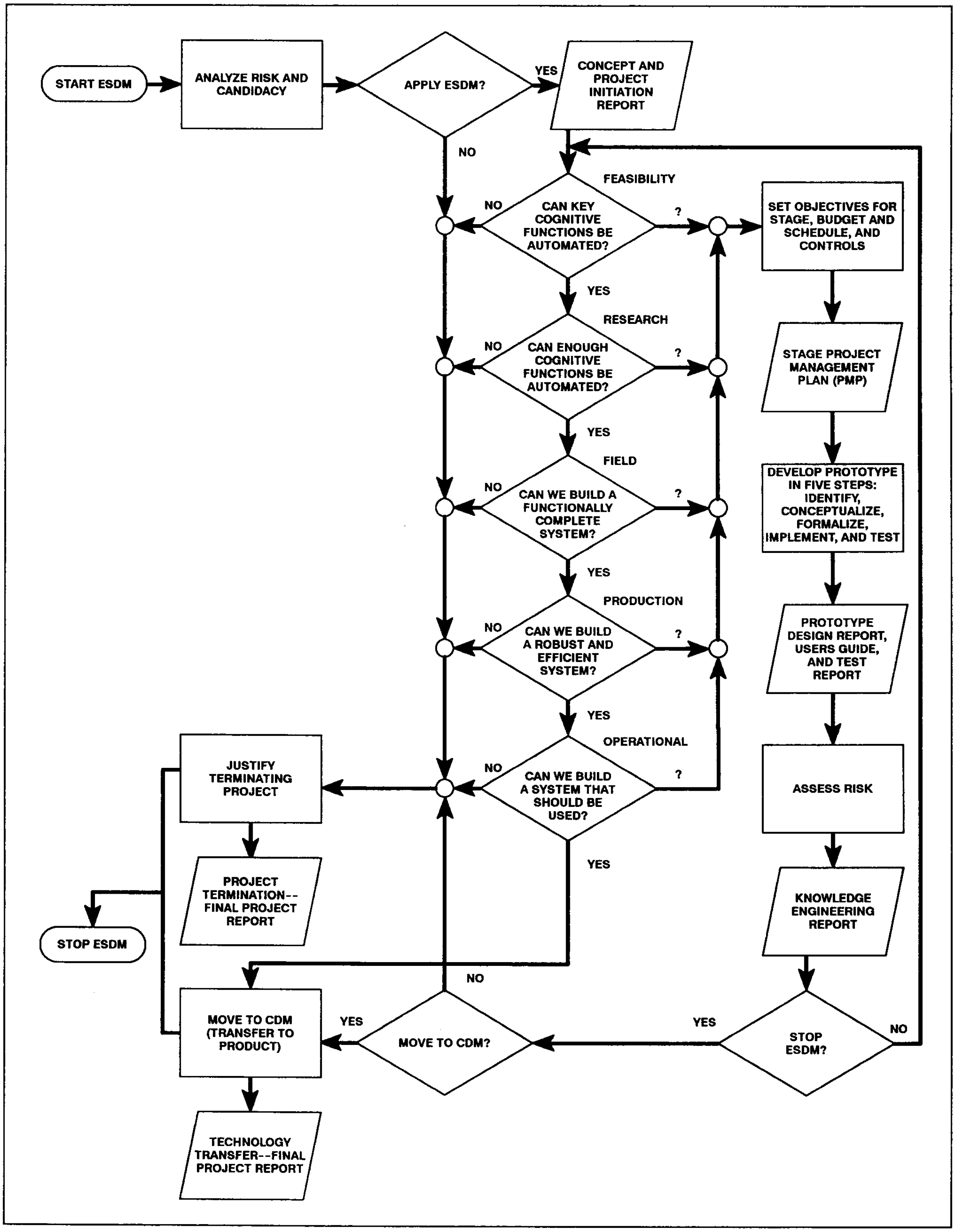

Figure 2. Expert System Development Methodology (ESDM)

on the use, rather than the structure, of the product.

3.2 The Five Steps

Within each of the expert system development stages, five steps accomplish the work. These steps are not necessarily performed in a sequential fashion (i.e., work can and does move from a later step to an earlier step as knowledge is acquired). These steps are highly iterative.

Step 1: Identify the Problem
: Determine what knowledge is used by the expert to perform the cognitive function(s) to be modeled.

Step 2: Conceptualize the Solution
: Acquire relations, information flow, constraints, and problem-solving strategies from the expert, organize these into concepts, and use them to model the cognitive functions.

Step 3: Formalize the Solution
: Map these concepts to a formal representation. This representation may be based on an already selected (mandated) implementation tool or language. If not, this formal representation should be used to select the tool or language.

Step 4: Implement the Prototype
: This task consists of expressing the already structured knowledge within the framework of the selected tool.

Step 5: Test the Prototype
: Validate the rules that embody knowledge; verify that the prototype implements the design; and test for robustness, speed of execution, and deployment risk.

3.3 Transition to Conventional Software Development

ESDM recommends that the expert system development project terminate when the feasibility and use risks are reduced to acceptable levels. When the decision is made to proceed with development and deployment of the expert system under a conventional software methodology, the expert system development findings are recorded in the Technology Transfer Report and serve to guide conventional system development. Transition from the ESDM expert system life cycle to conventional software development is illustrated in Figure 3.

ESDM recommends a formal transfer process and reassignment of some project personnel to the conventional software development team to handle problems or issues that might arise. Similarly, should the risks of either implementation or use be considered unacceptable, ESDM recommends that a final lessons-learned review be held and that the information acquired and the decision to discontinue development be recorded in the Project Termination Report.

3.4 ESDM Products

ESDM recommends that at least two products be generated at the project level: (1) the Concept and Project Initiation Report and (2) either the Technology Transfer Report or the Project Termination Report. The later two documents serve as the final project report. Which one is generated depends upon the outcome of the project. The Technology Transfer Report is generated when the decision is made to transfer further development and deployment of the expert system to a conventional software methodology. The Project Termination Report is generated when the expert system development is terminated whether because the risk of continuing is too high or because of external factors.

The Concept and Project Initiation Report is prepared following the initial risk and task suitability analysis. The document provides a high-level strategic plan for development, a description of the domain, a specification of the expert system to be developed, and the results of this initial risk and task suitability analysis. The Technology Transfer Report includes the project history, the lessons learned, a summary of each stage of expert system development completed, the knowledge acquired, the test cases developed, and the expert system requirements. The Project Termination Report is very similar except that the risk analysis and justification for project termination appears in place of system requirements.

Some products are produced for each stage in ESDM. These repeated products may be simple augmentations of their earlier counterparts or may be separate entities. If a stage in ESDM is iterated (as previously discussed), the products for that stage are also iterated. The products provided in ESDM following each stage of expert system development and some of their contents include:

- Knowledge Engineering Report
  - Summary of the knowledge acquired
  - Lessons learned during the stage
  - Recommendations for later stages
- Prototype Operations Guide
  - Installation instructions
  - System initialization
  - User interface description
  - Functional description
  - Scenario

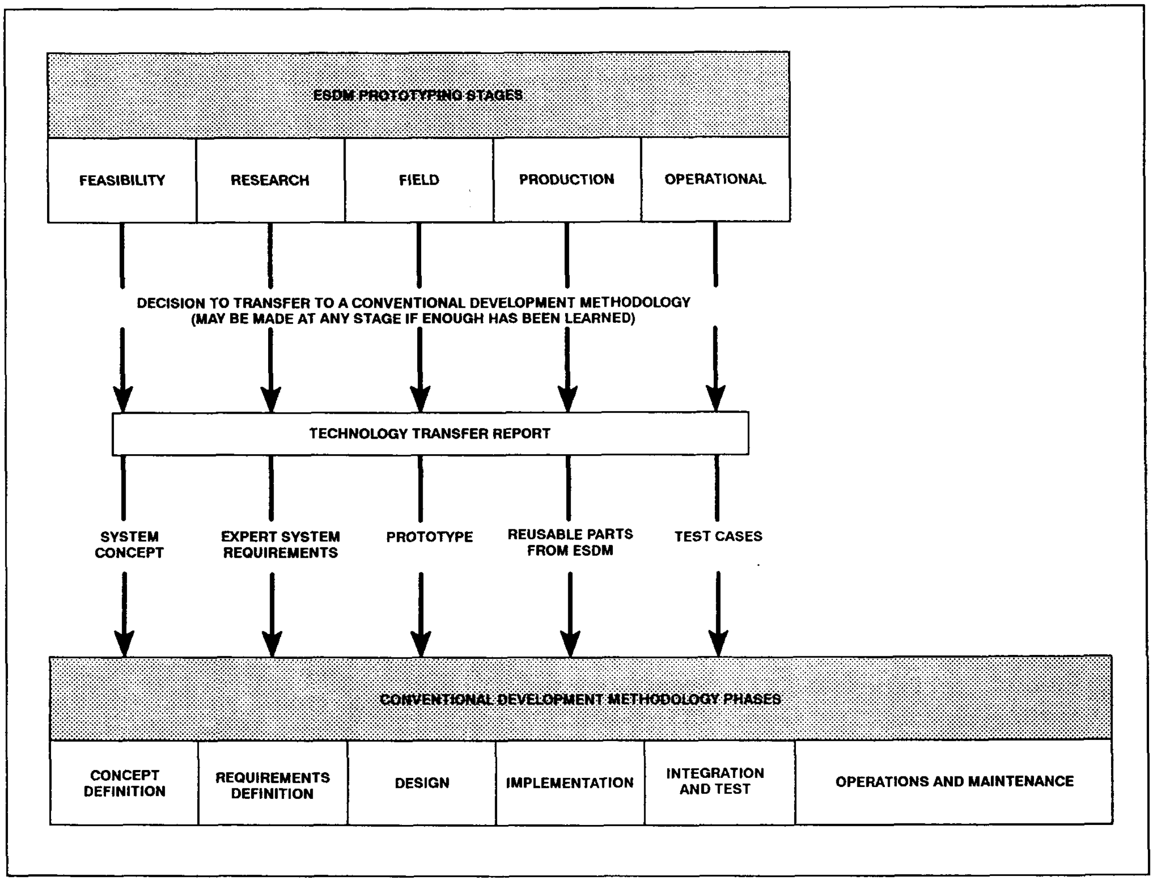

Figure 3. Transition to Conventional Software Development

- Prototype Design Report
  - Description of the stage prototype
  - Depiction of the interfaces
  - Requirements for the expert system
- Prototype Test and Evaluation Report
  - Test plan
  - Test cases
  - Test results
  - Evaluation
- Prototype

4 Current Activities

The purpose of an expert system development methodology is to provide guidance to managers and technical personnel. For managers to accept ESDM guidance, the methodology needs to provide effective insight into the development process and the means to measure and monitor progress. For technical personnel to accept ESDM guidance, the methodology must conform to and assist the extant development process. Because ESDM is also intended to serve as a standard for expert system development, the methodology needs to be comprehensive, complete, and unambiguous. Over the past two years, we have been evaluating the methodology to ensure that ESDM offers effective insight into and assists the development process. In particular, we have tried to ensure that our formal documentation of ESDM is both complete and understandable.

Our ESDM documentation is currently being updated to incorporate the recommendations, findings, and conclusions drawn from the methodology validation. Additional material concerning validation, verification, and testing of expert systems [13] is also being incorporated. The revised documentation will be available from the authors.

5 Summary

Our expert system development methodology provides both the life-cycle model and the formal, well-developed, risk management framework deemed by many expert system project managers to be essential for successful product development. In ESDM, the focus is on risk management and on driving out and validating system requirements. Among other features, ESDM requires the project manager to review progress, reassess risk, and make the conscious decision at the end of each step whether to proceed with development under ESDM, to transfer to conventional software development, or to terminate the project. We recommend that the project manager use a linear decision function to isolate the subjective step in this decision process and provide a specific example, the TAROT metric. ESDM also recommends and provides guidance for a set of project and step reports that can be used to bring the project under control (i.e., to monitor the process and assist in making the necessary decisions).

References

[1] Goddard Space Flight Center, *Expert System Development Methodology User Guide*, DSTL-90-004, Revision 1, August 1989.

[2] Goddard Space Flight Center, *Expert System Development Methodology Standard*, DSTL-90-005, Revision 1, August 1989.

[3] Goddard Space Flight Center, *Expert System Development Reference Manual*, DSTL-90-006, Revision 1, August 1989.

[4] Boehm, Barry W., "A Spiral Model of Software Development," *IEEE Computer*, May 1988.

[5] Boehm, Barry W., "A Spiral Model for Software Development and Enhancement," *ACM Sigsoft Software Engineering Notes*, August 1988.

[6] Medsker, L., LaSalle, A. J., and Hillmer, D., "Knowledge Acquisition and the Expert System Life Cycle," *Proceedings of the IEEE Conference on Managing Expert System Programs and Projects*, September 1990.

[7] Alpar, Paul, "Toward Structured Expert Systems Development," *Expert Systems With Applications*, Vol. 1, 1990.

[8] Kiss, Peter A., "A Model for a Knowledge-Based System's Lifecycle," *NASA Conference Publication 3059*, July 1989.

[9] Agarwal, Ritu, and Tanniru, Mohan, "Systems Development Life-Cycle for Expert Systems," *Knowledge-Based Systems*, Volume 3, Number 3, September 1990.

[10] Khan, A. F. Umar, "Mapping the Expert Systems Development Process to the Department of Defense Life-Cycle Model," *Proceedings of the IEEE Conference on Managing Expert System Programs and Projects*, September 1990.

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

[12] Waterman, D. A., *A Guide to Expert Systems*, Addison-Wesley Publishing Company, Inc., 1986.

[13] Gilstrap, Lewey, "Validation and Verification of Expert Systems," *NASA Conference Publication 3110*, May 1991.