

VALIDATION AND VERIFICATION OF EXPERT SYSTEMS

Lewey Gilstrap
Computer Sciences Corporation
Beltsville, MD

Abstract

Validation and verification (V&V) are procedures used to evaluate system structure or behavior with respect to a set of requirements. Although expert systems are often developed as a series of prototypes without requirements, it is not possible to perform V&V on any system for which requirements have not been prepared. In addition, there are special problems associated with the evaluation of expert systems that do not arise in the evaluation of conventional systems, such as verification of the completeness and accuracy of the knowledge base. The criticality of most National Aeronautics and Space Administration (NASA) missions makes it important to be able to certify the performance of the expert systems used to support these missions. This paper presents recommendations for the most appropriate methods for integrating V&V into the Expert System Development Methodology (ESDM) and suggestions for the most suitable approaches for each stage of ESDM development.

Introduction

Expert systems are mechanizations of the cognitive problem-solving capabilities of human experts. At the outset of expert system development, it is not known whether it is even possible to model and mechanize the mental processes of human experts, much less how long it would take and how much it would cost to do so. The procedures used to develop expert systems are directed toward extracting from the expert the knowledge and skills used in an expert task and toward preparing a series of successively more refined prototypes that mimic the behavior of the human expert. Expert systems tend to evolve rather than follow a planned course of development, and their life cycle differs significantly from the life cycle of conventional systems.

Because of the differences in the life cycles of expert and conventional systems and the need for sound management of expert system development projects, the Data Systems Technology Division of the National Aeronautics and Space Administration/Goddard Space Flight Center (NASA/GSFC) 4 years

ago undertook a program to formulate a methodology for expert systems to be developed at GSFC. The result of this effort was the Expert System Development Methodology (ESDM) (CSCa, 1988; CSCb, 1988; CSCc, 1988; Sary et al., 1990; CSC, 1989; Gilstrap, 1990).

Briefly, ESDM is a risk-driven development methodology. Areas of risk in the development are identified, and work on any one phase is directed toward reducing the next highest remaining risk. Risks due to uncertainty are inherent in any system development but are greater at the outset of expert system development because of the unknowns in the modeling of the human expert's cognitive behavior.

The system life cycle in ESDM is divided into five stages, each of which is further divided into five steps as shown in the spiral model in Figure 1. This spiral model is similar to that recommended by Boehm (1986) for conventional systems. The Boehm spiral model is also used for expert systems by Stachowitz and Combs (1987), Stachowitz et al. (1988), O'Keefe and Lee (1990), and others.

At the time the ESDM task was initiated, it was decided that the establishment of procedures and methods for performing validation and verification (V&V) would be deferred. The original focus was on methodology; however, V&V are concerned with both methodological and technical issues. Because of the criticality of NASA missions, it is necessary to be able to certify the behavior of expert systems and to verify their performance. A study, which is the basis for this paper, was undertaken to determine the steps necessary to integrate V&V into ESDM.

As pointed out by Green and Keyes (1987), V&V has seldom been done for expert systems. The primary reason is that requirements analyses are not always done for expert systems. Without requirements, it is, by definition, not possible to perform V&V. A secondary reason is that there are special problems, discussed in more detail later, in doing V&V for expert systems, even when requirements exist.

ESDM recommends that requirements be prepared in all expert system developments, but does not mandate them. However, requirements should not be

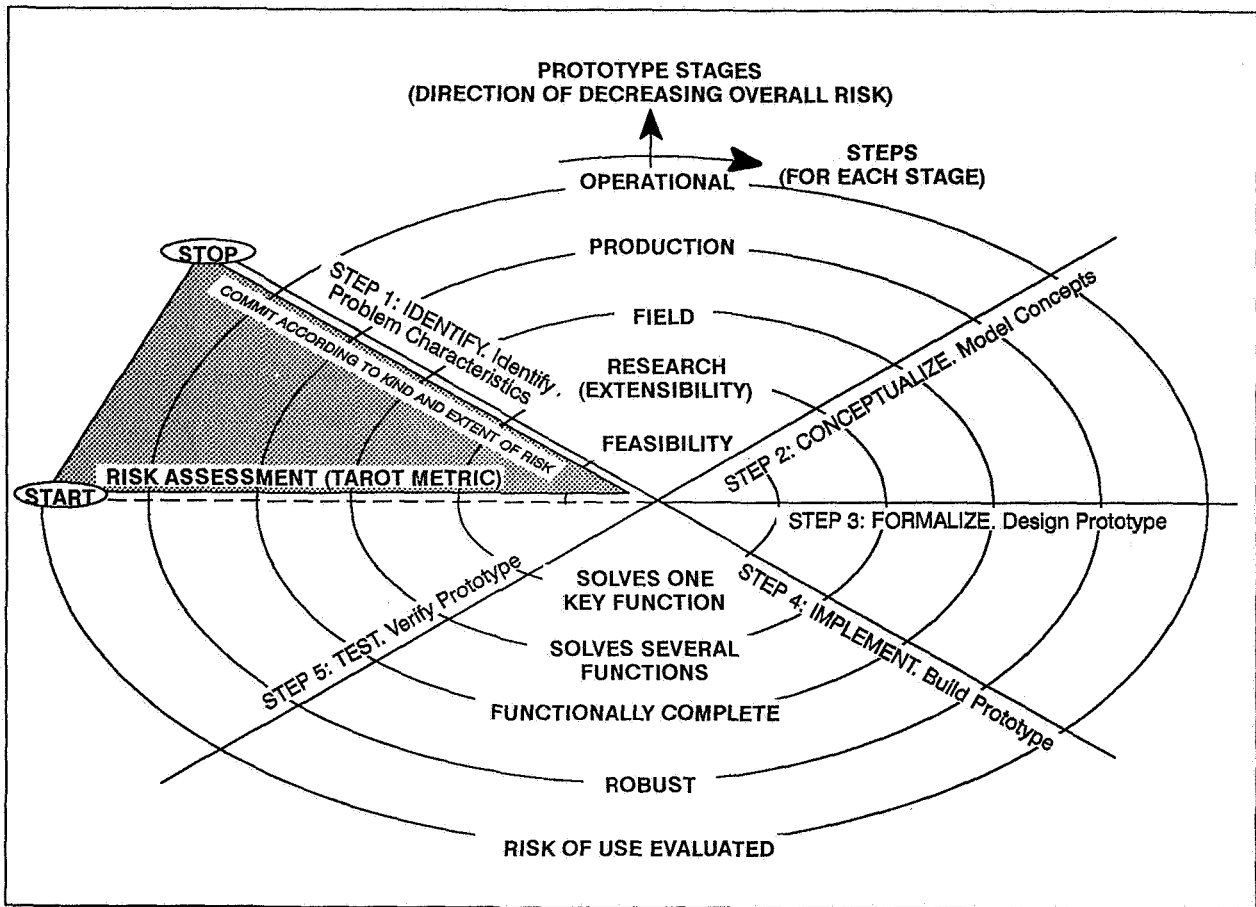


Figure 1. Spiral Model of ESDM

prepared until the major uncertainties about the system have been resolved. Typically, this does not occur until after the field stage of work has been completed.

Validation and Verification

The terms *validation*, *verification*, and *testing* are defined as follows in the Institute for Computer Sciences and Technology (ICST) special publication on V,V&T (1988):

“Validation, Verification and Testing (V,V&T): A process of review, analysis, and testing employed throughout the software development lifecycle...which helps ensure the production of quality software.

“Validation: The determination of the correctness of the end product (code) with respect to the software requirements, i.e., does the output conform with what is required?”

“Verification: The determination that each phase and subphase of the development life-cycle is correct, complete, and consistent with itself and with its predecessor product.

“Testing: The examination of program behavior by executing the program on sample or operational data sets to determine the correctness of the program.”

Verification is more concerned with the structure or form of a system, while validation is more concerned with behavior. In simple terms, as Boehm (1984) expresses it, in verification we ask

“Am I building the product right?”

and in validation we ask

“Am I building the right product?”

Green and Keyes (1987) point out that verification is, in many cases, a “paper” activity, while validation is a “live” activity involving testing. In conventional systems, validation and verification tend to find different types of errors; they are complementary processes, and neither is sufficient by itself to uncover all the errors in a system.

Testing and Inspection

The primary tools used in V&V are testing and inspection. The purpose of testing is to find errors, omissions, or unnecessary elements in a system. The most common types of tests used in conventional development are

- Unit tests—these are primarily path tests and are directed toward finding gross structural errors
- Integration tests—these are performed each time a new module is added to a system being developed
- System tests—these are performed to ensure that a system as constructed meets the requirements specified for the system
- Acceptance tests—these tests are performed by the customer or user on delivery of a system

Unit and integration tests help to verify that the system has been built correctly. System tests validate the system with respect to requirements. The validation of requirements (against the real world) may require special tests to be designed and executed. For example, in some conventional system development, field tests under actual conditions might be required to validate the system requirements. Such tests would be in addition to system tests for ensuring that the system satisfies requirements.

In addition to testing, software system inspection is also performed to uncover errors or to verify system consistency or correctness. Inspection techniques have also been used in developing expert systems, and a wide variety of special inspection techniques have been designed for expert systems. For example, the human expert is given the knowledge base for an expert system and asked to review it for errors of omission or commission. A number of these methods are listed and briefly described in the ESDM user’s manual (CSCa, 1988).

Provisions for Expert System V&V

Based on literature review and analysis of the problem, a methodology must provide for the following in order to support V&V:

- Requirements generation (O’Keefe & Lee, 1990; Culbert et al.[a], 1987; Culbert et al.[b], 1987; Barrett, 1990; Preece, 1990)
- V&V planning (Barrett, 1990), defining the following:
 - Objective of each test
 - Criteria for each test
 - Data required to measure attainment of objective
 - Data acquisition procedures for collecting required data
 - Analytical procedures for determining whether criteria have been met
- Guidelines (Green & Keyes, 1987) for the following:
 - Procedures for tests and inspections
 - Requirements-traceable tests
 - Parameters of system to be tested
- Special testing of knowledge-based systems, as required
- Automatic testing tools (Stachowitz & Combs, 1987; Stachowitz et al., 1988; Gupta & Biegel, 1990) to support the following:
 - Test requirements determination and documentation
 - Test planning
 - Execution of test and inspection procedures
 - Execution of special testing
 - Requirements tracing

Without requirements, V&V are simply not possible in the usual sense of the term. The cost of preparing requirements in some later stage of expert system development, after sufficient information has been acquired, is considered a good investment. Requirements provide the basis for testing all subsequent prototypes and are essential for testing critical applications in operational environments.

The methodology for developing expert systems must mandate test planning and provide for independent testing and inspection. It must also provide guidelines

and specify procedures for performing tests and inspections. Experience has shown that for best results, testing and implementation of expert systems should be independently managed, just as for conventional systems development.

Expert systems are not conventional systems, however, and we can test expert systems in ways that are impossible with conventional systems. The special kinds of testing that are possible for expert systems exist because of the knowledge base, the different types of knowledge representation, and the non-procedural methods of reasoning that may be used in expert systems. Examples of special tests include searching for contradictions in the knowledge base, ensuring that explanations given by the system are understandable and reasonable, and ensuring that the system does not produce patently absurd or illogical conclusions. Details of such tests may depend strongly on the specific knowledge representation used in the system and require more, perhaps much more, preparation than do tests to ensure adherence to system requirements.

Most expert systems developers would agree that it is desirable to have automated tools for testing, and some would argue that it is essential. Stachowitz and Combs (1987) state:

“It is our conjecture that software validation can be more easily performed in a knowledge-based system environment. In such an environment the number of life cycle steps is reduced from the traditional four (requirements development, specification development, design development, code development) to just the first two, resulting in a considerable reduction of the amount of validation work to be performed.”

Also, Gupta and Biegel (1990) list seven limitations of manual test planning and execution in support of expert system testing, ranging from possible lack of objectivity to the “astronomically large” number of test cases needed to extensively test a system. Certainly the likelihood is quite high that any complex expert system designed for highly critical requirements will require special tests that are not available from generic products. An expert system development methodology must make provisions for evaluating criticality and for developing special test procedures in parallel with the system, particularly in the high-criticality cases.

Special Expert System V&V Tests

Special tests, for which there may be no generic, commercially available testing tool, are conducted to resolve differences among multiple experts and to verify the following:

- Correctness of reasoning
- Inference engine
- Knowledge base
- Correctness and value of the output advice/actions
- Correctness of explanations
- Expertise of the expert
- System boundaries

In addition to validating or verifying the knowledge base, performance of one or more of the above tests may be necessary before the system can be certified for critical applications. Test design should be considered to be just as important as the design of the knowledge representation scheme, the inference engine design, or the knowledge engineering methods used on the project.

Culbert and colleagues discuss the issues involved in performing many of these special kinds of tests in the context of NASA systems (a, 1987) and present their own expert system development methodology (b, 1987), one that supports verification and validation. The approach they describe makes use of panels of domain experts, users, and managers with system responsibilities to ensure that all applicable viewpoints are represented, both during development work and during inspections (Culbert et al.[b], 1987). The life cycle used in this approach consists of four phases:

- Problem definition
- Initial prototype
- Expanded prototype
- Delivery/maintenance

The emphasis in this approach is not on the use of tests, automated or otherwise; rather, it is on ensuring that all relevant human skills are brought to bear in the development process—an important objective, and one that should be supported.

The mechanism used for verifying the correctness of knowledge bases depends on the kind of reasoning process used in the system. The framework for verifying knowledge bases that are developed for use with

ordinary logic is readily available. For example, resolution can always be used to mechanize the determination of rule-base consistency, although it is not the most efficient way. However, knowledge bases developed for nonmonotonic reasoning require an extension of the procedures developed for monotonic reasoning. Chang and colleagues (1990) discuss this problem and indicate how the automated testing tool, Expert Systems Validation Associate (EVA), could be extended to the nonmonotonic case.

The long-range goal of EVA was to construct an integrated set of generic tools to validate any knowledge-based system in any expert system shell (Stachowitz et al., 1988). It currently provides 11 tests that assist in verifying knowledge-based systems, ranging from structure and logic checks to test case generation and behavior verification. The addition of a capability for handling nonmonotonic logic would be a significant enhancement to the original system.

The problems of validating an expert or of resolving differences among multiple experts are of a very different order from the problems of verifying an expert system or features of an expert system. An evaluation technique called the *analytic hierarchy process (AHP)*, developed by Saaty (1980), was adapted by Liebowitz (1985) for evaluation of expert systems. The AHP technique makes use of pairwise subjective evaluations (Is A more important than B?) to achieve an integrated, global evaluation and ranking of many factors. A key feature of the method is that it is tolerant of intransitivity of relations (team A beats team B, which beats team C, which then beats team A). This technique can also be used to resolve the differences among multiple experts.

ESDM Modifications

The GSFC Expert System Development Methodology (ESDM), represented by the spiral model of Figure 1, currently supports some of the provisions of a methodology needed for V&V. The following lists the changes to ESDM needed to more fully support V&V:

- *Requirements generation.* ESDM does not now mandate the preparation of requirements. ESDM will be modified to make requirements mandatory in all high-criticality projects to ensure that V&V are possible.
- *V&V planning.* ESDM currently requires the design of tests to check for stage completion. ESDM will be modified to specify V&V for all systems for which requirements are

developed. ESDM documentation requirements will also be modified to make the test plan a formal project document.

- *Guidelines for procedures, tests, test parameters.* ESDM will be modified to provide guidelines for tests and testing procedures that can be used. The choice of tests for a particular project will be left to the project manager and the knowledge engineer.
- *Special testing.* ESDM will provide a checklist of special tests. This checklist will include the type of knowledge representation, the type of inferencing method, and the language or shell that is appropriate for each special test.
- *Automatic testing tools.* For critical projects, ESDM will specify an analysis of automated tool requirements. ESDM will also provide guidelines for estimating the costs, labor, and schedule to develop such automated tools in parallel with the expert system.

The above are the primary changes to ESDM needed to support V&V. ESDM will not mandate the specific tests that must be performed, but will provide guidelines that can be used by project managers and developers. In the future, the guidelines in ESDM are expected to be replaced by standards. At present, there is not sufficient experience with V&V in any expert system methodology to be able to define such standards. To force standards at this early stage might unduly restrict the technology of knowledge-based systems.

Evaluation of Expert Systems

Testing is a special case of evaluation of systems. As discussed by Liebowitz (1985), evaluations are helpful in determining whether an expert system is meeting its intended goals. The main difference between evaluation and testing is that evaluations are performed with respect to a specific set of objectives or purposes, whereas tests are evaluated with regard to correctness or performance. Evaluation usually requires the use of subjective judgment with respect to the objectives or purposes of the system:

“In your judgment, is system X suitable for application (or purpose) Y?”

The procedure for making evaluations is more elaborate than is implied by the above question. Usually, the features or characteristics of the system are

evaluated one at a time, and an overall evaluation is obtained by integrating or producing a weighted average of the estimates of suitability over all the features. In V&V, the objectives or purposes are specified by the requirements of the system, and subjective judgment is reduced to a pass/fail decision. The overall global decision to accept or reject is usually based on the percentage of cases that pass. For expert systems, it is not reasonable to require 100-percent success for all tests, because a human expert does not succeed 100 percent of the time.

The special tests that can be constructed for expert systems correspond to a set of pass/fail tests for the properties, characteristics, or features of expert systems. For evaluations of all sorts, it is recommended that the features selected for evaluation be both independent and mutually exhaustive of all relevant characteristics (i.e., they should cover all of the important properties of the system). A checklist of such features relevant to the evaluation of a potential expert system for suitability and worth was included in ESDM for use with the Test for Application of Risk-Oriented Technology (TAROT).

A number of authors have developed lists of expert system features, including Liebowitz (1985), Marcot (1987), Boehm et al. (1978), and Stachowitz and Combs (1987). The following list of system characteristics was obtained by consolidating and then eliminating duplications from these other lists. It is more general than the current ESDM list of system characteristics in that it covers more factors than those needed for evaluation of project suitability. It will replace the current ESDM list.

- Structural parameters
 - Completeness
 - Comprehensibility
 - Conciseness
 - Correctness (freedom from contradiction)
 - Consistency (freedom from anomaly)
 - Legibility
 - Modularity
 - Self-descriptiveness
 - Simplicity
 - Structuredness
 - Understandability

- Behavioral parameters
 - Utility
 - Accuracy
 - Effectiveness
 - Correctness (correctness of output)
 - Credibility
 - Intelligibility
 - Ease of use
 - Producibility
 - Speed
 - Operability
 - Efficiency
 - Flexibility
 - Interoperability
 - Maintainability
 - Portability
 - Reliability
 - Reusability
 - Robustness
 - Sensitivity
 - Stability
 - Quality
 - Completeness (breadth, depth)
 - Conciseness
 - Consistency
 - Correctness
 - Integrity
 - Maintainability
 - Reliability
 - Testability
 - Suitability
 - Worth
 - Risk
 - Benefits
 - Costs
 - Urgency/priority
 - Criticality

The special tests needed for a given expert system project are all related to one or more of the structural and behavioral characteristics listed above. System developers can use the checklist to help select those features that are critical in their application. Once selected, special tests must then be developed to examine each of the features or characteristics.

Summary

Although V&V are difficult to accomplish for expert systems, the main problem has been that requirements have, in many cases, been missing. Without

requirements, V&V are not possible. However, even when requirements are available, performing V&V on expert systems may not be easy because of the large number of different types of tests and inspections that are possible for such systems.

ESDM, the GSFC methodology for development of expert systems, is being upgraded to provide guidelines and checklists that assist managers and developers in planning, documenting, and performing V&V tests and inspections to certify expert systems for use in critical applications.

There is no one, simple, magic solution to the achievement of reliable, bug-free, certifiable software of any kind. High-quality, reliable software requires many different types of tests and inspections, as well as adherence to sound design and coding practices. The best way to achieve quality expert systems is for project managers and developers to be aware of the options available for testing and inspection and to plan on using the appropriate tools at the right time. ESDM modifications support these objectives.

References

- Barrett, B. W. (July 1990). A software quality specification methodology for knowledge-based systems, *Workshop Notes for AAAI-90 Workshop on Knowledge-Based Systems Verification, Validation, and Testing*, Boston, MA.
- Boehm, B. W. (January 1984). Verifying and validating software requirements and design specifications. *IEEE Software Journal*.
- Boehm, B. W. (March 1986). A spiral model of software development and enhancement. *ACM Software Engineering Notes*.
- Boehm, J., Brown, R., Kaspar, H., Lipow, M., MacLeod, G. J., & Merrit, H. H. J. (1978). *Characteristics of software quality*, North-Holland.
- Chang, C. L., Stachowitz, R. A., & Combs, J. B. (July 1990). Validation of nonmonotonic knowledge-based systems. *Workshop Notes for AAAI-90 Workshop on Knowledge-Based Systems Verification, Validation, and Testing*, Boston, MA.
- Computer Sciences Corporation (CSC) (a) (September 1988). Expert system development methodology (user's guide), DSTL 90-004.
- CSC (b) (September 1988). Expert system development methodology (policy document), DSTL-90-005.
- CSC (c) (September 1988). Expert system development methodology (reference manual), DSTL-90-006.
- CSC (March 1989). Expert system development methodology; framework for evaluation of ESDM.
- Culbert, C., Riley, G., & Savely, R. T. (August 1987). Approaches to the verification of rule-based expert systems. *Proceedings of SOAR'87: Space Operations—Automation and Robotics Conference*, Houston, TX.
- Culbert, C., Riley, G., & Savely, R. T. (September 1987). An expert system development methodology which supports verification and validation. *Proceedings of Fourth IEEE Conference on Artificial Intelligence Applications*, NASA/Lyndon B. Johnson Space Center, Houston, TX.
- Gilstrap, L. (May 1990). Evaluation of a proposed expert system development methodology: Two case studies. *1990 Goddard Conference on Space Applications of Artificial Intelligence*.
- Green, C. J. R., & Keyes, M. M. (1987). Verification and validation of expert systems. *Proceedings: WESTEX-87—Western Conference on Expert Systems*, IEEE, pp. 38-43.
- Gupta, U. C., & Biegel, J. (July 1990). RITCaG: A rule-based intelligent test case generator. *Workshop Notes for AAAI-90 Workshop on Knowledge-Based Systems Verification, Validation, and Testing*, Boston, MA.
- Institute for Computer Sciences and Technology (ICST) (1988). *Software validation, verification, testing, and documentation*, S. Andriole (Ed.), special publication on V,V&T, Petrocelli Books, NJ, p. 7.
- Liebowitz, J. (December 1985). Evaluation of expert systems: An approach and case study. The Second Conference on Artificial Intelligence Applications, Miami Beach, FL.
- Marcot, B. (August 1987). Testing your knowledge base. *AI Expert*, pp. 42-47.
- O'Keefe, R. M., & Lee, S. (July 1990). An integrative model of expert system verification and validation. *Workshop Notes for AAAI-90 Workshop on Knowledge-Based Systems Verification, Validation, and Testing*, Boston, MA.
- Preece, A. D. (July 1990). The role of specification in expert system evaluation. *Workshop Notes for AAAI-90 Workshop on Knowledge-Based Systems Verification, Validation, and Testing*, Boston, MA.
- Saaty, T. (1980). *The analytical hierarchy process*, McGraw-Hill, NY.

Sary, C., Gilstrap, L., & Hull, L. G. (May 1990). Expert system development methodology (ESDM). *Proceedings of Fifth Conference on AI for Space Applications*, Huntsville, AL.

Stachowitz, R. A., & Combs, J. B. (1987, January 6–9). Validation of expert systems. *Proceedings of Hawaii International Conference on Systems Sciences*, Kona, HI.

Stachowitz, R. A., Combs, J. B., & Chang, C. L. (1988). Validation of knowledge-based systems. In E. Heer, H. Lum (Eds.), *Machine intelligence and autonomy for expert systems*, AIAA, Washington, DC, pp. 125–142.