N91-22790

CAPTURING FLIGHT SYSTEM TEST ENGINEERING EXPERTISE: LESSONS LEARNED

Irene Wong Woerner
Jet Propulsion Laboratory
California Institute of Technology
Pasadena, California 91109

ABSTRACT

Within a few years, JPL will be challenged by the most active mission set in its history. Concurrently, flight systems are increasingly more complex. Presently, the knowledge to conduct integration and test of spacecraft and large instruments is held by a few key people, each with many years of experience. This expertise is unique to JPL and not readily available through academia or industry. JPL is in danger of losing a significant amount of this critical expertise, through retirements, during a period when demand for this expertise is rapidly increasing.

The most critical issue at hand was to collect and retain this expertise and develop tools that would help ensure JPL's ability to successfully perform integration and test of future spacecraft and large instruments.

The proposed solution was to capture and codify a subset of existing knowledge, and to utilize this captured expertise in knowledge-based systems. Such systems would be available simultaneously to numerous tasks and would also facilitate knowledge transfer to a new generation of engineers.

Consequences of not implementing a solution immediately were loss of expertise, thus potentially jeopardizing JPL's current preeminence in integration and test of spacecraft and large instruments. Further consequences of inaction were either a substantial increase in cost to adequately test future missions or inadequate test programs that significantly increase mission risk.

This paper describes first year results and activities planned for the second year of this on-going effort . Topics discussed include lessons learned in knowledge acquisition and elicitation techniques, life-cycle paradigms, and rapid prototyping of a knowledge-based advisor (Spacecraft Test Assistant) and a hypermedia browser (Test Engineering Browser). The prototype Spacecraft Test Assistant supports a subset of integration and test activities for flight systems. Browser is a hypermedia tool that allows users easy perusal of spacecraft test topics. This paper will also describe a knowledge acquisition tool called ConceptFinder which was developed to search through large volumes of data for related concepts and will be modified to also semi-automate the process of creating hypertext links.

Introduction

This paper describes an initial phase of an effort to capture a set of processes required for integration and test of flight systems, and to develop knowledge-based tools which utilize this captured expertise. This effort is part of the solution to help JPL preserve its expert knowledge of flight system test, make this knowledge readily available to less experienced engineers, and develop knowledge-based automated tools for future, more complex test programs.

Objectives

Major technical objectives for the first year were to 1) prove feasibility and effectiveness of combining multiple technologies, including expert system and hypertext, with test knowledge and models on a large scale; 2) establish an approach and architecture to facilitate systematic achievement of goals through building the necessary generic structure, tools, processes and technologies; and 3) capture, codify and automate a subset of current processes of integration and test.

These first year objectives were met by capturing a subset of information on flight system environmental testing, and using this captured expertise in two knowledge-based prototypes which assist in test engineering functions.

Background

Over the past 20 years, JPL's planetary program has successfully developed and tested a set of increasingly complex spacecraft. Until now, missions have been more or less serial in nature, such that integration and test of these spacecraft have been planned and accomplished by essentially the same set of key engineers and managers. Looking toward the future, two factors are converging that cause concern. First, as emphasis shifts from building spacecraft to building large-scale instruments, simultaneous test activities will be required. The number of experienced spacecraft and large instrument "test experts" is limited. Accordingly, major instruments would have test programs managed and conducted by less experienced engineers. Second, with only one exception, key lead individuals who comprise the experienced test team are all over 50 years of age.

JPL's risk of losing this capability has become more real. Last year, only 5 persons at JPL had knowledge and actual experience in managing a spacecraft test program. This year, there are only 4. There are 8 additional "test experts" of spacecraft and/or large instruments who could manage a large test program. Eleven of 12 experts identified are over the age of 50. The following chart illustrates the vast experience base, 420 cumulative years, JPL is in danger of losing.

	Experienced Managers		Test Experts (8)
	1989 (5)	1990 (4)	
Average Age	58	56.8	56
Total Years Experience	168	132	260
Average Years Experience	33.6	33	32.5

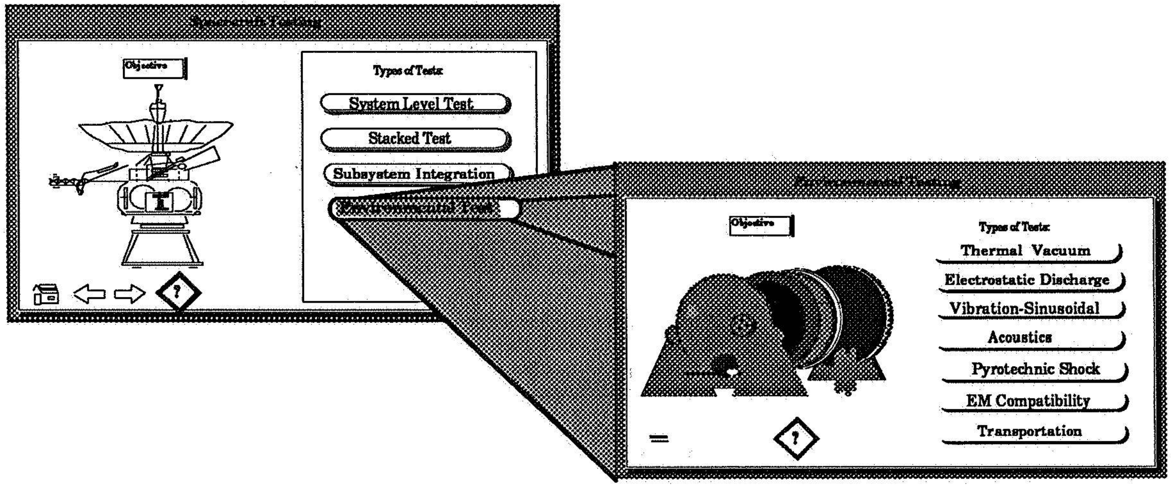
It is estimated that within 5 years, 80 percent of this expertise may retire. Within the first year, this body of experienced test managers has decreased by 20 percent, because one of our most experienced experts retired. The threat of losing our expertise is already being realized.

Technical Background

In expert systems, knowledge and control are separated. Knowledge is concentrated into a knowledge base, while control resides in an inference engine. A knowledge-based system capturing a subset of existing knowledge on flight system testing was viable for a number of reasons. For one, expertise exists and was available. This problem, loss of I&T expertise, was faced by an organization, rather than an individual. Also, there was a sufficient amount of information needed in decision making to justify the use of a "smart" information system, and the task of application was non-trivial, yet well-bounded and easily expandable. (Vassilio83)

Hypertext is a software methodology for presenting information in a non-linear fashion. Ted Nelson, a pioneer of hypertext, defined it as "a combination of natural language text with the computer's capacity for interactive branching, or dynamic display... of a nonlinear text... which cannot be printed conveniently on a conventional page." (Nelson67) An outstanding feature of hypertext is a physical realization of conceptual links which conventional text can only symbolize. (Monk88) A textual cross-reference to a related subject is an example of a conceptual link in conventional text.

Hypertext technology allows for retrieval of precise information needed for a specific task in an easy and cost-effective manner. For example in prototype Browser, if a user wants more information on environmental testing, pointing to that topic with a mouse and "clicking" brings up this information. From there, other environmental test related information can be accessed. In fact, a user can traverse to a detailed test procedure if that is the product desired. This example is shown pictorially below.

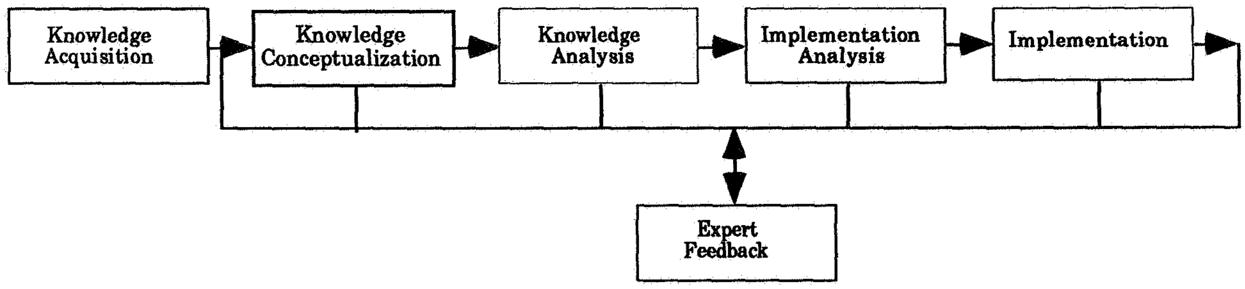


Adding hypertext technology to a rule-based system also provides an additional knowledge representation technique. The number of traditional rules required is reduced; specifically rules related to explanation. Feasibility of combining these two technologies has already been demonstrated in another JPL effort "Application of Expert System and Hypertext Technologies to Technical Document Preparation." (Wong90)

Approach

A knowledge engineering process was developed by tailoring and combining a process described by Hayward (Hayward88). and Barry Boehm's spiral model (Stachowitz87). Hayward defined a process that categorized knowledge engineering into five phases: knowledge acquisition, knowledge conceptualization, knowledge analysis, implementation analysis, and implementation. These phases were applied to Boehm's spiral model which essentially calls for a continuous iteration of these phases around an axis of time. This spiral process is typical of rapid prototyping development which involves multiple iterations. However, a departure from traditional rapid prototyping techniques is addition of expert feedback at each phase rather than solely during evaluation of a prototype. Rather than waiting until a prototype was implemented, experts were shown meta-products for verification of correctness. Rapid prototyping is a desirable approach, because it facilitates verification of implemented knowledge for correctness and completeness by a panel of experts, developers and management. With a rapid prototyping approach, meta-products and the prototype are incrementally evaluated and their effectiveness determined. Products are then improved iteratively, based on feedback, until experts and users are satisfied with results.

Involvement of experts in every phase of development was helpful on two levels. First, we could verify the correctness of what the knowledge engineer interpreted much sooner and more efficiently than waiting until a rough prototype was implemented. Although this required more "up front" time, errors are not propagated and are easier to correct; especially when the system is still a paper product. Second, experts are involved at every step of development and feel more ownership, because experts could see progression of system development and incorporation of information and data that they supplied. The following figure illustrates the modified approach:



In the first year, a subset of existing knowledge on integration and test of flight systems was captured in a structure-free, textual information base by transcribing one-on-one interviews with domain experts. Pre-conceived structure was purposely avoided so that information gathered would not be inhibited by how information is asked or stored. This information base is comprised of Wordperfect and ASCII files. Another benefit of a structure-free format is traceability of rules to pieces of raw data these rules were derived from. Furthermore, this process ensures gathered knowledge is not lost in translation from text to knowledge structure. It is also highly desirable to perform knowledge acquisition in such a way that results are reusable beyond present goals. For example, a different knowledge structure could be implemented at a later time, and the raw data would be available rather than just rules. Since a common word processing package was used for the transcription process, no training was required. This process has resulted in a raw information or "data" base that is equivalent to about 600 pages of text. Existing documentation which also contains a lot of embedded knowledge contributed another 500 pages of text.

There were several techniques which help in knowledge acquisition. Since the information base is large, tools were needed for information retrieval. Retrieval tools are a valuable aid to knowledge engineers, because the raw information base that needs to be searched is very large. Furthermore, existing technical documentation compounds the amount of data a knowledge engineer must sort through. A tool to help search for related "chunks" of concepts for encapsulation in a knowledge representation structure enables a knowledge engineer to transform data into information more efficiently. Nevertheless, retaining a complete raw information base is important.

Commercially available information retrieval tools were used to facilitate information retrieval from such information bases. However, a PC-based prototype, ConceptFinder, was developed and also used in conjunction. Reasons for developing ConceptFinder include an ability to search through either WordPerfect or ASCII files from within any application. ConceptFinder is a terminate and stay resident (TSR) program and related concepts are defined in a thesaurus-based fashion. For example, related concepts of thermal vacuum testing are hot/cold nominal testing, tailoring of chambers, and bake-out tests. Searching on thermal vacuum testing will also identify portions in the raw information base that mention bake-out tests or tailoring of chambers.

Furthermore, ConceptFinder was easily extendable to include hypertext authoring capabilities based on the same related concepts thesaurus. ConceptFinder gives a list of related concepts already defined in hypertext and facilitates establishment of links by generating the code needed to link a topic. This authoring capability paves the way for automatic generation of hypertext links later. A basic research question that still exists is how to establish context for automatic hypertext link generation. A technique we will try, within a limited domain, is statistical analysis of related words clustering. For example, if the word "power" is surrounded by words like "ground" or "voltage," links chosen would be different than "power" surrounded by words like "politics" or "money." However, this is not an effective method for establishing context in a general purpose or knowledge environment. (Salton89) Consequently, this is at best an inefficient work around until a better method is available.

After conducting several high level interviews on flight system test engineering to determine the general scope and relationships between large conceptual tasks, overall knowledge on integration and test was decomposed into logical modular units such as "integration of subsystems" and "environmental testing." Breakdown of a large task into manageable units help ensure objectives and design are robust, and easier to expand progressively and systematically in scope. The next level is to focus on one specific area; keeping in mind where this piece fits in the "big picture." This process helps codify what and how we do things. Stages of interview progress from orientation to identification to analysis. These stages represent the levels of detail that knowledge acquisition requires. Orientation is a grand tour which establishes an overall structure. Identification focuses in on one area, and analysis provides the rules and underlying relationships that exist for this one area of knowledge.

During the first year, the subdomain of focus was environmental testing, specifically thermal vacuum testing. This is a defined task which represents a subset of critical information necessary in most flight system test programs. Initial knowledge acquisition also included establishing a common language between expert and knowledge engineer. This common language was necessary to gradually minimize the ambiguities that accompany different and non-shared experiences. Consequently, the number of unfamiliar words or concepts diminish over time, and less time is spent in defining terms.

First year findings indicate knowledge acquisition can be categorized into three basic types: 1) past events - "I typically found dead shorts are caused by rework and miswiring;" 2) current activity - "Why do you check all grounds first? Because once the spacecraft subsystems are stacked, it's hard to access and test;" and 3) future events - "How would you test something like CRAF (Comet Rendezvous Asteroid Flyby)? I would first figure out what its mission is suppose to be, determine the kinds of instruments it has on board, etc." We found it easier to start with a retrospective question, "What did you do on x?" and integrating additional types subsequently. There are multiple reasons for this approach: 1) experts had just completed a major spacecraft project; 2) experts were not currently testing a systems and therefore, could not be shadowed to codify what they were doing and 3) future spacecraft were not adequately defined to determine details of a comprehensive test program. Consequently, the best types of questions to start with is very dependent on which activity is most immediate -- past, present or future.

To conceptualize knowledge, knowledge presentation diagrams were created. These diagrams are graphical representations of an expert's thought process as translated by a knowledge engineer. A knowledge presentation diagram is tangible proof to everyone that logically documenting expertise is possible. (Kearne90) It served as a focal point for further discussion with an expert and as a guideline for tool development. These diagrams were instrumental in translating existing data into rules, and were also used in creating hypertext links. They provided much of the control flow logic and also aided in creation of a top-level hypermedia navigation map. Studies have shown that a graphical representation of a "document's" structure vastly aided performance of a user's awareness of where they were in a hypertext document. (Simps89) This capability lessens the probability of a user getting lost in hyperspace. This condition occurs when a user has lost their point of initial reference, because of traversal through too many links.

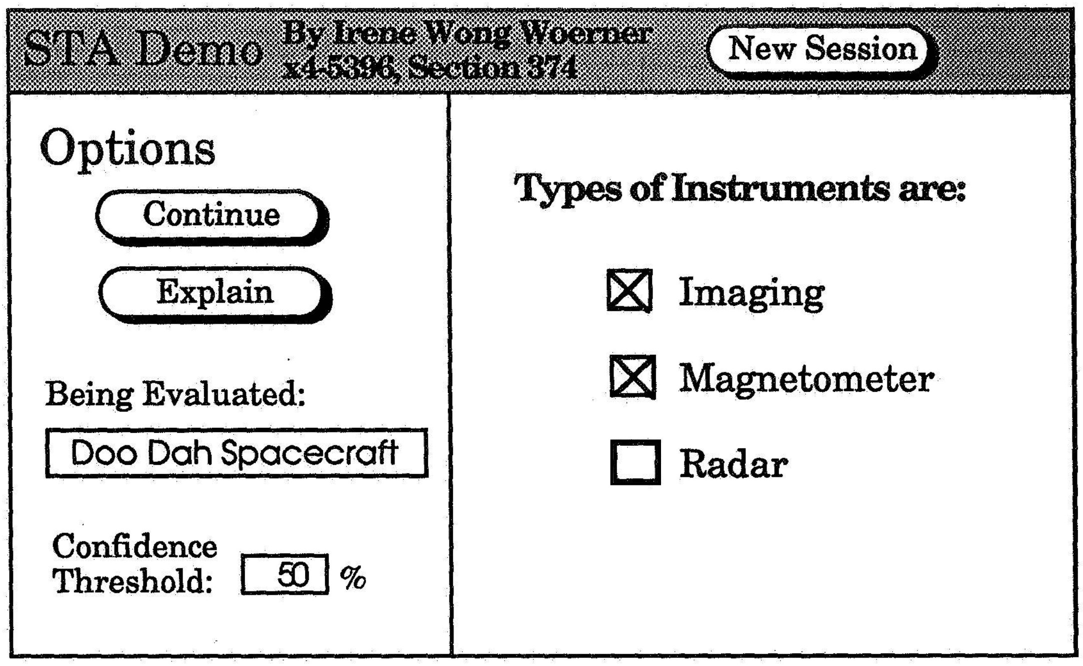
Knowledge analysis incorporated both epistemological and logical analysis. Epistemological analysis concentrates on examining structural properties of conceptual knowledge especially with reference to limits and validity (e.g. "Environmental testing includes vibration and acoustic testing, and you would use acoustic testing to simulate a launch profile."). Logical analysis determines control structure, and what factors are responsible for inference making (e.g. "If you have an imaging device, you'll most likely have strict contamination requirements"). Implementation is then based on results from analysis. Control structure typically lends itself to implementation in a set of rules. Conceptual knowledge, especially deep knowledge, is more difficult to represent in rules. Rules express surface knowledge fairly well, but the number of rules required to cover most potential possibilities or behavior that are manifested make rule-based systems difficult to manage and implement for large domains. One alternative planned for our second year activities is use of model-based reasoning; specifically causal models.

Another factor that has made this activity successful is the profile of knowledge engineers. Knowledge engineers are from a testing environment and therefore, have some understanding of the domain. This is significant because knowledge acquisition becomes a little easier. Since they already understand or speak the same language as the experts, communication is simplified and some time is saved in having to establish a common vocabulary. Knowledge engineering tasks are also divided up by pairs. A knowledge engineer who is more fluent in test activities and does well in interpersonal relationships conducts most interviews and performs knowledge analysis for focussing subsequent interviews, clarifying information and providing deduced rules and relationships to the developer. The other half of this team is a strong developer who also does knowledge analysis, but takes rules and relationships deduced and implements them in a working prototype. Both knowledge engineers, however, must work closely together and with experts.

Prototype Tools Developed

Based on knowledge captured in environmental testing, a prototype expert system advising engineers in one integration and test task was developed to demonstrate feasibility of utilizing captured knowledge in a tool to perform a specific task. The prototype Spacecraft Test Assistant (STA) serves as an assistant to a human, who evaluates the expert system's recommendations and acts based on confidence in those recommendations. By capturing existing specific knowledge, an "on-line expert" was available to assist less experienced engineers in testing well defined aspects of spacecraft and large instruments, thus freeing experts to solve more complex and anomalous problems.

STA is a rule-based system implemented in a commercial PC-based expert system shell. Rules deduced from environmental testing allow STA to provide a preliminary list of types of tests to conduct, how to perform these tests, and some diagnostic help in the event of failure. With online capabilities, a test engineer can determine needed environmental tests in less time than traditional manual methods. For example, if a spacecraft has an imaging instrument, STA infers with high probability that this instrument contains a charged coupled device (CCD) and therefore, has strict contamination requirements. If a user wants rationale for why this decision was reached, clicking on an explanation "button" will produce a window explaining that a CCD functions best at cold temperatures and therefore, acts as a cold trap for contamination which would impair its capabilities. Consequently, strict contamination requirements are needed to prevent this. Based on strict contamination requirements and other factors, STA would suggest types of thermal vacuum tests that must be done such as bake-out tests, which chamber to use, how to tailor this chamber to meet strict contamination requirements, how to run a bake-out test, where best to place sensors such a residual gas analyzer and how to measure for contamination. The following is a typical query screen for STA:



Being Evaluated: Doo Dah Spacecraft

Confidence Threshold: 50 %

Types of Instruments are:

☒ Imaging

☒ Magnetometer

☐ Radar

Another prototype developed was Browser. Browser is a hypermedia application implemented in a commercial PC-based tool. Although not a knowledge-based system per se, links created in Browser are very knowledge intensive. Based on knowledge presentation diagrams depicting decomposition of flight system test engineering into logical modular units, raw data is organized into hypermedia for easier browsing. Since relationships between information and data sets are very important, a hypermedia tool provides more insight than, for example, a hard-bound text book. A good source for structural information was existing technical documentation. Knowledge, especially process knowledge, is often embedded in documentation like management plans. For example, a test engineering management plan on a particular spacecraft integration and test program provided a good basis for a test taxonomy for spacecraft test and a source for determining the framework of how decomposition should occur. Often existing documentation can describe a "classic" model of processes. However, real world usually deviates from this model, and experts are required to provide the richness of knowledge that is required to tailor or understand the processes that actually happen. Much of this richness is in the form of heuristics derived from years of experience and lessons learned. There are examples that can be recounted of anomalies that were encountered that may require a change in the basic model. Design of knowledge-based tools should consider ease of changes an integral part of design, because changes can occur even at the most fundamental model level after prototype implementation has already occurred.

An important factor to consider before implementing a hypermedia tool is availability of resources to define and implement relational links. Cost-effective hypermedia systems will become more of a reality if more automated authoring tools are available, because the current method of establishing links is very labor intensive.

Future Activities

Some activities planned for the second year include interfacing STA and Browser. If a user is unsure of what a query is asking or wants even more information on why a decision was reached, they can browse through a large set of related data before continuing in STA.

Another capability is coupling these tools with commercial data acquisition systems to actually run a test and perform knowledge-based analysis on results. Several micro-based data acquisitions are currently being evaluated.

A major technical objective for the second year is integrating another subdomain, Spacecraft Assembly Facility testing, with the existing knowledge base and creating a systematic process for expansion of scope. Another area that requires work is conflict resolution. Currently, the scheme is to provide conflicting information separately and to indicate the associated expert. However, a technique we will try is interviewing several experts together to resolve conflict. Another technique we will try is recording interviews with video and audio media. We are hoping to discover insight and nuances provided through body language and expression. How this will be transcribed and searched is still an open issue. Video media may, in fact, be more cost-effective than micro-audio cassettes because of large availability.

Potential benefits from this effort are vast, because both its products, such as captured knowledge base and tools, and its processes, such as application of multiple technologies to a specific problem domain, are useful and have broad applications. First, captured expertise on flight system test and integration is irreplaceable and is now in a usable form. Furthermore, retaining this extensive experience base assures continued excellence in JPL's test programs by securing existing knowledge and evolving this knowledge base as new test engineering concepts are acquired.

The research described in this paper was carried out by the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration.

References

Hayward, S.A. et al. "Structured Analysis of Knowledge." Knowledge Acquisition Tools for Expert Systems. Boose, J.H. and Gaines, B.R. (editors), Academic Press Limited, 1988.

Kearney, Michele. "Making Knowledge Engineering Productive." AI Expert. July 1990.

Monk, A. F., Walsh, P. and Dix, A.J. "A Comparison of Hypertext, Scrolling and Unfolding as Mechanisms for Program Browsing." People and Computers IV. Cambridge University Press, Cambridge, 1988.

Nelson, T.H. "Getting It Out of Our System." Information Retrieval: A Critical Review. G. Schechter, ed., Thompson Books, Washington D.C., 1967.

Salton, G. and Buckley, C. "On the Automatic Generation of Content Links in Hypertext." Technical Report. Department of Computer Science, Cornell University, Ithaca, New York, April 1989.

Simpson, A. "Navigation in Hypertext: Design Issues." Proceedings from the 13th Annual International Online Information Meeting, London, England, December 1989.

Stachowitz, Rolf A. and Combs, Jacqueline B. "Validation of Expert Systems." Proceedings of the Twentieth Annual Hawaii International Conference on System Sciences, 1987.

Vassiliou, Yannis, et al. "Expert Systems for Business Applications - A Research Project at New York University." Database Engineering, Vol. 6, #4, Dec. 1983.

Wong, Irene. "Application of Expert System and Hypertext Technologies to Test Document Generation: An Update." Proceedings of the Seventh International Test Conference on Testing Computer Software, June 1990.