1

Design, Development and Validation of Expert Systems : A Survey of Developers

Daniel E. O'LEARY

ABSTRACT

This paper discusses a survey of developers of expert systems. Although previous researchers have developed frameworks and performed case studies, there is little empirical work on developers, views about the two primary development methodologies (prototyping and traditional software engineering), determining who performs the validation effort, the relationship between the validation budget and actual expenditures, and the use and effectiveness of various validation processes.

Prototyping is found to yield more robust models, without requiring more effort, than alternative methods. In addition, prototyping is found to be amenable to incorporation of activities such as requirements definitions, milestone setting and establishing completion dates.

It also is found that the effectiveness and the use of validation techniques are highly correlated. Interestingly, roughly 98% of the validation appears to be done using test data, direct examination of knowledge bases and parallel use of systems. Validation budget areas are used and sometimes exceeded, thus, indicating that validation is a critical activity that is performed by most developers.

As in knowledge acquisition, human participation in the validation process appears to be a major bottleneck. Experts appear to be the source of roughly 50%

of the validation effort. This suggests the need to develop techniques that allow greater leveraging of expertise. Finally, in an assessment of a wide range of potential problems, the completeness, correctness, and accuracy of knowledge bases were found to be the most important problems. This indicates where to focus validation resources and efforts.

1. INTRODUCTION

Much of the expert systems research [4] is theoretical or single case study oriented (e.g., the development of a single system). Little empirical work has been done beyond the analysis of single systems. The purpose of this paper is to mitigate that gap by investigating two basic sets of issues: design and development of expert systems (primarily through and examination of the acceptance and use of the prototyping approach) and validation of expert systems. This was done using a survey of developers.

1.1 Design and development of expert systems

There has been substantial research in the area of prototyping general information systems [9] and decision support systems. The extensive bibliography of [7] illustrates that wide-ranging literature. The work on prototyping research in information systems has ranged from theoretical to empirical. For example, [2] compared prototyping information system to more structured life-cycle approaches.

Many expert systems and artificial intelligence researchers was promulgated the use of prototyping [6] and others have examined the uses and controls appropriate for prototyping of expert systems [12]. Still, as expert systems have moved out of the research lab and into practical settings, the implications for implementation of prototyping in a "real world" setting have not been explored. For example, does the prototype approach interfere with the validation process ?

In addition, other researchers have begun to emphasize the use of more traditional software engineering approaches [3]. However, unlike general information systems, there has been little empirical research to compare the acceptance and use of those two methodologies in the development of expert systems. Mitigating that limitation is one objective of this paper.

1.2 Validation of expert systems

Closely associated with the development of an expert system is the validation of that system. Validation researchers have developed frameworks for the validation process [12, 13] provided case studies of the validation process [15] and analyzed the need for validation efforts in the development of expert systems [11].

Unfortunately, there is little evidence to indicate that any given validation approach is more effective or even which approaches are used.

In addition, there are a number of other implementation issues that have not been addressed in the literature. For example, do developers of expert systems have budgets for validation, and what is the relationship to actual expenditures.

Similarly theoretical analyses have been developed for who should perform the validation process, yet little empirical evidence exists on what is done in practice. Thus, a second objective of this paper is to gather empirical evidence as to the methods used to validate expert systems, examine the validation budget process and understand who actually performs the validation process.

1.3 Biggest problems

Since the validation process is so intertwined with the design and development processes, there is some interest in determining the relative importance of the validation process. One approach to that issue is to ask developers of expert systems to assess the importance of different potential problems that include both validation issues and more general issues of system performance. If validation issues are found to be important then their rank will either be about the same or more important than those other issues.

In addition, it is critical to be able to assess which validation issues are the most important. Since constrained resources are always an issue in a system development process, there is a need to obtain guidance on where to focus validation efforts. Unfortunately, there is little empirical evidence on what are the biggest potential system development and validation problems. This is a third objective of this paper.

1.4 Uniqueness of artificially intelligent systems and expert systems

This paper investigates the design, development and validation of expert systems. Thus, it is assumed that expert systems are different from other types of computer systems. It is not the purpose of this paper to explore those differences. However, those differences include : a focus on symbolic knowledge rather than numeric ; investigation of previously unstructured problems ; and inclusion of both symbolic and numerical information in the same program (e.g., rules of the form "if then" and uncertainty factors on the weights) ; and the general lack of a means by which to determine the quality of a solution, other than a comparison to a human's decision.

1.5 Plan of this paper

This paper proceeds as follows. The design and development and the validation are treated separately in the next two sections. Section 2 reviews the previous research on the design and development of expert systems using prototyping and software engineering approaches and summarizes the specific issues to be examined. Section 3 reviews some of the primary validation issues in expert systems and summarizes the specific issues examined in this research. Section 4 discusses the test instrument and sample that participated in the study. Section 5 discusses the findings. Section 6 briefly summarizes the paper.

2. DESIGN AND DEVELOPMENT OF EXPERT SYSTEMS

As noted above there are two basic theoretical design approaches that have been promulgated for the design and development of expert systems : prototyping [6] and [12] and software engineering [3,16]. Those studies provide a basis for the choice of some of the relationships examined in this paper.

In addition, there has been research on the use of prototyping in information systems and decision support systems. Although expert systems are unique, because they are computer systems, it is likely that some of the variables important to prototyping information systems and decision support systems will be also important for expert systems. However, this is not to say that the effects will be the same. Instead, it is likely that the unique aspects of expert systems will have an impact on the type of effect of the variables. These studies are another source of some of the variable studied in this paper.

The following summarizes the research issues addressed in the test instrument used as the basis of this paper. The subsection titles include in parentheses the questions in the research study discussed in those subsections. The questions are summarized in table 2.

2.1 Prototyping leads to robust models (1)

A critical aspect of an expert system is its "robustness". If the system is unable to respond to the demands placed upon it, then the system may not receive continued use. Thus, robustness generally would be viewed as a highly desirable characteristic. The interactive development associated with the prototyping approach would appear to lead to a more robust system since there would be constant user feedback on the quality of the system's responses.

However, Boehm et al. [2] found that prototyping yielded a slightly lower level of robustness for general information systems. There are at least two reasons why this finding probably would not hold in the area of expert systems. First, the finding may be, in part, a function of the level of the complexity of the information systems that were developed. If the systems are not complex then an initial requirements analysis may be able to anticipate the primary and additional needs of the users. Since an expert system is likely to be addressing a more complex or initially less structured problem, it would appear that the substantial user feedback associated with prototyping would contribute to the robustness of the system. Second, in the development of an expert system there is typically substantial asymmetry of expertise : the expert knows more about the domain than the developers of the system. Unless the knowledge engineer also is an expert, it would be difficult to anticipate additional user needs as may be done in more traditional information systems. Unless the system contains sufficient domain knowledge, it will not be robust. Thus, the interaction associated with prototyping could help establish a more robust system.

2.2 Prototyping facilitates development of specifications (2)

One of the differences between a traditional software engineering approach and prototyping is the generation of requirements. If the objectives and processes of the system are well known, a priori, then the development of the specifications is a relatively structured task. However, in the case of expert systems, the tasks the systems are to accomplish often are not, a priori, well structured. The notion of prototyping [6] is consistent with the iterative development of specifications. Prototypes offer a vehicle with which developer and client can use to agree on specifications. This can be particularly important when the system is being developed for some other company. The author is familiar with a consulting firm that has indicated that it uses prototypes as the basis of eliciting a set of specifications. Thus, prototypes may be a useful tool in delineating specifications in other firms.

2.3 Prototypes are difficult to manage and control (3)

Alavi [1] found that "the designers of the prototyped information system experienced more difficulty managing and controlling system experienced more difficulty managing and controlling the design process." That finding was attributed in part to the higher level of user participation and the more frequent changes in user requirements. There does not appear to be any unique aspect in assuming that expert systems should be any easier to manage and control with the prototyping process.

2.4 Software engineering as an approach for expert systems (4)

An alternative to the prototyping approach is the software engineering approach. As a result, recent researchers such as Bull et al. [3], have suggested the use of software engineering, rather than a prototyping approach. They report successful use with that approach. Thus, we would anticipate that software engineering would be a feasible tool for developing expert systems.

2.5 Management tools for prototyping (5, 6, 8)

Alavi [1] found that project managers suggested that prototyping was more difficult to manage, in part since " due to the newness of the nature of prototyping, there is a lack of know-how for planning, budgeting, managing and control". Thus, management may be more difficult because prototyping might not allow easy use of milestones, completion dates or allow for the development of formal deliverables, all part of the traditional software engineering process [16].

As a result, in a discussion on "how to prototype" Alavi suggested integrating devices of this type (milestones, completion dates and deliverables) into the prototyping process. In addition, researchers such as Bull et al. [3] have promulgated traditional software engineering development of expert systems, that include such devices. Thus, it would be expected that devices associated with the management and controlling of systems projects would be embedded into either a prototyping approach or software engineering approach and that they would facilitate management of the process.

2.6 Prototyping makes it more difficult to validate (7)

Since prototyping is less structured than more traditional software engineering methods and more difficult to control, it has been found in field studies of the use of prototyping in information systems [1] that testing activities could be bypassed or performed superficially. This would suggest that prototyping could inhibit validation.

However, in the case of expert systems, since expertise is present in the expert, the interactive approach of prototyping offers an important opportunity to validate the system. In particular, in some cases only the expert will be able to determine if the system is solving the problem correctly or solving the correct problem. As a result, the prototyping process is likely to facilitate a timely and constant validation effort.

2.7 Prototyping takes more effort (9)

In an experiment of seven software teams, Boehm et al. [2] found that prototyping yielded products that were about the same in performance as more traditional software engineering approaches, yet had 40 percent less code and 45 percent less effort. There is no apparent reason to assume that the expert system environment would be any different. Thus, prototyping of expert systems would be expected to take less effort.

3. VALIDATION OF EXPERT SYSTEMS

Researchers have developed frameworks in which to couch the validation efforts and they have provided a number of different methods to assist in the validation process. However, there is little empirical work available regarding which tools are used, how effective they are, who performs the validation efforts, do budgets exist for validation, what happens to validation budgets (are they used) and what are the primary validation difficulties.

3.1 Validation methods

O'Leary [11] identified a number of different methods for the validation of expert systems, including : testing with actual data, testing with contrived test data, direct examination of the knowledge base by either an expert or the knowledge engineer, parallel use of the system by either an expert or a non-expert, and contingency tables. Of concern are at least two sets of issues. First, to what extent are these validation tools used, if at all. Second, to what extent are these tools effective.

Testing with actual data, tests the system with the same type of requirements that it is likely to face if it is implemented. Since the validator is able to assess the system in "real life" situations (deriving some level of "comfort") and see the direct effectiveness of the system in those situations, the effectiveness may exceed the extent of the effort.

Testing with contrived data can focus the validation efforts on either extreme points or particular parts of the system of greatest concern. Unfortunately, the development of test data may take a substantial effort, while the success at ferreting out problems is not necessarily guaranteed. In fact, the difficulties of developing contrived data have led some to suggest that intelligent systems be developed to design that test data.

Direct examination of the knowledge base is another tool that can assist in the validation process. If the knowledge is in a readable form then it may prove quite useful for the expert to examine the system. Unless the knowledge engineer understands the domain a similar examination by the knowledge engineer may not yield useful insights.

Another approach that can be useful for validation of the system is to use the system in parallel with the decision maker. Such parallel usage may be done by the expert or a naive user. Since the expert typically is actively involved in the development of such systems, parallel use by the expert may not yield substantial insights. Alternatively, use by a non-expert may generate a substantial number of validation inquiries. This may called the "emperor's new clothes effect". A user unfamiliar with the system and many of the assumptions entailed may find many difficulties.

Finally, other statistical tools or contingency tables might be used to assist in the validation process. These validation methods are summarized in table 2.

3.2 Who performs the validation process

O'Leary [14] elicited a number of different potential validators including : the same expert from whom knowledge was gathered, a different expert than from whom the knowledge was gathered, the knowledge engineer, the end user, a sponsor of the system, or an independent validator. Each of these potential validators is listed in table 3.

Using the *same expert* from whom the knowledge was gathered has both advantages and disadvantages. In the first case, the expert has been a part of the project and thus, the effort has captured their expertise. In addition, the expert has thought about what it takes to structure the problem as a system. Finally, that expert is accessible. In the second case, since the expert has worked with the system this might indicate that overlooked assumptions continue to be overlooked and that the expert may be too close to the system design and development effort to note inconsistencies, incompleteness, etc.

A different expert offers advantages such as the potential for another view of the problem and an awareness of unstated assumptions in the model. However, there are also potential disadvantages such as a different model of the world (conservative vs. liberal), a lack of understanding of the problem that comes from the development of such a system or a lack of availability.

There is some evidence that the knowledge engineer becomes a "near expert" as they work on particular expert system projects [8]). This is critical to the validation effort, since if the knowledge engineer does not have an appropriate level of expertise then there may be only limited ability to validate the system. However, the knowledge engineer may be too close to the system or may have a vested interest in the system's use. In each case, these issues could influence the validation judgments of the knowledge engineer.

The ultimate *end user* may also play a part in the validation effort, depending on the extent of understanding of the problem. In many cases expert systems are developed so that decision making can be pushed down in the organization to lower level decision makers. In these cases the user may not provide substantial insight. However, for those cases where the systems are designed to assist a decision maker to ensure uniform quality or increased speed of processing, the user may provide substantial insights.

In some cases there may be a *sponsor* of the project. Again, as with the end user, the understanding of the sponsor can play a critical part. Probably the greatest independence of analysis would come from an *independent validator*. Unfortunately, there is little evidence to indicate that there is very much of that activity taking place.

3.3 Validation budget

As noted in Alavi [1], one concern with using prototyping of information systems is that testing could be either bypassed or performed in a superficial manner. A measure of whether or not an appropriate level of validation is performed are whether or not there is a budget for validation to meet or exceed budgeted expenditures.

3.4 Potential problems

In order to assess the relative importance of validation, in general, and some specific validation issues, in particular, a list of some of the more critical validation issues was developed. These validation issues, summarized in table 4, included completeness of the knowledge base, correctness of the knowledge in the knowledge base, and accuracy of the knowledge base in its interactions with the user. These issues are unique to the validation of expert systems since they all deal with the knowledge base of the system and include each of those delineated as knowledge base problems in O'Leary [13]. Inference engine issues were not included since many of the systems developed make use of existing expert system shells.

Another issue that is primarily a concern in the development of expert systems, sequencing of rules, was also included. This particular issue is a system construction issue to expert systems. It was included in order to assess the importance of such construction issues in relation to validation issues.

Finally, two generic information systems issues were also included. First, if a system is hard to use it can result in a lack of use of the system. Second, if the results generated by the system are difficult to interpret then that can also result in a lack of use of the system. These problems are often cited as major problems with information systems in general. Clearly, since these problems are critical issues. Their inclusion in the list of potential problems can provide insight into the general importance of validation issues.

4. TEST INSTRUMENT AND SAMPLE

Since there is little empirical evidence to substantiate many of the issues delineated in the previous two sections, a test instrument was developed and sent to a sample of developers of expert systems.

4.1 Test instrument

The test instrument gathered background information from the respondents, including "number of systems currently in use", "number of expert systems being developed" and industry classification".

The remainder of the questionnaire was divided into five different parts. Part 2, entitled "System Design and Development Approach," contained questions requesting a response to (-2, -1, 0, 1, 2), described as "Not True" (-2) to "True" (2).

Part 3 examined the validation budget asking the subject for the percentage of total development and the percentage of the implementation budget planned to go to validation and the percentage that actually goes to validation. Part 4 asked the user to rate eight different validation methods for the percentage of validation methods and for the percentage of validation effectiveness (e.g., testing with actual data). The percentages were required to sum to 100. Part 5 asked the subject to estimate the percentage of the total validation from whom knowledge was gathered. The percentages were required to sum to 100. Apparently, the section on validation is virtually complete, since two of the respondents reported informally that they used the instrument as a check list to assist in the validation process.

The final part asked the subject to assess "potential biggest problems". Subjects were asked to rank seven potential problems in terms of the impact on use of the system. A "1" was used to represent the biggest problems.

4.2 Sample

The test instrument was sent to 80 different subjects. The subjects were chosen because the author had accumulated either published information or private information indicating that the subjects had designed, developed and validated at least one expert system. Although this approach to choosing subjects for the test instrument was not random, it was viewed as necessary since the population of developers was small.

A single domain was chosen in order to normalize the impact of the domain. The domain, accounting and financial applications, was chosen because of the relatively large number of systems compared to other business domains, such as marketing.

The average respondent had 3.5 expert systems in use and 5.0 systems under development. If a subject responded and indicated that they did not develop expert systems they were removed from the sample. A single follow-up to the initial mailing was sent to those subjects that did not respond to the first mailing. The initial mailing was in January 1989 and the follow-up was in March 1989.

A total of 34 test instruments were received. This yielded a 42.5% response rate. Not all respondents answered each part of the questionnaire. The questions on the budget received the smallest response : 28 respondents answered the first two sections, while 31 respondents answered the remaining portion of the test instrument.

4.3 Question construction

The questions were constructed so as to encourage response. In addition, the questions were aimed at finding out what developers do or do not do.

As a result, the questions reflect the way a trial sample of developers viewed the world. For example, three of the questions use the terms "... it is difficult, infeasible or impractical..." Although this phrase uses three terms (difficult, infeasible and impractical) all are reasons that developers gave for doing something, e.g., using completion dates. Further, professionals were gauged as not likely to take the time to answer three questions differing only by a single word (first difficult, then infeasible, then impractical).

The questions were initially developed and discussed with two developers. Then a revised questionnaire was developed. The two developers who assisted in the design were not included in the sample.

5. FINDINGS

5.1 Design and development of expert systems

5.1.1 Analysis of data

The results are given in table 1. The sign (+ or -) of all the questions was as expected, based on the discussion in section 2.

A number of the questions had relatively large average scores for a range of (-2, 2). Questions 8 (formal deliverables) and 9 (prototyping takes less effort) had absolute mean values of about 1 or greater. Questions 1 (robustness), 5 (milestones), 6 (completion dates), and 7 (validation) had absolute mean values between .59 and .79.

A t-test was used to test the statistical significance of the individual mean scores. For each question, the null hypothesis was that the mean score was equal to zero against the alternative that the score was either less than or greater than zero, depending on whether the value was respectively less than or greater than zero. If a score were potentially to take on an average value of zero this would suggest that the statement was not either "true" or "false," as would be indicated by the respective sign on the average response. All but questions 2, 3, and 4 were significant at the .01 level.

5.1.2 Interpretation

Question 1 indicates that prototyped expert systems indeed are thought to be more robust. On the other hand question 9 indicates that prototyping does not take more effort than more formal methods. Further, it appears that when prototyping is used, various management devices (milestones and completion dates) are built into the process (questions 5 and 6), as recommended by Alavi [1]. As noted in discussions with firms, it appears that prototypes are used as a basis of generating requirements (question 8). Finally, prototyped systems were not judged to be more difficult to validate (question 7).

5.2 Validation of expert systems

5.2.1 Validation methods

The results of the inquiry on validation methods are contained in table 2. The rank correlation between percentage validation effort and percentage validation effectiveness was .967. Thus, there is a very strong relationship between use and effectiveness.

The greatest percentage change of the percentage validation effort (36 %)was with parallel use of the system by a non-expert. The change on direct examination by an expert was -23%. These changes indicate that some techniques appear to provide more insight into the validation process. In the aggregate, the results indicate that roughly 49% of the validation effort is with some kind of test data, while another 30.6% is with direct examination of the knowledge base and 18.6% is with parallel use of the system.

As a result, 49.2% of the validation effort is done using human participants. Although knowledge elicitation is often called a bottleneck because of the extent of human participation, it appears that validation faces a similar concern.

Further, over 98% of the validation is with rather basic methods – that is, not statistical or analytical. This suggests that there is substantial room for work in the development of techniques to assist in the validation of expert systems.

5.2.2Validation budget

The respondents reported an average validation budget of 22.46%, while the actual reported validation budget averaged 23.78%. There were 14 respondents that reported that the budget and actual percentage were the same, 5 reported that the budget was less than actual and 9 reported that the budget was more than actual.

Thus, there are at least two pieces of evidence that indicate that even though prototyping was used, the validation budget was not eliminated. First, there were 28 out of the 34 respondents that reported a validation budget. Second, in 23 out of the 28, actual was at least as large as the budget.

5.2.3 Who performs the validation process

The results on the "who" side of validation are contained in table 3. These results indicate that roughly 50% of the validation effort is accomplished by experts in this domain. Clearly, this is in contrast to virtually any other kind of computer-based system. In addition, this indicates that validation is a very expensive process that could benefit from using alternative techniques, as opposed to focusing all those expert resources on the task. Finally, these results also indicate relatively minor use of end users, sponsors and independent validation.

5.3 Potential biggest problems

5.3.1 Analysis of data

The mean observations are summarized in table 4. A statistical analysis of the pairwise relationships between some of the means of the distribution of responses for each of the problem pairs is contained in table 5. A pairwise approach was used because importance is only established relative to other problems on the list and one of the primary concerns in this research is the relative importance of validation issues.

A t-test was used to compare the mean values of the responses to the average response value for problems 4 and 5 (used to represent non-validation problems). This results are summarized in table 5. For each question (1, 2, 3) the hypothesis was that the average response for problems 4 and 5 was rated as not important, while the hypothesis for questions 6 and 7 was that problems 1, 2, 3 and 7 were found to be statistically significant. This indicates that the validation problems (1, 2, 3) are viewed as being more difficult than non-validation issues to which they were compared.

5.3.2 Interpretation

The results are summarized in table 4. The three most highly ranked problems were validation problems, each of which seem relatively unique to expert systems and other intelligent technologies. "Completeness of the knowledge base" was recognized as the most difficult problem in roughly one third of the responses and the first or second problem in almost one-half of the responses. It also had the lowest average score, and thus was recognized as the major problem. "Ensuring correctness of the knowledge base" had the second lowest average and the second highest number of being either first or second ranked. "System misses some possible opportunities" had the third lowest average score and tied for third place in total number of first and second rankings.

The next two lowest ranking occurred for "System is hard to use" and "Results from the system are difficult to interpret." These two alternatives are not unique to expert systems technology, but instead are problems of systems in general.

The two remaining problems in the list given to subjects are unique to expert systems : "System presents incorrect opportunities" and "sequencing rules." These problems were found to be ranked first or second in only five of the responses. The rankings also establish the importance of validation problems, since validation issues had three highest average rankings.

6. SUMMARY

This paper has presented the results of a survey of expert systems developers on designing, developing and validating expert systems. This study supplements previous theoretical analysis and case studies.

The prototyping development methodology is found to lead to more robust systems. However, prototyping apparently does not increase the difficulty of validating systems, probably because of the increased level of communications. Prototyping seems to take less effort than a software engineering approach. Further, it does appear that some developers have built various devices (completion dates and milestones) into the prototyping process. Finally, a prototyping approach can be used as the basis for defining the deliverables.

The survey provides empirical evidence for the use and effectiveness of particular validation methods. For example, it was found that roughly 50% of the validation effort employs a test data approach. In addition, the validation process appears very labour intensive, thus possibly becoming a bottleneck, as with knowledge acquisition.

The study also found a prevalent use of budgets for managing the validation process. Based on self-reporting in the study, those budgets are roughly adhered to.

Further, the study indicated who performs the validation process. Roughly 50% of the validation effort is performed by experts. This indicates that validation is an expensive process that could benefit from the development and use of validation tools.

In addition, the paper finds that validation problems are among the biggest problems facing developers. The issues of completeness, correctness and accuracy of the knowledge base are viewed as the most difficult.

Table 1
System Design and Development Approach*

No. Ques	Expected Sign	Mean	Std. Dev	Questions
1	+	0.76	1.059	We have found that prototyping leads to more robust expert systems than more formal software engineering approaches.&
2	+	0.32	1.489	We use prototyping development methodologies for expert systems because it is difficult, infeasible, or impractical to develop "specifications" for AI/ES projects.
3	+	- 0.17	1.294	We have found that prototyping an expert system is a more difficult approach to manage than software engineering approaches.
4	-	- 0.505	1.304	We have found that traditional software engineering methods just don't work in the development of AI/ES projects.
5	-	- 0.59	1.308	We use prototyping development methodologies for expert systems because it is difficult, infeasible or impractical to develop "milestones" for AI/ES projects. &
6	-	- 0.61	1.358	We use prototyping development methodologies for expert systems because it is difficult, infeasible, or impractical to develop completion dates for AI/ES projects. &
7	-	- 0.79	1.022	We have found that prototyping expert systems, rather than using traditional software engineering methods leads to systems that are more difficult to validate. &
8	-	- 0.97	0.923	We don't use prototyping because it doesn't allow for formal deliverables, which makes it difficult to know when the system is complete or when formal sign-off can be made. &
9	-	- 1.03	1.175	We have found that prototyping takes more effort than more formal methods to develop the same type of system. &

Not True				True
-2	-1	0	1	2

* Subjects were asked to "circle a number between -2 and 2.".
& Significantly different from 0 at .01 level.

Table 2
Validation Methods

Method	%Validation Effort	%Validation Effectiveness	Increase (Decrease)
Testing with actual data	31.1	33.5	2.4
Testing with contrived data	17.9	16.3	(1.6)
Direct examination of knowledge base by expert	17.6	17.1	(0.5)
Direct examination of knowledge base by knowledge Engineer	13.0	10.0	(3.0)
Parallel use of system by expert	11.6	12.1	0.5
Parallel use of system by non-expert	7.0	9.5	2.5
Decision tables/ contingency tables	1.4	1.1	(0.3)
Other	0.3	0.3	0.0
Total	99.9◊	99.9◊	

Table 3
"Who Generally Performs the Validation Process ?"

Percentage	Job Category
29.6	Same expert from whom knowledge was gathered
20.3	Knowledge engineer
20.1	Different expert than from whom the knowledge was gathered
12.4	End user
9.5	"Sponsor" of the project
7.5	Independent validator
.3	"Other"
99.7	Total (difference from 100 due to rounding error by some respondents)

◊ Differences due to rounding error by respondents.

Table 4
"Potential Biggest Problems" (subjects were asked to rank the problems from 1 to 7 with 1 used to denote the biggest problem).

Problem Number	Average Rank	Ranked First	Ranked Second	Problem
1	3.10	10	5	Determining cmpleteness knowledge Base
2	3.12	5	7	Difficult to ensure that the Knowledge in the system is correct
3	3.21	5	6	System does not present some possible opportunities to the user
4	3.89	8	3	System is hard to use
5	4.11	3	5	Results from the system are difficult to interpret
6	4.32	1	5	System presents some incorrect opportunities to the user
7	4.78	1	4	Difficult to sequence rules correctly

Table 5
Pairwise Level of Significance of Test of Similarity of Distributions for "Potential Biggest Problems"

Problem	Problem 4	Problem 5
1	0.025	0.01
2	0. 01	0.01
3	0.05	0.01
4	NA	NS
5	NS	NA
6	NS	NS
7	0.025	0.05

NS = Not significant and NA = Not applicable, otherwise significantly different at the specified.

REFERENCES

1. ALAVI M., (1984) : *An Assessment of the Prototyping Approach to Information Systems Development*, Communications of the ACM, Vol. 27, No. 6, June 1984, pp. 556-563.
2. BOEHM B., GRAY T., SEEWALDT T., (1984) : *Prototyping Versus Specifying : A Multiproject Experiment*, IEEE Transactions on Software Engineering, Vol. SE 10, No. 3, May 1984, pp. 290-303.
3. BULL M., DUDA R., PORT D. AND REITER J., (1987) : *Applying Software Engineering Principles to Knowledge-base Development*, Proceedings of the First Annual Conference on Expert Systems in Business, New York, November 10-12, 1987, pp. 27-38.
4. CHANDLER J., LIANG T., (1990) : *Developing Expert Systems for Business Applications*, Merrill Publishing, Columbus, Ohio.

5. GOLDBERG A., (1986) : *Knowledge-Based Programming : A Survey of Program Design and Construction Techniques*, IEEE Transactions on Software Engineering, Vol SE 12, No. 7, 1986, pp. 752-768.
6. HAYES-ROTH F., WATERMAN D., AND LENAT D., (1983) : *Building Expert Systems*, Addison-Wesley, Reading, Massachusetts.
7. JENKINS M., FELLER R., (1986) : *An Annotated Bibliography on Prototyping*, Institute for Research on the Management of Information Systems, Indiana University, June.
8. LETHAN H., JACOBSEN H., (1987) : *ESKORT -- An Expert System for Auditing VAT Accounts*, Proceedings of Expert Systems and Their Applications, Avignon, France.
9. NAUMANN J., JENKINS M., (1982) : *Prototyping : The New Paradigm for Systems Development*, MIS Quarterly, September 1982, pp. 29-44.
10. O'LEARY, D., (1990) : *Validation of Expert Systems*, Decision Sciences, Vol. 18, No. 3, 1987, pp. 468-486, in Chandler and Liang.
11 O'LEARY, D., (1990) : *Soliciting Weights or Probabilities From Experts for Rule-Based Systems*, International Journal of Man-Machine Studies, Volume 32, pp. 293-301.
12. O'LEARY, D., (1988-a) : *Expert Systems Prototyping as a Research Tool*, in Applied Expert Systems, E. Turban and P. Watkins, (eds) North-Holland.
13. O'LEARY, D., (1988-b) : *Methods of Validating Expert Systems*, Interfaces, November-December, 1988-b, pp. 72-79.
14. O'LEARY,D., (1987) : *Validation of Expert Systems*, Decision Sciences, Summer 1987, Vol. 18.
15. O'LEARY D., KANDELIN N., (1988) : *Validating the Weights in Rule-Based Expert Systems : a Statistical Approach*, International Journal of Expert Systems, Volume 1, N° 3, pp. 253-279.
16. PRESSMAN R., (1987) : Software Engineering, 2nd edition, McGraw-Hill, New York.