Verification and Validation of Multiple Agent Systems: Combining Agent Probabilistic Judgments

Daniel O'Leary
University of Southern California

Abstract

One of the principle issues in multiple agent systems is how to treat the judgments of the agents in those systems: should they be combined or treated separately? If the judgments are "substantially different" then that likely signals different models being employed by the agents. As a result, if the experts' judgments are disparate, then it is unlikely that the judgments should be combined.

However, developers of multiple agent systems have combined substantially different judgments by averaging. Such a combination is likely to provide a composite judgment that is inconsistent with each individual judgment. An important aspect of verification and validation of multiple agent systems is the analysis of the combination of such judgments. Thus, a critical issue in multiple agent systems is determining whether or not the judgments of the experts are similar or disparate. As a result, the purpose of this paper is to investigate the combination of probability judgments in multiple agent systems. Traditional statistics are used to investigate whether or not different judgments are substantially different. In addition, a new approach is developed to determine if probability distributions of agents are similar enough to combine or disparate enough to treat separately. A case study is used to illustrate the problems of combining multiple agent systems and to demonstrate the new approach.

1. Introduction

Combining judgments of multiple agents in probabilistic expert systems (ES) and influence diagrams (ID) is becoming an increasingly important issue in systems designed to capture the judgment of multiple agents. Unfortunately, the combination of multiple expert's judgments is not straightforward. Consider a system where two experts have probability judgments of 1 and 0 for the same event x and 0 and 1 for the same event $\sim$x. Such disparate judgments generally would signal that the experts have different models of the world. Alternatively, it may signal that there is an error in one of the assessments. In either case, combining these judgements, using approaches such as averaging, is likely to simply camouflage the disparate nature of the judgments. The resulting combination is likely to be representative of either agent. Unfortunately, developers of multiple agent systems have done just that.

Thus, the issue is under what conditions should multiple agents' judgments not be combined in an ES or ID? In particular, the purpose of this paper is to investigate the process of determining whether the judgments of multiple agents are similar enough so that they can be combined into a single, e.g., average judgment. In addition, a related issue is what metrics can be used to determine the extent of similarity between the two sets of judgments.

1.1 Importance

The combination of agents' judgments is an important issue in the development of multiple agent systems for a number of reasons. First, the combination of disparate judgments is likely to result in system behavior that is not sensible. For example, if there are two schools of thought as represented by multually exclusive probability distributions, what does it mean if the system presents a third average response.

Thus, from a development perspective it is important to find out when to combine multiple agents' assessments. Second, the combination of multiple disparate probability judgments is likely to result in difficulties when the system is verified and validated. Tests of the data at the extreme points (e.g., x and $\sim$x) will result in different responses from the human experts.

1.2 Expert Systems, Influence Diagrams and Multiple Agents

Research on multiple agent systems has been summarized in, e.g., Bond and Gasser [1988], Gasser and Hill [1990] and Gasser and Huhns [1989]. In many situations, multiple agent systems more closely model actual processes than single agent systems. In many real world situations, there are seldom single decision makers. Instead decision makers seek out the advice of others. In addition, multiple agent systems can be used to assist in many decision problems, such as monitoring different operations simultaneously, where the use of a single agent might be at substantial disadvantage.

Multiple agent systems can either integrate or choose between the judgments of multiple experts at basically two different times. The multiple agents' judgments are either aggregated at the time the system is built (e.g., Dungan [1983]) or at the time the system is run (e.g., Ng and Abramson [1991]). The first approach uses the assessments from multiple agents to establish a single system. The second approach provides more flexibility, allowing for evolving sets of agents.

Generally, the judgment of multiple agents may be aggregated in any of a number of ways. Although approaches such as negotiation between agents have been used, typically systems have combined the judgments of multiple agents by averaging the estimates (e.g., Dungan [1983] and Ng and Abramson [1991]).

1.3 Probability Distribution Judgments

This paper focuses on probability judgments of multiple agents. In particular, it is assumed that agents provide an estimate of a discrete probability distribution, for use in a multiple agent system. As is often the situation in influence diagrams, each expert would provide a discrete probability distribution across a number of categories. In the example in the introduction, experts provided probability estimates of x and $\sim$x of 1 and 0, and 1 and 0, respectively.

Traditional statistical analysis is used to investigate the problem. However, limitations in that approach within the context of multiple agent systems, suggest the search for an alternative. As a result, a new approach is developed to ascertain if probability judgments are similar enough to combine or disparate enough to signal the likelihood of different underlying models, and to provide a metric of that similarity. The basic problem and approach is illustrated in the context of a case study where disparate judgments were averaged.

1.4 Outline of This Paper

This paper proceeds as follows. Section 2 summarizes the case study from which the data used in this paper is generated. Section 3 provides a basic structure to analyze the problem of when to aggregate judgments. Section 4 investigates metrics for determining if the distribution estimates of two agents are similar enough to combine. Section 5 briefly summarizes the paper and analyzes some extensions.

2. Case Study: Pathfinder

Pathfinder1 is a multiple agent influence diagram,2 designed to support medical decision making. Pathfinder is discussed in detail, in a number of sources, including Heckerman et al. [1991]. Ng and Abramson [1991] list the

multiple agent probability distributions associated with a small portion of the system. The distributions for thirteen different symptom and disease "arcs" are summarized in Table 1.

TABLE 1
Complete Set of Probability Assessments

Arc #	Category					
	1	2	3	4	5	6
1.	.990	.010	.000	.000	.000	.000
	1.000	.000	.000	.000	.000	.000
2.	.990	.010	.000	.000	.000	.000
	1.000	.000	.000	.000	.000	.000
3.	.985	.015	.000	.000	.000	.000
	1.000	.000	.000	.000	.000	.000
4.	.985	.015	.000	.000	.000	.000
	1.000	.000	.000	.000	.000	.000
5.	.990	.010	.000	.000	.000	.000
	1.000	.000	.000	.000	.000	.000
6.	.990	.010	.000	.000	.000	.000
	1.000	.000	.000	.000	.000	.000
7.	.000	.010	.400	.500	.090	.000
	.000	.200	.600	.200	.000	.000
8.	.000	.000	.000	.000	.000	1.000
	.000	.000	.600	.200	.200	.000
9.	.980	.015	.005	.000	.000	.000
	.000	.200	.600	.200	.000	.000
10.	.900	.090	.010	.000	.000	.000
	1.000	.000	.000	.000	.000	.000
11.	.980	.015	.005	.000	.000	.000
	.900	.100	.000	.000	.000	.000
12.	.900	.090	.010	.000	.000	.000
	1.000	.000	.000	.000	.000	.000
13.	.000	.010	.400	.500	.090	.000
	.000	.800	.200	.000	.000	.000

@ For each "arc" the first line corresponds to expert #1 and the second line corresponds to expert #2.
Categories - Lacunar SR: 1 = Absent; 2 = Rare; 3 = Few; 4 = Many; 5 = Striking; 6 = Sheets.
Source: Ng and Abramson [1991]

An examination of the probability distributions in table 1 finds that in some cases the distributions are very similar, while in other cases they appear to be quite different. For example, the distributions for arc 1 appear to be about the same for both experts, while, the distributions for arc 9 appear substantially different for each of the agents. However, these are qualitative assessments, quantitative measures of the extent of similarity would be helpful in determining when the distributions of the agents are similar and "substantially different."

The developers faced the problem of constructing a single system that included information from two agents in the same system. The approach used in Pathfinder, discussed in Ng and Abramson [1991] was to form an average of the two different estimates for each of the arcs.

3. General Approach and Implications

The general approach used in this paper, for determining if individual judgments should be combined, is modeled using statistical reasoning (e.g., Freund [1971] and Edgington [1980]). First, the agent's probability assessments can be examined to determine if they are the same. If the judgments are identical then it does not matter which is used or if the judgments are averaged.

Second, if the judgments are not the same, they can be investigated to determine if they are "substantially different" or not. If the judgments are not substantially different, then it generally would be appropriate to average the particular judgments.

Third, alternatively, the different judgments may appear to be "substantially different." In that situation, it probably would not be reasonable to combine the distributions using averaging. Instead, either one or the

other would likely be used, or additional evidence would be gathered.

If the judgments are substantially different, then that might indicate an error. For example, O'Leary [1990] found that developers of expert systems had difficulty developing weights on rules in a manner consistent with probability theory. Virtually all systems reviewed in that paper had errors or anomalies in the probability estimates for the weights on the rules in an expert system. Thus, substantial differences may indicate errors in the judgments or recording of the judgments, etc. for at least one of the agents.

However, if the judgments appear "substantially different" and are not in error then that could indicate that the different judgments are representative of different models or assessments of the evidence and knowledge. In either case, the existence of substantial differences would suggest additional knowledge acquisition to more clearly or fully specify the underlying models or an analysis of the correctness of the probability distributions.

3.1 "Substantially Different"

Thus far, the term "substantially different" has been used to describe when the distributions are disparate enough so that they should not be combined. At the extremes we know that "identical" distributions are not "substantially different." Further, we know that if one agent indicates x has a probability of 1 and $\sim$x has a probability of 0, while the other agent gives x a probability of 0 and $\sim$x a probability of 1 then those two agents estimates are completely different, and thus "substantially different." In section 4, the paper draws on probability theory and a new approach to give more specific meaning to "similar" and "substantially different."

3.2 Combining Different Distributional Assessments

There are a number of different approaches that can be used to combine the judgments from multiple agents in the same model. The remainder of this section briefly reviews different techniques for integrating the judgments of multiple agents. However, the primary focus beyond this section is on determining if the expert assessments are substantially different.

First, each of the individual judgments on a set of events could be averaged. However, as noted in the introduction, where the two agents had probability of 1 and 0 on x, it may not be appropriate to integrate multiple agent's judgments from different distributions by averaging. The agents may have two different models or views. Averaging parameters from two disparate models can result in a model that has little meaning. For example, what does it mean to average the judgments of a conservative and a liberal?

Second, if agents' judgments on a rule or arc are substantially different, then it could indicate a need to choose between one (or more) of the models that the expert agents are using. There are a number of approaches that could be used. For example, when faced with a choice between a set of alternatives, consensus could be used to choose the majority model. As another example, the existence of a difference, might indicate the need to require negotiations among agents representing the different perspectives. Alternative agents or their representations might "argue" as to which set of judgments should be used.

Third, the existence of a difference may indicate that it is necessary to solicit additional information to better or further characterize the model. The differences may result because the model is ambiguous or underspecified. In that situation, additional knowledge acquisition or verification and validation is likely to be

appropriate to further clarify or specify the model. Ultimately, this could result in a different set of expert system rules or, in the case of influence diagrams, nodes or arcs or both.

Fourth, the existence of a difference in judgments may indicate a need to integrate situation-specific information into the system. If the system is underspecified then specific case information may be required to generate the necessary context. From a knowledge representation perspective, case-based reasoning might be used to adaptively choose the model that best meets the needs of the situation. Further, case-based reasoning might also be used to create a hybrid model that includes features from the other approaches.

4. Analysis of Agent Probability Distributions

The purpose of this section is to develop methods for determining whether or not two agents' probability distributions are substantially different. Two approaches are employed. First, a traditional statistical analysis, using correlation coefficients is employed. Second, a new approach, called cutpoints, is ddeveloped and discussed.

4.1 Statistical Analysis

Assume that for each of two agents, for each rule or arc, there is a probability distribution across a set of n points. We can use the correlation to measure the extent of similarity. The statistical significance of the correlation can be used to determine if the agents' distributions are "substantially different" or "similar."

In terms of the case, the correlation coefficients, between the two experts' distribution estimates are as follows: arcs 1-6 .999; arc 7, .686; arc 8, -.349; arc 9, -.345; arcs 10-12, .995; and arc 13, -.219. In the case of arcs 1-6, and 10-12, the arcs' correlations are highly statistically significant, at .03 and .01, respectively. Thus, we reject the hypothesis that the distributions are not correlated.

The correlation coefficient for arc 7 was not statistically significant. The correlation coefficients for arcs 8, 9 and 13 were negative and found not statistically significant. Thus, we cannot reject the hypothesis that there is not a correlation between the distributions, for arcs 7, 8, 9 and 13.

As a result, it would be reasonable to combine the distributions on arcs 1-6 and 10-12. However, the correlation coefficients for arcs 7, 8, 9 and 13 suggest that it would not be appropriate to combine the agents' probability distributions for those arcs.

Unfortunately, the analysis of the statistical significance of the correlation coefficient has some limitations in the context of multiple agent systems. First, in the generation of most multiple agent systems, the number of categories n, will be small. However, as n approaches 3 the measure of statistical significance approaches 0, since the factor (n-3) is used in the determination of the statistical significance (e.g., Freund [1971]). Second, this test of statistical significance of the correlation coefficient assumes a bivariate normal distribution. Unfortunately, that assumption is not always valid (e.g., Freund [1971]). Third, the correlation measures relatedness and not necessarily whether or not the two should agents' distributions should be combined. Thus, an alternative approach is discussed.

4.2 Cutpoints

This section presents a new approach for analysis of whether agent probability distributions, for a given rule, are substantially different. This approach, referred to as cutpoints, requires no distribution assumption as was used in the determination of the statistical significance of the correlation coefficient.

In terms of the discrete probability distributions on the individual arcs, such as those listed in table 1, each category will be referred to as an index number. Some of those indices have interesting properties that will help us determine if the distributions of the two experts are similar enough to, e.g., average.

Define a maximal cutpoint as an index (in the example ranging from 1 to 6) such that the difference in the cumulative probability, between the two distributions, at that index, is maximal. For example, in the case of arc 7, at category 3 the distribution for expert 1 has probability of .410, while that of expert 2 has probability of .800. The difference of .390 is larger than that of any other cutpoint, for n = 1, ..., 6. The complete set of maximal cutpoints, for the case, is given in Table 2.

TABLE 2
Maximal Cutpoints for the Sample of Probability Assessments

Arc #	Expert #1		Expert #2			
	x	x'	x	x'	Location	Amount
1.	.990	.010	1.000	.000	1	.010
2.	.990	.010	1.000	.000	1	.010
3.	.985	.015	1.000	.000	1	.015
4.	.985	.015	1.000	.000	1	.015
5.	.990	.010	1.000	.000	1	.010
6.	.990	.010	1.000	.000	1	.010
7.	.410	.590	.800	.200	3	.390
8.	.000	1.000	1.000	.000	5	1.000
9.	.980	.020	.000	1.000	1	.980
10.	.900	.100	1.000	.000	1	.100
11.	.980	.020	.900	.100	1	.080
12	.900	.100	1.000	.000	1	.100
13.	.010	.990	.800	.200	2	.790

Source: Ng and Abramson [1991]
"Location" refers to category at which maximal cutpoint occurs. "Amount" is the amount is the absolute value of (Pr(x for expert 1) - Pr(x for expert 2))

Define a zero cutpoint as an index where the cumulative probability for one distribution is zero and the cumulative probability for the other distribution is nonzero. There may be more than one zero cutpoint for a distribution. For example, in the case of arc 8, zero cutpoints occur at indices 3, 4, and 5.

Define a double zero cutpoint as an index, such that the cumulative probability for both distributions at a zero cutpoint is one. In that case, there is an index where all the probability for one expert is on one side of the index and all the probability for the other expert is on the other side of the index. For example, as shown for arc 8 there is a double zero cutpoint at the index 5. There also may be multiple double zero cutpoints.

4.3 Use of Cutpoints

These cutpoint concepts can be useful in the analysis of the similarity of two probability distributions on an arc. First, the occurrence of a double zero cutpoint is probably the most critical. Zero and double cutpoints define alternative ways to define the entire distribution, with two indices, say x and $\sim$x . That revised distribution, with a double zero cutpoint, has zero probability associated with x and $\sim$x for each of the two experts. This implies the two experts see certainty of mutually exclusive sets of events. Thus, rather than just defining level, there can be implications for structure: A zero probability between two events indicates no relationship between events.

Second, the maximal cutpoint provides insight into the similarity of the distributions of the two experts. The maximal cutpoint value provides a measure that allows us to assess the point of maximal difference between the experts. One approach would be to suggest that those distributions with a maximum cutpoint of .10 or lower (or .05 or .01, as in classic probability theory) would be viewed as similar, while those with a cutpoint larger than .10 would be viewed as disparate. This approach indicates that arcs 7, 8, 9 and 13 would be viewed as disparate at the .10 level. In this case the results are the same as the use of the correlation coefficient analysis.

Third, maximal cutpoints are useful in describing the index number behavior. In particular, the maximal cutpoints for a set of arcs provides a distribution of cutpoints. In the example, "1" is a maximal cutpoint ten times, "2," "3," and "5," (arcs 7, 8 and 13) are each cutpoints one time. As a result, we might assert that the comparison of the probabilities distributions for arcs 7, 8 and 13 behave differently than the comparison of the other arcs. This could suggest that the distributions of the two agents for those arcs are sufficiently different than the other distributions.

4.4 Summary

This section has presented a portfolio of methods for determining the existence of similarity or a significant difference between probability distribution estimates of multiple agents. In each case it was found that arcs 7, 8, 9 and 13 were disparate enough that they probably should not be combined.

The existence of such differences can be critical to the success of the system. Particularly for those systems designed to assist in making decisions with "life and death" consequences. Potentially camouflaging processes by averaging distributions may result in ignoring an important underlying process of critical importance.

5. Summary, Extensions and Contributions

This section briefly summarizes the paper, reviews some potential extensions and discusses some of the contributions of the paper.

5.1 Summary

This paper has investigated the issue of when probability assessments of multiple experts in the generation of ES and ID are similar or disparate. The correlation coefficient was used as a measure of similarity. In addition, a cutpoint approach was developed and demonstrated. It was found that the example system appears to combine disparate probability distributions.

Although the focus of this paper was on determining whether or not expert probability assessments are similar, the paper briefly discussed the alternatives that can be executed if the assessments appear to be substantially different. For example, an assessment may be in error; consensus may be used to determine which solution is appropriate; case-based reasoning might be used to determine a set of contingencies in which one or another model would be appropriate; or negotiations may be necessary to choose which set of assessments should be used.

5.2 Extensions

This paper can be extended in a number of ways. First, although the probability distributions were used in an influence diagram, this analysis is not limited to that context.

Second, in some cases we may be able to determine that the sets of expert probability distribution assessments come from specific distributions, e.g., poisson. In such a case, rather than using traditional statistical approaches to generate estimates of statistical significance, we could generate specific distribution-based, distributions of test statistics.

Third, nonparametric approaches could be used to evaluate the statistical significance. For example, computer intensive statistics (Noreen [1986]) could be used to generate a distribution of test statistics in order to assess the statistical significance of a particular correlation coefficient.

Footnotes

1. This paper does not criticize the systems referenced here. In fact, each of these systems is a path breaking system for a

number of reasons. Instead, this paper examines these systems in order to determine what kinds of problems can occur in the process of integrating multiple agent systems. In addition, these systems are used to provide data for the analysis used in the paper.

2. Influence diagrams capture knowledge in a graphic arrangement of nodes (events) and relationships between events (arcs). Arcs can include probability distribution information about the relationship between events.

References

Bond, A. and Gasser, L. Readings in Distributed Artificial Intelligence, Morgan Kaufmann, San Mateo, CA 1988.

Dungan, C., "A Model of Audit Judgment in the Form of an Expert System," Unpublished Ph. D. Dissertation, University of Illinois, 1983.

Dungan, C. and Chandler, J., "Auditor: A Microcomputer-based Expert System to Support Auditors in the Field," Expert Systems, October 1985, pp. 210-221.

Edwards, W., "Influence Diagrams, Bayesian Imperialism, and the Collins Case: An Appeal to Reason," Cardozo Law Review, Volume 13, Numbers 2-3, November, 1991, pp. 1025-1074.

Efron, B., "Bootstrap Methods: Another Look at the Jackknife," Annals of Statistics, Volume 7, 1979.

Edgington, E., Randomization Tests, Marcel Dekker, 1980.

Freund, J., Mathematical Statistics, Prentice-Hall, Englewood Cliffs, New Jersey, 1971.

Gasser, L. and Hill, R., "Coordinated Problem Solvers," Annual Review of Computer Science, volume 4, pp. 203-253, 1990.

Gasser, L. and Huhns, M., Distributed Artificial Intelligence, Volume II, Morgan Kaufmann, San Mateo, Ca, 1989.

Heckerman, D., Horvitz, E., and Nathwani, B., "Toward Normative Expert Systems: Part I. The Pathfinder Project. Methods of Information in Medicine, 1991.

Lilliefors, H., "On the Kolmogorov-Smirnof Test for Normality with Mean and Variance Unknown," Journal of the American Statistical Association, Volume 62, 1967, pp. 399-402.

Ng, K. and Abramson, B., "Probabilistic Multi-Knowledge-Base Systems," Unpublished working paper, 1991.

Noreen, E. An Introduction to Testing Hypotheses Using Computer Intensive Statistics, Unpublished paper, 1986, also published by John Wiley, 1990.

O'Leary, D., "Soliciting Weights or Probabilities from Experts for Rule-based Expert Systems," International Journal of Man-Machine Studies, 1990.

Theil, H., Statistical Decomposition Analysis, Amsterdam, North Holland, 1972.

Velleman, P. and D. Hoaglin, Applications, Basics and Computing of Exploratory Data Analysis, Belmont, CA, Duxbery, Press, 1981.

D. Hoaglin, Applications, Basics and Computing

Designing Testable, Heterogeneous Software Environments

Christopher Landauer, Kirstie Bellman

Key Phrases:
knowledge-based software engineering,
integrated software environments,
system architecture for testing

Over the last 8 years, we have developed techniques for designing, testing, and evaluating several new computer technologies, including knowledge-based systems (KBSs). However, even as we speak, the technologies that we need to contend with are changing; rarely do these new technologies come alone. Instead, we are in an era where the problems we are working on demand large software environments with toolsets and libraries composed of often very different types of components. We see fuzzy controllers combined with knowledge-bases and neural nets and all of these combined with standard graphic programs, user interfaces, computer algorithms, spreadsheet programs, editors, database management systems etc.

In this paper we introduce a methodology for constructing large heterogeneous software environments in such a way as to make them ''testable'' and maintainable. The paper is divided into two parts: first, we introduce our approach to engineering software environments, and then our approach to verification and validation of KBSs. Then we show how the V\&V methods can be applied directly to the KBSs that hold the ''wrappings'', and use them to analyze a simple example.

The ''wrapping'' methodology builds flexible environments by encapsulating both programs and data (ALL computational resources in a system) with the explicit knowledge of what type of resource they are, what they do, and with knowledge of their various styles of use. These wrappings provide standard interfaces to software resources, and provide knowledge about the resource, so that other tools in the environment can interact appropriately with the wrapped resource, either to provide it with information or to use its information effectively. These descriptions include not only the usual protocol and input requirements for applying a software resource (what we call ''assembly''), but also metaknowledge about the appropriate context for applying a resource and for adapting a resource to different problems (which is what we call ''integration''). The conceptual architecture we have developed has two main components: the wrappings, which are knowledge-based interfaces supporting the use of resources, and the Study Manager, which is a program that processes the wrappings to coordinate the problem study activities.

A major principle of our approach is to wrap everything (Everything!): All tools, data and other software resources are wrapped e.g., data files, user screens and other user interface elements, plans and scripts, databases, computational analysis tools, simulation programs, models and other external programs and data. Even the programs that interpret the wrappings of other software are wrapped so we can study the wrapping processors with the same system. Any part of a complex software environment is considered to be a software resource, and everything gets explicit descriptions.

The wrapping approach is essentially a knowledge-based approach to dealing with the problems of engineering large, heterogeneous software and modelling environments. We use explicit knowledge about the software resources in order to select, integrate, and adapt them to the needs of the user. At the moment, this knowledge is gathered a priori from whatever knowledge sources exist: the resource developers, experts (on that model, on that domain, on that numerical algorithm and so forth), documentation, texts, etc.. Eventually we can envision systems which are actually smart enough to be able to generate some of the knowledge we use in the wrappings about a resource. For example, even now we could construct programs that would analyze the equations in a given software program for non-linearity and simultaneity and add that information

to the wrappings for a software resource. However, regardless of how the knowledge is obtained, we are left in this approach with a number of rules and opinions for the use and adaptation of a software resource. The question then arises, ''How do we check the correctness of this metaknowledge?'' and even more so, ''How do we check the consistency of this knowledge with the knowledge in the other wrappings being processed in this system?'' In this part of the paper, we briefly review the verification and validation methods we have developed for testing KBSs and discuss how these methods can be directly applied to the databases that hold the ''wrappings''.

We summarize the few aspects of the wrapping approach that are used in the V\&V discussion. Problems are ''posed'', by the system or the user, and the Study Manager organizes the resources to solve the problem. The current state of knowledge is maintained as a list of context values, maintained in this implementation as pairs, with context component parameters and values. Many problems are information request problems, asking for the value for a particular context component. The context and requirement conditions are expressed in terms of value conditions on these context component names, and the products are described as context component names whose values are set by a resource.

We performed an example analysis on one of our small application projects, by interpreting the Wrapping KB (WKB) entries as rules. The application consists of a small number of programs, set up by user menu selections, and coordinated by means of a few sample scripts. We start by describing the basic entities in the application, then describe the WKB formats for this application and show some example entries. There are a few problems that can be posed, including ''network_study'', and auxiliary problems like ''need_execution_style'' that help collect information required to solve a problem. The resources include application programs (a few large analysis programs written in Fortran, a simulation package written in C, and a user interface and intelligent editor to one of the analysis programs, written in C++) and data files (typically input files for the application programs). These programs are often used together, and for the small number of scenarios we prepared for this application, the coordination was done with explicit scripts. Finally, the very simple user interface was a very limited menu interpreter. Each of the other programs has its own user interface, and all of them were made available.

We decided to get a quick assessment of the dynamic properties of this KB by using the CLIPS inference engine. CLIPS has advantages in wide availability, low cost, and reliability over other expert system shells, and is the only expert system shell we know with a validated inference engine. It thus affords an easily used tool for testing certain properties of rulebases. The differences between CLIPS and other inference engines is trivial compared to the advantage of having a focus for the analysis of a rulebase. The translation (done by hand) into CLIPS resulted in 75 rules.

The main point of doing the example V\&V analysis of the wrappings was to show that we can usefully find anomalies in the KBs, even at a very early stage of development. Despite the fact that we used a different inference engine, a different language, and an incomplete translation scheme, the exercise caught many errors in the KB. Each of the translation or analysis steps found different errors, and even different kinds of errors. It is our experience that any kind of formal analysis at all can find errors, and the more different kinds of analysis are tried, the better able they are to detect errors.

The correlation computations took about 4 seconds to run on a SUN SparcStation 2, and about 45 seconds on a SUN 3/60 (with 10 seconds or so to compile using gcc), so the time required for these calculations is trivial. Writing the program to compute the correlations, and editing the files used by that program, was a matter of a couple of hours in an afternoon, so the time required to prepare for this analysis was also quite small.

Although we only emphasize in this paper the structural analyses that may be applied to the wrappings, we believe that the wrapping approach will advance software engineering in several ways: (1) it provides explicit descriptions (and documentation) about each software resource, including what is in essence both a specification for that resource and practical advice on its acceptable and appropriate use; (2) it can provide traceability during dynamic testing, and an easy way to insert probes; (3) it allows standard structural testing of the wrappings, when these are stored together as a database; (4) it allows the possibility of incorporating on-line software checkers. The hope is that eventually we will have computer systems in which the means to test and evaluate the system are not peripheral, but rather an integral part of the software system.

Some References

Kirstie L. Bellman,
''The Modelling Issues Inherent in Testing and Evaluating
Knowledge-based Systems'',
Expert Systems With Applications J., Volume 1, pp. 199-215 (1990)

Kirstie L. Bellman,
''An Approach to Integrating and Creating Flexible Software
Environments Supporting the Design of Complex Systems'',
pp. 1101-1105 in
Proceedings of WSC '91: The 1991 Winter Simulation Conference,
8-11 December 1991, Phoenix, Arizona (1991)

Christopher Landauer,
''Correctness Principles for Rule-Based Expert Systems'',
Expert Systems With Applications J., Volume 1, pp. 291-316 (1990)

Christopher Landauer,
''Integrated Software Environments'',
pp. 164-170 in
Proceedings of CAIA '92: The 8th Conference on AI Applications,
2-6 March 1992, Monterey, California (1992)

Christopher Landauer, Kirstie Bellman
''Integrated Simulation Environments'' (invited talk),
Proceedings of DARPA Variable Resolution Conference,
May 1992, Herndon, Virginia (1992)