An Achievement Test for Knowledge-Based Systems: QUEM

Caroline Clarke Hayes
University of Illinois
Department of Computer Science
405 North Mathews Avenue
Urbana, Illinois 61801 U.S.A.

Michael I. Parzen
University of Chicago
Graduate School of Business
1101 East 58th Street,
Chicago, Illinois, 60637 U.S.A.

Abstract

This paper describes QUEM, a method for assessing the skill level of a knowledge-based system based on the quality of the solutions it produces. QUEM is demonstrated by using it to assess the performance of a particular knowledge-based system, P^3. QUEM can be viewed as an achievement or job placement test given to knowledge-based systems to help system designers determine how the system should be used, and in what capacity by what level of users. In general, it is difficult to find useful metrics for assessing a system's overall performance. Most literature on evaluation deals with validation, verification and testing in which the primary concern is the correctness and consistency in the databases and rule-bases. However, these properties alone may not be sufficient to determine how well a system performs its task. QUEM allows software developers to assess their system's performance by constructing a skill function based on human performance data that relates experience and solution quality. QUEM can be used to gauge the experience level of an individual system, compare two systems, or compare a system to its intended users. This represents an important advance in quantitative measures of over-all system performance that can be applied to a broad range of systems.

1 Introduction

When evaluating knowledge-based systems (KBSs) it is often difficult to find useful metrics for assessing a system's overall performance. Most literature on KBS evaluation deals with validation, verification and testing [Nazareth and Kennedy, 1993] in which the primary concern is with the correctness and consistency in the databases and rule-bases. Other systems address modifiability, ease of use and cost of the system. However, these properties alone may not be sufficient to determine how *well* a system performs its task. For example, a complete and consistent KBS may not necessarily create high quality solutions. It would be useful to have a method to estimate how well a KBS's performs its task on some absolute scale that would allow comparisons between systems to be made. However, it is not immediately obvious what type of scale should be used. In this paper we present QUEM (Quality and Experience Metric, pronounced "kwem"), a method for evaluating the *experience level* of a knowledge-based system and the *quality* of its solutions. In other words, QUEM estimates the performance of a system in terms of the years of experience a human would require to generate solutions of the same quality; experience level is the scale on which quality is assessed. QUEM can be considered to be an *achievement test* for KBSs since it estimates the level that a KBS has achieved. We use expert judges to assess the quality of solutions generated by human experts and KBS's. We then construct a "skill function" for the human data relating experience and solution quality. This skill function is used to estimate KBS experience level.

QUEM provides a *quantitative* way to estimate the experience level of a KBS, compare two KBS's, or compare the experience level of a KBS to that of its users. This type comparison is of particular importance if a KBS is to be used as an aid to human users. Understanding the skill level of the KBS relative to its users is important in determining how the system should be used and in predicting whether users will accept it. For example, it is often necessary that the skill level of the KBS equal or exceed that of its users. If the KBS produces solutions of lower sophistication and quality than the user can produce on his or her own, the user may consider the system to be a hindrance. Additionally, estimation of a KBS's experience level is one measure of how well developers have succeeded in capturing the domain expertise.

The development of QUEM arose from our desire to measure the quality of various knowledge-based systems which were under development. We were particularly interested in having a way to assess the quality of the problem solving abilities of prototype systems because it would be of great assistance in making development decisions. Before continuing a large programming effort to improve a given system, we wanted to have some assurance that the approach used in the prototype was a

reasonable one. If evaluation were to show that the best solutions produced by a given system were of low quality, then we would conclude that the approach was not a reasonable one and our efforts should be focused on re-structuring the approach and problem solving architecture. On the other hand, if the solutions produced proved to be of high quality then we could feel more confident that the approach was reasonable and that it would be worthwhile to put efforts into further development of the current system architecture.

In this paper we will outline the QUEM method developed for assessing average solution quality in terms of experience level, and demonstrate the use of QUEM to estimate the solution quality and experience level of a KBS for creating manufacturing plans. We used these results to assess the competence of our KBS problem solving approach, and to determine if we should continue on the same approach in future developments.

1.1 Challenges

Our first challenge was to identify a method for assessing solution quality in complex domains. Quality is the perceived utility that an artifact has to some set of people in a given context. A utility function is often used to provide a precise number to estimate quality, just as a watch provides a precise number to estimate the true time. Solution quality is in general hard to measure because it can be hard to quantify. This is particularly true in domains that are very rich such as architectural design, military battle planning, and manufacturing. There are usually many, sometimes conflicting factors that determine solution quality such as cost and accuracy, and esthetics. Even if one can generate a utility function to describe quality in a given domain, it may be hard to quantify the component factors that determine quality.

In our initial attempts to estimate solution quality for the manufacturing domain, we tried to construct a quality utility function composed of factors which our experts believed to be important. In the manufacturing domain, important factors included plan cost, feasibility and reliability. We attempted to generate a quality function for this domain based on these factors. However, we soon found this approach to be inadequate; it was not feasible to construct an accurate mathematical quality function because some important component factors, such as reliability, were very difficult to quantify.

To describe the example further, *plan reliability* is the likelihood that the operations within the plan will fail or will produce marginal results. Plans can fail in catastrophic ways resulting in physical damage to agents or equipment executing the plans, or in subtler ways, such as when the resulting product does not meet requirements. Predicting reliability requires knowledge of a wide variety of situations which are hard to capture without a large body of empirical data. Because of these difficult to quantify component factors, the task of constructing a reasonably accurate mathematical quality metric for KBS solutions is very difficult in practice, for many (if not most) rich and complex domains.

However, after some initial disappointment on finding that a good mathematical quality function was not feasible in our domain, it occurred to us that we did not actually need a quality function because we had a number of fairly robust quality measuring devices readily available to us: human experts. Human experts can succeed in assessing quality where a quality function may fail because experts are able to estimate hard-to-quantify quality factors, such as reliability, based on their broad empirical experience. Additionally, we found that although there was some variation in how judges assessed quality, their quality assessments were usually fairly close. Experts with similar experience tend to make similar quality assessments of a given solution. Furthermore, those assessments correlate very strongly with the experience of the problem solver. Thus, although there is a perception that humans are unreliable, we found that human experts were fairly consistent in assessments, and that their variability can be measured (for example, by having several experts independently rate the same solution) and taken into account. We decided look for a way to use the solution quality assessments produced by human experts to assess KBS solution quality.

Our second challenge was to devise a scoring system in which human judges could report their quality assessments. The scoring system must allow the quality assessments of different judges to be compared. Initially, we considered having the judges assign quality scores between 1 and 10, like Olympic sports judges, indicating the absolute quality of each plan. However, we decided against such an approach because experts do not have a standard or agreed upon method for assigning quality measures to plans. We were concerned that it might be difficult to compare scores assigned by two judges; if 10 is the best quality score, an enthusiastic judge might give many 10's while a conservative judge may rarely give a score better than 6. However, the first judges's 10 may mean the same thing as the second judges's 6. We decided that it would be more appropriate to have the judges rank order solutions from best to worst, rather than to assign them scores.

2 Related Work

As mentioned earlier, most literature on knowledge-based system (KBS) evaluation deals with validation, verification and testing (VVT) [Nazareth and Kennedy, 1993] in which the primary concern is with correctness, circularity, inconsistency, redundancy, modifiability, ease of use and cost [Lane, 1986], [Liebowitz, 1986]. However, these properties alone may not be enough to describe a system's competence in solving problems effectively. [Clancey, 1993] describes four perspectives useful for evaluating a system's competence: performance, articulation, accuracy and completeness. Other parameters important to system competence are: solution feasibility, solution quality, problem solving range, computer effort, and user effort.

Most competence evaluations provide *relative* mea-

sures of system performance. These evaluations provide the information that system x works better than system y, or human z. For example, when Aikins [Aikins, 1981] evaluated her system, Puff, a medical diagnostic system for cardio-pulmonary diseases, she compared the performance of her system against the diagnostic performance of three human doctors. She found that Puff's diagnosis agreed with the average diagnosis more often than did any of the individual doctors. From this she concluded that not only could Puff perform competently, but it was also more accurate on average than any of the individual experts in the study. Dixon, et al. evaluated their system, Dominic [Dixon *et al.*, 1987], by comparing its results against those of two other KBS's and a human expert. From this comparison they concluded that "Dominic is a reasonably capable designer ... although the two domain specific programs produced slightly superior performance."

However, simply knowing that one KBS produces better quality solutions than another does not necessarily tell the KBS developers if either produces particularly good solutions. For this reason we also felt it was necessary to develop a *quantitative* measure of KBS experience level which would allow one to make statements such as, "My KBS is estimated to have captured n years of experience." Such measures can better aid system developers in assessing whether their KBS is sophisticated enough for their purposes.

3 General Method

The QUEM procedure requires one or more *knowledge-based systems* for comparison, a set of *problems*, several *subjects* of various experience levels, and two or more *expert judges*. The expert subjects should have differing levels of experience. The expert judges should have experience equal to or greater than that of all subjects. The judges should not double as subjects in order for this test to produce meaningful results. Additionally, the domain of experience for the KBS, judges and subjects, must all be very similar, otherwise they may judge quality by very different criteria. The QUEM procedure for rating KBS experience level is:

1. **Solve**. Have all subjects and all KBSs each solve all problems in the problem set.
2. **Sort**. For each problem, put all solutions together in a group. If there are 3 problems, there will be 3 solution groups.
3. **Rank**. Have the expert judges independently rank order all of the solutions in each group from worst quality to best quality. Label the worst solution in each group as number 1. Successively number each solution, assigning the highest number to the best solution.
4. **Adjust Ranks**. If a judge ranks several solutions as having equal quality, the ranks must be normalized so that they can be compared to other rankings. For example, suppose Judge 1 is given 6 solutions which he ranks 1 through 6, while Judge 2 is given the same six solutions but she ranks two solutions as worst, three as intermediate, and one as best, producing the ranks of 1, 1, 2, 2, 2, and 3. The rankings of Judge 2 must be adjusted if they are to be compared to Judge 1's rankings. To adjust the rankings, they must be divided in to tied groups. Judge 2's rankings would be divided into three groups: (1, 1) (2, 2, 2) (3). All data points must be renumbered starting from the lowest number, such that each has a separate consecutive rank: (1, 2) (3, 4, 5) (6). Next, the average rank of each group is computed, and each member of a group is assigned the value of its group average. Thus, Judge 2's adjusted rankings would be: 1.5, 1.5, 4, 4, 4, and 6.
5. **Compute subject averages**. Compute the average quality ranking for each subject and KBS across all problems using the adjusted rankings.
6. **Plot subject averages**. Put the KBS data aside for a moment. Plot each human subject's experience on the y axis and his or her average quality ranking on the x axis.
7. **Fit a skill function to the data**. Fit a line or curve to these data using linear regression or other method appropriate to the data. Call this the *skill function*.
8. **Construct confidence bands** to indicate the amount of variation one can expect in individual performances at any given experience level. A point estimate of experience is not useful without some idea how accurate the estimate is. To compute these bands let x_m denote the average quality rank of a KBS. Using the linear regression model described above, our experience estimate of the KBS is $y_m = b_0 + b_1 x_m$. A 95% confidence interval for this estimate is given by

$$y_m \pm t_{(n-2,.025)}\sqrt{(\frac{1}{n} + \frac{(x_m - \overline{x})^2}{\sum_{i=1}^{n} x_i^2 - n\overline{x}^2})s_e^2}$$

where $t_{(n-2,.025)}$ is the 95%confidence coefficient based on the t−distribution and s_e^2 is an estimate of the amount of variability in the relationship between experience level and average quality rank. All these quantities are standard output results from statistics packages.

9. **Construct an experience estimate and interval.** For each KBS in the study,
   (a) Plug the KBS's average quality ranking (x) into the skill function to obtain the *experience estimate* for the KBS.
   (b) Plug the KBS's average quality ranking (x) into the equation for the upper confidence band. Repeat for the lower confidence band. The two numbers produced represent the *experience interval* for the KBS.

The results of this process are:

- A *skill function* for humans, relating length of experience to solution quality.
- *Confidence bands* around the skill function showing the expected range of skill for practitioners having a given length of experience.
- An *experience estimate* for the KBS. This value indicates the most likely value for the experience-level of the KBS.
- An *experience interval* showing the range of human experience levels expected to achieve the same average quality as the KBS, with 95% confidence.

3.1 Applications of QUEM

QUEM can be used in a variety of ways:

1. **Estimate the experience level of a single KBS.** When applied in this way, solutions created by a single KBS are ranked along with solutions created by a range of humans.
2. **Identify a change in experience level** between an old and a new version of a KBS. Solutions of two or more versions of the same KBS are ranked along with solutions created by humans.
3. **Compare two or more KBSs in the same domain** using the same method described in (2).
4. **Compare two unrelated KBSs** that operate in different domains. In order to compare two unrelated KBS's, two separate QUEM tests must be performed, and the resulting experience levels compared. A separate group of judges and subjects with appropriate domain knowledge must be selected for each test.
5. **Estimate the amount by which a computer assistant raises the skill level of a user.** Run two problem solving sets of the same difficulty on the same user: one set without the aid of the KBS and one with the KBS. A separate problem set must be used for each trial to avoid learning effects. Use the same analysis method as (2) - treat the user's two trials as one would treat two different versions of the same KBS.

3.2 Selecting judges, subjects, problems

In order to perform a test, the experimenter will need to take some care in selecting judges and a range of subjects. Selection of problems turned out to be a less difficult issue. We found that in the domains which we studied (manufacturing and software development) even very simple problems were of sufficient complexity to show strong differences between practitioners ranging between 0 and 10 years of experience. This is probably true for most complex domains, although it may not be true for very simple or toy domains.

Subject and judge selection. The judges should preferably have 10 or more years of experience. (MacMillan [MacMillan *et al.*, 1993] refers to such experts as "super experts.") However, given the rarity of highly experienced experts, one may have to settle for what one can get. The subjects' and the judges' experience area should closely match the domain of the KBS being evaluated.

Range of subjects. Ideally, one would like to select subjects so that the experience level of the KBS falls within the subjects' experience range. The method will still work even if the KBS falls slightly outside the range of the subjects' experience, however if it falls too far outside their range then the experience interval may become too broad to supply a useful experience estimate. For example, if the experience level of the KBS is 5 years, one may want to select subjects ranging from 2 to 10 years of experience. Unfortunately, before applying QUEM, one does not know the experience level of the KBS, so one must make an initial educated guess as to what the range of experience levels should be for the subjects. It may be necessary to conduct 1 or more pilot studies to identify the appropriate experience range for the subjects. The first time we tested the manufacturing KBS (described in the later example), we did not guess the subject range correctly. We selected subjects having between 2 and 5 years of experience, but found that the KBS's experience level was above the range of these subjects. This provided useful information, but it did not allow us to put an upper bound on the system's experience level. After two additional years of development on the system, we conducted a second test in which we selected subjects between 2 and 24 years of experience. This time we found that our KBS's experience level did fall inside the range of the subjects (approximately at the 8 year mark). These two previous studies enabled us to select the correct range of subjects (2 - 10 years) for the study in this paper.

Graceful degradation. There are many sources of variation in the data. Variations may arise from differences in the way judges make assessments, motivation levels of the subjects, and other factors. The total variation in the data from all such sources is reflected in the width of the confidence bands and experience interval. This representation of variability makes QUEM robust to noise to an extent, and provides QUEM with the property of graceful degradation. Thus, if the experimenter accidentally introduces additional variation by poor selection of one judge or subject, it may broaden the experience interval (reducing the precision of the answer) but it will not greatly change the result.

3.3 Limitations of the method

QUEM can provide useful information for a domain only when practitioners in the field show a distinct improvement in solution quality over time. An example of a domain where this relationship is known to exist is management planning of software development projects[Fiebig, 1997]. However, experience may not bring quality improvements in all domains. The existence of such a relationship can easily be determined by testing if a simple function can be found which fits the data well. The converse, that no relation exists, is harder to determine. If

Problem Solver	Years of Experience	Judge 1			Judge 2			Average Solution Rank
		P1	P2	P3	P1	P2	P3	
Subject 1	2	2	2	8	1	1	1	2.50
Subject 2	2	1	1	5	2	5	5	3.17
Subject 3	5	3	-	4	7	-	2	4.00
Subject 4	5	5	3	7	4	4	4	4.50
Subject 5	7	4	5	6	3	3	3	4.50
Subject 6	8	8	8	1	8	8	7	6.67
Subject 7	10	-	7	9	-	6	-	7.33
KBS	*	7	4	2	6	7	8	5.67

Figure 1: Quality rankings assigned to solutions

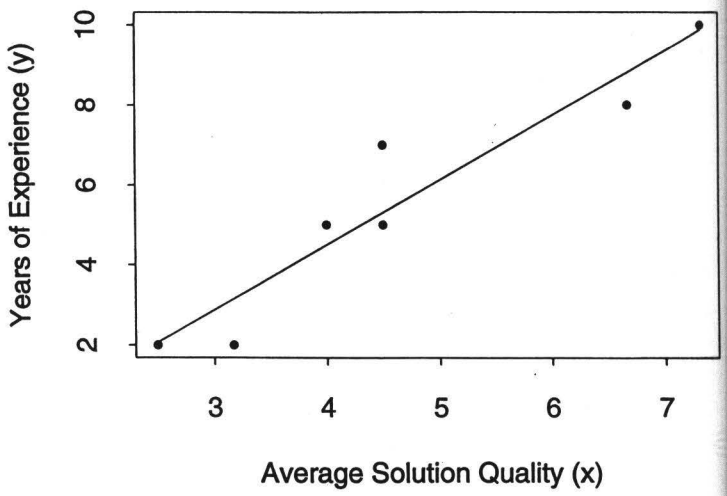

Figure 2: Average quality rankings and skill function.

no clear relationship is found in the data, it does not necessarily mean that one does not exist. It could also mean that the subjects or judges were not chosen well, the range of experience levels was too narrow, or increased skill manifests itself in ways other than through increased solution quality (such as increased speed in producing a solution).

4 Example: Evaluation of a KBS

QUEM was used to evaluate a manufacturing KBS, MACHINIST (later called P^3 [Hayes, 1996]). Experts in this domain are highly skilled and require as much as 8 to 10 years of intensive practice to achieve master level status. Seven subjects and two judges were selected. The subjects ranged between 2 and 10 years of experience. They had 2, 2, 5, 5, 7, 8, and 10 years of experience respectively. The two expert judges had 15 and 18 years of experience respectively. We prepared 3 problems for the subjects to solve, P1, P2 and P3, all of approximately the same difficulty level.

4.1 Evaluating a KBS with QUEM

1. **Solve:** We had each of the subjects and the KBS solve all three problems. We wrote up all solutions in a uniform format and handwriting (to disguise their source).
2. **Sort:** We sorted the solutions into 3 groups: each group contained all solutions to a specific problem.
3. **Rank:** We had two expert judges independently rank the plans in each group, from worst to best. The worst plan was given a score of 1. The ranks assigned to each plan are shown in Figure 1. P1, P2 and P3 are problems 1, 2 and 3. The missing data points resulted when subjects were unable to complete all three problems when they called away due to immediate job demands.
4. **Adjust Ranks**. This step was not necessary for these data because each plan had a unique rank.
5. **Compute Subject Averages**. The average quality ranking received by each subject across all three problems was computed. These values are shown in the last column of Figure 1. The lowest average score, 2.50, was received by the subject 1 who had only two years of experience. The highest average score of 7.33 was received by subject 7 who had 10 years of experience. The KBS received a quality ranking of 5.67. A factorial analysis performed on the data showed experience to be statistically significant, but not judge nor part (which confirmed our expectations).
6. **Plot**. The average quality rankings or the human subjects were plotted on the graph shown in Figure 2.
7. **Fit a Skill Function to the Data**. Several types of curves were fit to these data, including a logarithmic function and several types of polynomials. However, a simple linear regression fit the data best. The regression yielded the following equation for the model: $y = -1.97 + 1.67x$. We use this as the *skill function,* shown in Figure 2 as a heavy diagonal line.
8. **Construct confidence bands**. 95% confidence bands are shown in Figure 3 as curved bands flanking the skill function.
9. **Plot the KBS average quality rank**. The average quality rank for the KBS, x(m), was plotted on the quality (x) axis.
10. **Construct an experience estimate and interval.** The average plan quality rating of the KBS, 5.67, is plugged into the skill function. This produces a value which estimates the KBS's experience level at 7.20 years of experience. This is the *experience estimate.* Using the equations for the confidence bands, it was determined that the *experience interval* is 6.03 to 8.36 years of experience. This means that the true experience level of the KBS lies

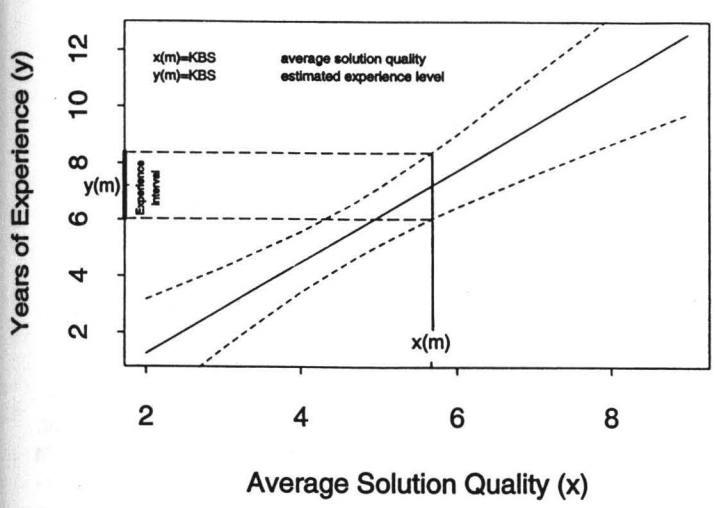

Figure 3: The KBS experience estimate and interval.

somewhere between 6.03 and 8.36 years, with 95% confidence.

Figure 3[1] shows the experience estimate and experience interval for the KBS on the y axis. From this evaluation we concluded that our system exhibits a very high experience level. On the basis of these results we confirmed that our problem solving approach was a reasonable and effective one. We decided that our basic approach was sound and that we should proceed with development along the same lines. Information on how to change the system to improve it further was derived from further knowledge engineering and protocol analysis.

5 Summary and Conclusions

In this paper we present QUEM, a general method for measuring the experience level of a KBS, and assessing the quality of its solutions relative to human practitioner. This method allows researchers to answer the question, "How expert is my expert system?" Assessing the solution quality and experience level of a KBS system is important in helping developers decide if their approach is sufficient, how the system should be used, and with what level of user it should interact.

The method utilizes expert judges to rank order solutions produced by both KBSs and human subjects. The rankings for the human subjects are used to construct a skill function describing the relationship between experience and solution quality. Lastly this skill function, and its confidence bands can be used to estimate the KBSs' experience level, and to bound the true value.

Previous methods for evaluating a KBS performance involve *qualitative* comparisons, such as "system x performs *better than* system y," which does not say if either system performs well at all. The QUEM procedure allows a system developer to make a *quantitative* assessment of the solution quality and experience level of a KBS. This measure allows system developers to answer the questions such as, "*How much* better is system x than system y?" or "*How many* years of experience does my KBS capture?"

QUEM can be used in any domain in which increased experience leads to measurably increased solution quality (which is presumably most complex domains). Additionally, it can be used to assess a partially complete KBS which can construct solutions but which may not be complete or correct in all aspects. It can be used to measure the experience level of an individual KBS, to compare several KBSs which operate in the same domain, or to compare the experience levels of several KBS's that operate in unrelated domains. QUEM represents an important advance in providing quantitative measures of system performance that can be applied to a broad range of complex domains in which solution quality may otherwise be hard to quantify.

[1] Figure 3 courtesy of M. C. P. Dorneich.

References

[Aikins, 1981] J. S. Aikins. Representation of control knowledge in expert systems. In *Proceedings of the First AAAI*, pages 121–123, Stanford, CA., 1981.

[Clancey, 1993] W. J. Clancey. Acquiring, representing and evaluating a competence model of diagnostic strategy. In et al Buchanan, Bruce G., editor, *Readings in Knowledge Acquisition*, pages 178–215. Morgan Kaufman Publishers, San Mateo, California, 1993.

[Dixon *et al.*, 1987] J. R Dixon, A. Howe, P. R. Cohen, and M. K. Simmons. Dominic 1: Progress toward domain independence in design by iterative redesign. *Engineering with Computers*, 2:137–145, 1987.

[Fiebig, 1997] C. Fiebig. The development of expertise in complex domains. Master's thesis, University of Illinois, Urbana, Illinois, May 1997.

[Hayes, 1996] C. C. Hayes. p^3: A process planner for manufacturability analysis. *IEEE Transactions on Robotics and Automation*, 12(2):220–234, April 1996.

[Lane, 1986] N. E. Lane. Global issues in evaluation of expert systems. *Proceedings of the 1986 IEEE International Conference on Systems, Man and Cybernetics*, pages 121–125, 1986.

[Liebowitz, 1986] J. Liebowitz. Useful approach for evaluating expert systems. *Expert Systems*, 3(2):86–96, 1986.

[MacMillan *et al.*, 1993] J. MacMillan, E. B. Entin, and D. Serfaty. Evaluating expertise in a complex domain – measures based on theory. In *Proceedings of the Human Factors and Ergonomics Society*, pages 1152–1155, 1993.

[Nazareth and Kennedy, 1993] D. L. Nazareth and M. H. Kennedy. Knowledge-based system verification, validation and testing. *International Journal of Expert Systems*, 6(2):143–162, 1993.