QUEM: An Achievement Test for Knowledge-Based Systems

Caroline C. Hayes, *Member*, *IEEE*, and Michael I. Parzen

Abstract—This paper describes the QUality and Experience Metric (QUEM), a method for estimating the skill level of a knowledge-based system based on the quality of the solutions it produces. It allows one to assess how many years of experience the system would be judged to have if it were a human by providing a quantitative measure of the system's overall competence. QUEM can be viewed as a type of achievement or job-placement test administered to knowledge-based systems to help system designers determine how the system should be used and by what level of user. To apply QUEM, a set of subjects, experienced judges, and problems must be identified. The subjects should have a broad range of experience levels. Subjects and the knowledge-based system are asked to solve the problems; and judges are asked to rank order all solutions—from worst quality to best. The data from the subjects is used to construct a skill-function relating experience to solution quality, and confidence bands showing the variability in performance. The system's quality ranking is then plugged into the skill function to produce an estimate of the system's experience level. QUEM can be used to gauge the experience level of an individual system, to compare two systems, or to compare a system to its intended users. This represents an important advance in providing quantitative measures of overall performance that can be applied to a broad range of systems.

Index Terms—Knowledged-based systems, expertise, performance measures, knowledge engineering, solution quality.

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 (VVT) [18] 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. A complete and consistent KBS may not necessarily create high quality solutions.

It would be useful to have a method to estimate a KBS's overall competence. *Competence* is used in this context to mean the system's ability to perform in a value adding manner within its problem-solving context. In this paper, we address the issue of competence in terms of experience level and solution quality. We present the QUality and Experience Metric; we abbreviate it QUEM and pronounce the abbreviation "kwem." Put succinctly, QUEM is a method for evaluating the *experience level* of a knowledge-based system and the *quality* of its solutions. QUEM can be considered to be an *achievement test* for KBSs. We used expert judges to assess the quality of solutions generated by human experts and KBSs. We then constructed a "skill function" for the human experts relating experience and solution quality. We used the skill function and the KBS's quality ranking to estimate the KBS's experience level.

QUEM provides a *quantitative* way to estimate the experience level of a KBS, compare two KBSs, or compare the experience level of a KBS to that of its users. This last 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. 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 produces on his or her own, the user may consider the system to be a hindrance. Additionally, estimation of a KBS's experience level also allows developers to gauge how well they have succeeded in capturing the domain expertise.

We will demonstrate use of QUEM to estimate the solution quality and experience level of two versions of a KBS, Machinist [11] which produces optimized manufacturing plans. We used this measure to test the basic soundness of our KBS problem-solving approach, and to determine if we should continue on the same approach in future developments.

The following is a guide to the paper: In Section 2, we discuss our motivations in developing the method and explain our choices and assumptions in creating it. In Section 3, we review related work. In Section 4, we outline the QUEM method. In Section 5, we demonstrate QUEM by using it to evaluate the skill level and solution quality of two different versions of Machinist. Section 5 also describes the results of this evaluation. Section 6 summarizes these results, and Sections 7 and 8 discuss future work and conclusions, respectively.

- *C.C. Hayes is with the Department of Computer Science, University of Illinois at Urbana-Champaign, Urbana, IL 61801, and the Beckman Institute for Advanced Science and Technology, 405 N. Mathews Ave, Urbana, IL 61801. E-mail: hayes@cs.uiuc.edu.*
- *M.I. Parzen is with the Graduate School of Business, University of Chicago, 1101 E. 58th St., Chicago, IL 60637.*

Manuscript received 5 Dec. 1994; revised 13 May 1995.
For information on obtaining reprints of this article, please send e-mail to: tkde@computer.org, and reference IEEECS Log Number 104420.

2 A Short History of the Development

2.1 Motivations

The development of QUEM arose from the desire to measure the quality of Machinist, a knowledge-based system for manufacturing planning [12]. Machinist creates manufacturing plans for machined parts when given a geometric specification of the part. Initially, we implemented a prototype system to test the basic method.

However, before pushing ahead with a large effort to improve the completeness and robustness of the system, we wanted to be able to evaluate the basic soundness of the approach. In other words, if evaluation were to show that the best solutions produced by the system were of low quality, then we would conclude that our efforts should be focused on re-evaluating our basic approach and problem-solving architecture. On the other hand, if the solutions produced proved to be of high quality then we could feel confident in putting our efforts into further development of the current system architecture. To make this judgment, we needed to find a way to measure the solution quality produced by our evolving system.

2.2 Challenges in Developing a Quality Metric

Quality is in general hard to measure because it is hard to quantify. Even if one can generate a function to describe quality, it may be equally hard to quantify the component factors. In our initial attempts to estimate solution quality we attempted to construct just such a quality function composed of factors that our experts believed to be important: plan cost, feasibility, and reliability. However, we soon found this approach to be inadequate. After much tweaking and adjusting of the quality function we found that its assessments still did not agree very closely with those of the experts. Furthermore, we realized that it was unlikely that it would be possible to come up with an accurate quality function for the reason that many of the component factors, such as reliability, were very difficult to quantify accurately.

To describe the problem 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, for example when tools break, or in subtler ways, such as when the the resulting product does not meet tolerance 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 factors, the task of constructing a solution quality metric proved very difficult.

However, we found that experts were able to make quality assessments, and that they tended to agree strongly with each other in those assessments. One reason that experts can succeed in assessing quality where a quality function fails is that experts are able to estimate hard-to-quantify quality factors, such as reliability, because they have a broad range of empirical experience to draw on. Human experts vary some in their assessments, but that variability can be measured (for example, by having several experts independently rate the same solution) and taken into account. Advantages of this approach are that an experimenter can still measure quality without explicitly knowing the quality function.

Similar difficulties in measuring solution quality arise in many other domains, such as design and scheduling, in which solution quality is judged on a broad range of ill-defined characteristics. The measures described here are well suited for measuring quality in any domain in which a strong correlation between experience and solution quality can be demonstrated.

Next, we needed 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 machinists do not have a standard or agreed upon method for assigning numerical 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 10s while a conservative judge may rarely give a score better than 6. However, the first judges' 10 may mean the same thing as the second judges' 6. We decided, instead, to have the expert judges rank order the plans from best to worst. This makes it easier to compare the quality assessments of different judges.

3 Related Work

As mentioned earlier, most literature on knowledge-based system evaluation deals with validation, verification and testing (VVT) [18] in which the primary concern is with correctness, circularity, inconsistency, redundancy, modifiability, ease of use, and cost [14], [15]. However, these properties alone may not be enough to describe a system's competence. A complete, consistent, and nonredundant system may not solve very sophisticated problems, nor cover a broad range of problems. Conversely, a system which performs at a high level of competence, may be neither complete nor consistent.

Clancey [6] describes 4 perspectives useful for evaluating a system's competence: performance, articulation, accuracy (in terms of closeness to human reasoning), and completeness. Other parameters important to system competence are: solution feasibility, solution quality, problem-solving range, computer effort, and user effort. Computer effort refers to the speed at which the computer solves a problem and the number of decisions it must make [17]. User effort refers to the degree to which an expert computer tool increases (or decreases) the effectiveness of a human expert in problem-solving.

MacMillan et al. [16] report on an experiment in which expertise was assessed in a complex domain in which there are no agreed upon expertise measures, and no single right answers to problems. This study has certain similarities with QUEM in the nature of the domain studied and in the way in which data was collected. They studied various solution properties and problem-solving behaviors, and correlated them with expertise. Their goal was to eventually use these factors to measure human expertise, although

they had not constructed such a measure yet. They had several "super experts" act as judges, where super experts were retired four-star generals with extensive experience in a military tactical domain. Twenty-six military officers served as subjects and were given three tactical situations to solve. The judges rated the subjects overall level of experience and the judges' ratings were compared. They found that experts focus immediately on critical unknowns, build and use a richer mental model than nonexperts, use mental models to explore more possible outcomes, and develop more robust and flexible plans.

The most common way of evaluating a system's performance is simply to demonstrate that it generates *feasible* solutions for some family of problems. The result of such tests is usually binary: either the the system performs sufficiently or it does not. For example, Baykan and Fox demonstrate the feasibility of WRIGHT [3] on a set of five kitchen design tasks, a household layout problem, and a bin packing problem. These solutions are compared against standard solutions found in a kitchen designer's manual (which was presumably created by a human kitchen design expert.) In all cases, WRIGHT found the design in the kitchen manual.

Some evaluations provide *relative* measures of system competence. These evaluations provide slightly more information than simply saying "the system works" or "it doesn't work." They can also provide the information that system x works better than system y, or human z. Mostow, et al. [17] compared the effort expended by users of their system, BOGART, verses VEXED [19]. Dixon et al. evaluated their system, Dominic [8], by comparing its results against results generated by two other KBSs and a human expert. They tested the systems and the human on a set of six v-belt design problems, four heat sink problems, a rectangular beam design problem, and a tube truss design. From this comparison they concluded that "Dominic is a reasonably capable designer... although the two domain specific programs produced slightly superior performance." When Aikins 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 [1]. From this evaluation she concluded that not only could Puff perform competently, but that it was also *more* accurate on the average than any of the individual experts in the study.

However, simply knowing that one KBS produces better quality solutions than another KBS does not necessarily tell the KBS developers if either produces particularly good solutions. Both may produce very good solutions or poor solutions. For this reason we also felt it was necessary to develop a *quantitative* measure of KBS experience level. One would like to be able to say, "My KBS produces solutions equivalent in quality to an expert with *n* years of experience, or simply, "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.

Early work in providing assessments of KBS experience level includes work by Hayes [10], [11], in which a KBS's solutions are compared against those produced by humans at various experience. In the first of these two studies, it was found that the KBS, Machinist, performed better on average than a set of human practitioners having between 2 and 5 years of experience. However, since experts beyond 5 years of experience were not included in the study the exact level of the KBS could not be determined. In the later study, a new version of the KBS was developed and a wider range of experts was used. It was found that the new version of Machinist performed better than particular experts having 2, 2, 5, 5, 7, 11, and 24 years of experience, respectively, but less well than particular experts having 8 and 11 years of experience. The conclusion reached was that the new version of Machinist performed at an experience level between 7 and 8 years of experience.

However, these studies left many open questions. For example, if the KBS performed better than particular experts having 7, 11, and 24 years of experience but worse than a particular expert having 8 years of experience, what did that mean? Was the 8-year machinist particularly good for his experience level or were the 11- and 24-year machinists particularly bad? If the experts in the study deviated from the average, how sure could one be about the KBS's estimated experience level? Is is important to know how much confidence to assign to the KBS's estimated experience level. QUEM provides a way to determine this information by constructing confidence bands to show the expected range of variation in performance at any level.

4 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 judges should have experience equal to or greater than 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.

4.1 The QUEM Method

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 three problems, there will be three solution groups.
3) **Rank:** Have the expert judges independently rank order all solutions in each group from best quality to worst 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 equal in 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 6 solutions but she ranks 2 solutions as worst, 3 as intermediate, and 1 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 methods appropriate to the data). Call this the *skill function*. For example, if we have n human subjects and our data is of the form (x_i, y_i), $i = 1, \ldots, n$, with x_i denoting the average quality ranking of the ith subject and y_i the corresponding years of experience. We may model a linear relationship between x and y using simple linear regression resulting in the skill function $y = b_0 + b_1 x$ where

$$b_0 = \bar{y} - b_i \bar{x}, \quad b_1 = \frac{\sum_{i=1}^{n} x_i y_i = n\overline{xy}}{\sum_{i=1}^{n} x_i^2 = n\bar{x}^2},$$

and $\bar{x}$ and $\bar{y}$ are the respective sample means.
8) **Construct confidence bands**: Construct 95 percent pointwise confidence bands around this function. These bands show the variation in individual performances that one can expect to find at any given quality level.

Pointwise confidence bands are crucial to the analysis since a point estimate of experience is not useful without some idea how accurate the estimate is. 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 percent confidence interval for this estimate is given by

$$y_m \pm t_{(n-2,0.025)} \sqrt{\left(\frac{1}{n} + \frac{x_m - \bar{x}^2}{\sum_{i=1}^{n} x_i^2 = n\bar{x}^2} \right) s_e^2}$$

where $t_{(n-2,0.025)}$ is the 95 percent confidence coefficient based on the t–distribution and s_e^2 is an estimate of the amount of noise in the relationship between experience level and average quality rank. All these quantities are standard output results from statistics packages. Note that the width of the confidence interval is dependent on sample size, noise in the system and the distance of x_m from the average of the human subjects' average quality rank.

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) Again, take the KBS average quality ranking (x) and plug it into the equation for the upper confidence band. Repeat for the lower confidence band. The 2 numbers produced represent the *experience interval* for the KBS.

The results of this process are:

- 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 that might be capable of achieving the same average quality as the KBS, with 95-percent confidence.
- A *skill function* for humans, relating length of experience to solution quality.
- *Confidence bands* showing the expected range of skill in practitioners having a given length of experience.

4.2 Applications of QUEM

QUEM can be used in a variety of ways. It can be used to:

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**. Same method as 2.
4) **Compare 2 unrelated KBSs** that operate in different domains. In order to compare two unrelated KBSs, 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 trials on the same user: one 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 you would two versions of a KBS. Create the skill function using a set of subjects other than the user. Use the skill function to estimate the skill level of the user with and without the KBS.

4.3 Selecting Judges, Subjects, and 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 problem domains which we studied (manufacturing and software development) that 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 real-world domains. However, it may not be true for formal games or domains that have been simplified for formal study because much of the richness may have been removed.

Subject and judge selection. The judges should preferably have more than 10 years of experience. (MacMillan et al. [16] 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 which is being evaluated.

Range of subjects. Ideally, one would like to select subjects so that the experience level of the KBS falls in the middle of the subjects' experience range (although the method will still work even if the KBS falls slightly outside the range of the subject's experience). If the KBS falls too far outside the range of the subjects, the experience interval may become too broad to supply a useful 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 guess at what the experience levels of the subjects should be. It may be necessary to conduct one or two pilot studies in order to find the right experience range for the subjects. The first time we tested Machinist [10], we did not guess right. We selected subjects between 2 and 5 years of experience, but found that Machinist's experience level to be above the range of these subjects. We conducted a second test on Machinist [11] 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. These two previous studies enabled us to select the correct range of subjects (2 to 10 years) for the study in this paper.

Problem selection. There was some initial concern in this study that it would be difficult to select appropriate problems. A range of four problems was selected that were intended to differ in difficulty. However, the analysis showed that three of the problems were of equal difficulty and the fourth was too hard for the majority of subjects. The study also showed that it was not necessary to use problems of varying difficulty to distinguish novices and experts. For any one problem, the structural differences their solutions was so great that virtually any problem would have served equally well to distinguish skill gradations between 2 and 10 years. In other work, Fiebig and Hayes [9] found that this is also true for experts in the software management domain.

4.4 Graceful Degradation

There are many sources of variation in the data. Variations may arise from differences in the way judges make assessments or slight mismatches between the judge's domain of experience and that of the subjects or KBS. Variations in skill between two subjects having the same reported length of experience may be due to differences in talent or motivation during training. Additionally, it is often difficult to pinpoint the start of an expert's training precisely; a machinist who's family runs a machine shop as a business may have picked up a certain number of skills through observation long before the start of formal training. Thus, two subjects with the same reported length of experience may not have the same actual length of experience. The total variation in the data is reflected in the width of the confidence bands and the experience interval.

This representation of variability makes QUEM robust to noise to an extent. If the experimenter accidentally introduces additional variation by poor selection of one judge or subject, it will probably not greatly affect the results. In the worst case, the experience interval may become so broad as to provide little useful information.

4.5 Limitations

QUEM can provide useful information for a domain only when practitioners in the field show a distinct improvement in skill (measured through solution quality) over time. Experience may not bring skill (or wisdom) in all domains. However, researchers wishing to evaluate a KBS may not know a priori if an experience/solution quality relationship exists in the domain. Since systematic measurements of this relationship are not typically taken, it is not definitively known in most domains whether such a relationship exists. The existence of such a relationship can be determined by applying QUEM; if a simple function can be found which fits the data well then a relationship exists.

The converse, that no relation exists, is harder to determine. If 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 that increased skill manifests itself in ways other than through increased solution quality (such as increased speed in producing a solution).

5 Example: Evaluation of a Manufacturing KBS

Task domain. QUEM was used to evaluate a particular KBS, Machinist. Machinist is designed to automatically generate manufacturing plans given a description of an artifact. *Machining* is the art of shaping metal with a variety of tools. In this particular task, parts were to be created on a A CNC machining center, which is a computer controlled machine tool that can perform a variety of different types of machining operations such as drilling, milling, or reaming. Humans who perform this task are highly skilled individuals requiring as much as 8 to 10 years of intensive practice to achieve master level status. To create a manufacturing plan, the machinist must select and sequence the manufacturing operations. He or she must also choose particular cutting tools clamps, and workpiece positions for each op-

eration. Skills that must be acquired by machinists in order to create high quality plans include the ability to select appropriate operations, detect interactions, and optimize the overall plan.

KBSs. In evaluating Machinist we examined and compared two versions of the system: an early version, frozen 2-1/2 years after the start of system development, and a later version, frozen 5-1/2 years after the start of development. We will call the early version Machinist 1 and the later version, Machinist 2.

Subjects and judges. 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.

Problems. We prepared three problems for the subjects to solve. All three problems were of approximately the same difficulty level. For each problem, subjects were given a drawing of a part, a description of the stock material from which the part was to be made, and a list of tools and machines they could use in manufacturing the part. The subjects were then asked to generate a manufacturing plan for fabricating the part using the given materials and equipment.

5.1 Applying QUEM

1) **Solve:** We had each of the subjects and the two KBS solve all three problems. We wrote up all solutions in a uniform format and handwriting (to disguise their source).

   Fig. 1 shows an example of a solution created for problem number 3 by a machinist with 5 years of experience. The solution is a manufacturing operations plan. For each step in the plan, the changing shape of the part is shown on the left. The operations performed are shown in the center. The comments made by two experienced judges are included in the right margin.
2) **Sort:** We sorted the solutions into three 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 Table 1. P1, P2, and P3 are problems 1, 2, and 3, respectively. The missing data points resulted when subjects were unable to complete all three problems due to being called away to attend to immediate job demands.
4) **Adjust ranks**. This step was not necessary for this data because individual judges did not judge any of the plans to be equal in quality.
5) **Compute subject averages**. Next the average quality ranking received by each subject across all three problems was computed. These values are shown in the last column of Table 1. The lowest average score, 2.50, was received by the machinist with only 2 years of experience. The highest average score of 5.67 was received by the machinist with 10 years of experience. The early version of Machinist, Machinist 1, had an average quality ranking of 4.67 and the later version, Machinist 2, was 5.67. A factorial analysis performed on the data showed experience to be statistically significant, but not judge nor part, which is what we had hoped would be true.
6) **Plot**. The average quality rankings received by the humans only, were plotted on the graph shown in Fig. 2.
7) **Fit a skill function to the data**. We fit several types of curves to these data, including logarithmic and several types of polynomials, but found that a simple linear regression fit the data quite well. The regression yielded the following equation for the model:

$$y = -1.98 + 1.62x$$

   This is the *skill function*. The skill function is shown in Fig. 2 as a heavy diagonal line.
8) **Construct confidence bands**. Fig. 3 shows 95 percent confidence bands as curved bands flanking the skill function. The formula for the bands is given as a function of x, the average quality rank. Numerically, the formulas are for a given x value,

$$\text{high band} = -1.98 + 1.62x + 2.57\sqrt{\left(\frac{1}{7} + \frac{(x-4.67)^2}{18.54}\right)1.05},$$

$$\text{low band} = -1.98 + 1.62x - 2.57\sqrt{\left(\frac{1}{7} + \frac{(x-4.67)^2}{18.54}\right)1.05}.$$

9) **Plot the KBS average quality rank**. The average quality ranks for both Machinist 1 and Machinist 2 were plotted on the quality (x) axis. However, only the latter version, Machinist 2, is shown in Fig. 3 to prevent it from becoming too cluttered.
10) **Construct an experience estimate and interval.** or each KBS, Machinist 1 and Machinist 2, the result of this step will answer the question, "If the program's solutions were created by a human, how much experience would an expert judge estimate that human to have?"

   Machinist 1 received an average plan quality rating of 4.67. If one plugs this value into the skill function as x, the equation yields y = 5.58. This means that Machinist 1 is estimated to have an experience level of 5.58 years. This is the *experience estimate*. This figure is supported by an earlier study [10] which indicated that the Machinist 1 system performed at a level superior to humans with 5 years of experience. This level of competence was produced after only 2-1/2 person-years of development.

   Machinist 2 received an average plan quality rating of 5.67. Using the skill function as above, it was determined that Machinist 2 is estimated to have 7.20 years of experience. This is the *experience estimate*. Using 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 Machinist 2 lies somewhere between 6.03 and 8.36 with 95 percent confidence.

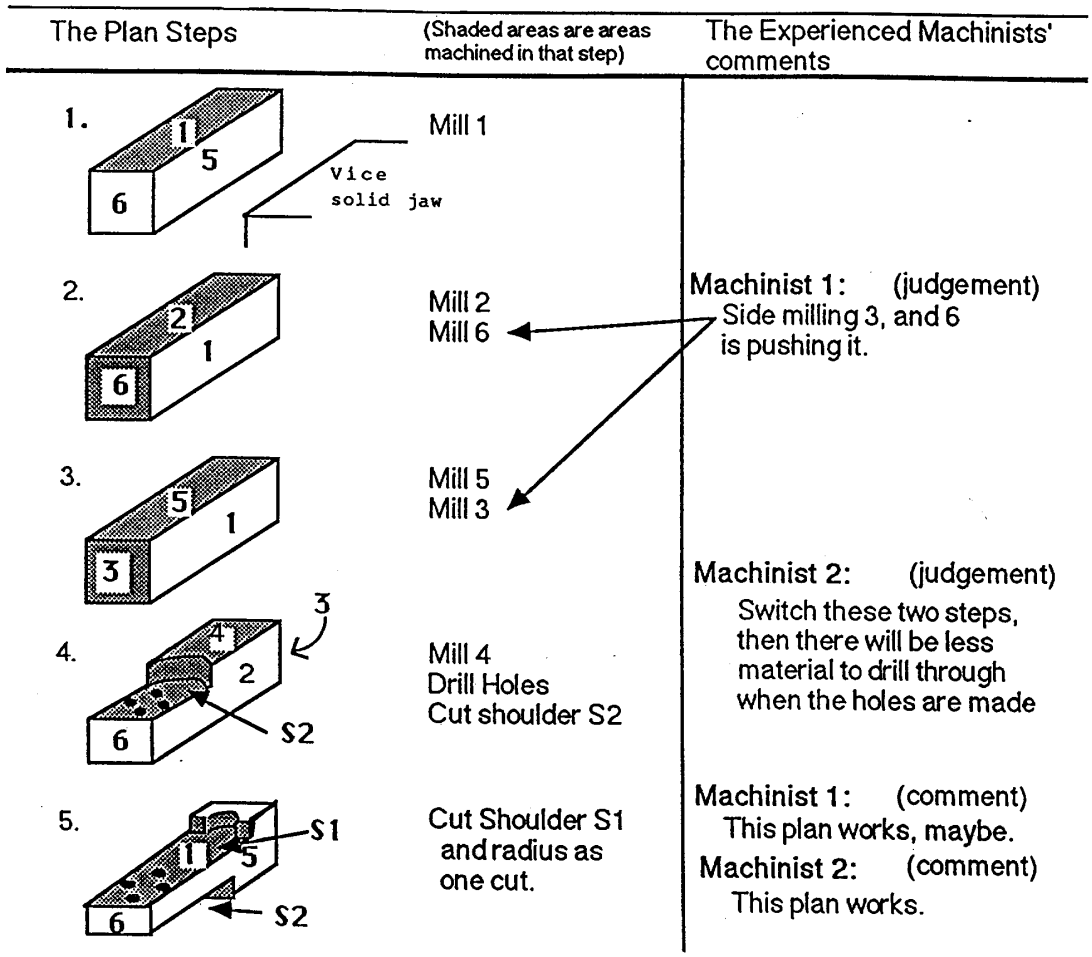

Fig. 1. An operations plan generated by a machinist with 5 years of experience.

TABLE 1
QUALITY RANKINGS ASSIGNED BY JUDGES TO SOLUTIONS

		Judge 1			Judge 2			
Problem Solver	Years of experience	P1	P2	P3	P1	P2	P3	Average Solution Quality Rank
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
Machinist 1	*	6	6	3	5	2	6	4.67
Machinist 2	*	7	4	2	6	7	8	5.67

In Fig. 3, we display both the experience estimate and interval for Machinist 2. Both the earlier and the later versions of the Machinist system exhibit a very high experience level. On the basis of these results we confirmed that our problem-solving architecture 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.

6 DISCUSSION

6.1 Changes in the Rate of KBS Skill Improvement

We had mixed feelings about the findings reported in the previous section. On the positive side, the skill level of the

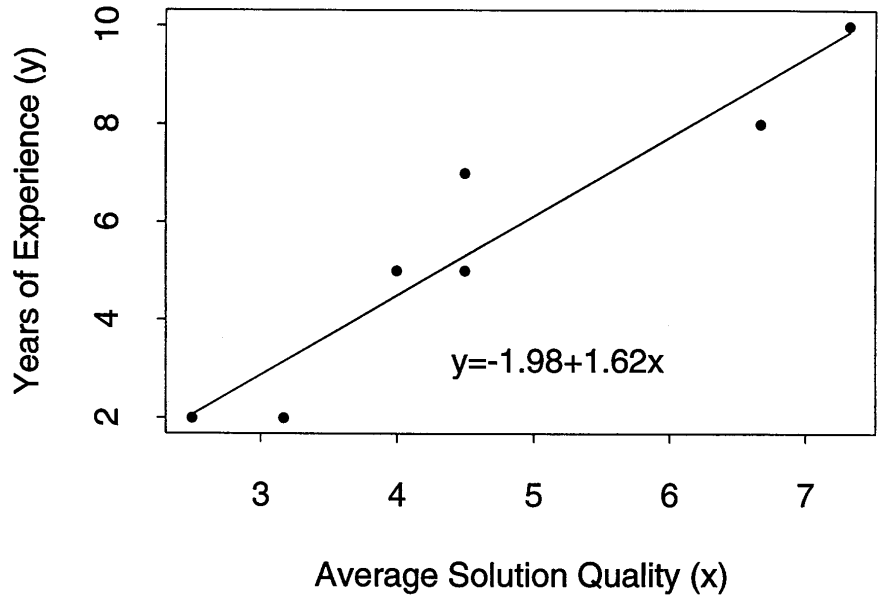

Fig. 2. Plot of average quality rankings and skill function.

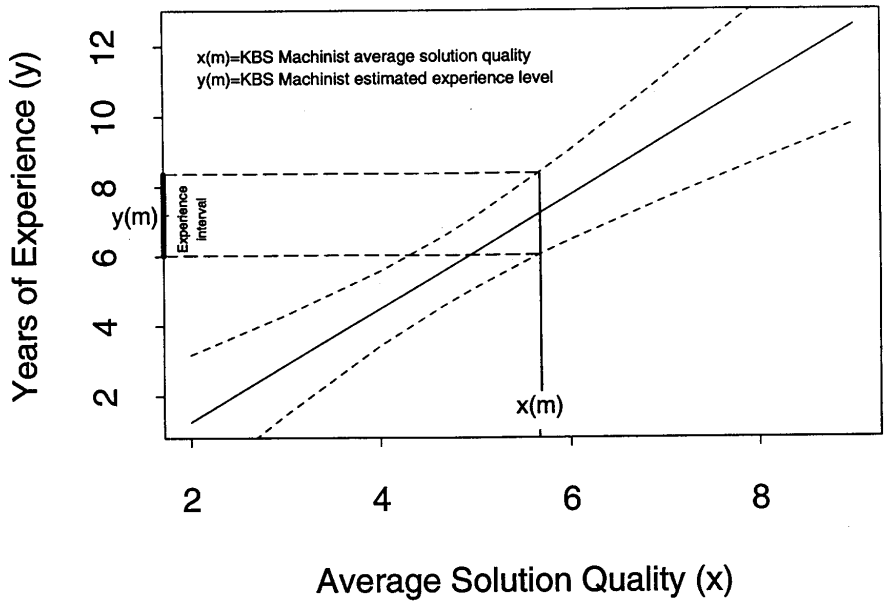

Fig. 3. The experience estimate and interval for the KBS, Machinist 2.

KBS at all stages was greater than the number of person years required to develop the system. On the negative side, after 3 additional years of intensive development on Machinist 1, the experience level of the system was only improved by another year (although we had broadened the problem range quite a bit.)

There are several interpretations we could put on these results. One is that the true experience levels of the two versions of Machinist 1 and Machinist 2 are not really 5.58 and 7.20 years. Perhaps the true experience level of Machinist 1 lies somewhat lower (but within the confidence interval) and Machinist 2 lies some what higher.

A second interpretation (which does not exclude the first interpretation) is that this study only estimates the impact of increased experience on solution quality. However, it does not reflect *all* advancements in the system's overall competence. Problem *range* is also an important part of competence. Machinist 2 can solve a much broader range of problems than can Machinist 1. If one had a good metric for measuring range, it might also be possible to estimate experience with respect to the range of problems a KBS can solve. To get a complete picture of all changes in a system's overall competence, it may be desirable to estimate experience with respect to *several* factors.

Now that we have introduced this method for measuring skill levels of knowledge-based systems, we hope that other researchers will perform similar studies on other systems. We will be very interested to see if other

KBSs exhibit similar nonlinear changes in experience level over time.

6.2 Cost of a QUEM Evaluation

We have frequently been asked by other researchers how much it cost in terms of time to conduct a QUEM evaluation. Our evaluation of two versions of the Machinist system required approximately 17 hours. This will depend partly on the domain and the difficulty of the test problems selected. Some people are daunted because they believe that they must collect a large group of experts, who's services are difficult to obtain, in order to conduct a QUEM evaluation. However, most of the subjects used in the study are required to be nonexperts because the method uses subjects having a variety of experience levels. Those at lower experience levels can be found in relative abundance. It is only finding highly experienced judges that presents a challenge.

The 17 hours to perform our tests was completed, intermittently, over a period of about three weeks. We found that experts and apprentices were in general quite interested in participating in the study. Each of the seven subjects in the study took between 50 and 70 minutes to complete all solutions to all three test problems. We also had two additional subjects attempt solutions, but found that their experience was not appropriate. In our prescreening of these subjects, they claimed to have the appropriate CNC experience, but after examining their solutions and questioning them further, we found that their CNC experience was actually very weak. Although these two subjects were not included in the study, we feel it is important to mention them because time to collect unused data still needs to be calculated in the total time required.

The two judges each took approximately 90 minutes each to critique and rank all solutions. All together, it took approximately 17 hours to collect the data, have judges rank the solutions and analyze the results. We found that the information we got from this evaluation was well worth the small investment of time.

6.3 Advantages

The QUEM method for measuring the experience level of a KBS has several advantages. It allows measurements to be taken on a partially developed KBS system without requiring the KBS to be complete, correct, or broad in problem coverage. Such measures are important for allowing KBS system developers to test the basic validity of an approach before spending additional effort making the system more complete and robust. Additionally, it can be used in domains in which a quality function is hard to quantify precisely.

7 Future Work

In future studies, we would like to construct models of the many stages of problem-solving behavior and the changes that occur in the transition from novice and expert. Additionally, we would like to examine more closely what happens between 10 and 20 years of experience. This study does not use data in the range beyond 10 years of experience because any experts at that level were used as judges. Consequently, it is unknown what happens in skill development beyond 10 years of experience. Is there a leveling off point in later years beyond which further experience does not necessarily lead to better quality plans?

We did examine the plans of 1 subject who had 22 years of experience. The judges rated his plans as slightly lower in quality than the 10-year expert. However, on further consideration, it was not clear if this was a fair comparison, because the 22-year expert was trained on one style of machine (manual) while the plans were being judged as if they were made for computer controlled (CNC) machines. It is possible that the reason we found it difficult to find experts 20-plus years of relevant experience in the manufacturing domain is that technology advances at a rate fast enough to make experience acquired 20 years ago obsolete.

8 Summary and Conclusions

In this paper, we presented QUEM, a general method for measuring the experience level of a KBS, and assessing the quality of its solutions relative to human practitioners. This method allows researchers to address the question, "How expert is my expert system?"

The method involves having expert judges rank order solutions produced by both KBSs and human subjects, constructing a skill function describing the relationship between length of experience and solution quality in the human subjects, then using function and the KBS's quality rankings to estimate the KBS's experience level.

Previous methods for evaluating a KBS performance involve *qualitative* comparisons. For example, "System x performs *better than* system y," which is not to say that either system performs well at all. The QUEM procedure allows a system developer to make a *quantitative* assessment of the 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?"

Some other advantages of QUEM are that it can be used in any domain in which increased experience leads to measurably increased solution quality. Additionally, it can be used on a system that is under development that may not be entirely complete or correct in all aspects, as long as it can construct solutions. It can be used to measure the experience level of an individual KBS, compare several KBSs which operate in the same or in unrelated domains, or estimate the amount by which a computer assistant raises the skill level of the user.

Assessing the experience level of a KBS system is important in helping developers to decide if their approach is sufficient, and how the system should be used, and with what level of user it should interact.

Acknowledgments

A special thanks to all the machinists who provided data or served as judges for this study, including Jim Dillinger, Dan McKeel, Jack Rude, Steve Rosenberg, Bill Knight, Dare Hire, and Mike Westjohn. Another thanks to Bill Burtle for his data collection, to J.R. Hayes for his psychological data analysis advice, and to Michael Dorneich for his editorial comments.

REFERENCES

[1] J.S. Aikins, "Representation of Control Knowledge in Expert Systems," *Proc. First AAAI*, pp. 121-123, Stanford, Calif., 1981.

[2] J.R. Anderson, "Development of Expertise," *Readings in Knowledge Acquisition*," B.G. Buchanan et al., eds., Morgan Kaufmann, San Mateo, Calif., pp. 61-77, 1993.

[3] C. Baykan and M. Fox, "WRIGHT: A Constraint-Based Spatial Layout System," C. Tong and D. Sriram, eds., *Artificial Intelligence in Eng. Design*, vol. 1, ch. 13, pp. 395-432, Academic Press, San Diego, 1992.

[4] D. Card, "What Makes for Effective Management," *IEEE Software*, Nov. 1993.

[5] *The Nature of Expertise*, M. Chi, R. Glasser, and M.J. Farr., eds. Hillsdale, N.J.: Lawrence Erlbaum Associates, 1988.

[6] W.J. Clancey, "Acquiring, Representing, and Evaluating a Competence Model of Diagnostic Strategy," B.G. Buchanan et al., eds. *Readings in Knowledge Acquisition*, pp. 178-215, Morgan Kaufmann, San Mateo, Calif., 1993.

[7] P. Cohen, T. Dean, Y. Gil, M. Ginsberg, and L. Hoebel, "Handbook of Evaluation for the ARPA/Rome Lab Planning Initiative," *Proc. Workshop ARPA/Rome Laboratory Knowledge-Based Planning and Scheduling Initiative*, Tucson, Ariz., Morgan Kaufmann, Feb. 1994.

[8] J.R. Dixon, A. Howe, P.R. Cohen, and M.K. Simmons, "Dominic I: Progress toward Domain Independence in Design by Iterative Redesign," *Eng. with Computers*, vol. 2, pp. 137-145, 1987.

[9] C. Fiebig and C.C. Hayes, "Modeling the Development from Novice to Expert Human Planners," *Proc. IEEE Conf. Systems, Man, and Cybernetics*, Bejing, China, 1997.

[10] C.C. Hayes, "Observing Machinists' Planning Methods: Using Goal Interactions to Guide Search," *Proc. Ninth Ann. Conf. Cognitive Science Soc.*, pp. 952-958, Seattle, 1987.

[11] C.C. Hayes, "Machine Planning: A Model of an Expert Level Planning Process," PhD thesis, Carnegie Mellon Univ., Pittsburgh, Penn., May 1990.

[12] C.C. Hayes and H. Sun, "Reasoning About Manufacturability Prior to Reaching the Shop Floor," *Proc. SIGMAN Workshop Reasoning about the Shop Floor*, Seattle, 1994.

[13] J.S. Lancaster and J.L. Kolodner, "Problem Solving in a Natural Task as a Function of Experience," *Proc. Ninth Ann. Conf. Cognitive Science Soc.*, pp. 727-736, Seattle, 1987.

[14] N.E. Lane, "Global Issues in Evaluation of Expert Systems," *Proc. IEEE Int'l Conf. Systems, Man, and Cybernetics*," pp. 121-125, Atlanta, 1986.

[15] J. Liebowitz, "Useful Approaches for Evaluating Expert Systems," *Expert Systems*, vol. 3, no. 2, pp. 86-96, 1986.

[16] J. MacMillan, E.B. Entin, and D. Serfaty, "Evaluating Expertise in a Complex Domain—Mesures Based on Theory," *Proc. Human Factors and Ergonomics Soc.*," pp. 1,152-1,155, 1993.

[17] J. Mostow, M. Barley, and T. Weinrich, "Automated Reuse of Design Plans in BOGART," C. Tong and D. Sriram, eds., *Artificial Intelligence in Eng. Design*, vol. 2, ch. 2, pp. 57-104, Academic Press, San Diego, 1992.

[18] D.L. Nazareth and M.H. Kennedy, "Knowledge-Based System Verification, Validation, and Testing," *Int'l J. Expert Systems*, vol. 6, no. 2, pp. 143-162, 1993.

[19] L.I. Steinberg, "Design as Top-Down Refinement Plus Constraint Propegation," C. Tong and D. Sriram, eds., *Artificial Intelligence in Eng. Design*, vol. 1, ch. 8, pp. 251-272, Academic Press, San Diego, 1992.

Caroline C. Hayes received her BS degree in 1983 in mathematics, her MS degree in 1987 in knowledge-based systems, and her PhD degree in 1990 in robotics, all from Carnegie Mellon University in Pittsburgh, Pennsylvania. Her PhD was the first of its kind ever awarded from a robotics department. Since 1991, she has been a member of the Department of Computer Science at the University of Illinois, Urbana-Champaign, and at the Beckman Institute for Advanced Science and Technology, also located in Urbana.

In February of 1998, she will be starting in the position of associate professor in the University of Minnesota's Department of Mechanical and Industrial Engineering. Her awards and honors include a Carnegie Mellon postdoctoral fellowship in 1990, the University of Illinois Incomplete List of Teachers Rated as Excellent, the W.C. Gear Outstanding Junior Faculty Award in 1996 and, in 1998, the Richard and Barbara Nelson chair. Her research focus is in creating tools for decision support and automation in complex domains such as manufacturing planning, architectural and mechanical design, and military planning. She is a member of the IEEE, the IEEE Computer Society, and the Editorial Board of *IEEE Transactions on Knowledge and Data Engineering*.

Michael I. Parzen received the BSc degree in mathematics from Carnegie Mellon University in Pittsburgh, Pennsylvania, in 1987, and the DSc degree in biostatistics from Harvard University in 1993. He is currently an associate professor of statistics in the Graduate School of Business at the University of Chicago. His main research interests are statistical computing, robust estimation, and survival analysis.