MEASURING THE VALUE OF KNOWLEDGE [1]

Yoram Reich
Department of Solid Mechanics, Materials and Structures
Faculty of Engineering
Tel Aviv University
Ramat Aviv 69978
Israel

Key words (beside the title):

evaluation metrics, measurement theory, verification and validation of expert systems, software engineering, design, machine learning, knowledge acquisition, research methodology

ABSTRACT

The quality of knowledge a system has substantially influences its performance. Often, the terms knowledge, its quality, and how it is measured or valuated, are left vague enough to accommodate several *ad hoc* interpretations. This paper articulates two definitions of knowledge and their associated value measures. The paper focuses on the theory underlying measurements and its application to knowledge valuation; it stresses the issue of constructing meaningful measures rather than discussing some of the desirable properties of measures (e.g., reliability or validity). A detailed example of knowledge valuation using the measures is described. The example demonstrates the importance for system understanding and the difficulty of valuating knowledge. It shows the importance of employing several different measures simultaneously for a single valuation. The paper concludes by discussing the scope of and relationships between the measures.

1 INTRODUCTION

In a world with information highways, many kinds of data, information, or knowledge become commodities whose trade will be based on, or requires methods of, valuation (Mowshowitz, 1994). The study of knowledge valuation methods is also motivated by more immediate reasons. The first and most general motivation is related to education and knowledge acquisition: Knowledge valuation methods can help identify good knowledge to be used by people or for inclusion in computer systems. The second motivation is methodological: Knowledge valuation methods can support the evaluation of systems developed in research or practice and the determination of their relative merit. This evaluation is essential for providing feedback on research progress and for supporting the refinement of ideas. In some situations, such as when developing systems by prototyping or developing learning systems, knowledge valuation methods must be used within projects and not only for comparing between them. A third motivation is related to building integrated systems: When a complex computer system has several competing modules for solving each of its task, knowledge valuation can identify which module to invoke for solving the task.

There may be other motivations to study knowledge valuation but in this study, we limit the discussion to the second function: The valuation of knowledge embedded in, or to be used by, computer support systems. This motivation is related to the general need to evaluate intelligent systems and to the active field of verification

[1] An different version appeared in the Proceedings of the Seventh Banff Knowledge Acquisition for Knowledge-Based Systems Workshop, Banff, Canada, 1992

and validation of expert systems. The state of evaluating intelligent systems was documented by Green and Keyes (1987) and seemingly has not progressed much since (Guida and Mauri, 1993; Gupta, 1993). Green and Keyes described the reluctance of system developers to verify and validate their products. The lack of effort spent on evaluating software systems in general, and expert systems specifically is also discussed by Adelman (1991) and by Cohen and Howe (1989) who also illustrate the benefits from evaluation to research and the need to support evaluation.

When evaluation of systems is addressed in studies, it is often performed in an *ad hoc* manner (Guida and Mauri, 1993). The evaluation criteria are often ill defined or even meaningless. Even recent proposals about evaluation (e.g., (Guida and Mauri, 1993)) or group projects aimed at evaluation (e.g., the Sisyphus project (Linster, 1992)) do not address the theoretical issues underlying measurements we list below.

This status is not surprising given that in the more general field of software engineering the status is the same. For example, Zuse and Bollmann (1987) discussed the chaotic state of measures in software engineering. Most of the software measures appearing in the literature do not conform to the notions of measurement theory (Pfanzagl, 1971; Roberts, 1979; Stevens, 1946). The argument is not that some measures are superior to others for different purposes or under certain conditions, but that measures must have certain properties that determine their meaningful status, and that those properties are violated in most studies. This paper attempts to remedy this situation by presenting key issues that must be addressed when developing measures for system evaluations.

The topic of knowledge valuation introduces several prerequisite issues. First, how is something in general being valuated or measured? Tied with this question are the concepts of *measure* (or metric) and *scale* we review in Section 2. Second, what do we mean by knowledge? Third, how can valuation methods be applied to knowledge. Section 3 provides two definitions of knowledge and Section 4 proposes four methods for valuating knowledge: structural qualitative and quantitative, and functional qualitative and quantitative. Section 5 describes the experimental design support system BRIDGER that is used to illustrate the valuation measures. Section 6 provides a detailed example of measuring the bridge design knowledge that BRIDGER has. The extent of the valuation allows the appreciation of the many aspects affecting the selection and application of different valuation measures of knowledge. Section 7 analyzes the example, discusses the construction of measures, and outlines some future work.

The discussion in this paper should be viewed as outside the controversy about whether computer can think. We will define knowledge without relation to human knowledge or intelligence and similarly in the example, will refer to design knowledge without reference to human design. By stating this, we intend to avoid McDermott's (1981) criticism about the misuse of terminology, refrain from dealing with unresolved controversies that are orthogonal to the topic of the paper, and maintain the focus on the methodological aspects of system evaluation, in general, and knowledge valuation, specifically.

2 MEASURES AND SCALES

2.1 Basic definitions

In order to measure something meaningfully, one has to use an appropriate measure or a scale.[2] These concepts can be formalized. First, consider two relational systems, *empirical or observed* denoted by $\mathcal{A}$ and *formal* denoted by $\mathcal{B}$. $\mathcal{A}$ is defined by

$$\mathcal{A} = \{A, R_1, \ldots, R_n, O_1, \ldots, O_m\}, \text{ where} \quad (1)$$

A is a non − empty set of objects,

[2]The following discussion on the theory of measurement borrows much from Zuse and Bollmann (1987).

R_i are relations on A, and

O_j are binary operations on A;

and $\mathcal{B}$ is defined similarly, with the set of objects B, relations S_i, and binary operations P_j on B. For example, an observed relational system may contain the set of physical objects, the relation *heavier-than*, and the binary operation of *assembling-two-objects*. The formal relational system can consist of the set of positive real numbers, the relation *larger-than*, and the *addition* operator.

Second, there must be a function μ, called *metric or measure*, from the observed to the formal relational system. If this metric preserves the relations and binary operations, i.e., $\forall i, j$ and $\forall a, b, a_1, \ldots, a_k \in A$

$$R_i\,(a_1, \ldots, a_k) \quad \Leftrightarrow \quad S_i\,(\mu(a_1), \ldots, \mu(a_k)) \text{ and} \tag{2}$$

$$\mu(a\ O_j\ b) \quad = \quad \mu(a)\ P_j\ \mu(b), \tag{3}$$

it is called a *homomorphism*. The tuple $(\mathcal{A}, \mathcal{B}, \mu)$ where μ is a homomorphism is called a *scale*. A mapping that assigns each object its weight is a homomorphism and therefore a scale.

Intuitively, preserving the relations (equation 2) means preserving the equivalence classes of the original relation, so that the ordering between entities is maintained. This is a basic requirement for measures. Preserving the binary operations (equation 3) means that if we want to create entities from others, that we need not measure the new entities but we could assess their value from those of their building blocks. In the area of knowledge acquisition, machine learning, or validation of expert systems, this aspect is often ignored, but is most critical because the systems being measured continually evolve and increase their knowledge "content."

2.2 Types of scales

There are several types of scales: nominal, ordinal, interval, and ratio. The meaning of the different scales is intuitive. They can be defined formally depending on the type of transformations, $g : \mu(A) \rightarrow B$, that can map the scale $(\mathcal{A}, \mathcal{B}, \mu)$ to another scale of the same type. For example, a ratio scale admits only a similarity transformation, i.e.,

$$g(x) \quad = \quad ax,\ a > 0, \tag{4}$$

whereas an interval scale admits the following transformation

$$g(x) \quad = \quad ax + b,\ a > 0. \tag{5}$$

For example, the mapping that assigns each object its weight is a ratio scale; it can work with *Kg* or *lb* as the measuring units. The translation between them works according to equation 4.

The type of scale determines its uses for valuation. For example, we can calculate the arithmetic mean of interval and ratio scales, but can only calculate percentage with a ratio scale. Thus, it is *meaningful* to calculate the percent increase of weight, but *meaningless* to calculate this for temperature which is based on an interval scale. The types of scales can be ordered depending on their "informativeness" where the ratio scale is the most informative, i.e.,

$$nominal\ <\ ordinal\ <\ interval\ <\ ratio. \tag{6}$$

To be meaningful, measures used to valuate knowledge should have all the aforementioned properties of scales, especially the preservation of the binary operations that is often neglected.[3] These are the binary operations that give hope for the ability to combine small building blocks such as pieces of knowledge into larger, more "knowledgeable" one. To illustrate the benefit from this requirement, consider that instead of the observed system with the physical objects we had an observed system whose set of objects consisted of sets of rules, its relation was *more-knowledgable*, and its binary operation was the *union* of rule sets. We could have mapped a set of rules to a positive number denoting its *knowledgeability* or IQ. If we were able

[3] Roberts (1979) briefly mentioned that there is a controversy about whether the binary operations are required for the definition of measures or not.

to devise a mapping that would be a homomorphism, we would have been able to tell what is the IQ of a combination of several rule sets by adding their corresponding IQs.

One note about quantification of metrics is due. The informativeness of a scale correlates with it being quantified. Thus interval or ratio scales are much sought after compared to nominal or ordinal scales. Often, the quantified scales are perceived to be related to more "scientific" procedures, or to reflect a more mature understanding of a subject matter, or even to be objective. We note that in many cases, there can be many quantified scales between two relational systems and there may be situations that no perfect quantified scale can be defined,[4] yet some are used nonetheless. If such a choice is not considered subjective, it is theory-laden and intersubjective, or at best, interpersonal (Kyburg, 1984). Thus, no valuation can be objective.[5]

Also, often a quantified valuation is impossible and sometimes it is inappropriate for some measurements. In such cases, combined qualitative and quantitative valuation methods are appropriate (Kaplan and Duchon, 1988). In all cases, however, independent of the type of scale used, the valuation methods must be designed to have the properties of scales. That is, the construction of a good metric or measure for knowledge involves creating the observed and the formal systems as defined in equation 1 and making sure that the metric is a homomorphism. In particular, it is critical to guarantee the preservation of the binary operations (equation 3).

2.3 Direct and indirect measures

Some properties of objects can be measured directly.[6] Weight and length have direct scales. Length is the prototypical direct measure. We can "observe" its value and the measurement directly corresponds to our observations. It is almost as if we do not need the formal system for defining the metric since it is almost identical to the observed one. We cannot, however, measure directly intelligence or the knowledgeability (or expertise) of a person or a system. We may have some intuition about who is smarter or more expert than another but there is no dependable way that we can define an ordering and be able to generate equivalence classes for defining a relation in the observed system and subsequently a scale to allow for a direct measure of intelligence (Kyburg, 1984).

Since we have a tendency to seek quantifications, we may look for an *indirect* way to measure intelligence, such as using IQ or performance tests. An indirect measure has to satisfy some necessary conditions. It has to conform to our intuition about the direct measure. That is, it has to be *valid*. Since we have no way to "combine" human intelligence (unless we talk about committees of experts, and if we do, there is no way to predict what is the "intelligence" of a committee from that of its members), validity only relates to our intuition about the relations in the observed system. Thus, the indirect measure must be some monotonic transformation of the direct measure. However, with respect to the indirect measure, such as an IQ test, any monotonic transformation of it will also fit our direct judgment and potentially many other non-monotonic transformations of IQ. Therefore, validity may not be such a strong condition. The second necessary condition is *reliability*. It relies not merely on replicability of measurements but on our believe that the property being measured is somehow invariant. A reasonable assumption when dealing with systems that do not learn or evolve through development.

Measuring human intelligence is different than measuring the knowledge of intelligent systems. As we said, we cannot "combine" human intelligence. Therefore, for human intelligence, the binary operations in

[4]Consider the study in (Adelman, 1989) reporting no (or almost no) significant differences in knowledge base quality when it was developed by varying the domain expert, elicitation method, and the knowledge engineer. Since we would have anticipated differences, the results suggest that quality scales are extremely hard to formulate.

[5]We disagree with Gaines and Shaw (1989) reference to "*objective knowledge*" as the knowledge arrived at by consensus among experts; we view such knowledge as been intersubjectively created within a constructivist viewpoint.

[6]This section borrows from Kyburg (1984).

equation 1 are ignored. This omission neglects the major difficulty underlying the meaning of measures that arises in the presence of the binary operations, and leaves us to deal with a task similar to the validation of intelligent systems by various types of performance experiments as discussed elsewhere (Adelman, 1991).

In contrast, for the knowledge sharing initiative, it is required that knowledge be combined to yield better knowledge. Thus, the binary operations are critical and must be taken into consideration when developing measures. Consequently, indirect measures of systems' knowledge are much harder to devise than those for human intelligence.

3 WHAT IS KNOWLEDGE?

Before discussing the value of knowledge and how it may be measured we must define what do we mean by the term knowledge. The two definitions offered are not a contribution of this work, they merely reflect existing perspectives. In defining knowledge it must be understood that the definition dictates the type of valuation measures that can be applied.

Many researchers or studies on knowledge-based systems avoid the question of what knowledge is by discussing knowledge representation.[7] Implicitly, these studies define knowledge as:

STRUCTURAL DEFINITION: Knowledge is *whatever is represented*.

Knowledge is therefore a static entity; it may include facts, axioms, derivations, causal relations, mathematical models, etc. Knowledge may be measured directly. There is also a definition of knowledge as a dynamic entity:

FUNCTIONAL DEFINITION: Knowledge has a purpose. Knowledge is *what a system has that allows it to attain goals*.

Thus, knowledge cannot be observed (or measured) directly, but rather, indirectly through observing the intelligent behavior of a system.

Doyle (1988) termed these definitions as: (1) *explicit knowledge* which is what is represented; and (2) *implicit knowledge* which is what can be deduced from explicit knowledge which, in turn, depend upon the inference mechanism. Doyle argued that adopting the explicit definition has several limitations: (1) it cannot explain actions that are not logical, default or nonmonotonic reasoning; (2) it cannot explain some psychological phenomena; and (3) it cannot handle inconsistencies that naturally arise, for example, in knowledge generated from several experts.

Defining knowledge in the structural way and considering the reasoning mechanisms that operate on it *is different* than defining knowledge as a functional entity. It is only with very simple knowledge representation schemes, small knowledge bases, and simple inference mechanisms that a prediction about the expected performance of knowledge will be reasonable based solely on structural inspection; and even then, it will only be a prediction. In more complex situations, (and in agreement with Doyle,) these predictions will fail. In general, it is undecidable to determine performance from structural knowledge even when considering the mechanisms that manipulate it. Finally, the distinction between structural knowledge plus reasoning mechanisms and functional knowledge is analogue to the distinction in science between a hypothesis and its testing — definitely two distinct concepts.

[7] It is interesting to recall a recent exchange in the KAW electronic mailing list about what knowledge is and whether it is even necessary to answer this question. This exchange demonstrated the confusion and differences in understanding the concept of knowledge.

3.1 Structural Definition

The structural definition has several appealing properties. The main one is knowledge sharing (and/or trading). If knowledge is what is represented, then knowledge can be abstracted from the system using it and shared with another reasoning mechanism. Two "quantities" of knowledge could be added to yield a larger knowledge base, or knowledge could be transferred between knowledge systems (Neches et al., 1991). A knowledge interface format such as KIF (Genesereth and Fikes, 1992) is expected to facilitate this transfer. Another appealing property of the structural definition is that it facilitates easy valuation performed by simply inspecting the declarative structure of knowledge (with or without considering the inference mechanisms). This is much cheaper than executing behavior assessment experiments.

There are some difficulties with the structural definition. It detaches knowledge from its method of acquisition, although almost by definition, the act of acquisition determines the meaning of knowledge. When knowledge acquisition terminates, some meaning may be unrecoverable. For example, in the process of knowledge acquisition, uncertainties regarding the knowledge are established and their manipulation is not axiomatized declaratively but embedded in the inference mechanism. Later it may be impossible to recover the sources or the exact meaning of the uncertainties even if they were axiomatized. To illustrate, it has been observed that if the uncertainty handling mechanisms are changed, the ability to use the knowledge or the performance may decrease (Shortliffe, 1976). This difficulty suggests that one cannot generate knowledge by one mechanism that employs certain probabilistic models and "plug" it into an expert system that uses different models and expect the system to function properly.

Another difficulty with the structural definition arises since even though this definition focuses on what is represented, the ultimate aim of knowledge sharing or exchange is the solution of complex problems. Therefore, it is acknowledged that knowledge has a purpose which is not tested and, at best, can only be hypothesized from the structural measure.

Last, it is unclear that knowledge can be plugged into a system and function well. Humans require education and learning to assimilate new knowledge rather than reading declarative structures. This was the idea behind developing GUIDON2 to use the knowledge of NEOMYCIN to teach students (Clancey, 1988) without letting the students look at the low level knowledge representation. Similarly, it may be required to train systems with new knowledge rather then plug it into their memory.

3.2 Functional Definition

Newell (1982) opposed to the structural view of knowledge. He argued that "*knowledge* is a distinct notion, with its own part to play in the nature of intelligence," independent of representation. In order to define knowledge, Newell defined the "principle of rationality: If an agent has knowledge that one of its actions will lead to one of its goals, then the agent will select that action." This principle governs the use of knowledge for making the appropriate actions. Knowledge is therefore defined as "whatever can be ascribed to an agent, such that its behavior can be computed according to the principle of rationality. [...] Knowledge is a competence-like notion, being a potential for generating action;"[8] therefore, knowledge manifest and should be evaluated functionally. This means that knowledge can be described in terms of its operation to satisfy the goals of the design system. If a system cannot use a piece of information in its reasoning, then the system *does not have* this knowledge. The relation between knowledge and representation is clear. First, "representations exist at the symbol level" and not at the knowledge level. Second, "knowledge serves as

[8] See Fetzer (Fetzer, 1990) (1990, p. 127-130) for a critic on this definition. In addition to this criticism, continental philosophy has significantly different ideas about the nature of knowledge, its interpretation, and understanding (Mueller-Vollmer, 1985). See also (Mallery et al., 1986), for a review of hermeneutics related to computer understanding.

the specification of what a symbol structure should be able to do."[9] Even though, the actual implementation of the symbol structure is only an approximation of the knowledge level.

4 VALUATION MEASURES

The valuation of knowledge depends on what we define as knowledge. For each of the definitions, two complementary measures, qualitative and quantitative, are proposed.

4.1 Structural Measure

The process of *structural* valuation is similar to a "brain surgery." Knowledge is measured based on its internal structure. Since a crucial aspect of knowledge is its use in performance tasks, the structural measure will probably involve *projecting* how the knowledge will perform in solving problems.

Qualitative measure. The qualitative structural measure (QLS) is based on the assumption that humans can understand declarative knowledge embedded in systems. The assumption is based on past practice with small expert systems having simple knowledge representation schemes such as traditional production rules. In rule-based systems (albeit without uncertainties) knowledge can be verified to be free of redundancy, conflicts, circularity, and incompleteness (Nazareth, 1989). Much work on the evaluation of (rule-based) expert systems deals with these kinds of verifications. Once problems are discovered, it is the task of the expert (or knowledge engineer) to assess the situation and address it. This assessment is done qualitatively.[10]

Similarly, in machine learning, the comprehensibility of knowledge generated is described as an important measure by some researchers (Michalski, 1986). This measure, however, has evolved from research on concept learning where the structure of knowledge is sufficiently simple and the task is a single-step classification that is easy to comprehend and evaluate, and it is often easy to predict the knowledge performance from its structure.

Contrary to small and simple systems, in large and complex systems, it is hardly possible to understand what is the role of a small piece of knowledge and envision its potential run-time interactions with additional knowledge. Thus, such systems are hard to validate structurally.

It will become clear from the example that it is difficult to define the QLS to have an ordinal scale so it can be used to compare between different knowledge contents. The straight forward solution of using expert or user ratings will probably not support a homomorphic measure.

Quantitative measure. The quantitative structural measure (QNS) quantifies the structure of knowledge. Viewing knowledge as information allows such quantification (e.g., (Shannon, 1948; Boulton and Wallace, 1973)). A heuristic measure for assessing the quality of a classification, which is relevant to the example described later, was proposed by Gluck and Corter (1985). A generalized form of this measure, called *knowledge utility* (*KU*) is used in the illustrative example (Section 5).

In the context of statistical decision theory, (and with several other assumptions,) one can attempt to quantify the value of knowledge. Skyrms (1990) defined knowledge as something that allows making informed

[9] The view of knowledge as a specification for its structural description was used, for example, by Levesque (1984) for building knowledge representation that can support certain functions; and by Kyburg (1988) to draw conclusions on how uncertainties should be represented to support decision making.

[10] Verification may fit well into the quantitative measure discussed next if it only deals with the detection of problems without their corrections.

decisions; he then used this definition to valuate knowledge as the increase in the expected utility of making a choice due to this knowledge. This valuation was possible due to the simple nature of the knowledge involved: a single piece of evidence.

A QNS is easy to apply with a knowledge representation formalism whose content could be quantified. Nevertheless, from the example it will be clear that it is difficult to create a meaningful quantified measure (i.e., a homomorphism). Compared to the QLS, it is even less clear what is the relation between this measure and the expected performance of a system.

4.2 Functional Measure

This measure valuates knowledge based on its performance in various tasks. Cohen and Howe (1989) discussed the evaluation of large artificial intelligence systems. They suggested several performance experiments that can be used for this purpose. They showed how performance evaluations guided their research through three generations of the system DOMINIC, which deals with routine design of mechanical devices (Howe et al., 1986).

More generally, Adelman (1991) classified performance experiments into three classes: (1) *experiments* which are suited for the early stages of system development and may generate fully reproducible results; (2) *quasi experiments* which are for the operational stage of systems and consists of fully controlled artificial studies; and (3) *case studies* which are opportunistic and wholly unconstrained to be used for operational system. The first two evaluations can support the definition of quantitative measures and the latter can support qualitative measures.

The focus of these studies were the illustration of the kinds of experiments that could be done for certain assessments and their reliability and validity properties. There are many other studies discussing various types of qualitative and quantitative performance measures (Gupta, 1993). Nevertheless, these studies, do not address the issues pertaining to measurement theory discussed here which govern the meaningfulness of measures.

Qualitative measure. The qualitative functional measure (QLF) can be viewed as “performing protocol analysis” (Ericsson and Simon, 1980) on a system for evaluating its knowledge. The system is used to solve a variety of representative problems, and its problem-solving behavior, including the intermediate results is coded and analyzed (Chandrasekaran, 1983). This is one of the important techniques for evaluating systems and can lead to significant insight about a system’s behavior. It can also tell whether a system’s performance is a success or merely a fluke (Pople, 1985).

As with the QLS, it is difficult to define a QLF to have an ordinal scale (i.e., homomorphic ordinal measure) so it can be used to compare between different levels of knowledge functionality.

Quantitative measure. The quantitative functional measure (QNF) is based on the performance of a system over many problems that span the range of problems the knowledge of the system is expected to solve. The performance of the system is compared with the solutions generated by human experts or normative theories such as decision theory or other acceptable solutions.

In general, the performance measurement of systems and their comparisons is not straight forward. In the context of performance evaluation of learning programs, Kononenko and Bratko (1991) suggested that one cause for this difficulty is that the types of answers from different programs are not exactly the same. In addition, two systems may not solve exactly the same problem.

Similar to the dependency of the qualitative measures on the particular system or knowledge measured, functionality measures are equally contextualized. For example, Gaines (1989) measured the quality of the initial state of knowledge of a learning program by counting the number of examples the system needed for training (in addition to its initial knowledge) to achieve a predetermined performance level. The initial knowledge was ordered on an intuitive ordinal measure and reflected issues most relevant to supervised concept learning systems. The performance measure roughly corresponded to the intuitive measure. Gaines suggested that a value measure could be defined based on the logarithm of the data used for training the system. Note that in this measurement, the issue of the binary operation was not considered.

Since the QNF summarizes the results in statistics, it tends to hide some details of the system's behavior. Many controlled studies are needed to uncover behavior patterns that can be identified (even though only qualitatively) by the QLF. This measure may be perceived as easy to construct; nevertheless, as we see from (Gaines, 1989) and from the example, creating a measure that is a homomorphism is not straight forward. Its creation is part of the process of understanding the performance of the system whose knowledge is valuated.

The QNF could be integrated with the previous measures as in the evaluation of the quality of concept descriptions created by learning (Bergadano et al., 1988). This evaluation included three measures: (1) accuracy (i.e., QNF) that tested knowledge on previous training examples; (2) comprehensibility (i.e., QLS) that was operationalized by a syntactic complexity measure thus really became a QNS; and (3) cost (i.e., QNS) that measured the storage and computation needed to manipulate the learned knowledge. These measures were ordered and assembled into one evaluation with a lexicografic evaluation function. Note that there was no reference to any binary operation on knowledge by these measures. In contrast, in the example that follows, the focus will be the development of homomorphic measures of knowledge.

4.3 Relation to other definitions of measures

The measures presented are not new. They have been mentioned elsewhere in different names. For example, the structural measure corresponds to static analysis of software (Howden, 1978) or domain validation of intelligent systems (Benbasat and Dhaliwal, 1989). The functional measure corresponds to dynamic analysis of software or procedural validation of systems. Also, the QLS, the QLF, and the QNF correspond to content validity, construct validity, and empirical validity of systems, respectively (Hollnagel, 1989).

Another similarity is between QNF, QLF, QNS, and QLS and a black-box, white-box, consistency, and completeness methods, respectively for testing expert systems (Kirani et al., 1992). For each of these methods there are several different strategies that can be used to execute it, each with its own advantages and disadvantages for particular purposes. The above is demonstrated through a comparative study of the methods.

There are many other existing measures or evaluation methods that resemble the measures we discussed. While their collection and analysis are important, they are beyond the scope of this paper.

5 BRIDGER

In order to ground the different valuation measures and their tradeoffs we illustrate them in a concrete valuation of the design knowledge accumulated within BRIDGER, a system developed to explore the potential of knowledge acquisition techniques for building a design assistant for the preliminary design of cable-stayed bridges.

The section starts by describing the bridge domain and then reviews the system's architecture and operation. Since the purpose of this paper is to focus on knowledge valuation and not on BRIDGER, only the necessary parts required for the demonstration are described.[11]

5.1 Domain of Cable-Stayed Bridges

Figure 1 describes the main components and dimensions of a cable-stayed bridge. It is composed of a superstructure and a substructure. The superstructure is composed of a deck, towers, and stays that are attached to the towers and support the deck. The figure shows some of the properties which are used to describe a cable-stayed bridge; additional properties include: SPAN-N, the number of spans of the bridge; DECK-A, DECK-MI, TOWER-A and TOWER-MI, the cross-sectional areas and moment of inertia of the deck and the tower, respectively; DECK-M, the material of the deck; and STAY-A, the cross-sectional area of the stays.

Put figure 1 about here

In a design scenario the requirements are expressed as a set of specification property-value pairs (e.g., the required length of the bridge). Design is then executed by making design choices as assignments of design properties (e.g., SPAN-M=500ft., STAY-N=20).

A rough illustration of the complexity of the domain can be conveyed by the number of properties used to describe various aspects of the problem: 9 properties describe a specification; 30 properties describe a design, 15 properties describe the analysis results, and 4 properties describe the evaluation of a bridge.

5.2 BRIDGER's Architecture

BRIDGER's architecture, shown in Figure 2, consists of two main systems: synthesis and redesign. The synthesis system is responsible for synthesizing several candidates from a given specification. Synthesis knowledge is generated by learning from existing designs and from successful design examples that are selected by the user.

Put figure 2 about here

Candidate designs are transferred to a module that analyzes them based on US code for bridge design and submits them to a redesign module, if necessary. The redesign module retrieves the best design modifications for the bridge. The user can override the redesign modifications and supply explanations that enhance redesign knowledge. The results of the redesign system are acceptable designs. The designer evaluates the results and can submit a subset of them to the synthesis system for further training.

ECOBWEB, an enhanced version of COBWEB (Fisher, 1987), is the learning program that implements the synthesis system. It acquires synthesis knowledge and uses it to synthesize new bridges. ECOBWEB represents knowledge in a classification hierarchy. It has several operators that build the classification from examples. Learning and synthesis progress by one-step look-ahead search in the space of classification hierarchies directed by an evaluation function to select the best operator.

[11] See (Reich, 1991a; Reich, 1991b; Reich, 1993; Reich and Fenves, 1995) for further details on BRIDGER.

The evaluation function, called *category utility* (*CU*) (Gluck and Corter, 1985), evaluates a classification of a set of designs into mutually-exclusive classes $C_1, C_2, \ldots, C_n$, by:

$$CU = \frac{\sum_{k=1}^{n} P(C_k) \sum_i \sum_j P(A_i = V_{ij}|C_k)^2 - \sum_i \sum_j P(A_i = V_{ij})^2}{n} \tag{7}$$

where C_k is a class, $A_i = V_{ij}$ is a property-value pair, $P(x)$ is the probability of x, and n is the number of classes. The first term in the numerator measures the expected number of property-value pairs that can be guessed correctly by using the classification. The second term measures the same quantity *without* using the classes. Thus, the category utility measures the expected *increase* of property-value pairs that can be guessed *above* the guess based on frequency alone. The measurement is normalized with respect to the number of classes. The higher is the value of *CU*, the better the quality of the classification is.

BRIDGER has a variety of synthesis strategies, ranging from case-based to prototype-based design and from extensional to intentional strategies. To simplify the discussion we use the simplest strategy: an extensional case-based strategy. In this approach, BRIDGER *retrieves* a pre-determined number of candidate designs from the classification hierarchy. The candidates are complete descriptions of previously designed bridges. BRIDGER then *adapts* these candidates to fit the new specification by performing various scaling operations.

6 EXAMPLE

This section describes a detailed valuation of BRIDGER's design knowledge as it develops through learning. Learning allows to make observations about preserving the binary operations in equation 1 in a natural manner. Four knowledge hierarchies, K_1, K_2, K_3, and K_4, were generated by learning. Hierarchy K_1 was generated from a set of 96 bridge examples that were compiled from the (partial) description of existing bridges around the world. Hierarchy K_2 was generated from the 96 examples after their completion, analysis and redesign to satisfy the US code for bridge design. Therefore it contains higher quality and uniform examples. Hierarchies K_3 and K_4 were generated from K_2 by training it with 48 and 96 good quality artificial descriptions of bridges, respectively.

We hypothesize that a reasonable "unit" of knowledge will be one training example that is used by ECOBWEB. Each additional training example modifies the knowledge and the system's subsequent performance. Thus, this unit is a theory-laden choice natural to ECOBWEB, but one that may not make sense for other systems. We will test the hypothesis and subsequently modify it.

In the terminology of measurement theory the empirical relational system is:

$$\begin{array}{lll} \mathcal{A} & = & \{A, R, O\}\ , \text{ where} \\ A & & \text{is the set of knowledge hierarchies } K_i, \\ R & & \text{is the } \textit{better} - \textit{quality} - \textit{than} \text{ relation on } A, \text{ and} \\ O & & \text{is the } \textit{appending} - \textit{hierarchies} \text{ operation on } A. \end{array} \tag{8}$$

There are some assumptions underlying this definition and thus, this empirical system is only approximate. First, the knowledge quality of BRIDGER depends heavily on the quality of its training examples. Thus, we have to control the uniformity of the quality of the training examples; in what follows, they are designs that satisfy bridge design codes but, nonetheless, are not optimal. Also, since hierarchy K_1 is equal to K_2 in terms of the number of examples, but was trained with poor quality examples, we cannot treat it as part of the empirical system. Nevertheless, for reference purposes we included its valuation as well.

Second, the relation *better-quality-than* remains undefined. It may denote many intuitive and subjective direct measures of knowledge. Thus, we have some degrees of freedom when we attempt to construct an indirect measure that will fit it.

Third, the binary operation *appending-hierarchies* can be defined in several ways. One definition takes one hierarchy and trains it by the examples used to generate the second hierarchy. This is an order dependent procedure that is undesirable because the operation will not be commutative or worse, non-unique. Another definition creates a new root and attaches to it the two hierarchies. This definition will fail to support a scale, unless gradually, the split and merge operators of ECOBWEB will re-structure the new hierarchy.

Fourth and related to the previous item, we note that for the same set of examples we can create many different hierarchies since ECOBWEB's learning is order dependent. Therefore, the number of examples in a hierarchy does not determine any of the measures uniquely; moreover, there may be significant variations between the values of the measures even though they can partially be mitigated by some techniques (Reich, 1991a).

The formal relational system and the homomorphic mapping remain to be defined for each of the four measures we discuss next.

6.1 Structural Measure

The structural content of knowledge was evaluated in two ways: qualitative and quantitative. The qualitative measure attempts to analyze the knowledge generated from a domain perspective; and the quantitative measure is based on the evaluation function employed by ECOBWEB for assessing classifications.

Qualitative measure (QLS). Figure 3 shows the K_2 synthesis hierarchy generated from 96 examples of bridges. The classes are described with some of their properties. Some properties are shown in bold font; these are the characteristic properties. Intuitively, characteristic property values of a class are those property values that are very common in the class and rarely appear in the other classes of the same level. The figure also shows the name of each class and in parenthesis the number of bridges used to generate it.

Put figure 3 about here

The hierarchy is subdivided into two large subclasses: class C which contains long bridges (i.e., long LENGTH and SPAN-M properties) with many stays, and class B which contains short bridges with fewer stays. Further subdivisions mainly reflect differences in the LENGTH, CROSS-L, SPAN-M, SPAN-N, and DECK-M properties. Several patterns emerge in the hierarchy. They can be interpreted using domain knowledge, and may point to some design heuristics. The number of examples, however, is not sufficient to allow learning to discover strong patterns; any explanation should cautiously be accepted.

For example, class B contains only bridges with steel decks and class C contains 44% concrete-deck bridges; in addition, the average main span of class B is shorter than that of class C. These trends point to a preference for using concrete for longer bridges and steel for shorter bridges. The first is correct, but the second is not.[12] A close look at the subclasses of C shows that H and I, which contain only steel bridges, have longer average main span than the two other classes, containing mainly concrete-deck bridges. Therefore, the preferences stated before no longer apply. The conclusion is that the average main span value of C is only a common value of the class but does not necessarily provide a good characterization of the class.

This example demonstrates the subjective and imprecise nature of a qualitative inspection of knowledge. It is unclear which aspects in the class description are important and how they should be interpreted. It is

[12] See Table 3.3 in (Podolny and Scalzi, 1986) showing lower bids for concrete bridges as opposed to steel bridges for several recent long bridges. Also see Figure 3.8 of that reference showing that concrete is preferred to steel for short spans.

unclear how the *value* of the measure is to be defined. Since the valuation depends on personal inspection, we may try to operationalize it by experts ratings of the 4 knowledge hierarchies through some controlled experiment, even though we are now aware of its limitations. In this case, the formal system will include a set of some ordinal values such as *poor*, *modest*, *good*, and *excellent*; and the relation *better-quality-than* on this set will be the experts intuitive ordering between them. Unfortunately, we can hardly think of a reasonable binary operation on these values that will reflect the *appending-hierarchies* binary operation on knowledge hierarchies. A binary operation in which, for example, *poor* "+" *modest* "= " *good* will be very poor. In general, when experts ratings are used, the binary operation issue is neglected.

The best that we can do with the binary operation is salvage some information by inspecting a hierarchy as it grows by training (as a special case of applying the binary operation). Such inspection may potentially explain some of the behavior revealed by the other measures. Figure 4 illustrates the growth pattern of the synthesis hierarchy. It reflects the organization of knowledge rather than its content, but for uniform, good quality, examples, the organization may be indicative of the content. Initially, the hierarchy is "flat," consisting of the root node and its leaves. When additional information is accumulated, a second level starts to grow. Approximately twice the number of examples is required to form that second level. This pattern of growth continues later.

Put figure 4 about here

Given this, the design performance (i.e., quality and time) is not expected to improve continuously, but rather in stages. To illustrate (see Figure 4), assume that a design is initiated with hierarchy (a) and that the best candidate is class C_2. If BRIDGER is asked to synthesize *n* candidates, it will consider all the sons of C_1 and output the *n* best matches to the new specification. After additional training, hierarchy (b) is generated and used for the same design; synthesis progresses from C_3 to C_4, and finally, to C_5. Now, synthesis chooses the best *n* sons of class C_4 as candidate designs. The sons of C_4 form a more homogeneous class than the sons of C_1, but this has required doubling the number of training examples. Additional training leads to the generation of of hierarchy (c). If synthesis progresses through the path C_6, C_7, C_{10}, and C_{11}, then candidates are generated from the sons of C_{10} (*n* out of the 6, assuming than $n \leq 6$). If the path ends at class C_9, the candidates are generated from the sons of C_7 (*n* out of the 20, assuming than $n > 2$). The first case will demonstrate an overall performance improvement, but the second will show similar performance quality and degraded time performance to that demonstrated by hierarchy (b). These differences suggest that learning is not continuous, although it may seem so when performance is averaged over many cases.

This valuation suggests that the number of training examples in a hierarchy may not be an adequate measure for knowledge value and that the depth of the hierarchy may.

Summary of QLS.
This measure is subjective; it can provide qualitative insight about the behavior of knowledge mainly, since the internal mechanisms of the system were known to the evaluator. However, in general, the internal mechanisms of a system are unknown or too complex and it may be hard to extrapolate the functionality of knowledge from this valuation. Due to these properties, a coarse scale could be devised based on subjective experts ratings of knowledge hierarchies. This scale will probably ignore the binary operation of *appending-knowledge-hierarchies*.

Quantitative measure (QNS). In order to quantify knowledge, we can attempt to construct a formal system as follows: The set of entities is the real numbers; the relation is the *larger-than*; and the binary operation is the *addition* of numbers. Knowing the internal mechanisms of ECOBWEB — the system that created the knowledge — we seek a measure that will reflect them.

We define the quantity of knowledge by a measure, called *knowledge utility (KU)*, that estimates the *increase* in the number of properties that can be predicted for a given specification when using the *knowledge hierarchy*, relative to the number of properties that can be predicted by using values' frequency. We recall that the category utility function (*CU*) which governs part of ECOBWEB's behavior, estimates that increase for a *classification* but not for a *hierarchy*. Therefore, we apply *CU* recursively, starting from the root of the hierarchy, and obtain the following measure:

$$\begin{aligned} &knowledge-utility\ (class): \\ &\quad if\ \ class\ is\ a\ class,\ \ return\ 0.0; \\ &\quad else,\ \ return\ \ CU \times n + \sum_{C_k \in sons\ of\ \ class} P(C_k) \times knowledge-utility\ (C_k), \end{aligned} \tag{9}$$

where *CU* and the other symbols are defined as in equation 7. After the calculation, the value is normalized by the number of properties describing artifacts.

Figure 5 shows the knowledge utility as a function of the examples learned by ECOBWEB. The value of about 0.1 reached after learning 192 examples suggests that approximately 8 out of the 58 properties describing designs can be predicted accurately. This may seem to be a rather disappointing result since it is difficult to envision that a knowledge with such a low utility can be helpful.

Put figure 5 about here

It is clear that *KU* does not preserve *any* intuitive *better-quality-than* relation since it is not a monotonic function of the number of examples within a hierarchy. Moreover, we will see later that poor *KU* values are not good predictors of the good performance we observe later. Thus, *KU* is a poor measure of knowledge even though it was intuitively good.

Summary of QNS.
This measure is abstracted from the mechanisms that manipulate knowledge; even though, in our example it relies on *CU* which certainly governs parts of the system's behavior. The bad values obtained by this measure are in contrast to the good performance reported later. This discrepancy illustrates the difficulty in formulating good QNS that can be used to predict performance. Whereas in the QLS we could have suggested experts ratings as a qualitative measure, we cannot do this with the quantitative one.

It is extremely hard to find a QNS that will be a homomorphism. We predict that basing QNSs for other knowledge representations on counting items such as rules, frames, conditions per rule, etc., will result in similar poor measures.

6.2 Functional Measure

The functional value of knowledge was measured by evaluating BRIDGER's performance in synthesis activities performed on 48 test specifications (see (Reich, 1991a) for details). The four knowledge hierarchies, K_1, K_2, K_3, and K_4, were used to synthesize 4 new candidates for each specification. The 192 (48×4) synthesized bridges were used in the qualitative and quantitative functional evaluations.

Qualitative measure (QLF). Instead of analyzing the complete trace of synthesis, we focus the valuation on one important synthesis step: the retrieval of candidate designs. Table 1 shows the number of different existing designs retrieved, and the names of the designs most commonly used. The number of times each existing design was used in the generation of candidates, out of the 192 new candidates, is given below each name. The small number of designs retrieved reflects an internal tendency that characterizes the knowledge. In light of the discussion on Figure 4, it is not surprising that the structure of the hierarchy can lead to such behavior at the early stages of learning.

Put table 1 about here

The difference in the number of designs retrieved suggests a synthesis pattern similar to that presented in Figure 4. In particular, the synthesis pattern emerging from the K_1 and the K_2 hierarchies is probably similar to the path C_6, C_7, C_8, and C_9. Such a path forces the retrieval of designs from classes of designs higher in the hierarchy. Since a large class is used as a source of existing designs, the selection would usually favor a small number of 'strong' matches. If the path is from C_6 to C_{11}, the selection would be from smaller and different groups of classes, leading to the retrieval of a larger number of distinct designs. This is the case when K_3 and K_4 are used.

Figure 6 shows the four designs most often used when synthesizing with the K_2 hierarchy. These designs are listed in Table 1. The two designs on the right are scaled down by a factor of two. All four designs are two-span bridges with average main span (224 m). The range of spans is large, allowing the retrieval of designs that are relevant to a new specification therefore do not require significant scaling.

Put figure 6 about here

Figure 7 shows the 12 designs most often used when synthesizing with the K_3 hierarchy. Most of them are three-span bridges. The average length of the main span is 179 m. A surprising observation is that most of the bridges have a small number of stays. This fact and the observation that almost no three-span bridges were used by the K_2 hierarchy point to the existence of a *shadowing* phenomenon. Certain bridges are not retrieved since they reside on hierarchy branches that are rarely visited. But once these branches become accessible, their leaves start being used as candidates.

Put figure 7 about here

Figure 8 shows the 8 designs most often used when synthesizing with the K_4 hierarchy. There is a better balance between two- and three-span bridges, and more variation in the number of stays. The length of the main span of these designs is longer than before (345 m) and its variability is slightly less then that observed for the K_2 hierarchy. The increasing average length helps design large bridges without compromising the design of bridges with small spans.

Put figure 8 about here

As with the QLS, it is hard to create a metric from the above discussion. As before, experts ratings could be used for creating an ordinal scale that will most probably neglect the binary operation.

Summary of QLF.
This measure can be used to estimate the QNF by generalizing over system's behavior patterns. It can also be used to confirm the QLS. A more detailed QLF can point to important issues that need to be addressed in improving the system; for example, should the *shadowing* effect discussed before and confirmed here be handled?

Quantitative measure (QNF). This measure valuates the performance of BRIDGER while designing candidates for the 48 test specifications. It measures the two main sources of power of BRIDGER's synthesis process: (1) the retrieval of designs closely related to the new specification; and (2) the adaptation of candidates with scaling values.

The retrieval process is evaluated by the amount of *scaling* of the main span of the retrieved design needed to satisfy the new specification; it measures how close is the retrieved design to fulfilling the dimensional specification which is the most significant controlling parameter over the design. The candidate adaptation process cannot be tested independently. The combination of the two processes is tested by measuring the *quality* of candidate designs which is a weighted summation of the constraints that a design violates (Reich, 1991a).

The formal system for this measure is created with the set of real numbers that denote either the scaling or quality values; the relation is the *larger-than*; and the binary operation is the arithmetic *addition*. The homomorphism maps a hierarchy to its performance values.

Table 2 provides the statistics of the *scaling* needed to adapt the candidates to the specifications of the 48 test problems and the *quality* of the designs synthesized. The columns denoted by *total* provide the average of these measures. The columns denoted by *lower*, *average*, and *upper*, provide the results for three groups of specifications corresponding to far-lower-than, similar-to, or far-higher-than, the average specification of bridges in the training examples. These groups roughly divide the set of 48 test problems into three equal parts.

Put table 2 about here

A MANOVA (Hays, 1988) analysis was performed to assess the statistical significance of the differences in the performance levels observed. This is performed to assess whether the measure fits our intuition about the quality of knowledge in the observed system. The total scaling values satisfy: $K_2, K_3 >_{0.01} K_4$;[13] where the $>_{0.01}$ indicates that the scaling values of K_2 and K_3 are greater (greater are worse) than K_4 with statistical significance at the $p < 0.01$ level and that the difference between K_2 and K_3 was not statistically significant. Therefore, the improvement is not a smooth function, but occurs in steps as predicted by the QLS. The total quality values satisfy: $K_2 >_{0.01} K_3, K_4$. The group of specification also influences the results. The scaling values satisfy: $lower <_{0.01} average <_{0.01} upper$, whereas the quality values satisfy: $lower, average <_{0.01} upper$. This is in agreement with a known engineering heuristic stating that it is relatively easy to design artifacts that are similar to past experience or slightly scaled down and harder if designs are to be scaled up.

In terms of measurement theory, the fact that some measures were not different in a statistically significant

[13] As mentioned before, the results of K_1 are not analyzed.

manner suggests that the mapping between the observed and the formal system is inadequate since the major differences in the hierarchies (i.e., different number of examples) did not result in different values of performance. This suggests that performance (or the value of knowledge) is not proportional to the size of the hierarchies or the number of examples used to generate them.

If we summarize results from previous measures we note that the two qualitative measures suggested that it is the depth of hierarchy, rather than the number of examples, that may govern the system performance. If we recall the power law of practice, where there is a power (or a log-log) relationship between training and performance (Newell and Rosenbloom, 1981), we may consider that performance (or the value of knowledge) varies as a power function of the number of examples. Or that the logarithm of performance varies as a logarithmic function of the number of examples or as a linear function of the depth of a hierarchy.

Another MANOVA analysis was run with this model. The results of the *scaling* remained as before, but, the results of the *quality* were more conclusive: $K_2 >_{0.01} K_3 >_{0.01} K_4$ and $lower <_{0.01} average <_{0.01} upper$. Therefore, we may conclude from this analysis and the supporting evidence from previous measures that a power law model better explains the data than the linear model.

Summary of QNF.
This measure is the most precise of all measures, but it also depends on the mapping between the observed and the formal systems used to define it. Therefore, the meaning of the statistical results is also subjective. The measure used in the example is system and domain dependent; other systems or domains will lead to the creation of different measures. This exercise further demonstrates: (1) the difficulty in creating accurate quantitative measures; and (2) the need to test quantitative measures and contrast their valuation with other measures.

7 DISCUSSION AND SUMMARY

Starting from a review of key concepts in measurement theory, we gave two definitions of knowledge, each leading to two types of value measures. The four measures were demonstrated in the evaluation of the design system BRIDGER. We saw that none of the measures was perfect, but that together they supported each other in providing a better understanding of the system behavior and the relation between its behavior and its knowledge. This understanding led us to refining a QNF measure of knowledge (for BRIDGER) that looks meaningful: the value of BRIDGER's knowledge varies as a logarithmic function of the number of examples or as a linear function of the depth of its knowledge hierarchy.

We would like to suggest that a measure based on the logarithm of data, e.g., logarithm of the number of rules in a rule-based system or the number of problem spaces (or rule clusters) rather the number of production rules in an architecture such as Soar (Laird et al., 1987), may be more universal than was demonstrated. Nevertheless, this is just a hypothesis.[14]

Table 3 summarizes the different measures used in the example according to the terminology of measurement theory. A question mark denotes an unspecified entry. The only measure that managed to preserve roughly the binary operation of *appending-hierarchies* was the QNF measure, and only after is benefited from insight from the two qualitative measures. The two qualitative measures were totally unconstrained in their textual output and the QNS failed.

Put table 3 about here

[14] Gaines (1989) also suggested it but without providing a rationale.

We note that the valuation context as determined by a system's characteristics and the domain of interest has a major influence over the types of measures that may be used for knowledge valuation. There are no objective or truly system independent measures. For example, accuracy or completeness are operationalized by testings that depend on the system's characteristics and the domain of interest. If so, what can be transferred from this exercise to other systems or domains? Were the measures used in this exercise *ad hoc*?

Since, none but the QNF succeeded, even if roughly, to be a homomorphism, we need only discuss it. The first unit of knowledge posited, i.e., one training example, was *ad hoc* and proved to be wrong. The depth of the hierarchy (or the logarithm of the number of examples) was better as a unit of knowledge, but it is only the corroboration of this hypothesis with data from the other qualitative measures that allowed us to understand *why* this unit worked. Due to this understanding we can claim that this measure is not *ad hoc*.

None of the measures can be transferred as is to other systems or domains. Therefore, how does one construct measures for different contexts? The key lies in understanding the *process* of measure construction. It is an iterative process of hypothesizing several measures and testing them, during which a better understanding of the system's knowledge, mechanisms, and behavior emerges. While some measures may be fruitless, others may together provide enough data to assist in the specification of better measures that should be tested again.

The kind of measure hypothesized or model posited (e.g., knowledge utility or power law of practice) has a significant impact on the success, failure, or meaningfulness of the valuation, or on the number of iterations required to achieve some interesting results. Often, finding good models is *the* significant research problem. When a hypothesized measure works as a homomorphism, we say that it provides a representation of the observed relational system (Roberts, 1979). We can appreciate the difficulty of constructing such a representation given the stringent conditions it needs to satisfy (i.e., equations 2 and 3). We can also agree that it is not only the measurement of knowledge as an end that is important but the construction of measures is equally critical for understanding the system whose knowledge is being valuated.

Even though the measures are context-dependent, there is still some insight that can be learned and may be transferred to other contexts. Table 4 summarizes some general observations about the 4 measures. The QNS is mainly driven by the representation formalism, it is the most difficult to construct and will probably not be cost-effective to develop. The qualitative measures require domain expertise for understanding the knowledge represented or the system's solution traces. While the QLF can be used if such traces are provides, the QLS can be used only on small chunks of knowledge. Occationally, once problems or issues are uncovered by the QLF, the QLS can address them with a narrow focused valuation.

In the example, the qualitative measures were summarized by text and some numeric data. This does not suffice for defining a scale. Nevertheless, they can be turned into an ordinal scale by devising questionnaires and having several experts relatively rate the systems under investigation. Such ratings could be used to create an ordinal scale. Different questionnaires could be developed for different experts such as system maintainers or expert users, and give rise to values that depend on the task of these experts in relation to the systems (see (McGraw and Harbison-Briggs, 1989) for more details).

Put table 4 about here

Future work includes several important tasks. The first task is the collection of data on evaluations of additional systems or studies comparing between different evaluation methods. This data will discuss different ways to operationalize vague criteria such as "appeal: usability; how well the knowledge base matches our intuition and stimulates thought; ..." (Marcot, 1987, p. 442); and it will address the relative utility of different methods for different evaluation purposes as well as their failures. This data will assist the future contextualized selection of measures and will lead to a better understanding of the issues underlying the valuation of knowledge.

Another task deals with the formation of guidelines for system development that will facilitate appropriate valuation. It is best if a system allows for all four types of valuations to be performed. Therefore, detailed traces must be provided on demand and simple representation formalisms are preferred to more complex ones.

Finally, research should address the valuation of knowledge embedded in a decision-support setting where a human expert or a user is cooperating with its computer assistant. After all, is it the improved performance of this "team" that was the motivating goal for developing the system.

ACKNOWLEDGMENTS

This work was supported in part by the Engineering Design Research Center, a National Science Foundation Engineering Research Center, Carnegie Mellon University.

REFERENCES

Adelman, L. (1989). Measurement issues in knowledge engineering. *IEEE Transactions on Systems, Man, and Cybernetics*, 19(3):483–488.

Adelman, L. (1991). Experiments, quasi-experiments, and case studies: A review of empirical methods for evaluating decision support systems. *IEEE Transactions on Systems, Man, and Cybernetics*, 21(2):293–301.

Benbasat, I. and Dhaliwal, J. (1989). A framework for the validation of knowledge acquisition. *Knowledge Acquisition*, 1(2):215–233.

Bergadano, F., Matwin, S., Michalski, R. S., and Zhang, J. (1988). Measuring quality of concept descriptions. In Sleeman, D., editor, *Procedings of the Third European Working Session*, pages 1–14, Aberdeen. Pitman.

Boulton, D. M. and Wallace, C. S. (1973). An information measure for hierarchical classification. *The Computer Journal*, 16(3):254–261.

Chandrasekaran, B. (1983). On evaluating AI systems for medical diagnosis. *AI Magazine*, 4(2):34–37.

Clancey, W. J. (1988). Acquiring, representing, and evaluating a competence model of diagnostic strategy. In Chi, M. T. H., Glaser, R., and Farr, M. J., editors, *The Nature of Expertise*, pages 345–420. Lawrence Erlgaum Associates, Hillsdale, N.J.

Cohen, P. R. and Howe, A. E. (1989). Toward AI research methodology: Three case studies in evaluation. *IEEE Transactions on Systems, Man, and Cybernetics*, SMC-19(3):634–646.

Doyle, J. (1988). Implicit knowledge and rational representation. Technical Report CMU-CS-88-134, Computer Science Department, Carnegie Mellon University, Pittsburgh, PA.

Ericsson, K. A. and Simon, H. A. (1980). Verbal reports as data. *Psychological Review*, 87(3):215–251.

Fetzer, J. H. (1990). *Artificial Intelligence: Its Scope and Limits*. Kluwer Academic Publishers, Dordrecht, The Netherlands.

Fisher, D. H. (1987). Knowledge acquisition via incremental conceptual clustering. *Machine Learning*, 2(7):139–172.

Gaines, B. R. (1989). The quantification of knowledge–formal foundation for acquisition methodologies. In Ras, Z. W., editor, *Methodologies for Intelligent Systems, 4*, pages 137–149, New York. North-Holland.

Gaines, B. R. and Shaw, M. L. G. (1989). Comparing the conceptual systems of experts. In *Proceedings of The Eleventh International Joint Conference on Artificial Intelligence*, pages 633–638, Detroit, MI. Morgan Kaufmann.

Genesereth, M. R. and Fikes, R. E. (1992). Knowledge interchange format, version 3.0 reference manual. Technical Report Logic-92-1, Department of Computer Science, Stanford University, Stanford, CA.

Gluck, M. and Corter, J. (1985). Information, uncertainty, and the utility of categories. In *Proceedings of the Seventh Annual Conference of the Cognitive Science Society, Irvine, CA*, pages 283–287, San Mateo, CA. Academic Press.

Green, C. J. and Keyes, M. M. (1987). Verification snd validation of expert systems. In *Proceedings of the Western Conference on Expert Systems (WESTEX-87)*, pages 38–43, Los Alamitos, CA. IEEE Computer Society Press.

Guida, G. and Mauri, G. (1993). Evaluating performance and quality of knowledge-base systems: foundation and methodology. *IEEE Transactions on Knowledge and Data Engineering*, 5(2):204–224.

Gupta, U. G. (1993). Validation and verification of knowledge-based systems: A survey. *Journal of Applied Intelligence*, 3(4):343–363.

Hays, W. L. (1988). *Statistics. Fourth edition*. Holt, Rinehart & Winston, New York.

Hollnagel, E. (1989). Issues in the reliability of expert systems. In Hollnagel, E., editor, *The Reliability of Expert Systems*, pages 303–329, New York, NY. Wiley.

Howden, W. E. (1978). Introduction to software validation. In Miller, E. and Howden, W. E., editors, *Tutorial: Software Testing and Validation Techniques*, pages 1–2, New York, NY. IEEE Computer Society Press.

Howe, A., Dixon, J., P., C., and M., S. (1986). Dominic: A domain-independent program for mechanical engineering design. *International Journal For Artificial intelligence in Engineering*, 1(1):23–29.

Kaplan, B. and Duchon, D. (1988). Combining qualitative and quantitative methods in information systems research: a case study. *MIS Quarterly*, 12(4):571–586.

Kirani, S., Zualkernan, I. A., and Tsai, W. T. (1992). Comparative evaluation of expert system testing methods. In *Proceedings of the 1992 IEEE International Conference on Tools with AI (Arlington, VA)*, pages 334–341, Washington, DC. IEEE Computer Society Press.

Kononenko, I. and Bratko, I. (1991). Information-based evaluation criterion for classifier's performance. *Machine Learning*, 6(1):67–80.

Kyburg, H. E. J. (1984). *Theory and Measurement*. Cambridge University Press, Cambridge, UK.

Kyburg, H. E. J. (1988). Knowledge. In Lemmer, J. F. and Kanal, L. N., editors, *Uncertainty in Artificial Intelligence 2*, pages 263–272. North-Holland, Amsterdam.

Laird, J. E., Newell, A., and Rosenbloom, P. S. (1987). Soar: an architecture for general intelligence. *Artificial Intelligence*, 33(1):1–64.

Levesque, H. J. (1984). Foundations of a fucntional approach to knowledge representation. *Artificial Intelligence*, 23(2):155–212.

Linster, M., editor (1992). *Sisyphus '92: Models of problem solving*. Vol. 630 of Technical report of GMD. GMD, St. Augustin.

Mallery, J. C., Hurwitz, R., and Duffy, G. (1986). Hermeneutics: From textual explication to computer understanding. Technical Report A.I. Memo No. 871, Massachusetts Institute of Technology, Artificial Intelligence Laboratory, Cambridge, MA. Also apppeared in *The Encyclopedia of Artificial Intelligence*, John Wiley & Sonse, New York, 1987.

Marcot, B. (1987). Testing your knowledge base. *AI Expert*, 2:42–47.

McDermott, D. (1981). Artificial intelligence meets natural stupidity. In Haugeland, J., editor, *Mind Design*, pages 143–160, Cambridge, MA. MIT Press.

McGraw, K. L. and Harbison-Briggs, K. (1989). *Knowledge Acquisition Principles and Guidelines*. Prentice-Hall, Englewood Cliffs, N.J.

Michalski, R. S. (1986). Understanding the nature of learning: issues and research directions. In Michalski, R. S., Carbonell, J. G., and Mitchell, T. M., editors, *Machine Learning: An Artificial Intelligence Approach, Vol 2*, pages 3–41. Tioga Press, Palo Alto, CA.

Mowshowitz, A. (1994). Information as a commodity: assessment of market value. In Yovits, M. C., editor, *Advances in Computers, Vol. 38*, pages 247–316, London. Academic Press.

Mueller-Vollmer, K., editor (1985). *The Hermeneutics Reader*. Continuum, New York, NY.

Nazareth, D. L. (1989). Issues in the verification of knowledge in rule-based systems. *International Journal of Man-Machine Studies*, 30(3):255–271.

Neches, R., Fikes, R., Finin, T., Gruber, T., Patil, R., Senator, T., and Swartout, W. R. (1991). Enabling technology for knowledge sharing. *AI Magazing*, pages 37–56.

Newell, A. (1982). The knowledge level. *Artificial Intelligence*, 18(1):87–127.

Newell, A. and Rosenbloom, P. S. (1981). Mechanisms of skill acquisition and the power law of practice. In Anderson, J. R., editor, *Cognitive Skills and Their Acquisition*. Erlbaum Associates, Hillsdale, N.J.

Pfanzagl, J. (1971). *Theory of Measurement*. Physica-Verlag, Wurzburg, 2nd edition.

Podolny, W. and Scalzi, J. B. (1986). *Construction and Design of Cable-Stayed Bridges. Second edition*. John Wiley and Sons, New York.

Pople, H. (1985). Evolution of an expert system: From Internist to Caduceus. In De Lotto, I. and Stefanelli, M., editors, *AI in Medicine*, pages 179–208, Amsterdam. Elsevier.

Reich, Y. (1991a). *Building and Improving Design Systems: A Machine Learning Approach*. PhD thesis, Department of Civil Engineering, Carnegie Mellon University, Pittsburgh, PA. (Available as Technical Report EDRC 02-16-91).

Reich, Y. (1991b). Design knowledge acquisition: Task analysis and a partial implementation. *Knowledge Acquisition*, 3(3):237–254.

Reich, Y. (1993). A model of aesthetic judgment in design. *Artificial Intelligence in Engineering*, 8(2):141–153.

Reich, Y. and Fenves, S. J. (1995). A system that learns to design cable-stayed bridges. *Journal of Structural Engineering, ASCE*. (in press).

Roberts, F. S. (1979). *Measurement Theory with Applications to Decisionmaking, Utility, and the Social Sciences*. Encyclopedia of Mathematics and its Applications, Vol. 7. Addison Wesley, Reading, MA.

Shannon, C. E. (1948). A mathematical theory of communication. *The Bell System Technical Journal*, XXVII(3):379–423,623–656.

Shortliffe, E. H. (1976). *Computer-based Medical Cunsultation: MYCIN*. Elsevier, New York, NY.

Skyrms, B. (1990). The value of knowledge. In Savage, C. W., editor, *Minnesota Studies in the Philosophy of Science: Volume XIV: Scientific Theories*, pages 245–266. University of Minnesota Press, Minneapolis, MN.

Stevens, J. (1946). On the theory of scales and measurement. *Science*, 103:677–680.

Zuse, H. and Bollmann, P. (1987). Software metrics: Using measurement theory to describe the properties and scales of static software complexity metrics. Technical Report RC 13504, IBM Research Division, Yorktown Heights, N.Y.

Table captions

1. Summary of retrieved designs
2. Scaling and Quality statistics of candidates
3. Measures of design knowledge
4. Observed and formal systems

Table 1:

Knowledge	# of different designs	designs retrieved # of times retrieved (out of the 192)							
K_1	12	E10	E12	E49	E60				
		45	45	45	45				
K_2	8	E46	E55	E78	E91				
		43	43	43	43				
K_3	19	E2	E10	E32	E88	E19	E22		
		16	16	16	16	12	12		
		E24	E26	E29	E49	E115	E111		
		12	12	12	12	12	9		
K_4	19	E80	E144	E192	E3	E159	E162	E168	E135
		25	25	25	15	13	13	13	13

Table 2:

Knowledge	Scaling				Quality			
	lower	average	upper	total	lower	average	upper	total
K_1	1.25	3.58	5.85	3.074	1188.31	35.63	62.22	278.36
K_2	0.97	2.50	4.08	2.154	0.34	4.61	325.81	50.19
K_3	0.97	2.53	3.57	2.092	0.57	2.55	5.67	2.89
K_4	0.88	1.32	2.85	1.325	0.41	0.73	3.06	1.20

Table 3:

	Observed	Formal			
		Structural		**Functional**	
		Qualitative QLS	Quantitative QNS	Qualitative QLF	Quantitative QNF
Set	knowledge hierarchies	ordinal values	positive real numbers	ordinal values	positive real numbers
Relation	better-than	intuitive	$>$	intuitive	$>$
Operation	appending hierarchies	?	+	?	+
Mapping		expert inspection	KU	expert inspection	performance tests

Table 4:

		Structural		Functional	
		Qualitative QLS	Quantitative QNS	Qualitative QLF	Quantitative QNF
1	Require	domain expertise	precise formalism	domain expertise, solved test cases	domain expertise, solved test cases, normative theories
2	Apply on (after satisfying row 2)	simple representation, manageable size knowledge	simple representation	systems with detailed traces	everything including a black box
3	May be used to	approximately predict performance, uncover simple problems	scant relation to performance	predict performance, uncover behavior patterns	quantify performance
4	Summarized by	textual information	quantitative data	textual + quantitative information	concise quantitative data
5	Best possible scale	ordinal	interval	ordinal	interval
6	Method to construct best scale	expert ratings	system dependent (very hard)	expert ratings	performance tests (hard)

Figure captions

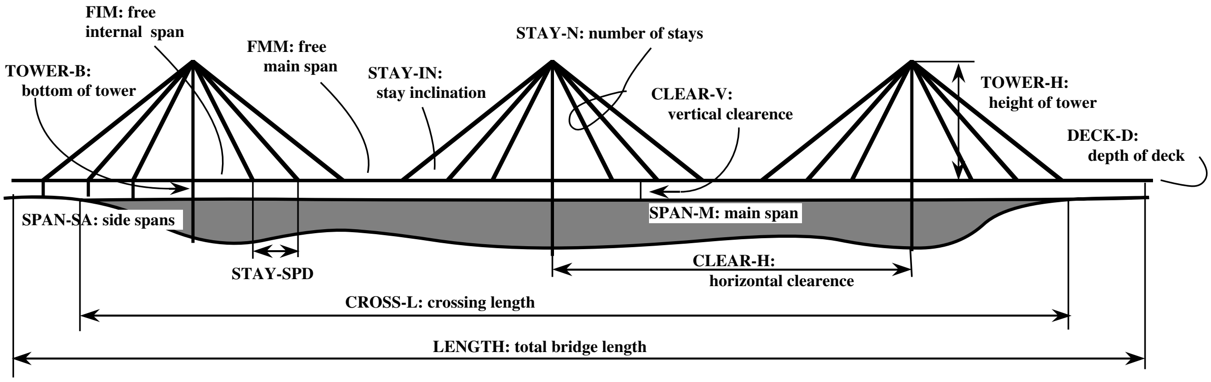
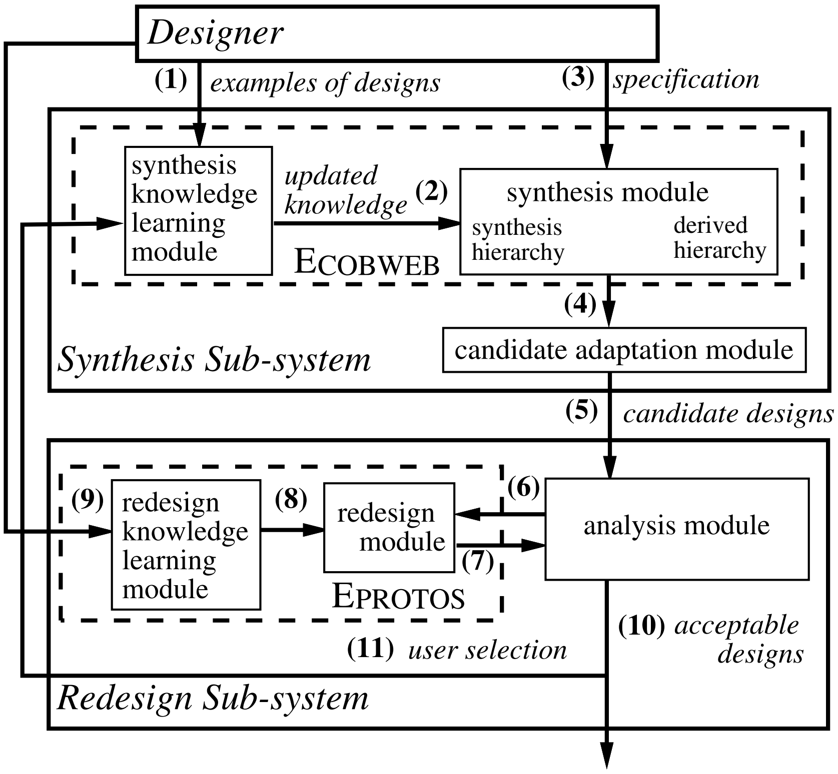

FIM: free
internal span
FMM: free
main span
TOWER-B:
bottom of tower
STAY-IN:
stay inclination
STAY-N: number of stays
CLEAR-V:
vertical clearence
TOWER-H:
height of tower
DECK-D:
depth of deck
SPAN-SA: side spans
SPAN-M: main span
STAY-SPD
CLEAR-H:
horizontal clearence
CROSS-L: crossing length
LENGTH: total bridge length

Figure 1:



Designer
(1) examples of designs
(3) specification
synthesis knowledge learning module
updated knowledge
(2)
synthesis module
synthesis hierarchy
derived hierarchy
ECOBWEB
(4)
candidate adaptation module
Synthesis Sub-system
(5) candidate designs
(9)
redesign knowledge learning module
(8)
redesign module
(6)
(7)
analysis module
EPROTOS
(10) acceptable designs
(11) user selection
Redesign Sub-system

Figure 2:

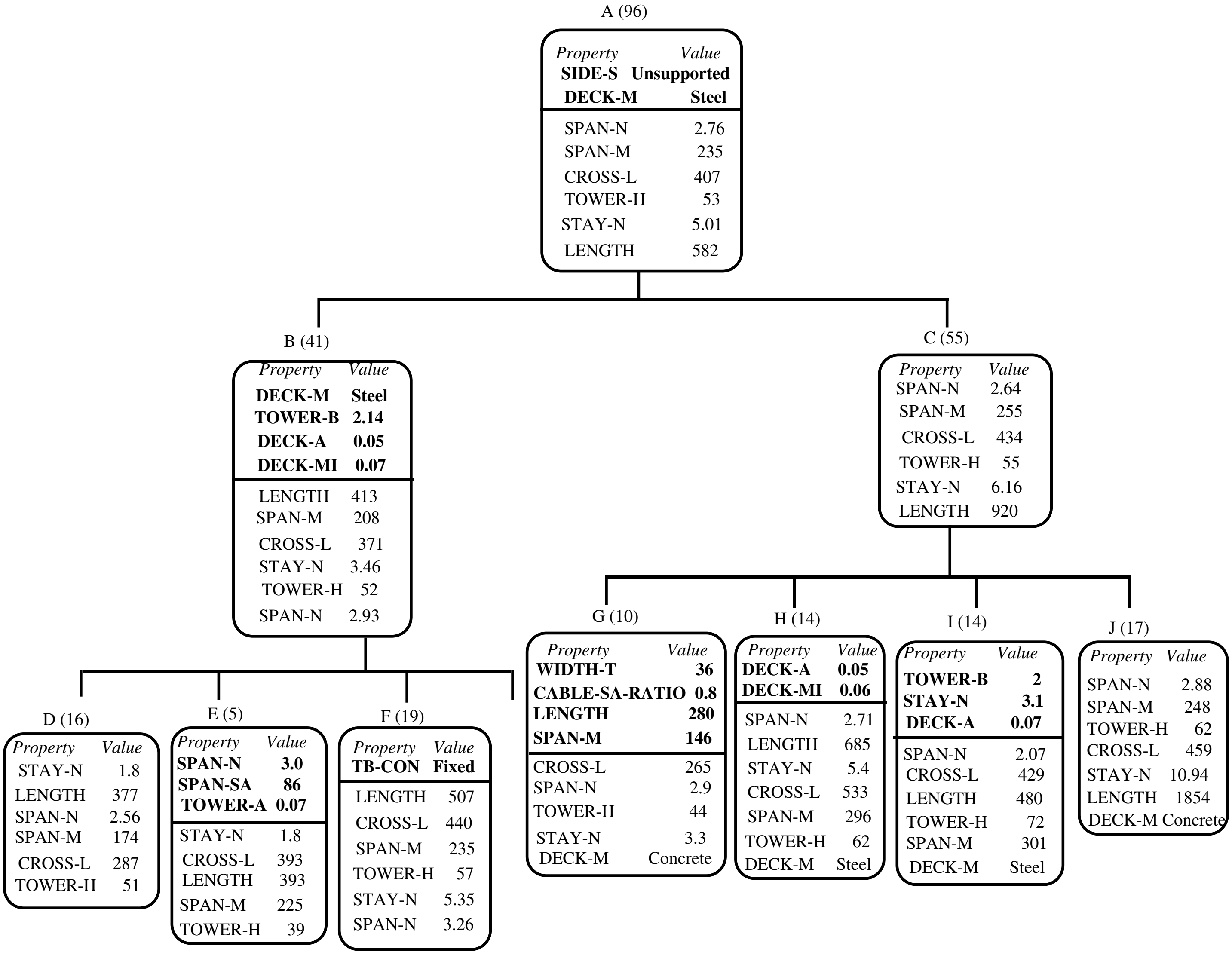
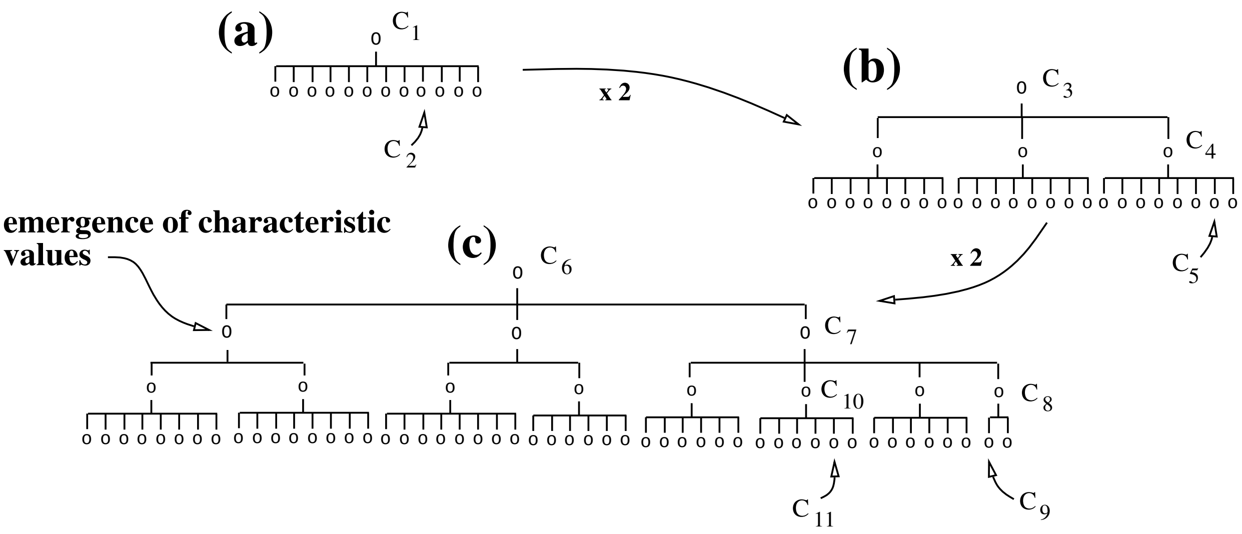

A (96)
Property Value
SIDE-S Unsupported
DECK-M Steel
SPAN-N 2.76
SPAN-M 235
CROSS-L 407
TOWER-H 53
STAY-N 5.01
LENGTH 582
B (41)
Property Value
DECK-M Steel
TOWER-B 2.14
DECK-A 0.05
DECK-MI 0.07
LENGTH 413
SPAN-M 208
CROSS-L 371
STAY-N 3.46
TOWER-H 52
SPAN-N 2.93
C (55)
Property Value
SPAN-N 2.64
SPAN-M 255
CROSS-L 434
TOWER-H 55
STAY-N 6.16
LENGTH 920
D (16)
Property Value
STAY-N 1.8
LENGTH 377
SPAN-N 2.56
SPAN-M 174
CROSS-L 287
TOWER-H 51
E (5)
Property Value
SPAN-N 3.0
SPAN-SA 86
TOWER-A 0.07
STAY-N 1.8
CROSS-L 393
LENGTH 393
SPAN-M 225
TOWER-H 39
F (19)
Property Value
TB-CON Fixed
LENGTH 507
CROSS-L 440
SPAN-M 235
TOWER-H 57
STAY-N 5.35
SPAN-N 3.26
G (10)
Property Value
WIDTH-T 36
CABLE-SA-RATIO 0.8
LENGTH 280
SPAN-M 146
CROSS-L 265
SPAN-N 2.9
TOWER-H 44
STAY-N 3.3
DECK-M Concrete
H (14)
Property Value
DECK-A 0.05
DECK-MI 0.06
SPAN-N 2.71
LENGTH 685
STAY-N 5.4
CROSS-L 533
SPAN-M 296
TOWER-H 62
DECK-M Steel
I (14)
Property Value
TOWER-B 2
STAY-N 3.1
DECK-A 0.07
SPAN-N 2.07
CROSS-L 429
LENGTH 480
TOWER-H 72
SPAN-M 301
DECK-M Steel
J (17)
Property Value
SPAN-N 2.88
SPAN-M 248
TOWER-H 62
CROSS-L 459
STAY-N 10.94
LENGTH 1854
DECK-M Concrete

Figure 3:



(a)
C1
C2
x 2
(b)
C3
C4
C5
x 2
emergence of characteristic values
(c)
C6
C7
C8
C9
C10
C11

Figure 4:

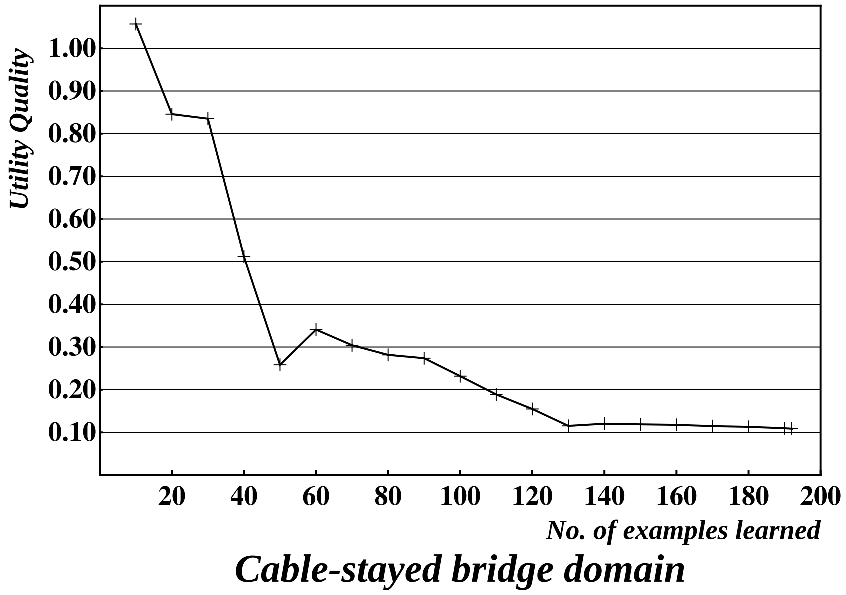

Utility Quality
1.00
0.90
0.80
0.70
0.60
0.50
0.40
0.30
0.20
0.10
20
40
60
80
100
120
140
160
180
200
No. of examples learned
Cable-stayed bridge domain

Figure 5:

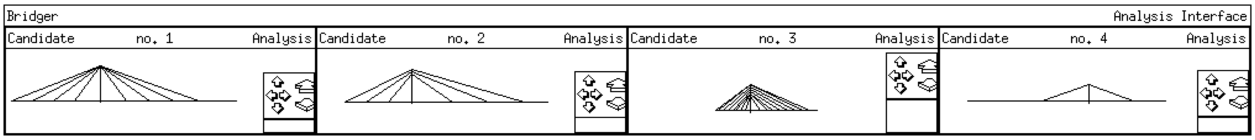

Bridger
Analysis Interface
Candidate no. 1 Analysis
Candidate no. 2 Analysis
Candidate no. 3 Analysis
Candidate no. 4 Analysis

Figure 6:

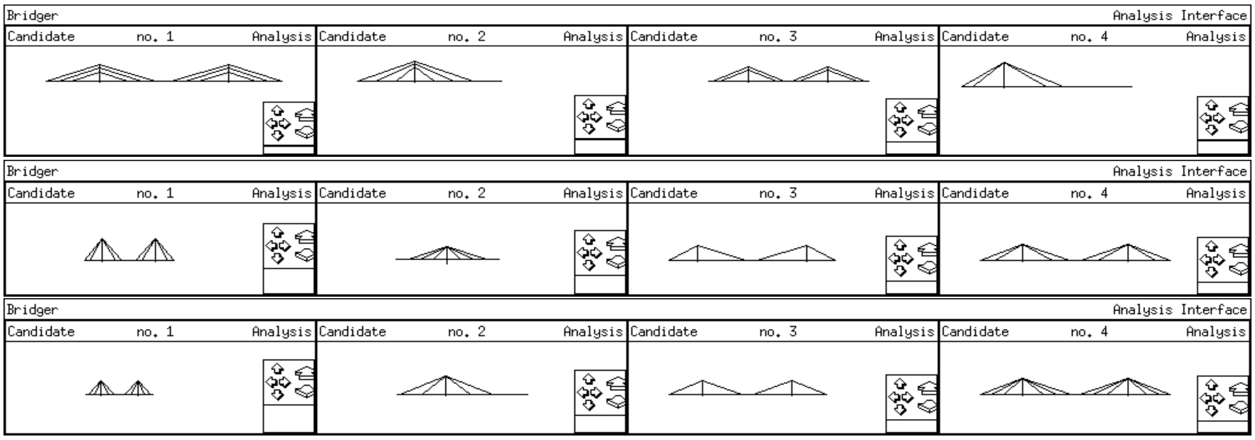

Bridger
Analysis Interface
Candidate no. 1 Analysis
Candidate no. 2 Analysis
Candidate no. 3 Analysis
Candidate no. 4 Analysis
Bridger
Analysis Interface
Candidate no. 1 Analysis
Candidate no. 2 Analysis
Candidate no. 3 Analysis
Candidate no. 4 Analysis
Bridger
Analysis Interface
Candidate no. 1 Analysis
Candidate no. 2 Analysis
Candidate no. 3 Analysis
Candidate no. 4 Analysis

Figure 7:

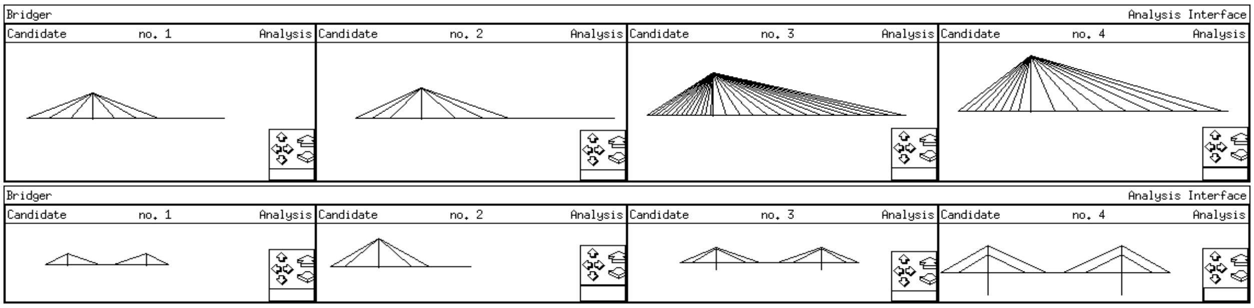

Bridger
Analysis Interface
Candidate no. 1 Analysis
Candidate no. 2 Analysis
Candidate no. 3 Analysis
Candidate no. 4 Analysis
Bridger
Analysis Interface
Candidate no. 1 Analysis
Candidate no. 2 Analysis
Candidate no. 3 Analysis
Candidate no. 4 Analysis

Figure 8: