M. G. Kahn

Division of Medical Informatics
Washington University School of Medicine
St. Louis, USA

Synopsis

Knowledge-Based Systems

The creation of a knowledge-based system (KBS), like the development of any other complex software program, proceeds in relatively discernible (though frequently iterative) steps: problem identification, solution conceptualization, program design and implementation, and finally system evaluation and deployment. Of the six papers selected for the *Knowledge-Based Systems* section of the 1993 IMIA Yearbook of Medical Informatics, the papers by Musen [1] and by Sandblad et al. [2] focus on solution conceptualization, the paper by Heckerman and Nathwani [3] focuses on program design and implementation, and the papers by Koski et al. [4], Verdaguer [5], and Kotzke and Pretschner et al. [6] focus on predeployment system evaluation.

In his paper *Dimensions of knowledge sharing and reuse* [1], Musen notes that the development of comprehensive, validated knowledge bases (KBs) such as QMR is laborious and expensive. The research challenge is to devise new KB development methods that enable KBs to be shared among different sites and to be reused for new or unique purposes. Musen examines five dimensions which must be addressed to promote knowledge reuse:

1. reusable lexicons (the linguistic terms used to describe the domain of discourse),
2. reusable ontologies (the descriptions of objects in the world, their properties and relationships to other objects),
3. reusable or interchangeable inference syntaxes (the symbolic notations used to represent the lexicons and ontologies),
4. reusable tasks (the application problem to be solved), and
5. reusable problem-solving methods (the strategies or methods which allow an application to solve a task).

The paper then focuses on the development of shareable problem-solving methods as one approach of separating how to represent a solution (rules, logic, probabilities) or how to program a behavior (forward chaining, *modus ponens*, Bayes formula) from how to solve a class of problems using a library of basic problem-solving methods.

Sandblad and Meinzer, in their paper *Modelling and Simulation of complex control structures in cell biology* [2], are concerned with modeling cell dynamics. Their focus is on constructing, executing, and evaluating complex process-control systems. Their modeling methodology abstracts processes into a collection of units which are constructed from two components: a process and a management unit. The management unit, in turn, is composed of a process controller and a process monitor. A model is constructed in a top-down fashion, using decomposition techniques to continually refine features until the model reaches a level of detail sufficient for its purpose. A model of the intestinal crypt/villus system is used as an example of a complex biological system which requires modeling of both static and dynamic features.

Based on reading only the titles and abstracts, the papers by Musen and by Sandblad and Meinzer would seem to be miles apart in the concepts and concerns raised by the authors. Yet I have categorized both papers as emphasizing solution conceptualization. The following three quotes, which appear in the Sandblad paper ([2], p. 37), could also have appeared in Musen's paper [1]:

- A model is a formal representation of those aspects of a studied system that are relevant to the intended use of the model.
- A model must always be formulated in some modelling language.
- We lack a suitable language to build such models and methods to analyze them.

Both papers focus on issues of languages, models, and representations. Musen does so to examine barriers to sharing and reusing knowledge; Sandblad and Meinzer do so to introduce a modeling methodology. Although Sandblad and Meinzer have a specific application in mind for their methodology, the issues they raise are confronted by all model builders, including builders of KBSs. Similarly, Musen's concerns about selecting lexicons, ontologies, inferences, tasks,

and problem-solving methods appear in one form or another in Sandblad's paper, even though Sandblad and Meinzer do not discuss knowledge sharing or knowledge reuse.

Heckerman and Nathwani, in their paper entitled *Toward normative expert systems: Part II, probability-based representations for efficient knowledge acquisition and inference* [3], describe a new probabilistic representation scheme, called similarity networks, as a method to ease the construction of large belief networks, and a representation method, called partitions, as a method to reduce the number of probability assessments. Both new techniques were implemented in the construction of Pathfinder, an expert system that assists surgical pathologists with the diagnosis of lymph-node diseases. The paper includes a brief tutorial on traditional belief networks. A similarity network is an extension of belief networks. It consists of an undirected graph whose vertices represent mutually exclusive diseases and whose edges contain local belief networks with features relevant to the discrimination of the two diseases connected by that edge. The overall belief network is constructed by the graph union of all the local belief networks. Similarity networks are particularly useful for representing conditional independence, and this strength is used to create partitions of diseases based on diagnostic features. Given a partition, only one set of probabilities needs to be elicited for all diseases in that partition, thereby greatly reducing the number of probabilities that need to be assessed. The paper also presents a brief evaluation of the efficiency of this method for constructing large belief networks and for assessing probabilities.

The largest set of papers focuses on KBS evaluation. Evaluation methodologies for computer-based decision-support systems are primitive, relative to the evaluation methodologies routinely used to study other clinical interventions. In the *Introduction to Decision-Support Systems* for the 1992 IMIA Yearbook of Medical Informatics ([7] p. 49), Van Bemmel and McCray noted:

"The central question in all evaluation studies is and will remain: what is the Objective Reference or Gold Standard by which to evaluate the systems?"

In a prior paper, Wyatt and Spiegelhalter stratify program-evaluation studies into two stages: laboratory and field testing [8]. Each stage has different study objectives and outcome measures which require different types of gold standards. Wyatt would classify the programs described in this section as "laboratory"-level systems. All three evaluators in this collection of papers faced the problem of no accepted gold standard; each paper adopts or proposes a different gold standard.

In their article entitled *Development of an expert system for haemodynamic monitoring: computerized symbolization of on-line monitoring data* [4], Koski et al. evaluated the performance of a program which transforms filtered real-value physiologic signals into symbolic labels and trends. Using data obtained from 10 clinical cases and a single clinical expert "gold standard," Koski compared their program's symbolic labels to the expert's labels. Complete agreement in the symbolic classification was used as the only performance metric. Koski's evaluation methodology is frequently used early in a program's development cycle to ensure that the evolving system's performance is "in the right ballpark." The observed 99.4% agreement on symbolizing signal levels is encouraging, whereas the 93.0% observed agreement on symbolizing trends may be problematic. For example, Compton performed serial evaluations of the GARVAN-ES1 system, a rule-based expert system which interpreted thyroid function tests [9]. The initial evaluation demonstrated 96% accuracy. The system eventually achieved 99.7% accuracy, but only after the knowledge base had more than doubled in size. Thus, despite the seemingly gratifying 93% observed agreement, Koski may face a daunting task to improve the performance of the trend symbolization module.

In their paper *Validation of the medical expert system PNEUMON-IA* [5], Verdaguer et al. asked five specialists to analyze 76 clinical cases in order to evaluate the performance of their diagnostic expert system. In an interesting methodologic twist, the evaluators never defined a "gold standard" answer for each clinical case. Instead, three distance measures were defined for comparing pairs of answers. The distance measures were selected to investigate different types of potential discrepancies (e.g., many small differences versus any large difference). Cluster analysis was used to describe groupings of distances. Because the clinical specialists had varying levels of expertise, the results of cluster analysis could be used to discern which grouping of physicians was most similar to the expert system. The evaluators never counted the number of "right" or "wrong" diagnoses, yet the information derived from the cluster analysis gave a broad overview of the clinical performance of their system.

Kotzke and Pretschner, in their paper *Possibilities of software phantoms for quality control of KBS in nuclear medicine* [6], are concerned with the lack of a sufficient range of clinical cases to ensure that all aspects of the clinical decision-support system are

tested. They describe the development of a software phantom, a parameterized computer program designed solely to generate test cases for evaluation. An evaluation system uses the parameters to generate cases with differing "known" features. The expert system evaluates these generated cases. An expert evaluator or an automated evaluation system compares the expert-system generated results to the parameters used by the phantom to create the case. To evaluate the degradation characteristics of the expert system, the software phantom generates artifacts or unrealistic cases. The paper describes the development of a software phantom for evaluating an expert system which interprets cardiac scintigrams for heart motion abnormalities.

With a parameterized software phantom, the evaluator systematically probes all aspects of a problem domain. An interesting question left unanswered in the paper is: How is the software phantom evaluated? Certifying the performance of an expert system using an unvalidated software phantom seems problematic. The development of the software phantom by the same set of programmers who developed the expert system also may cause concern. For example, subtle inaccurate assumptions about the data which may have crept into the expert system could easily be included in the development of the software phantom. Should the software phantom be created by an external group? If so, how can the two separate groups ensure that the software phantom generates test cases which can be understood by the expert system, without the expert-system group imposing their lexicon or ontology onto the software-phantom group? Conversely, if the two groups are free to define separate lexicons, how can we ensure that the answers generated by these systems are comparable?

Koski et al. [4] employed a single domain expert, Verdaguer et al. [5] employed a panel of experts and a cluster analysis technique, and Kotzke et al. [6] employed a computer-generated gold standard. The difference in the use of a single expert by Koski versus the use of multiple experts by Verdaguer reflects the difference in the maturity of the evaluated systems. Koski evaluated a specific task which is to be integrated into a larger, evolving decision-support system, whereas Verdaguer evaluated a large, completed diagnostic expert system. Matching the size and complexity of the evaluation methodology to the state of system development has been emphasized previously by others [10].

This collection of articles demonstrates the breadth of issues faced by KBS investigators and implementors. The days of the isolated computer programmer who "whips up" a system which is imposed upon the users are long past. Medical informatics is a multi-disciplinary team effort. The diversity of the issues presented in this section reconfirms this perspective.

References

[1] Musen MA. Dimensions of knowledge sharing and reuse. Comput Biomed Res 1992;25:435-67.

[2] Sandblad B and Meinzer HP. Modelling and simulation of complex control structures in cell biology. Meth Inf Med 1992;31:36-43.

[3] Heckerman DE and Nathwani BN. Toward normative expert systems: Part II. Probability-based representations for efficient knowledge acquisition and inference. Meth Inf Med 1992;31:106-16.

[4] Koski EMJ, Mäkivirta A, Sukuvaara T and Kari A. Development of an expert system for haemodynamic monitoring: computerized symbolization of on-line monitoring data. Int J Clin Monit Comput 1991;8:289-93.

[5] Verdaguer A, Patak A, Sancho JJ, Sierra C and Sanz F. Validation of the medical expert system PNEUMON-IA. Comput Biomed Res 1992;25:511-26.8

[6] Kotzke K and Pretschner DP. Possibilities of software phantoms for quality control of KBS in nuclear medicine. Meth Inf Med 1992;31:126-34.

[7] Van Bemmel JH, McCray AT, Eds. Yearbook of Medical Informatics 1992. Stuttgart: Schattauer, 1992.

[8] Wyatt J, Spiegelhalter D. Evaluating medical expert systems: What to test and how? Med Inform 1990; 15: 205-17.

[9] Compton P, Horn K, Quinlan JR, Lazarus L, Ho K. Maintaining an expert system. In: Quinlan JR, ed. Applications of Expert Systems. Glasgow, Scotland: Turing Institute Press, 1989: 366-84.

[10] Miller PL, Sittig DF. The evaluation of clinical decision support systems: What is necessary versus what is interesting. Med Inform 1990; 15: 185-90.