An answer to commentators on the paper "Generic tasks as building blocks for knowledge-based systems: the diagnosis and routine design examples"

B. CHANDRASEKARAN

I thank the commentators for their time, and generally positive remarks on the promise of the task-specific approach, in particular the generic task (GT) proposal.

Johnson and Zualkernan would like a methodology for mapping domain knowledge onto one or more generic tasks so as to solve problems efficiently. This stage of problem and domain analysis in which the kind of reasoning that goes on needs to be analysed in a vocabulary of generic tasks is very important, and in our laboratory we identify this as the *epistemic analysis* stage. For specific classes of problems we have developed guidelines on how to perform this mapping. For example, Bylander and Smith (Bylander and Smith, 1985) describe a set of criteria and guidelines for mapping medical knowledge into CSRL-like structures for diagnostic reasoning. Similarly Brown (1984) describes criteria for mapping design knowledge into DSPL-like structures.

David, in addition to a recapitulation of the motivations for task-specific approaches, which I fully endorse, makes several points. First is that the GT theory does not uniquely specify how to decompose a complex task into GT-like components. Second is that the spirit of the GT theory, viz its proposal about identifying generic information processing strategies and supporting them directly by means of knowledge and inference primitives for system construction and explanation, is more important than the specific GTs that we propose. I fully agree with the first point. We have repeatedly pointed out in our work that tasks such as diagnosis and design are compound tasks, and depending upon the domain and the knowledge available, different decompositions into generic tasks would be appropriate. Enforcing one decomposition will in fact limit the utility of the GT view. Regarding the second point, I agree in principle. However, the tasks of classification, matching, abductive assembly and plan refinement have a certain ubiquitousness to them that are likely to make them widely useful in conjunction with other GTs.

Iwasaki, Keller and Feigenbaum (hereafter IKF) agree with the practical usefulness of the GT view in helping construct knowledge systems efficiently. However they raise the concern that the GT view may lead to the "construction of many isolated knowledge bases ... whose knowledge may be encoded in different forms but may arise from the same body of fundamental knowledge of the domain". They go on to identify our tools with "shallow" knowledge and propose that "deep reasoning" or "first principles" representations will be needed in conjunction with "task specific inference engines" especially for reusability of knowledge and consistency maintenance.

IKF are right that the examples of use that I presented in the paper, namely the use of GTs in diagnosis and design, in fact use "shallow" knowledge, i.e. knowledge specifically compiled for these tasks. But my paper covers only a portion of the work in our laboratory and our point of view on building effective and efficient knowledge-based systems. In fact, the GT methodology is not restricted to shallow knowledge systems. In our laboratory, over the last several years, we have been concerned with the process by which diagnostic knowledge can be derived from a knowledge of structure and function of a device. We have developed a representation called the *functional representation* (FR) which provides the primitives for representing how the structure of a device is related to its behaviour and hence its function (Sembugamoorthy and Chandrasekaran, 1986; Sticklen, 1987). We have shown how diagnostic knowledge in the form of malfunction hierarchies and matching knowledge (i.e. in a form that can be represented by CSRL) can be derived from FR by means of a form of knowledge transformation. We have also developed methods by which device FRs can be used in the design task (Goel and Chandrasekaran, 1988).

IKF's concern for consistency between knowledge bases is well taken. In our work, where we use the FR for representing the relationship between structure and function and use a compiler to generate diagnostic knowledge, properly linking FR to the diagnostic and design knowledge derived from it, is important in keeping the knowledge in various forms consistent.

The FR and the inference processes that accompany it are a generic task in the sense in which generic tasks have been defined in the paper, i.e. specific input/output relations define the tasks for which the FR can be used, and there exist well-specified knowledge and inference forms for the representation. The GT view is not intrinsically restricted to shallow reasoning systems, nor should the ideas automatically lead to the development of disparate knowledge bases. If, for example, the design of an artifact would change, the FR can be updated, which will then result in the appropriate portions of the diagnostic knowledge structure encoded in CSRL to be also changed automatically.

IKF seem to argue that task-specific knowledge structures are "shallow" representations and the so-called first principle knowledge representation is "deep". They further seem to imply that there is a more general knowledge representation that applies to first principle knowledge and knowledge transformers can convert knowledge from the general form to task-specific form.

I believe that several ideas are getting confounded here. It is true that our knowledge of physics is deeper than say engineering in the sense that some of the latter can be derived from the former, but that does not imply that deep knowledge in this sense is *encoded* in the cognitive system using a different knowledge representation language. It is important not to conflate depth in the sense of science with depth in a cognitive sense. There are not any set of knowledge representations that uniquely capture first principle scientific knowledge and another set to capture the so-called task-specific knowledge. Algebraic equations can capture scientific laws, engineering equations are derived from them, and also purely empirical knowledge. By the same token, classification hierarchies, frame structures, and first order predicate calculus can all be used to represent scientific knowledge, engineering knowledge derived from it, and purely empirical knowledge that has no known basis in science. A "plan" as an epistemic object can be part of a domain's compiled knowledge or the first-principle knowledge: one might talk of plans that genes use to construct biological parts, and these plans would naturally be part of the "first-principle" knowledge in that domain. DSPL can be used to capture this first principle knowledge. CSRL can similarly be used to represent the hierarchical structure of the Periodic Table. By the same token, "structure-behaviour" representations (which IKF use as examples of first principle knowledge) can be used to represent a shaman's mental model of how the structure of the voodoo doll causes certain behaviours. Thus there is really no one-to-one correspondence between scientific knowledge and forms of so-called deep representations on the one hand, and between task-specific knowledge and shallow representations on the other. It is important not to equate task-specificness in the sense of "diagnosis" and "design" with the task-specificness in the sense of the GT approach. Classification and plan instantiation and refinement are generic information processing strategies. While diagnosis is certainly a generic activity, note that there is no GT in our paper corresponding to diagnosis. Some diagnostic knowledge may be in the form of classifications, some in the form of diagnostic plans, etc. Similarly some design activity may actually take place as instantiating and refining plans, while others may take place as parameter optimization by hill climbing. Thus the dimension in which strategies such as "classification" and "plan refinement" exist are orthogonal to the dimension in which tasks such as "diagnosis" and "design" as real world tasks exist.

IKF share the hope of many in the AI community (represented both by the logic community as well by others who do not see themselves as sharing the logicist viewpoint) that there exists a general knowledge representation language, and that knowledge stated in it can be transformed into various kinds of problem solving knowledge. Our view is that such a language can only be developed by paying close attention to how knowledge can be effectively and efficiently used. The effectiveness and efficiency of knowledge use in our view arises because there is a close connection between the information processing task in the service of which knowledge is used, the form of the knowledge and the inferences that are appropriate for it. It is in this sense that we regard CSRL, DSPL, FR and other knowledge structures as providing parts of this general knowledge representation. But unlike the logicists or other formalists, we think that the way to obtain the general representation language is as a union of languages each of which is appropriate for a generic information processing task. We also want to help the system builder by explicitly pointing out what types of knowledge representations are good for what information processing tasks, and what inference patterns are associated with what knowledge primitives. Thus the GT view asserts that there is a natural family of inference that goes with classification hierarchies, and that it is a good idea to explicitly link knowledge in the form of classification hierarchy to this inference and control knowledge. In contrast, general representation languages have usually obtained their generality by suppressing distinctions that are task-specific, viz the links between knowledge and how that knowledge should be used.

The GT proposal is entirely consistent with, and in fact supportive of, the desire of IKF to build "wide-ranging knowledge bases", since we consider ourselves as pursuing the same enterprise and vision. We think the GT languages provide some of the primitives in terms of which these wide-ranging knowledge bases can be effectively constructed.

This is not to say that the GT view is not itself due for some revisions as we understand better how to keep the advantages of the GT view while avoiding some of its disadvantages. One of our current goals is to understand how to bring additional flexibility in the use of knowledge and make integration of different generic tasks simpler and more principled. To do this, we are experimenting with SOAR (Laird, Newell and Rosenbloom, 1987), which provides the architectural facilities for flexibility and integration. While SOAR includes support for the so-called "weak methods", it has no theory of types of "strong", i.e. knowledge-intensive methods. The GT view on the other hand is a theory of the latter. We are currently looking into how the GT view and the SOAR view can be united for obtaining the advantages of both.

Acknowledgements

I thank Tom Bylander, Ashok Goel, and John Josephson for their very useful comments on the draft of this response.

References

Brown, D C, 1984. *Expert Systems for Design Problem-Solving using Design Refinement with Plan Selection and Redesign*, PhD thesis, The Ohio State University, August.

Bylander, Tom and Smith, Jr Jack W, 1985. *Mapping Medical Knowledge Into Conceptual Structures* Ohio State University, Laboratory for AI Research, Technical Report.

Sembugamoorthy, V and Chandrasekaran, B, 1986. "Functional representation of devices and compilation of diagnostic problem solving systems" In: *Experience, Memory and Reasoning*, J. Kolodner and C. Riesbeck (Eds.), Lawrence Erlbaum Associates, pp. 47–73.

Goel, A and Chandrasekaran, B, 1988. "Integrating model-based reasoning and case-based reasoning for design problem solving" *Ohio State University Laboratory for AI Research Technical Report*. Also appears in *Proc. AAAI Workshop on AI in Design*, American Association for Artificial Intelligence, August 1988.

Sticklen, Jon, 1987. *MDX2: An Integrated Medical Diagnostic System* June. PhD Dissertation, Department of Computer & Information Science, The Ohio State University, Columbus, OH, USA.

Laird, J E, Newell, A and Rosenbloom, P S, 1987. "SOAR: An architecture for general intelligence" *Artificial Intelligence* **33**, pp. 1–64.