expert system interfaces and social and organizational issues in the use of computers. The AI chapters may currently be of limited interest for practitioners but the organizational issues are widely recognized as important even though most people don't deal with them.

The handbook contains a total of 52 chapters and I must admit to not having read nearly all of them. Those I have read have mostly been of excellent quality.

For example, the chapters by Dennis Egan on individual differences and by John Whiteside, John Bennett and Karen Holtzblatt on usability engineering are very likely to be the standard references for these two topics in the future. People who only have time to read a single article on usability in their life might be recommended to read John Gould's chapter "how to design usable systems" – this single chapter contains enough checklists to form the basis for a quantum jump in usability in most development organizations.

Finally some comments on the usability of the handbook as a handbook: The number of printing errors in the book is a disgrace. No other word will do to describe the extent of typos in this book. The table of contents is somewhat unwieldy. The index (which is especially important in a handbook) is also too poorly done. For example there is a reference from the index term "thinking aloud" to page 963 (where thinking aloud is discussed in the context of knowledge elicitation for expert systems) and a reference from "software comprehension, thinking aloud" to page 108 where the method is mentioned in passing in a chapter on how programmers understand code. The index does not have a reference to the discussion by John Gould on p. 771 of thinking aloud in the context of testing and improving user interfaces. There are a further four references to "think aloud", all of which are to chapters mentioning some result from a thinking aloud study and not to discussions of the method itself. Having three different index terms for a single concept without cross references significantly lowers the probability that readers will find what they are looking for.

In spite of the various problems with the handbook, my conclusion is that it contains so many excellent chapters with original material or good surveys and extensive literature references that it will often be the first place I will turn to to look up information about topics in the human–computer interaction field.

Knowledge Acquisition for Knowledge–Based Systems (Volume 1)

Gaines, B.R. & Boose, J.H. (Eds.) Academic Press, 1988.

Knowledge Acquisition Tools for Expert Systems (Volume 2)

Boose, J.H. & Gaines, B.R. (Eds.) Academic Press, 1988.

Reviewed by: Linda Cook, Lockheed AI Center, 2710 Sand Hill Road, Menlo Park, CA 94025.

One of the major obstacles to the design and implementation of knowledge–based systems has been the knowledge acquisition bottleneck. A considerable amount of both basic research and application effort have gone into an analysis of this problem. This two volume series represents a compilation of these diverse efforts as reported at the first annual Knowledge Acquisition for Knowledge–Based Systems Workshop, held in Banff, Canada (November, 1986). While all of the papers are of the calibre that deserve mention, space precludes a review of all. As a best alternative, the general subjects covered and those papers which were unique, or offered unusual coverage of an interesting topic are presented in more detail. Volume 1 in this series (*Knowledge Acquisition for Knowledge–Based Systems*) describes research in three major areas: the cognitive aspects of expertise, the pragmatic aspects of knowledge acquisition skills, as well as, teaching those skills and, finally, inductive learning approaches to knowledge acquisition.

Articles classified under the cognitive aspects of knowledge acquisition offer some very unique ways of looking human problem solving processes and how that view might impact knowledge engineering practices. For example, Kornell, in his article entitled "Formal and Narrative Thought" suggests there are two distinct types of thought processes that should interest both knowledge engineers and those who are attempting to automate the knowledge acquisition process. Formal thought is characterized by closed worlds with legally defined laws of operation (i.e. predicate calculus), while narrative thought is described as a mode of thinking that is gestalt in nature. Transformations within formal thought are obvious, however, transformations within narrative thought can only be instantiated through "patterns of reasoning" (which corresponds to decomposing and arranging chunks of reasoning) and "kinds of reasoning" (handling the chunks of reasoning). In applications, patterns relate to the grouping and sequencing rules, embodying assumptions about goals or contexts. Kornell claims that these processes are not well understood and have been largely ignored by those in the knowledge acquisition community who tend to focus more on facts and heuristics. Another article by Bylander and Chandrasekaran (Generic tasks for knowledge–based reasoning) describes the search for generic reasoning tasks which can serve as a guide to the design and implementation of knowledge–based systems. Since process. Discussed in detail are two generic tasks: hierarchical classification and object synthesis by plan selection and refinement. As a whole, these articles present relevant issues for both

researchers and those involved in applications development.

A good indication of the maturity of any field is the adequacy of the educational practices offered for novices. Given the rather amorphous world of knowledge engineering one might conclude this aspect of knowledge acquisition to be somewhat lacking. However, several articles suggest the field is on the road to making good headway in the training of knowledge engineers. LaFrance's article (The knowledge acquisition grid: a method for training knowledge engineers) describes the major problems faced by knowledge engineers: a lack of understanding about human expertise, poor interviewing skills and a limited repertoire of questioning strategies. Her knowledge acquisition grid organizes "types" of expert knowledge and knowledge engineer questioning strategies as separate but interacting dimensions. The former represent how an expert's know-how might be stored while the latter are strategies available to the knowledge engineer for making that knowledge explicit. An excellent article for practicing knowledge engineers, as well as, those looking for guidance in developing those skills. The focus of additional relevant articles include: discourse analysis (analyzing the interactions of individuals requesting document retrieval) and protocol analysis (a methodology for eliciting and analyzing "think aloud" problem solving situations). The latter illustrates protocol analysis in a step-by-step manner using a real world example.

One of the benefits of this type of compendium of research and application efforts is the opportunity to learn from others experiences in developing expert systems. Johnson offers insight on the use of alternative knowledge-acquisition procedures in the development of a knowledge-based media planning system. Finally, the inductive approach to knowledge acquisition includes research on learning techniques, inductive rule generation and learning under uncertainty.

Contained in Volume 2 of this series (Knowledge Acquisition Tools for Expert Systems) is a review of the state of the art in automated knowledge acquisition. Most of the major knowledge environments are covered in detail (AQUINAS, KREME, KRITON, SALT, INFORM, KNACK and KITTEN) describing what the current state of development is and the directions for future research efforts. One of the advantages of the detail found in these reviews is that their method of automation is clearly presented and could be implemented manually if so desired. A good example is Boose's discussion of AQUINAS, a knowledge acquisition environment best suited for structured selection problems. Using their method of generating solution and trait hierarchies, one can impose a more structured knowledge engineering plan within manual interviews. A similar example is SALT, (Marcus, et al) a system for iterative design problems such as configuration.

Overall, this two volume series offers a state-of-the-art look at knowledge acquisition and due to the breadth of coverage, there is something there for everyone. As one final note, a large percentage of these articles appear in the *International Journal of Man-Machine Studies* special issues on knowledge acquisition.