same apparent authority and breadth of background reading. Furthermore, Reichgelt admits that while the bulk of the book describes methods that accept the knowledge representation hypothesis, the claim that any intelligent process contains knowledge that may be identified by external observers, there is much to be said for connectionist methods, which explicitly reject such a thesis. This is an extremely surprising admission for one who clearly believes in the logicist approach.

The evenhandedness extends to the assessment of each of the techniques covered. Reichgelt explains the advantages and disadvantages of each method in connection with a series of criteria set out in the first chapter. This allows comparisons to be made between the approaches, and goes some way towards providing a critical review of the subject; something which seems to be sorely needed by a subject of such critical importance as knowledge representation.

To illustrate the excellence of the book, consider the section on the representation of uncertainty, a subject with which I am very familiar. The section begins with a balanced discussion of the reasons why uncertainty is important, and three references for further reading, two of which seem appropriate. Then there is a detailed explanation of the certainty factor model which, while it has been heavily criticized, is still in wide commercial use. This is followed by a clear and concise account of Dempster–Shafer theory, an account which I have not seen bettered. Finally, and most commendably of all, there is a discussion of non-numerical formalisms, techniques which are only just beginning to catch on amongst those working in uncertainty. Indeed, Reichgelt goes so far as feel he needs to justify talking about uncertainty in quantitative terms, when in most circles it is assumed without question that uncertainty can only be handled with the use of numbers. An enlightened man indeed.

So, the overall verdict has to be that if you want a text that explains the basics of knowledge representation in plain English whilst being written with authority and no perceptible bias, and which provides clear criteria for discussing and choosing between formalisms, then look no further. Buy this book.

Reviewed by: Simon Parsons, Department of Electronic Engineering, Queen Mary and Westfield College, Mile End Road, London, UK.

Inheritance hierarchies in knowledge representation and programming languages edited by M Lenzerini, D Nardi and M Simi, John Wiley, Chichester 1991, pp 310, £22.50. ISBN 0 471 92741.

Reviewed by: Andrzej Glowinski, Advanced Computation Laboratory, Imperial Cancer Research Fund, London, UK

"Classification is a method of generalisation. Several classifications may, therefore, be used with advantage; and the physician, the pathologist, or the jurist, each from his own point of view, may legitimately classify the diseases and the causes of death in the way that he thinks best adapted to facilitate his inquiries, and to yield general results.

The medical practitioner may found his main divisions of diseases on their treatment as medical or surgical; the pathologist, on the nature of the morbid action or product; the anatomist or physiologist on the tissues or organs involved; the medical jurist on the suddenness or the slowness of death; and all these points well deserve attention in a statistical classification."
William Farr, Registrar General of England and Wales, Sixteenth Annual Report, 1856

The desire to impose structural order upon our knowledge of the world can be traced back to the origins of science and philosophy, but ancient schemes have long been abandoned in favour of the more objective and systematic approach introduced in the mid-eighteenth century by Linnaeus in his *Systema Naturae*. In essence, although the range of topics covered has expanded well beyond Linnaeus' classification of plants, the way in which concepts are organized into classes and hierarchies has not altered substantially since his time, with construction and use of the classificatory structures tending to employ methods which are informal, usually variable, often ill defined and sometimes uncomfortably *ad hoc* . More recently, however, there has been a steady growth of

interest in the properties and characterization of hierarchical structures, particularly with the mechanisms of passing information from one stratum in the structure to another—and in view of the subject's biological origins is of no surprise that this has been termed 'inheritance'.

The fields of artificial intelligence and object-oriented programming in particular have proved great stimuli to such studies. The need for a compact yet tractable means of analysing and representing real-world knowledge, especially in domains where multiple levels of complexity are encountered, have led to work on frames, semantic nets and classification-based reasoning, with David Touretzky's studies of inheritance perhaps the most widely known. At the same time, the increasing availability of computational power has made the use of object-oriented systems a very practical proposition, leading to a greater recognition of the relevance of inheritance hierarchies and increasing interest in their application. The publication of this text would thus appear to be very timely.

As William Farr pointed out, there is at least one enduring characteristic that plaques classifications and inheritance: no two people will seek to use a classification for the same purpose, hence the structures required will differ (as may the accompanying inheritance systems). The editors of this book aim to search out and clarify some of the differences between concepts of inheritance in various disciplines, and to provide both an introduction to research in the field together with a collection of technical reports on the state-of-the-art. This is an almost impossibly wide brief, but, with certain significant shortcomings, it is largely achieved.

The book is a collection of edited papers from a workshop held in Italy in early 1989 (there is no indication as to why the text was so long in publication), and shares much in common with a set of proceedings. Overall, the structure is more coherent than is usual with a collection of conference papers, but the organization of the book tends to suffer because of the variation in the content and style of individual contributions. In several places it is difficult to see the connections between chapters, even with the help of the overview given in the introduction (it may have been helpful to provide a preamble or commentary on papers or sections). The quality of presentation is as variable as one has come to expect of proceedings, with different type faces, some grammatical mistakes, and unfortunately several errors in citations. The long interval to publication should have allowed these to be corrected.

The variation between chapters is one of the most striking features of the book, and this is perhaps reflected in the lack of a precise identification of who it is aimed at. Contributions vary in complexity from very general, introductory overviews of topics that, although of interest to newcomers, will be of no value to those well versed in the field to detailed discussions that make substantial assumptions about the knowledge of the reader. This particularly so in the case of the technical papers, for which familiarity with logic and logical notation is a necessity. Given this proviso, most of the discursive contributions are well presented and reflect the standing of the individual authors in their respective fields. The contributions by Horty and Stein on credulous and sceptical inheritance, for instance, provide two quite different views of some of the theoretical problems which beset inheritance networks and usefully broaden the perspective to include reasoning and the management of uncertainty, while Selman and Levesque help to reintroduce some realism with their discussion of computational tractability. Several papers are very formal indeed, dealing with focused but important topics such as the application of the closed world assumption. These are definitely not for the novice, and serve as a reminder of how difficult it is to decide on the intended readership of a book such as this.

As with all workshops, there is a smattering of papers devoted to describing individual programmes and new programming languages. Although this meets the editors' aim to encompass state-of-the-art work, it is not as successful as one might expect, largely because of the tendency of the authors of these papers to concentrate on the details of the system to the detriment of more general discussion of the implications of their work. These comments apply equally to the sections dealing with knowledge representation and with object-oriented programming, but not to the final paper (on logic programming knowledge bases, by Finin and McGuire, which is effectively in a section of its own). This shows how one can successfully illustrate the discussion of principles by the

description of an implementation; it is also an example of the resurgent interest in the long-disfavoured hierarchical data model, particularly in the realm of deductive databases.

The book represents a worthy but flawed attempt to fill a gap in a field that is steadily becoming more important. Although it contains many good papers which provide a detailed overview of several important and controversial issues, it suffers from variation in presentation and editing—had the knife been wielded more actively a shorter but more effective text may have resulted. I am also concerned about the delay in publication, for this is a very active area of research; in addition, the presence of some of the overly descriptive papers will cause the contents to appear to age much more quickly than would otherwise been the case.

However, there is another and entirely different reason for a feeling of expectation unfulfilled which has more to do with the maturity of the field than this book. In the introductory chapter, Peter-Schneider raises two questions: what purpose does inheritance serve, and is it adequate? These would seem to be much easier to answer in the realm of object-oriented programming (although even this is not as certain as originally thought with the recognition of differences between inheriting behaviour and inheriting code or implementation details) than in knowledge representation. With a few exceptions, these questions seem to have been avoided, with a concentration on the properties of inheritance mechanisms being the order of the day. Perhaps another workshop is required to provide insights and answers to these questions.

"*Inheritance Hierarchies in Knowledge Representation and Programming Languages*" contains much worthwhile material for anyone interested in the field. However, the book's flaw is in attempting to cover too much ground—the intended audience is too broad, and most readers will have to be rather selective as they may find that some chapters are either rather introductory or too specialized.