construction of the program models, then it should be the models that are maintained and not the code.

The practical nature of the book is emphasized by the many exercises and case studies. My only criticism here is the way COBOL is used in the solutions. Whilst sympathizing with the reasons for using GO TOs and not PERFORMs, I still feel that what is being encouraged are programs that will be difficult to maintain, a crime further compounded when it is considered that the book is aimed partly at programmers.

Notwithstanding these criticisms, I found overall the book to be an improvement on its predecessor, I would feel happy in recommending this book as a supporting text book to undergraduate students on Jackson courses, especially because of its well thought out practical material and examples.

C. DAVIES
Manchester

ERA REPORT 90–0693R.
Formal Methods for Software Development—their significance to industry. Technology Limited. £150.

For some years Formal Methods have been proclaimed as helping in the production of good software. Do such methods have any use and significance in industry? The aim of this report is to: provide an impartial assessment of the current capabilities and significance of Formal Methods, and to enable industrial project managers to make an informed decision about whether, and how to adopt this emerging technology. They define a Formal Method as a mathematically based method for specification and verification.

The report is divided up into the following sections: the considerations in choosing a Software Engineering Method, a review of Formal Methods, application of Formal Methods to parts of a typical software life-cycle, results of a survey of the use of Formal Methods in industrial applications, a summary with recommendations. The report also has a glossary of terms for readers unfamiliar with the subject.

In considering how to choose a Software Engineering method they use a definition of a Software Engineering Method taken from the STARTS guide [1] as the basis of their discussion. Although parts of this section are poorly explained it draws out the relevant points. This is then used to categorize the Formal Methods they review.

Their review of Formal Methods is brief and useful. The table they present comparing the capabilities of typical Formal Methods is badly drafted, uses criteria which are not clearly defined and is not correct. The review is now also out of date, e.g. RAISE [2] is now available commercially although it was not when this report was written.

The section which describes how Formal Methods could be integrated into the software life-cycle gives a fair assessment of their benefits and difficulties. It does not, however, say much about animation and some of its examples are not clear. It is also unclear why they think a "safe" language could be used in place of a Formal Method.

The survey which ERA have conducted is described briefly and confirms their general conclusions. They also point out the lack of objective evidence of the contribution that Formal Methods make to the software engineering process. The tables used to summarise the results of the survey seem to include MALPAS [3] as a Formal Method, which would be ruled out by their definition and I would regard as a tool.

The report's summary in places goes beyond what has been said earlier in the report, e.g. it states that Formal Methods lack expressive powers for most problems which is not true. The overall summary is conservative about how Formal Methods should be used, restricting them to providing a secondary specification of critical applications, which I think even on the basis of this report could be extended to the specification of most software.

Glossaries are always easy to criticise but it was not clear which words would be in it and some, that were in the report, that I looked up were not in it, e.g., multivariable.

My main criticisms of the report are that: it needs updating, tables and diagrams are badly laid out and confusing, some of the detailed information is either unclear or incorrect, it does not give information on how a manager might go forward if he decided to use Formal Methods, e.g. no information on training courses.

In spite of these criticisms the report meets its stated aims and provides a good starting point for an industrial project manager. It is also good value for money when the alternative is consultancy.

REFERENCES

[1] National Computing Centre, *The STARTS Guide. A Guide to the Methods and Software Tools for the Construction of Large Real-time Systems*, second edition. NCC (1987).
[2] The RAISE Language Group, *The RAISE Specification Language (The BCS Practitioner Series)*. Prentice-Hall, Englewood Cliffs, NJ (1992).
[3] Rex, Thompson and Partners Ltd, *MALPAS User Guide, MALPAS Release 4.1*, issue 3 edition. RTP/4009/UG +(1988).

G. I. PARKIN
Teddington

MALCOLM CLARK
TeX applications, uses, methods. Prentice Hall, Simon & Schuster International Group, 66 Wood Lane End, Hemel Hempstead, Hertfordshire, HP2 4RG. 13–912296–6. £44.95.

This book, a collection of presentations made at the 1988 Exeter TeX User's Group meeting, is effectively 4