

8. Some benefits of the AC as a software system

What would be the benefits of having the AC as a software system?

1. There is an application model which is the reference point for both user and the DPS and which remains in existence during the whole of the life cycle of the application. User and DP activities are actually based on it. It is recommended to include in the application model also procedures of the BS which have no bearing on the DPS. Thus the application model may become a model of the whole of the BS.
2. The logical simplicity of an application and hence its comprehension by a user and systems analyst are greatly enhanced by the AC advocating (a) a separation of the general case of activities in the Condition/Action File from the individual case, i.e. the occurrence of activities, messages, data in the Status File; (b) the isolation of the time aspect of an application in the Condition/Action Handler. This component contains a 'timetable' of the application in the shape of its polling facility; (c) a separation of triggering functions from processing functions. The triggering functions are the domain of the Condition/Action Handler, while the processing functions are the domains of the user or the DPS.
3. Through the AC we would be able to simulate the behaviour of the DPS and the application as a whole, while we implement them, and to predict their performance under various load conditions. The state networks in the Condition/Action

File represent a model of the application on which simulation can be directly performed. To this end actions in the state networks are assigned resource consumption values. The frequency of activities and messages can be derived from the usage pattern of the application. The Condition/Action Handler is the driving device in the simulation runs, which can take place as the development work progresses. Thus the simulation capability of the AC would ensure at various stages of the development process that the application design is not only functionally sound, but also meets the performance requirements as laid down by the user.

4. We may visualise any BS as consisting of a subsystem of physical activities, such as the production of goods or the provision of services, and of a control subsystem. The control subsystem comprises the decision procedures, data files and the information flow between resources. Tuning this control system means finding the 'optimum' time pattern for making decisions and then constantly adjusting it to the changing requirements of the BS. The AC has a range of tools for setting and adjusting the control cycle in a BS. The main tool is the polling list of the Condition/Action Handler, as it specifies, by linking Condition/Action Handler processes to time conditions, how quickly activities are moved on in the user environment and the DPS.

Hence the AC as a software system would perform for the application what a DBMS performs for data and an operating system for computer resources: It would control the application more efficiently than systems have hitherto done.

References

- STECHER, P. (1977). Proposal for an interface system between the business and data processing systems, *The Computer Journal*, Vol. 20 No. 3, pp. 194-201.
- STECHER, P. (1978). On the Interface Between Business Systems and Data Processing Systems, Ph.D. Thesis, University of London.
- IBM (1). HIPO-Draw, Program-no. 5796-BFF.
- IBM (1974). HIPO, A Design Aid and Documentation Technique, GC20-1851.
- IBM (1977). TSO-3270 Display Support and Structured Programming Facility (SPF), Version 2, General Information Manual, GH20-1974.
- IBM (1978). Information Management System/VS, System/Application Design Guide, SH20-9025.

Book reviews

Logic for Problem Solving, by Robert Kowalski, 1979; 287 pages. (The Computer Science Library/North-Holland, \$18.95, \$9.95 paper)

One of the interesting attributes of artificial intelligence is that it has stimulated the creation of novel formalisms that are both fundamental and applicable to a wider area. Robert Kowalski's work in the field is a good example of this, and for several years he has seen formalism in logic as a fundamental way of expressing notions in problem solving. The central theme of the book is a form of 'clausal' logic which is related to top-down reasoning rather than the more conventional inferential schemes as found in classical texts such as Quine's.

The introductory chapter centres around the way that one expresses statements in clausal logic, while Chapter 2 begins to relate clausal forms to traditional AI topics such as data base searches and semantic networks. Chapter 3 discusses the parsing problem in the framework of clausal logic, while Chapter 4 arrives at tree searching in problem solving applications. This appears to be the intellectual epicentre of the book, which then goes on to consider computational procedures and plan formation, heading resolutely towards provability and ending with a brief consideration of the dynamics of informational systems which '... attempts to combine the traditional rôle which logic plays in epistemology and the philosophy of science with its new rôle in computing'.

For the browser, the book may be somewhat heavy going, while for the teacher of AI or Computer Science Theory it is an essential addition to the bookshelf.

I. ALEKSANDER (Uxbridge)

Simulation: Principles and Methods, by W. Graybeal and U. W. Pooch, 1980; 249 pages. (Prentice-Hall, £12.95)

Simulation is a technique which involves the application of an assortment of mathematical tools to the solution of a problem. It is beset with pitfalls for the inexperienced and unwary. Graybeal and Pooch have described the assortment of mathematical techniques and the pitfalls, together with various approaches and simulation languages in one single volume. In consequence, the treatment of any topic can at best be described as superficial. However, there are virtues in this approach. It illustrates a methodical approach to simulation in outline and, provided the simulator is already acquainted with various branches of mathematics, or is prepared to work hard at becoming so acquainted, could act as a guide and catalyst. It must be emphasised that it is not a practical how-to-do-it book. In particular the last chapter raises the problems to be encountered after a simulation model is completed and validated, namely the design of experiments using the model. This is a much neglected area in the world of simulation.

The presentation of material is clear, with mathematical results in general being assumed. Concepts are both explained and described mathematically and there are a number of simple arithmetic examples illustrating these and other ideas.

This book is designed to support a graduate course (in America) for potential users, which explains the inclusion of three pages on analogue computing (completeness?) but not the absence of activity diagrams.

R. E. SMALL (London)