$$cost_3 = \frac{1}{\text{comb}(B, b_2)\,\text{comb}(B, b_3)} \sum_{v_2=0}^{b_2}$$
$$\times \sum_{v_3=0}^{b_3} (b_1+v_2+v_3)\,\text{comb}(B-b_1, v_2)$$
$$\times \text{comb}(b_1, b_2-v_2)\,\text{comb}(B-b_1-v_2, v_3)$$
$$\times \text{comb}(b_1+v_2, b_3-v_3).$$

After applying the binomial coefficient properties it is derived that:

$$cost_3 = cost_2 + \frac{1}{\text{comb}(B, b_2)} \sum_{v_2=0}^{b_2} \sum_{v_3=0}^{b_3}$$
$$\times v_3\,\text{comb}(B-b_1-v_2, v_3)$$
$$\times \text{comb}(b_1+v_2, b_3-v_3).$$

Therefore, finally the total cost for a logical query decomposable in n simple ones is:

$$Totcost_n = \sum_{b_1=1}^{r_1} \cdots \sum_{b_n=1}^{r_n} \prod_{i=1}^{n} \mathrm{P}(b_i)\, cost_n,$$

where $cost_n$ is a recursive function of the form:

$$cost_n = cost_{n-1} + \frac{1}{\text{comb}(B, b_n)} \sum_{v_2=0}^{b_2} \cdots \sum_{v_n=0}^{b_n}$$
$$\times v_n\,\text{comb}(B-f_n-1, v_n)\,\text{comb}(f_n-1, v_n).$$

The quantity f_{n-1} gives the number of blocks which have been already accessed by the previous $(n-1)$ queries and is defined as:

$$f_n = b_1 + \sum_{i=2}^{n} v_i.$$

3. Numerical results

The following tables depict some simple numerical results. It is assumed that $R = 100$ records and $B = 20$ blocks; therefore, the block capacity is $bc = 5$. Cost A is the mean cost for answering the two queries independently of each other, while Cost B is the mean cost if the two queries are answered by using the method described in the second section. Table 1 concerns two logical user queries. As expected, greater are the query sizes greater is the gain. Table 2 gives a simple example to show that, as expected again, the gain is greater for greater number of logical queries.

Acknowledgement

Many thanks are due to Antonis and Christos Aletras for their help in experimentation.

Y. MANOLOPOULOS

Division of Electronics and Computer Eng., Department of Electrical Eng., Aristotelian University of Thessaloniki, 54006 Thessaloniki, Greece

References

1. I. Ahn and R. Snodgrass, *Partitioned Storage for Temporal Databases, Information Systems*, Vol. 13, No. 4, pp. 369–391 (1988).
2. B. G. Claybrook, *File Management Techniques*, John Wiley (1983).
3. S. B. Yao, Approximating Block Accesses in Database Organizations, *Communications of ACM*, **20**, (4), 260–261 (1977).

Book Review

JAMES W. HOOPER and ROWENA O. CHESTER, *Software Reuse Guidelines and Methods*, Plenum Press, New York and London, 1991 $45.00. ISBN 0-306-43918-2

The idea of reusing software has its origins in the very earliest days of computing when subroutine libraries were first introduced, but its conscious development as a technique with the potential for substantially reducing software development costs is much more recent. Although the concept of reducing costs through extensive software reuse seems at first to be simple and obvious enough, formidable practical obstacles have to be surmounted. These range from purely technical issues such as ways of generalising subroutine interfaces to significant managerial and organisational issues such as the best way of organising a large software production operation to take maximum advantage of any potential for reuse.

These issues have been addressed in a large number of research projects addressing different aspects of the problem of encouraging the introduction of widespread software reuse, but reuse involves so many widely different issues that it is often hard to interrelate the different aspects of the overall problem. With the present book, a general picture of the whole field emerges, and it is hard to overestimate the importance of this for further progress in making widespread reuse a reality. The book arises from a research project sponsored by the US Army: it comprises a very detailed and well-structured commentary on the technical literature on reuse, with accounts of case studies and recommended guidelines for future practice added at appropriate points.

After an introductory chapter, separate chapters deal with managerial and with technical guidelines for the introduction of software reuse, concluding with a brief chapter on 'Getting Started'. The accounts of relevant technical literature are very perceptively written, and many sections conclude with recommended guidelines for future practice – all the guidelines are collected together in a Appendix, with guidelines for reusable Ada code given separately. An extremely comprehensive range of references is given: somewhat confusingly references are given both at the ends of the chapters and in an integrated bibliography at the end of the book.

Anyone who thinks seriously about the problems of software reuse will realise that this is a complicated problem involving both technical and managerial issues, and that different contributions to the subject are often hard to relate even to one another. With this book, there is at last a unifying overview of the whole range of problems relating to software reuse that are carefully and perceptively related to each other and distilled into a series of detailed guidelines for future practice. As an overall tutorial introduction to software reuse and its problems coupled with its guidelines for future practice, the book deserves a wide readership.

PETER WALLIS
Bath

Announcement

1–5 MARCH 1993

CAIA-93, the Ninth IEEE Conference on Artificial Intelligence for Applications, Disneyworld Hilton – Orlando, Florida

The conference is devoted to advancing the application of artificial intelligence techniques to real world problems. This conference provides a forum for synergy between applications and AI techniques. Emphasis at this year's conference will be on new AI paradigms that can have or have had an impact on applications.

Partial list of invited speakers: Patrick Winston, MIT and Ascent Technology, 'Learning and Database Mining'; Wendy Lehnert, University of Massachusetts, 'What we've learned from the DARPA Natural Language Initiative'.

- 1–2 March 1993: Conference tutorial programme
- 3–5 March 1993: Conference technical programme

For registration and additional conference information, contact:

CAIA-93, IEEE Computer Society, 1730 Massachusetts Avenue, NW, Washington, DC 20036-1903. Phone: 202-371-1013.

Downloaded from https://academic.oup.com/comjnl/article/35/6/668/352998 by guest on 10 April 2024