SAPERE • AUDE

University of Otago
Te Whare Wananga O Otago
Dunedin, New Zealand

Software Development, CASE Tools And 4GLs - A Survey of New Zealand Usage (Part 1)

Stephen G. MacDonell

The Information Science Discussion Paper Series

Number 93 / 4
November 1993
ISSN 1172-6024

Software Development, CASE Tools and 4GLs – A Survey of New Zealand Usage.

Part 1: 750 New Zealand Organisations

Dr Stephen G. MacDonell[1]
Department of Information Science
University of Otago

November 1993

Abstract

This paper reports the results of a recent national survey which considered the use of CASE tools and 4GLs in commercial software development. Responses from just over 750 organisations show a high degree of product penetration, along with extensive use of package solutions. Use of 3GLs in general, and of COBOL in particular, is still relatively widespread, however. In terms of systems analysis and design techniques under a CASE/4GL environment, screen and report definition is the most preferred technique, although both data flow analysis and data modelling also feature strongly.

[1] Address correspondence to: Dr S.G. MacDonell, Research Fellow, Department of Information Science, University of Otago, P.O. Box 56, Dunedin, New Zealand. Fax: +64 3 479 8311 Email: stevemac@commerce.otago.ac.nz

1 Introduction

In general terms fourth generation languages (4GLs) and computer aided software engineering (CASE) tools have been in existence for a decade or more. During this period they have received extensive attention from academia and industry as well-publicised solutions to the 'software crisis', that is, the increasing backlog of new system requirements coupled with rising demands for maintenance. Early local industry perceptions of the benefits of 4GLs were typical of those reported worldwide—"Application development with fourth generation languages (4GL) has been shown to be at least 10 times faster than BASIC or COBOL." (as reported in *NZ Computer Interface*, April 1984).

The extent to which these products have since been adopted for use in development, however, is unclear. As recently as 1987 McClure suggested that just 2% of potential users were employing CASE tools. Furthermore, Pressman (1992) has remarked that even though CASE tools are more important than hardware for attaining greater levels of quality and productivity, they are still not widely used within software development. Locally, a 1987 survey found that less than 15% of total users were employing a 4GL (Botica, 1987).

A number of reasons may have contributed to this apparently slow acceptance: one possibility would appear to have been the absence of clear definitions for the terms '4GL' and 'CASE'. These and other related terms have been used very widely as marketing tools while the direct meaning of the words, in terms of product functionality, has remained vague (CIS, 1989). This is clearly reflected in the findings of a 1987 Australian survey: of 507 respondents, 19.6% *weren't sure* whether they were using a 4GL or not, due to problems of definition (Pacific Reporter, 1987).

Another possible cause may have been the expense associated with the adoption of CASE/4GL tools—Huff (1992) has suggested that many organisations simply found the move to CASE too costly, resulting in the cancellation of many CASE-based projects. Maria (1991) has stated that CASE tools have been slow to gain acceptance because, in general, the tools have failed to address the requirements for maintenance of existing systems, in spite of the fact that maintenance has continually consumed significant amounts of developer effort, and consequently, development budgets. Conversely, other reports have suggested that the acceptance of development tools has been steadily increasing (e.g. see Chen and Norman (1992) or CIS (1989)). In general, however, no empirical data has been provided to confirm or refute these suggestions.

The main objective of the survey described here, then, was to ascertain whether this situation of low product adoption has changed over the past five years. One of its principle aims was therefore to determine the current levels of 4GL and CASE tool usage in this country. In a similar vein, the survey was also concerned with related changes in software development practices. A 1987 survey of 280 organisations (resulting in 120 replies) has provided a baseline for comparison (McAulay, 1987).

2 The Survey

Development and distribution of the survey was carried out in April and May of this year. A further four months were then allocated to the data collection phase of the project. The procedures used during this period are described in the next subsection, followed by the presentation and discussion of the survey results.

2.1 Methodology

The survey instrument was developed with two conflicting objectives in mind. First, it needed to be comprehensive enough to ensure that the required data would be provided. Second, it needed to be relatively short so that potential respondents would not be discouraged from completing it. A set of (at most) eight sections were therefore included, considering issues relating to product use, development approaches, analysis and design methods and system types.

The survey was distributed to two different samples. The main sample of more than 1000 sites was made up of a cross-section of New Zealand companies and organisations, details of which were obtained from a number of sources, including the New Zealand Business Who's Who, the NZ Post Business Directory and conference attendance lists. Site selection was performed in an attempt to ensure that organisations of varying size and function were surveyed. (The second sample was a much smaller set of software houses. The responses obtained from this sample will be reported at a later date.) The distribution of sites in the main sample included organisations from nineteen different functional classes. For comparative purposes, Table 1 shows the percentage values for the number of sites in each class against the number of employees in the same class, as taken from the 1993 New Zealand Yearbook. Clearly this is only one classification method, and others, such as those based on contribution to GDP or on annual turnover, may in fact be more relevant. Accurately obtaining financial data of this kind over nineteen aggregated classes, however, is extremely difficult. Therefore the employee-based approach was adopted as a relatively crude but basically informative classification method, at least for the purposes of the survey.

Some marked differences appear in Table 1, but in general these are not overly

Site Type Classification (Percentages)			
Site type	Employees	Survey sites	Difference
01 Primary Production	2.5	2.3	− 0.2
02 Oil, Gas, Minerals, Electricity	2.4	9.1	+ 6.7
03 Chemicals, Pharmaceuticals	1.0	2.6	+ 1.6
04 Manufacturing (Non-food)	12.0	18.9	+ 6.9
05 Agricultural/Horticultural Products	3.4	3.7	+ 0.3
06 Retail, Wholesale, Distribution	21.5	12.2	− 9.3
07 Food and Beverage Manufacturing	2.6	4.6	+ 2.0
08 Hospitality, Tourism	5.7	2.6	− 3.1
09 Transportation, Storage	5.6	6.1	+ 0.5
10/11 Central and Local Government	7.8	8.2	+ 0.4
12 Health Care	8.5	1.3	− 7.2
13 Construction, Engineering	7.6	3.5	− 4.1
14/20 Legal and Business Services	9.2	5.1	− 4.1
16 Banking, Finance	3.7	4.3	+ 0.6
17 Insurance	1.3	5.1	+ 3.8
18 Communications, Media, Publishing	4.2	4.8	+ 0.6
19 Automotive Assembly and Sales	1.3	1.4	+ 0.1

Table 1: Distribution of survey sites across industry classes

unusual. Classes 06, 12, 13 and 14/20, for example, show significantly lower percentages of sites when compared to the percentages of category employees. These classes, however, represent labour-intensive industries (Retail, Wholesale, Distribution, Health Care, Construction, Engineering, and Legal and Business Services), in which the proportion of software development units to total personnel could be expected to be small. In contrast, the proportions of sites for classes 02 and 04 are much greater than the corresponding percentages of employees. Given that this cannot be attributed to low survey response, this might be taken to suggest that the number of employees per functional unit (and consequently per software development unit) is relatively low in these industry types (Oil, Gas, Minerals, Electricity, and Manufacturing (Non-food)). Support for the inclusion of a large number of sites from the Manufacturing class is provided by a recent survey of the information technology departments of 2000 Australasian organisations (Philipson, 1993), in which sites from the manufacturing industry also made up the largest percentage of the sample (at 23.7%). Thus the only potentially questionable difference in proportions occurs for class 02, which included all of the recently established electricity and gas distribution/retail authorities, hence the high percentage of sites in comparison to the number of employees. Over the entire sample, however, the distribution of site types is approximately proportional to the employee distribution. This would suggest that the responses of the selected sample should provide an appropriate pointer to those we could expect from the entire development industry in this country. (Results broken down according to industry type will be published in a subsequent paper.)

The survey was distributed by mail to Information Systems Managers over a period of two months. Each approach included a cover letter, explaining the motivation for the study, a copy of the survey and a pre-addressed reply-paid envelope. Every site that had not replied within six weeks of posting was contacted again in the same manner, but with a slightly different cover letter.

The response to the survey was extremely satisfying and quite unexpected. Of the original sample of 1068 sites 95 were found to be repeats—that is, more than one letter was sent to the same organisation, usually under a different name. A further 38 sites were found to be invalid—they were no longer trading, or declined to complete the survey. This left an actual sample of 935 sites. Of these, 753 usable replies were received. This represents a response rate of just over 80%. It is to be hoped that such a large number of responses will help to ensure that the responses are representative of the development population.

2.2 Results

The first response section of the survey included spaces for the respondent's name, title, address and so on. Of some interest here were the job titles of the respondents. The twelve most frequently used titles are shown in Table 2.

The frequencies shown in Table 2 account for just over 400 of the 753 replies. Responsibility for software development and use would appear to be well-defined, with jobs specific to computers or information systems making up nine of the top twelve job classes. Moreover, 449 of the responses were from managers, illustrating the relative importance that is now afforded to positions concerned with the control of software systems.

The first two software-related questions in the survey were concerned with system usage and overall system implementation strategies. The results of these questions are shown in Tables 3, 4 and 5.

When compared to past reports, certain trends can be identified from the results presented in Tables 4 and 5. McAulay (1987) found that of 120 New Zealand respondents, the average proportions of systems implemented under the approaches considered in Tables 4 and 5 were: 56% of systems were developed in-house, 27%

Respondent's Job Title	
Title	Frequency
IS Manager	109
MIS Manager	51
(Unspecified)	49
Systems Manager	32
DP Manager	30
Company Secretary	25
Financial Controller	22
General Manager	21
IT Manager	18
Consultant	15
Systems Development Manager	14
Computer Manager	13
Computer Services Manager	12

Table 2: Frequency of respondent job titles

Use of Information Systems		
	Yes	No
Use information systems?	736	17

Table 3: Use of computer-based information systems

were implemented from pre-packaged solutions, and 8% were developed by software houses.

Based on further questions, McAulay stated that the dominance of in-house development was not surprising, but that changes were anticipated. More than 50% of her respondents expected package use to increase relative to other methods, whereas only 36% expected in-house development to increase. Costs of in-house development and a lack of time were cited as the main reasons for moving away from in-house development to packaged solutions. Of the 120 respondents, 13% felt that the proportion of package use would decrease, due mainly to the need for integrated systems and for systems that satisfied unusual user requirements. McAulay (1987) remarked, however, that the first of these two reasons was likely to become less prominent given the trend towards integration-oriented packages.

By 1989 it was reported that most New Zealand companies would look for a package solution to fulfil application requirements before they would consider in-

System Implementation Strategies		
	Yes	No
Perform in-house development?	441	280
Purchase packages off-the-shelf?	626	88
Systems supplied by software house?	441	273
Use other implementation approaches?	59	655

Table 4: Methods of system implementation

System Implementation Strategy Proportions						
	1-20%	21-40%	41-60%	61-80%	81-100%	Average
	Cases:	Cases:	Cases:	Cases:	Cases:	
In-house development	176	60	50	57	98	28%
Packages off-the-shelf	222	99	81	95	129	42%
Software house	197	64	45	61	74	26%
Other	14	8	6	12	19	5%

Table 5: Proportions of systems implemented under various strategies

house development (CIS, 1989). This trend appears to have continued, given the results of the current survey. When compared with McAulay's results (1987), it can be seen that in-house development has fallen from 56% to 28% on average, and that both package use and software house development have increased markedly, from 27% and 8% to 42% and 26% respectively. These results confirm the expectations of McAulay's respondents with respect to changes in development strategies.

The remainder of the survey was concerned with in-house development so only those respondents that undertook their own development (a total of 456 sites) continued answering. The next question in the survey asked respondents who did not use CASE tools or 4GLs to specify the development languages that they did use. This generated valid responses from 187 sites. Just under half of this set of respondents (84) said they used one language, 56 respondents said that two languages were employed, 22 sites said they utilised three development languages, 13 cited use of four languages and 12 said that five or more languages were used for software development. The actual languages used are shown in Table 6.

Table 6 illustrates that, at those sites not using CASE tools or 4GLs, COBOL continues to be the most widely used development language. Even considering that CASE tool/4GL users were excluded from this question, the results provide some support for Pressman's assertion (1992) that languages like COBOL and, to a lesser

Development Languages Used			
Language Used:	Cases:	Language Used:	Cases:
COBOL	62	SQL	7
RPG	48	FORTRAN	6
DBASE	32	LOTUS 123	6
BASIC	21	UNIX SCRIPTS	5
C++	18	PL/1	4
C	16	ASSEMBLER	3
PICK TOOLS	14	CL	3
PARADOX	11	DATAFLEX	3
CLIPPER	10	QUEO	3
FOXPRO	10	Twelve different languages	2
PASCAL	9	Forty-three different languages	1
MS ACCESS	8		
(Unspecified)	8	Distinct languages specified	76

Table 6: Development languages used by non-CASE/4GL users

extent, FORTRAN, now nearly thirty years old, are still widely used. The results also illustrate a continuation of the trend reported by Botica (1987) for language use in this country. He cited a 1984 study in which 26% of respondents said that COBOL was used and 23% said that they used BASIC. This pattern was echoed in a 1987 survey of New Zealand developers (unfortunately, no sample size was given): COBOL 26.6%, BASIC 18.9%, Others 13.8%, RPG 13.1%, Fortran 6. 4%, Assembler 4. 8%, PL/1 2.6%. All of these languages feature strongly, in almost the same order of usage, in the responses to the current survey.

The final two sections of the survey were addressed only to those sites that employed CASE tools and/or 4GLs. It was originally intended that respondents would firstly specify the CASE tools they were using and then the 4GLs they were using. After an examination of both trade and academic literature, however, it was decided that the two product classes should in fact be left in one category, for the following reasons:

- As suggested in the introduction to this paper, the terms ‘CASE tool’ and ‘4GL’ have been defined in very vague terms. It was considered that this would make any objective and consistent categorisation of products unlikely.

- When first introduced, many tools were oriented towards assisting specific development tasks—some were related only to front-end activities, that is, analysis and design tasks (Burkhard and Jenster, 1989; CIS, 1989). In contrast, the focus of other tools was on the back-end of development, to assist personnel in the coding and testing phases (Norman and Chen, 1992). In recent years, however, increased product integration has meant that, in many cases, diverse products have been pooled to create a single development environment. Moreover, several products now address the entire development life cycle. This could be taken as confirmation of the assertion that 4GLs would be subsumed by CASE (Kolodziej, 1988; Stamps, 1989). It could equally suggest, however, that 4GL-type products have simply been provided with more effective front-end interfaces. Irrespective of these arguments, consistent classification of products given these circumstances was again thought to be unlikely.

Thus the remainder of the survey was completed by respondents whose organisation used a CASE tool and/or a 4GL—a set of 265 sites. This represents approximately 35% of the original respondent total (753 replies) and 60% of those who said they performed in-house development. It is difficult to compare these figures against those of previous surveys and reports as, in general, it is not clear whether the proportions cited in these reports referred to general organisations or only to those sites that performed in-house development. For illustrative purposes, however, some of the results are included here. King (1992) reported the results of two UK industry surveys, with both illustrating the failure of CASE to gain widespread industry acceptance. One survey of 250 UK companies found that 41% had no plan to use CASE, 27% were planning to use it, 25% were using it and 7% had used and rejected it. Similarly, a 1987 report from Australia revealed that, based on the responses of more than 500 sites, fewer than one in five Australian installations were using 4GLs, and only 4.6% intended to install a 4GL in the following year (Pacific Reporter, 1987). In terms of usage trends in this country, McAulay (1987) found that 38% of 120 New Zealand companies were using a 4GL of some sort. Botica (1987) reported a similar result—he found that fourth generation tools were being used by 31.3% of sites, based on an independent survey of an unspecified number of installations. Another 5% were considering using them, and 63.7% had no intention of using them.

Returning to the results of the current survey, more than 150 of the 265 CASE tool/4GL users said that they employed one CASE/4GL product, 57 sites cited use of two products, 32 respondents said that three products were used, seventeen organisations stated that they used four tools and five sites said that five or more CASE/4GL products were being utilised. The breakdown of usage over the various products is shown in Table 7.

CASE Tools/4GLs Used			
Product Used:	Cases:	Product Used:	Cases:
POWERHOUSE	33	SYSTEM ARCHITECT	5
ORACLE	28	AREV	4
LINC	23	ASSET	4
ADW	21	CUE-BIC	4
IEW	17	DEFT	4
ORACLE CASE	17	EXCELERATOR	4
PARADOX	17	IE:ADVANTAGE	4
(Unspecified)	17	SAS	4
FOXPRO	16	SUPERBASE	4
INFORMIX	15	ASAP	3
INGRES	12	CLARION	3
MS ACCESS	10	CQCS	3
LDA	8	EASYCASE	3
PROGRESS	8	GUPTA SQL	3
SYNON	8	IEF	3
DATAFLEX	7	NATURAL CONSTRUCT	3
NATURAL	7	PICK TOOLS	3
PACE	7	RTMAS	3
SPEED	7	Twenty-two different CASE tools/4GLs	2
PRIME INFORMATION	6	Forty-nine different CASE tools/4GLs	1
ALL	5		
LBMS	5		
RALLY	5	Distinct CASE tools/4GLs specified	111

Table 7: CASE tools and 4GLs used for in-house software development

The results presented in Table 7 illustrate the diversity of the CASE tool/4GL market, with more than 100 different development products cited as currently in use. The responses are dominated, however, by Oracle product users (45 sites), KnowledgeWare product users (ADW and IEW – 38 sites) and PowerHouse users (33 sites).

The next question in the survey asked respondents to indicate whether they used their CASE tool and/or 4GL products during more than one phase of development. The responses to this question are shown in Table 8.

It can be seen from Table 8 that the tools are used most frequently for assistance with coding—more than 80% of the CASE tool/4GL users stated that one or more products were utilised in this phase of development. Five of the remaining six phases also experienced strong CASE tool/4GL use, with between 50% and 72% of sites

CASE Tool/4GL Use During Phases			
	Yes	No	Blank
Feasibility study?	63	180	19
Analysis?	134	115	13
Design?	192	63	7
Coding?	220	33	9
Unit testing?	175	77	10
Implementation?	176	80	6
System testing?	165	82	15

Table 8: Development phases during which CASE tools/4GLs are used

using the tools at those times. Only the feasibility study showed an overall absence of tool use, with just 23% of product users stating that CASE tool/4GL support was employed for the relevant tasks. Although this could be interpreted as a reflection of a lack of product support for the feasibility study, it may simply be evidence of the assertion that support is generally unnecessary at this time, as the importance of this stage is relatively low (CIS, 1989).

The final product-related question in the survey was concerned with development organisations' analysis and design methods. The popularity of structured analysis and design techniques is widely acknowledged, as is the general expectation that CASE support for these techniques leads to increased product quality (CIS, 1989; Tse and Pong, 1989; Vessey *et al.*, 1992). Moreover, 4GLs are notably strong in their support for screen and report definition, often as part of a prototyping methodology (Keuffel, 1991; Jones, 1988). Given this type of support, widespread use of these methods was expected. The first part of the question therefore addressed the methods that were supported by the tools, and the second part considered actual use of the methods (irrespective of tool support). The associated responses are shown in Tables 9 and 10.

McAulay (1987) found that 47% of the 120 respondents to her survey did not use any structured analysis technique—data flow methods were employed by 25% of the sites, while data analysis techniques were in use in just 21% of the responding companies. The results presented in Tables 9 and 10 show that the use of data analysis procedures has increased markedly in the intervening six years. Even if it is assumed that none of the non-CASE/4GL sites use data modelling, the ratio of developers using data analysis has still increased by 9% to 30%. In contrast, the use of data flow diagrams (DFDs) appears to have remained stable at around 25% (under the worst case scenario). These results would appear to provide industry

Analysis and Design Methods With Tool Support			
	Yes	No	Blank
Data models?	127	103	26
Data flow diagrams?	103	124	29
Database operation definition?	115	99	42
Functional decomposition?	114	102	40
Screen/Report definition?	226	20	10
Other?	26	230	0

Table 9: Analysis and design methods supported by the CASE tools/4GLs used

Analysis and Design Methods Used			
	Yes	No	Blank
Data models?	132	76	48
Data flow diagrams?	106	102	48
Database operation definition?	100	93	63
Functional decomposition?	105	97	54
Screen/Report definition?	207	33	16
Other?	26	230	0

Table 10: Analysis and design methods used in development

support for the assertions that both data models and data flow diagrams are used extensively in development (Tate *et al.*, 1992; Gray *et al.*, 1991). Clearly the most widely employed method, however, is screen and report definition. Nearly 80% of the CASE tool/4GL users said that this technique was used as part of their development process.

The last pair of questions in the survey were rather more exploratory in nature, in that they addressed the number and type of systems produced using the organisations' CASE tools and 4GLs over the last three years. The results of these questions are shown in Tables 11 and 12.

3 Conclusions

If it is reasonable to conclude that 750 responses constitute an adequate sample then the results just presented provide a relevant and useful insight into software development in this country. These findings have implications not only for those directly involved in software development but also for those who teach it.

In terms of system implementation methods, the last five years have seen a

Number of New Systems Completed			
Number of systems:	Cases:	Number of systems:	Cases:
None	15	Eight	4
One	33	Nine	2
Two	37	Ten	15
Three	41	11-15	12
Four	32	16-20	2
Five	23	21-40	0
Six	20	41-60	3
Seven	6		

Table 11: Number of new systems completed with CASE tools/4GLs in last 3 years

Types of Systems Developed	
Type:	Cases:
Interactive transaction processing	112
Interactive reporting	88
Real time	71
Batch reporting	33
Interactive	23
All types	19
Batch	10
Batch transaction processing	9

Table 12: Types of systems developed using CASE tools and 4GLs

significant increase in the amount of package use when compared to other methods. Not only is the number of sites using packages for system implementation very high, at 85% of the respondents, but packages are now also used to implement the greatest proportion of systems. There may be several reasons for this situation—probably the greatest contributors are the increasing availability of high-quality packages and the high costs of in-house development. These findings also suggest that, along with traditional training in software development, a reasonably large component of package-based education might be equally appropriate in preparing students for the information systems workplace.

A large number of sites undertaking in-house development continue to use tools and languages developed prior to the CASE tool/4GL 'era', with 76 such languages being specified by 187 respondents. The strong use of COBOL and RPG may indicate that organisations are still making extensive use of some older applications and that the long-term investment in these systems continues to create a cost-based reluctance to migrate crucial and/or highly complicated systems to a new environment. Moreover, given that junior development staff are often assigned to maintenance tasks it would appear that teaching at least some COBOL remains a relatively sensible education strategy.

Just over 60% of the survey respondents who said they performed in-house development also stated that they used at least one CASE tool or 4GL, a figure much higher than expected given the slow initial acceptance of these products. This result indicates a significant shift towards the use of these products when compared to the situation just six years ago. Given that the proportion of in-house development has also decreased in this time, it could be concluded that these advanced tools have been purchased for the development of strategic systems, while the concurrent development of less important operational systems has continued either under a 3GL-type environment or through the use of package solutions.

The breakdown of CASE tool/4GL products in use highlights the diversity of the development tool market, with more than 100 products mentioned. The listing also illustrates the problem of product classification (as discussed earlier in the paper)—of the 41 CASE tool/4GL products in use at more than two sites, 17 of these also appeared in the non-CASE tool/4GL product list.

Of the development phases considered in the survey, tool support was used most frequently to assist coding tasks. This is almost certainly a result of the large number of programming aid tools available and in use relative to the number of front-end or full life cycle tools. The solid support for and use of various analysis and design methods provided no real surprises. A few respondents stated that they used a technique that was not supported, and *vice versa*, but in general those methods with tool support were widely utilised. These results also provide reasonably strong evidence of the continued use of structured analysis and design techniques, with data analysis and modelling being the most popular of these approaches. This represents something of a turnaround to the situation six years ago when data flow techniques were more widely used. In terms of development personnel education and training, these results suggest that equal attention should be given towards data and process analysis techniques, along with a consideration of screen and report prototyping methods.

In summary it can be concluded that the adoption of CASE tools and 4GLs has increased significantly over the last five years, so much so that these products now appear to be an integral component of the suite of tools used by organisations involved in software development. Given the tremendously high degree of response to the survey we intend to make it a regular event, so that trends can be identified quickly and accurately. Such an approach should enable all those involved—vendors, developers, managers and educators—with effective and up-to-date indications of industry product requirements.

Acknowledgements

The author is grateful for the support provided by the Information Science department at the University of Otago.

References

Botica, A., "Number Eight Style is Strength", *Computerworld New Zealand*, 27 July 1987, pp. 21-31.

Burkhard, D.L. and Jenster, P.V., "Applications of Computer-Aided Software Engineering Tools: Survey of Current and Prospective Users", *ACM SIGBDP Data Base*, Fall 1989, pp. 28-37.

Chen, M. and Norman, R.J., "A Framework for Integrated CASE", *IEEE Software*, March 1992, pp. 18-22.

CIS, *CASE Project - Report on Computer Aided Software Engineering in New Zealand*. Case Study Report, Center for Information Science, University of Auckland, May 1989.

Gray, R.H.M., Carey, B.N., McGlynn, N.A. and Pengelly, A. D., "Design Metrics for Database Systems", *BT Technology Journal* 9 (4), October 1991, pp. 69-79.

Huff, C.C., "Elements of a Realistic CASE Tool Adoption Budget", *Communications of the ACM* 35 (4), April 1992, pp. 45-54.

Jones, R., "A Quantum Leap in Languages: Major Gains in Using 4GLs", *Computerworld New Zealand*, 31 October 1988, pp. 18-19.

Keuffel, W., "Transformation Strategies", *Computer Language (USA)*, October 1991, pp. 35-42.

King, S.F., "Making CASE Work". Unpublished Research Proposal, Advanced Technology Centre, University of Warwick, Coventry, April 1992.

Kolodziej, S., “Fighting the 4GL Identity Crisis”, *Computerworld New Zealand*, 14 November 1988, pp. 32-34.

Maria, A., “CASE Technology: Today’s Reality”, *Journal of Systems Management*, February 1991, pp. 18, 26.

McAulay, K., “Information Systems Development and the Changing Role of MIS in the Organisation”, *Proceedings, 1st New Zealand MIS Management Conference*, Wellington/Auckland, March 1987, pp. S1.1-S1.16.

Norman, R.J. and Chen, M., “Working Together to Integrate CASE (Guest Editors’ Introduction)”, *IEEE Software*, March 1992, pp. 13-16.

Pacific Reporter, “4GL Sales Hype Baffles Aussie Users”, *Computerworld New Zealand*, 1 June 1987, p. 6.

Philipson, G. (ed.), *MIS 2000*. Strategic Publishing Group, Sydney, 1993.

Pressman, R.S., *Software Engineering: A Practitioner’s Approach*. McGraw-Hill (3rd ed.), New York, 1992.

Stamps, D., “CASE vs. 4GLs”, *Datamation*, 15 August 1989, pp. 29-32.

Tate, G., Verner, J. and Jeffery, R., “CASE: A Testbed for Modeling, Measurement and Management”, *Communications of the ACM* 35 (4), April 1992, pp. 65-72.

Tse, T.H. and Pong, L., “Towards a Formal Foundation for DeMarco Data Flow Diagrams”, *The Computer Journal* 32 (1), 1989, pp. 1-11.

Vessey, I., Jarvenpaa, S.L. and Tractinsky, N., “Evaluation of Vendor Products: CASE Tools as Methodology Companions”, *Communications of the ACM* 35 (4), April 1992, pp. 90-105.