7

Coping with Diverse Contexts for Decision Support

Andrew M. McCosh
University of Edinburgh
50 George Square, Edinburgh EH8 9JY, Scotland, UK
Tel.: +44131 6503 801, Fax: 44131 668 3053
E-mail: A.McCosh@ed.ac.uk

Abstract

A DSS which was written with one situation in mind may occasionally be moved to a different organisational environment. This will usually happen when there is a 'champion', or 'principal' for the DSS who wants to obtain the benefits in a new place. How should he go about moving it across so that it will continue to be a success? The paper presents stories about such transfers, and seeks to isolate the championing behaviours which worked well, the championing behaviours which did not, and the reasons.

Keywords

Champions, DSS, Adaptation, Organisational Environments, Contexts

1 INTRODUCTION

The focus of this book is upon the context within which decision support systems are created and used, and, especially, on the problems which arise when the contexts change. The contextual changes might take place in a single location as time passes, or they may occur because the DSS is relocated from one organisational setting to another. The present paper is addressing a number of changes of the second variety, and deals with a number of decision support systems which I have been involved in developing over a lengthy period of time. Some of these models have been relocated to other companies than the one which commissioned the model first, and, in some instances, I have had the opportunity to see what happened. Perhaps, the most interesting aspect of these relocations has been the way in which the person who wanted to use the model in a new location sought to introduce it to the second site.

Context Sensitive Decision Support Systems
D. Berkeley, G. Widmeyer, P. Brezillion & V. Rajkovic (Eds.)
© 1998 IFIP. Published by Chapman & Hall

The papers which discuss the theory of championship are immensely valuable in understanding how to introduce change. The entire profession of Organisational Design is dedicated to this complex task. The leadership literature is similarly blessed with a considerable selection of options for inspiring leaders and aspiring leaders to consider, as they try to work out the best way to bring the organisations they serve up to date. This paper does not purport to add to either of these literatures. The behaviour of the champions which I have observed in the cases to be presented below were not easy to classify in the terms of the change literatures as I have understood them. In fact, there is a very big gulf indeed between what some of the champions observed actually did, and what the change theory papers seem to be saying they ought to have done. An attempt will be made to leap this gulf towards the end of the paper.

Most of the models to be discussed are of a financial nature, and this fact has two consequences which may be quite important for the interpretation of the stories to follow. In the first instance, the financial orientation of the DSS models may make them slightly less organisation specific and slightly easier to move from one context to another than might be true if they had a different composition. If this assumption is valid, we shall see that it did not help very much, in terms of achieving a successful transition in two of the cases. Secondly, the models were used by financially specialised people and by a number of relatively senior general managers. This may mean that the results would not transfer to DSS which were intended for larger groups within a company workforce.

The four models have already been discussed in earlier conference proceedings, and will therefore be described in relatively brief form later in this paper. The first model was a mainframe merger analysis system, discussed analytically in Glover et al (1967). The second model to be discussed in this paper was an international bond switching algorithm, considered analytically by McCosh in Borcherding et al (eds) (1990). The third model (in order of appearance in this paper) was a price setting profit optimising model, initially discussed as chapter eight in McCosh and Scott Morton (1978). The fourth and last model was another merger analysis routine, which has gone through a large number of developments over many years, with the first description being offered analytically in Hawkins et al (1969) and, most recently by McCosh in Humphreys et al (eds)(1997). Each of these models has been put into use in one or more businesses, either by the present author or by a team which the present author could observe closely. In addition, each of these models was introduced into another business, usually without the active involvement of the present author or of the other members of the team of first installation. Usually, the only reason the present author knew about the second installation was because the champion was seeking permission to make the second installation or because the champion wanted help with the second installation, or both. The champion had moved to a new job, and wanted to use the DSS to assist him or her in fulfilling it.

The approaches adopted by the champions when seeking to make the second installation were interestingly different. I shall characterise their approaches as

being the survival adaptation routines of the mountain gorilla, the chameleon, the flounder, and the termite, respectively. I will try to demonstrate that the termite approach is the best. When confronting a situation or an environment which is new, uncertain, and might turn out to be dangerous, all animals have one or more standard procedures for coping with the new environment. The mountain gorilla will try to hide first, but if that does not work it will go on the attack, drumming on its chest and shouting war-cries in the hope of frightening the other party into submission or flight. This approach works fairly well for gorillas; it does not work quite so often for DSS champions, but it continues to be widely employed. The chameleon will normally react to an environmental change by making no movements whatever, and making no adaptations of its behaviour, but by changing the packaging. It has the capability to alter its skin colour to match its surroundings, and, thus, to blend, temporarily, into its new environment. This route toward achieving acceptance of a DSS in a new environment is very popular with consultants and software package salesmen.

The flounder is obliged to adapt when it reaches maturity. It is born with one eye on each side of its head, and feeds for the first part of its life on tiny sea creatures which it can catch. When it gets bigger, and hungrier, it finds it is not fast enough to catch its prey, and is forced to evolve into a bottom feeder. One eye moves round its head to join the other, so they are on the same side of its head, and it learns how to bury itself in the sand, ready to jump when prey come close. This thorough adaptation to the new situation is very effective indeed, but is achieved at great cost and considerable effort. Many flounders are taken by predators during the transition. The DSS champion using this approach will also find the cost to be acute, but the result may well be worth while.

The termite does not act as an individual. If there is a threat to the nest or hive, especially if an invading swarm of termites manage to breach the defence wall, the defending termites will divide into two groups. The construction crew will concentrate on rebuilding the wall and sealing it. The suicide squad will remain outside, fighting off the attackers, until they are all killed. An individual termite is disposable. The continuity of the hive is the thing that matters. A DSS champion using this approach will tend not to try to move a DSS to a new environment. Instead, he will adapt to context changes by throwing away the old DSS and using the ideas learned from it, perhaps also using the equations in it, to create a new one that fits the new situation.

2 RECOMMENDED BEHAVIOURS FOR DSS CHAMPIONS

The literatures on information systems and computer systems and office systems and decision support systems all contain advice on how to install the system most efficiently and effectively, and how to make changes. Donald Heany (1968) and Mumford and Banks (1967) got us off to a good start on the proper treatment of social processes in computer systems development and design and installation

problems. The two principal instructions they drew attention to and emphasised as vital were 'top management involvement in the project is essential if the project is to get anywhere' and 'the users must be involved in the design phase if the project is to be socially acceptable and, therefore, a success'.

It is apparent that these instructions have not been followed, or, alternatively, that they were not sufficient for the job at hand. Bashein (1994) has written a doctorate thesis entitled 'Re-engineering the credibility of Information Systems Specialists', which suggests the existence of a serious problem, whether his solution works or not. Allen (1995) has written about 'Succeeding as a Clandestine Change Agent', which seems to imply that getting a computer system changed has taken on the trappings of a military operation, and an undercover one at that.

The problems which these and other authors point to are numerous, but there are two propositions which cause specially serious trouble. First of all, the top managers who have been asked to take a leading role in the project, and to act as 'patrons' to the project champion, do this task with initial enthusiasm, but, then, start to wilt badly when the project starts to run late or begins to run into trouble in some way. This puts more pressure on the champion who may not have enough clout to keep the project in motion. Secondly, the involvement of the user population in a series of computer projects has equipped these users to rebel against the systems more effectively than they could in the earlier days when they did not understand what was going on.

There are, obviously, quite a few ways in which companies and information specialists and DSS champions can react to this situation. In this paper I merely report on two of the solutions which have been advocated by Markus and Benjamin (1997). First, they warn about the problems of the traditional approach. This depends on the thesis that the technology itself will bring about the change, and that the information systems person, whether he is the DSS champion or is acting with and on behalf of the champion, is not taking any action to bring about organisational change except to write the new system or to revise the old one. This attitude was the source of the problems which Heany, Mumford, Banks, and quite a few others, were complaining about thirty years ago. It is sad that Markus and Benjamin still find it necessary to regard this viewpoint as the basic one, from which we still have to make advances. I should make it clear that I do not suggest that Markus and Benjamin are wrong; merely that it is sad that no sufficient change has taken place.

The second change agent role that Markus and Benjamin discuss is one they call a '*Facilitator*', which is an approach drawn from the organisational design and organisational development literatures, such as the well known text of Cummings and Huse (1989). The orientation here is that of an OD consultant. This role is a very difficult one to learn how to fill because it requires the consultant to present himself or herself as an expert in getting things to happen, not as an expert in computing nor as an expert in the industry the company works in. If this method is done well, it is really quite effective because the consultant is there to help people

bring out the best of their own ideas and to try to get these to jell with the best ideas of the others. While I salute the two or three people I have met who can do this job, I regret to report that I just do not think there are enough of them to fill the need.

The third and last of the Markus and Benjamin list of change agent roles is that of *advocate*. The advocate agrees with the OD consultant that the technology does not change anything, it is the people who do that. The advocate does not see his job as being confined to bringing the best out of the employees with whom he is temporarily associated. The advocate has ideas for the direction the change should take. The advocate is not afraid to press his ideas for improving the situation on the champion, the patron, the employees, and anyone else who will listen. This approach can be very effective, given the right sort of advocate, provided the advocate is not believed to be a 'captive' of one of the parties to any disputes in progress. It is easier to fiind good people to fill this role in the business world than to find good OD consultants, in my experience.

When we come to consider the case stories, the principal character will invariably be the end user, the patron, or the champion, or all three. As such, the role to be fulfilled is clearly one of advocacy, in the sense Markus and Benjamin have used the term, but it also involves allocation of budgetary resources, and control of, at least, a part of the corporate agenda. The problem of having a failure of top management support did not arise in these situations, or, at least, it had been overcome before any observations of the situation could be made. We will call this role the '*principal*'.

3 THE DSS TRANSFER SITUATIONS

In the descriptions which follow, we will look at a condensed discussion of four imperfectly observed situations. It is important to note that this was not a controlled experiment. Acting as a practitioner, over a lengthy period of years, I worked with several colleagues on providing DSS models to interested users in a broad range of companies and government departments. Almost all of the ones I got involved in were financial, and financial executives and employees tend to be quite mobile. When they move to a new job they would require certain facilities, including some DSS support in the form of languages or staff or models. Occasionally, they would decide to buy, for their new company, a copy of a model which had been installed in the company from which they had moved. In four of these cases, the second company's purchase became known to me, usually because the executive wanted to ask a question about how the system worked or how he could alter it or whether we would change it for him or her.

In each instance, we will look at what the model did, what the nature of the first company was, what the nature of the second company was, what steps the principal took in trying to get the system to work successfully in the second company, and what sort of success it enjoyed in its second home.

Case One: A Mainframe Merger Search Model

The model was initially created for a highly acquisitive engineering firm, specialising in the manufacture, maintenance, and repair of huge power generating machinery, during the very active merger wave of the mid nineteen sixties. The model was designed to help the chairman of the company decide which company to buy next. The financial history of the engineering company and those of sixty possible acquisition candidates were loaded. A series of merger deals would be projected for each of these candidates. The model then printed out the ones which looked as though they might be sensible to acquire, together with a draft of the terms of the deal, designed to make it profitable for the parent firm and attractive to the shareholders of the candidate. (If you want to know a lot more about the model, please read Glover et al, 1967. If you do not want to know anything more about the model, please skip to the end of the italicised section just below. To learn a little more about the model, please read on.)

The model was written initially in Fortran II, with later editions in later editions of Fortran, a few versions in Algol, and the present edition is in C++. The first edition used a punched card data file, while now the data files are normal disk memory files. About three hundred to five hundred data items are required for each of the companies being examined for acquisitions, together with eight hundred for the parent company. The share prices changed by the hour, some of the other variables changed weekly or monthly, and most of them changed only once a year. In the early editions, this required a lot of card punching. In the present edition, the data are stripped automatically from a database purchased from one of the numerous data supply companies which serve the investment industry. Every day, at least in the early years, the model would be run and a printout presented to an aide to the chairman of the engineering company, who would decide whether there was enough new news in it to take it in to the boss. The model was designed to take the financial figures, and certain assumptions about possible cost saving possibilities and capacity economies. These would be combined for the two companies, and estimates would be made for the value of the sales and earnings patterns for future years. These future earnings streams would be evaluated against the quantity of the shares and debt instruments which were to be issued to make the acquisition. In the early editions, ten standardised acquisition packages were tested for each assumed future earnings pattern for each of the sixty companies for each of the next ten years. In later editions, a larger number of packages was tested, and a smaller output was printed about each of them. The taxation assumptions were incorporated on the basis of the tax laws prevailing at the time the model was being run, and many of the later editions were only different from their immediate predecessor because of a tax law change.

The aide to the chairman of the engineering company left the company in 1974 and went to work, as Vice-President for Development, for a company which made instrumentation and electronic devices of various kinds and applications. I received a letter from him instructing me to send a copy of the model to his new address and ordering me to instruct his new team of subordinates in its use. I attended on the appointed day, and was allowed four hours to explain the model to the team. Thereafter, I was instructed not to visit the premises again because the data were extremely confidential, and the VP wanted to make sure that there were no possible leaks, and also wanted it to be easily seen by his President that there was no possibility of a leak via a non-employee. I did not mind this; confidentiality is a very big issue in merger work. More controversially, however, he insisted that any questions his troops wanted to ask me had to be fed through him. He had no computing skills, and garbled some of the technical questions rather seriously.

It was obvious that I was not going to obtain any information on this case from him or from his team. However, one of them quit from the company, and happened to choose to undertake doctoral studies in a university at which I was then employed, though not under my supervision. The rest of this report rests on what the doctoral student told me.

The model worked fairly well at the outset. The electronics company had nearly as many possible acquisition candidates as the engineering company had. The financial data were easy enough to find, and the tax computations were identical. The company subscribed for the next edition when the tax law changed, but, otherwise, had absolutely no contact with me or with any of the partners in the software company. None was felt to be necessary. The Vice President did not allow any discussion of the procedures that were to be used. The model was to be run each day. The data files were to be kept up to date by a separate team, who did not see any of the output. The man who quit to become a PhD student had the task, amongst other things, of adding companies to the data set which were not being considered for acquisition, so as to keep the data team as confused as possible about what was happening.

Eventually, however, the special characteristics of the model which made it work well for the heavy engineering company began to be annoying to the electronics company President. The model worked on the assumption that the principal synergies arising from the merger would be cost savings, and, possibly, capacity consolidation. This was, by and large, valid for heavy electrical engineering. It, certainly, was not valid for instrumentation, and was valid for only a small part of the electronic equipment division. Most of the synergies in that company arose in the area of marketing, distribution, and sales. The real reason for buying out another electronics firm was to increase the speed of its growth above what its own finances could handle. It was not a matter of shutting down a duplicated plant.

During this period, quite a few mergers were put into effect. The sums worked out quite well. The model was being used in a rather basic fashion, but reasonable

merger candidates were identified. It was only later that the need for a different way of working became apparent. It took time before it became clear that the model had a cost-saving bias and that the growth financing approach could only be handled in a clumsy fashion. The acquisition candidates chosen by the model were perfectly sound cost saving or cost reduction or capacity elimination merger candidates, but, at that time, the parent company should have been going for innovation and growth opportunities. The situation was not a failure, but it could be looked upon, from the corporate view, as a missed opportunity.

Within the Development Department, it might be thought that the mountain gorilla approach had worked quite well, at least from the point of view of the mountain gorilla himself. Any member of the Development team who argued had been shown the door. Anyone who asked too many questions was told to look after his/her own duties and not get above themselves. Quite a few mergers were put into effect. The sums worked out quite well. The model was being used in a rather basic fashion, but reasonable merger candidates were identified. The bias in selection, towards cost reduction, did not become obvious for quite some time. None of the employees told the VP about this problem. It was not seen as safe. This is not really too surprising. Who is going to tell a mountain gorilla that he is making big mistakes? Especially when some definite victories are in the record?

Case Two: An International Bond Switching Decision Model

The second model was developed for a small international trading bank. The model was designed to assist the 'International Fixed Interest Unit' of the bank to decide when they should move from one market to another. Specifically, it was designed to tell them when to swap a Japanese bond for an American bond, and when to swap back again. The yield on a bond will vary considerably from day to day, but there is a tendency for macro-economic reasons and for reasons of risk and security for the yields to move roughly in step with one another. They do not move precisely in step, however, which is the reason why a bond switching model is worth creating. The 'yield gap' between a Japanese bond and a US bond varied from three per cent to five per cent during the period in the late eighties when the model was first built. This variation is enough to make it worth while for a trading bank to operate a switching model. The rule of thumb was to buy the relatively high yielding bond. At yield gap levels near five, that bond would be the American one, while at a yield gap near three, the relatively high yield bond would be the Japanese. Similar models could be developed for other pairs of currencies, and larger scale models are available for managing multiple currencies at once.

Although the bank was small, the unit was a very active one, and traded its bond portfolio frequently, within a limited range of fixed interest investments, with which the relatively small team felt they could stay completely familiar. (Readers who want to know more about the model are referred to McCosh (1990). This model was too complex for the creation of an italicised paragraph of the kind provided for models one and three).

The fixed interest team became rather well known in the medium ranked money centre which was their base, and, inevitably, this led to attempts by other institutions to persuade members of the team to change allegiance. One such attempt, which succeeded, involved the person who was second in command of the team, who was 'poached' by a fixed interest investment company in 1990. She asked permission to purchase a copy of the model from the bank and this was arranged. The bank did not consider itself to be in competition with the investment company and was quite willing to recoup some of its DSS development expense.

This person had been active in the unit at the time the DSS was first developed and was one of the two persons who were the product champions for its creation. She wrote to me when she got the new job, reported that she was taking the model with her, and said that it would make just as much money for her firm in the future as it had in the past. She had persuaded her new company to buy the DSS, she reported, because it was 'a good investment tool', and, therefore, naturally of interest to an investment company. I was a bit suprised to hear this.

This was an illustration of the chameleon approach to DSS context change management. Leave the system or model exactly the same as it was before, but change its packaging. Give it a new name. Stop calling it 'an international bond switch timing and trading DSS' and start calling it 'an international fixed interest medium term investment selection model'. Some of the people you are talking to will think they are the same thing anyway. Some of the people will not understand either phrase and will take your word for it.

There may, however, be some who will wonder how a model that is designed to facilitate short-term trading in a banking situation, where the portfolio is turned over two or three hundred times a year, can be the right answer for an investment company which holds its investments for lengthy periods, turning the portfolio over only two or three times a year. There may, alternatively, be some who wonder how a yield gap approach to investment management, requiring a large number of trading transactions, can possibly be more productive in the long run than a simple strategy of buying the security and holding it.

Two years later, I met this person at a conference. Her conference name badge revealed a new corporate affiliation. Had she brought the model with her to this third company? No, her new bank already had one which was similar. Had she found the model useful at the investment company? 'Well, not really, we seemed to use it so seldom that I kind of forgot how it worked between applications and, I am afraid, it wound up in a cupboard'.

Being a chameleon is clearly a hazardous business. Sometimes just changing your colour or your packaging is not enough to do the trick. A DSS which has been written for one situation and, then, repackaged for use in another situation without changes to its character and specification and without being subject to a very serious overhaul and a significant amount of re-tooling is to be viewed with great suspicion. As a moderately active, though part-time, consultant, I am rather ashamed to admit that quite a few chameleon-type DSS context changes have

originated from the consultancy profession. I have not deliberately done that myself, though it looks as though I contributed to the occurrence of this one, unwittingly.

Case Three: A Consumer Product Price Setting Model

The third case was a model which was developed as a conceptual DSS for a chapter of a book on DSS applications. The model was designed to cope with the problem that the best price for a company to set for a product is a function of a considerable array of variables. The overall demand for the product in physical volume terms is likely to be a function of the price charged for it, relative to the prices of competing products; it is likely to be a function of the quality of the product, relative to competing products; and it is likely to be a function of the amount of promotional effort put behind the product, relative to competition. It is possible, of course, to add further features but the present model is confined to the three listed.

The book chapter (McCosh & Scott Morton 1978, Chapter 8) was spotted by a consumer products firm in Holland, and a rather major project was undertaken by them to create a model for one of their product ranges. (If you want to know a lot more about the model, please read Glover et al, 1967. If you do not want to know anything more about the model, please skip to the end of the italicised section just below. To learn a little more about the model, please read on.)

The model was written initially in Dartmouth Basic and remained in that language for many years. It was, eventually, converted to Algol and then to C. The procedure was for the relationship between physical demand and price to be estimated by a graphical interactive estimation process which led to the creation of a hyperbola linking the two values. Then, a separate hyperbola was estimated linking demand to advertising expenditure. Finally, the manager was asked to estimate the linkage between demand and the quality of the product. In practice, this turned into an estimation of the link between demand and product cost. When these three hyperbolae had been estimated, a model was assembled of the form:-

$$D = K \times Y_{price}^{a} \times Y_{adv}^{b} \times Y_{Qual}^{c}$$

The Y values were obtained from the three separate single variable hyperebolae, each of which took the form (Demand - K1)(Price -K2)=K3. The three hyperbolae were used in combination to show the manager an estimate of the demand for a given price level, a given advertising spend, and a given product cost. He was able to make an input to the effect that the estimate was too high or too low, and the a, b, and c values would be reestimated, and a new value produced. The process was repeated until the manager was satisfied with the model's performance, or until he decided to change one of the three basic models, or until the general model became unstable, in which case it was necessary to restart. The outcome was a general

pricing model which enabled the optimal price for the product to be estimated, given a planned promotion spend and a known quality specification.

The Dutch company employed this model for consumer products for at least ten years, and may still be using it. One of the employees of that company, who was very close to the substantial task of creating the model in its operational form, was transferred to another consumer products company within the same group of companies. This Dutch company is a very substantial enterprise and has several dozen subsidiaries making consumer products, and each of these is a rather significant firm. The transferred employee could have, very easily, attempted a chameleon change in this situation. The product ranges of the two subsidiaries were quite similar, and it must have been tempting to try the model out, virtually unaltered, in its new home, to see how it went. However, she did not do this. She adopted the flounder strategy for change, the most thorough of the four.

The young flounder goes through a complex, lengthy, and rather dangerous period of transition as it matures into its new life as a bottom feeding flatfish. It totally changes its lifestyle and redesigns itself radically so as to be in the optimal condition to cope with its life as a mature flounder. The transformation undertaken by the pricing model, under this champion's direction, was of the same level of profundity and, also, it must be said, of the same order of cost.

The task of adapting the pricing model to its new context was the subject of a full corporate project, with a budget, with specific staffing assigned to it, with a separate billing number in the accounting system, and with a hierarchical reporting relationship which placed it under the subsidiary's director of marketing.

The first stage of the conversion involved trying the model out in its existing format, to see whether it recommended the same prices as were being generated by the existing procedure. This was a subjective, judgemental, and manually operated system under which the senior executives would consider the results of internal tests of all competitive products before setting the prices. One of the team called the process 'informed intuition', or II. The actual selling prices during this period were determined by the II approach, not by the model. In these tests, quite significant differences were observed between the model price and the II price. Each of these differences was examined and an attempt made to explain what had caused it. This comparison ran for a four month test period.

At that time, the model was revised. The new edition was named 'Prototype 2', which made it clear that this was still a trial. The most obvious difference was that the three relational equations were expanded to four, with quality and cost being treated as separate items. A less obvious change was that the equations were changed from being hyperbolae. The price equation became a quadratic, the quality equation became a Gompertz curve, and the other two became parabolae. Another four months of experimentation were conducted. At a meeting of the subsidiary board after that experimental period, the decision was taken to make further

alterations to the model, to continue the II system, but to use the model values for some of the real selling prices.

Prototype 3, as this version was named, differed from prototype 2 in that the two parabolae were replaced by Gompertz curves. After a further test period, Prototype 3 was renamed as 'The Pricing Model', and it was used, from then on, as the main way to set a price. Some of the testing procedures from the old system were continued as a basis for informing the quality equation.

The flounder approach to context change management is expensive. I tried to find out how much they had spent on the conversion exercise but they would not tell me. I asked if it had been within the budget, and they would not tell me that either. I asked if it had been a success and if it had been value for money, and they said it certainly had been both of these.

Case Four: An Interactive Merger Search and Deal Structuring Model

This fourth model has a substantial amount in common with model one, in that it is designed for the task of assisting a company to locate a possible acquisition target. It differs, however, in that it went much further through the process, and allowed for a much greater degree of analysis of the specific deal structure. This model has been through so many editions and versions that I do not know where we are up to in 1998. The first version was discussed in Hawkins et al (1969), and a fairly recent version is discussed in McCosh (1997). The theory of mergers and acquisitions, as a financial exercise, is very clearly set out by Franks in the Financial Management Hnadbook (1988) and there is a good discussion of the American equivalent in Brealey, Myers and Marcus (1995) at chapter 22.

This model is a classical DSS. The two principal components of any DSS are the manager who has a wealth of experience in the problem at hand, and the computer which can hold the models and the data and can do some of the leg work. In this instance, the man-machine synergy is particularly significant. The interaction between the manager and the model is continuous, as each of the possible acquisition candidates is studied in turn, and as the thousands of possible permutations of the deal are explored. An acquisition may be made using cash, or shares, or preference shares, or bonds, or debentures, or warrants, or incentive payments, or future bonuses, or any mixture of these, or even (conceivably, though I have never seen it done) all of them. The premium on acquisition over the previous market capitalisation of the candidate may be anywhere, from minus ten per cent to plus five hundred per cent.

In the early days of DSS, this model could confer a competitive advantage on the merchant bank that had one. It did not take long before every merchant bank had one, and ever more complex routines were built in to them to cope with the tax and competition regulations which governed the transaction.

It became clear, not just to me but also to others who were engaged in creating these models, that the termite approach to coping with context change was the right one. The idea of a disposable model has been with us for many years. The concept

of prototyping has been in the literature for a long time, but the idea of having a model that was deliberately created as disposable is more recent. The termite approach entails sustaining the essence of the DSS while letting each specific instance of the concept live its life and then die. Protect the hive (the DSS essence) against all attacks, but let individual termites (the specific models) be killed off.

The first instance of this model was put into operation for an investment bank in the USA and was, later, sought by a number of client companies which did not want their search routines to be known to their investment bankers. The clients were also very reluctant to be dependent on the investment bankers for their search and analysis methods as many of the bankers worked for their potential enemies as well as for them. The confidential elements were, therefore, to do with the application of the model, and the data sets loaded into the model, and the deal structures created through using the model, rather than the mathematics performed by it. The models had to be designed and built so that they could interact with data sets, some of which were in the form of private files and others were commercially available reference sources.

The models had to be designed, for presentation purposes, to conform to the expectations and preferences of assorted boards of directors, who had very varied opinions about what was important. The output from the models varied in scope and scale: from a postcard, bearing only a single target company name and a total bid value, all the way through to an encyclopaedia, covering hundreds of companies with up to a dozen deal possibilities each.

The first approach taken was that of the flounder. The first model, as in Hawkins (1969), was transferred over to the second client, and a thorough programme of checking and adaptation was commenced. The whole episode was really quite like case three above. It was a successful installation but it took a very long time. The third application of this technology was, therefore, done on a termite basis, and this approach has been used ever since, at least if I have been doing the project.

The second edition was not carried over to the third site, but was abandoned. A completely new model was assembled, using components and using ideas which came from the theory or came from the first edition or came from the second edition. The components were, gradually, separated into sections. Linkages to data sets were created anew. The social needs of the various boards of directors for acceptable data forms was met by rather slavish copying of the report formats which they had been receiving before. No attempt was made to take a complete DSS and relocate it. It is, perhaps, important to note that there were substantial differences between these models. The companies were interested in different things, and the manufacture of a DSS to suit these interests was essential to the success of the project.

A crucial development took place when it started to become necessary to put in a new model in the companies which were already using an earlier edition. It was at this point that the termite system really came into its own. The old termite was

left to die. A new system was built, from scratch, using new technology, new laws, some new data-sources, and, sometimes, a new computer language. At the same time it continued to use the familiar data sources and to generate familiar looking reports. It was very much quicker and very much easier to test for accuracy than would have been the case if we had used the flounder route, and it was, invariably, less costly.

4 A COMMENT AT THE END

I do not think it proper to claim that I am going to arrive at conclusions, so I call this section a comment. I have offered this paper as a series of stories about the problems which arise when a DSS is moved from one context to another. Even in the rather narrow confines of finance, the job of getting a DSS to work in a new situation is subtle, complex, and difficult.

I referred earlier to the ideas of Markus and Benjamin and others concerning how to be a change agent. They were suggesting the roles of 'facilitator' or of 'advocate' as the best means of achieving improvements in a company's fortunes through information systems. I really doubt the feasibility and effectiveness of a facilitator approach when the problem at hand is a change of context of a financially orientated DSS of some technical complexity. There seems to me to be a very definite need for an advocacy approach, where the principal gives positive leadership to the project from a standpoint of some authority. That authority may be derived from knowledge, from experience, or from a hierarchical role. The facilitator idea is appealing, in bringing out the talents of the team-in-place. The problem is one of credibility. It just does not seem likely that a facilitator could achieve any respect among the teams unless he had significant knowledge authority. Conversely, if the change agents had that knowledge, they would be very likely to have advocate-like (and fairly firm) opinions on what should be done.

If it is accepted that advocacy is likely to be the approach with the best chance, we still have to select among the four roles we have considered. Academic readers may feel that choosing among the four methods is very easy. They might say that the mountain gorilla approach is doomed from the start, that the chameleon is virtually fraudulent, and that the only debate is between the flounder and the termite, each of which might be thought applicable in different situations. Business readers might well differ. There is nothing, they might argue, quite like a good gorilla for getting a satisfactory but, perhaps, not optimal solution into place on time and under budget. The chameleon illustrated, they might continue, just got a bit unlucky, and could easily have been successful if she had adopted a slightly wider search pattern. Flounders and termites sound a bit expensive. The jury is still out on the question. I hope the characterisations are helpful as you consider your own views on the topic. What do I think? No question about it. I have been a termite for years.

5 REFERENCES

Allen, C.D. (1995) Succeeding as a Clandestine Change Agent. *Communications of the ACM*, **38-5**, 81-86

Alter, S. and Ginzberg, M. (1978) Managing Uncertainty in MIS Implementation. *Sloan Management Review*, **20-1**, 23-31.

Bashein, B.J. (1994) Re-engineering the Credibility of Information Systems Specialists. California State University at San Marcos, Working Paper.

Brealey, R.A., Myers, S.C. and Marcus, A.J. (1995) Fundamentals of Corporate Finance, International Edition. McGraw Hill, New York.

Cummings, T.G. and Huse, E.F. (1989) Organisational Development and Change, 4th edition. West Publishing, St Paul Minn USA.

Curley, K.F. and Gremillon, L.L. (1983) The Role of the Champion in Dss Implementation. *Information and Management*, **6-4**, 203-217.

Franks, J.R. (1983) Acquisitions and Mergers, Chapter 16 in *Financial Management Handbook*, (eds J. Broyles, I. Cooper, and S. Archer) Gower, Aldershot, UK.

Glover, J.D., Hawkins, D.F. and McCosh, A.M. (1967) Computers in Merger Analysis. *Mergers and Acquisitions*, **2-4**, 17-32.

Hawkins, D.F., McCosh, A.M. and Lampe, J.C. (1969) Time-shared in Merger Analysis. *Mergers and Acquisitions*, **4-1**, 11-19.

Heany, D. (1968) Development of Information Systems. Ronald Press, New York.

Markus, M.L. and Benjamin, R.I. (1996) Change Agentry - The Next IS Frontier. *MIS Quarterly*, **20-4**, 385-407.

Markus, M.L. and Benjamin, R.I. (1997) The Magic Bullet Theory in IT-enabled Transformation. *Sloan Management Review*, **38-2**, 55-68.

McCosh, A.M. and Scott Morton, M.S. (1978) Management Decision Support Systems. Macmillan, London.

McCosh, A.M. (1990) Supporting Group Decisions in Bond Trading, 333-352 in *Contemporary issues in Decision Making*, (eds K. Borcherding, O. Larichev, and D. Messick), North Holland, 12th (Moscow) Conference on Subjective Probability, Utility, and Decision Making, Amsterdam.

McCosh, A.M. (1991) Support for an Expertise-based Group, 257-276 in *Environments for Supporting Decision Processes* (eds H.G.Sol and J. Vecsenyi), North-Holland, IFIP Working Group 8.3, Amsterdam.

McCosh, A.M. (1997) Corporate Upsizing: The Evolving Role of DSS in Mergers and Acquisitions, 211-222 in *Decision Support in Organisational Transformation* (eds P. Humphreys, S. Ayestaran, A. McCosh and B. Mayon-WHite), Chapman Hall, IFIP Working Group 8.3, London.

Mumford, Enid and Banks, Olive (1967) The Computer and the Clerk. Routledge and Keegan Paul, London.

6 BIOGRAPHY

Andrew M. McCosh has taught at the Universities of Harvard, Michigan, Manchester and Edinburgh. His teaching specialities have mainly been in finance, accounting, ethics, and information systems. He is now Professor of Finance Emeritus at Edinburgh University, Scotland. He is also a director of the Lexington Mutual Funds of New York. He was elected Vice-Chairman of Working Group 8.3 of IFIP in 1996, having been a participant in the working group since the Vienna conference in 1982. His research interests have been concentrated in DSS applications in the fields of finance and investment and business analysis, in the theory and practice of management accounting, and (most recently) in the new field of financial ethics.