Pergamon

Journal of Accounting Education, Vol. 15, No. 1, pp. 53–69, 1997

0748-5751/97 $17.00 + 0.00

PII: S0748-5751(96)00042-5

COMPUTER PACKAGES AS COGNITIVE PARADIGMS: IMPLICATIONS FOR THE EDUCATION OF ACCOUNTANTS

Michael Wood
Philip Cahill
James R. Hicks
UNIVERSITY OF PORTSMOUTH

Abstract: Computer packages are playing an increasingly dominant role in the life of most business organisations, which in turn is reflected in a greater role for packages in education. These packages both presuppose knowledge on the part of the user, and also "contain" knowledge of accounting as well as mathematical and other concepts. This paper suggests that packages function not just as information processing tools, but as "paradigms" which structure the approaches taken to cognitive tasks—although there may be an important distinction between expected and actual paradigms. The change to package-based paradigms represents a substantial shift in cognitive framework. The paper discusses the implications of this for package users and for the educational process. Three important themes are emphasised: first the importance of students learning about packages on a "meta" level rather than simply learning to use them; second the suggestion that treating packages as "black boxes" may sometimes be inevitable and desirable; and third the possibility that computer packages may, to some extent, serve as a substitute for education. © 1997 Elsevier Science Ltd

INTRODUCTION

Computer packages are playing an increasingly dominant role in the life of most business organisations, which in turn is reflected in an increasing use of computer packages in the education of students. This paper argues that these packages function not just as information processing tools, but as the basis of cognitive paradigms which are substantially different from their predecessors. For example, packages will have implicit and/or explicit "rules" which may impose a particular mode of analysis. The use of menus —which inevitably constrains user choice—is a good example of this. In addition, different types of packages may support different approaches to problem solving. The ability of spreadsheets to deal with large volumes of data in a highly structured and systematic way may encourage the "brute force" application of arithmetic to problem solving when in fact more appropriate mathematical techniques may be available as viable alternatives.

Package-based paradigms represent a significant change in how people reason and process information. Any such changes must have a substantial impact on the educational process. For example, packages presuppose knowledge on the part of the user and also "contain" knowledge—both of which raise obvious questions about what potential users of the package need to know. As it is likely that long-term attitudes to the new computer-based paradigms are developed in the education process, this paper suggests that there is a need for particular sensitivity to differences between the new paradigms and the old ones. We single out three important themes: the importance of students developing a critical perspective on the strengths and limitations of packages as knowledge structuring devices; the need to recognise the limits of this critical perspective since the complexity of packages and of the background knowledge behind them means that students will often need to treat them as black boxes; and the possibility that computer packages should be regarded as a substitute for some aspects of training or education.

It is necessary to mention briefly what this paper is not about. It is not concerned with "computer assisted learning" (CAL) in any of its guises, but is concerned with the influence of business packages on the educational process. Our main conclusions will concern what is worth learning, not how it is best learned.

COMPUTER PACKAGES AND KNOWLEDGE OF ACCOUNTING: PACKAGES AS COGNITIVE PARADIGMS

Computer packages and their users have much in common with "normal science" as described by Kuhn (1970). Packages may be said to define cognitive paradigms in a similar way to that in which Newtonian mechanics defines a scientific paradigm.[1] According to Kuhn "normal" scientists are essentially "puzzle solvers" working within a well defined tradition, or "paradigm". These puzzles typically concern "the proper way to connect [the scientist's] own research problem with the corpus of accepted scientific knowledge" (Kuhn, 1981, p. 108). The education of normal scientists consists of training in standard techniques from textbooks and working through "examples". In this way students learn to solve puzzles by employing standard, accepted approaches. Part of the body of knowledge learned is explicit rules and part of it is "tacit knowledge".[2]

---

[1]We use the term paradigm to refer to the entire information system and approach to the processing of accounting data and the solving of accounting puzzles. In the case of package-based paradigms this incorporates users and packages (viewed as partners who co-operate in various cognitive tasks) and the whole constellation of explicit and tacit knowledge and conventions surrounding the use of the package. Where a package is not employed the accounting paradigm incorporates users and the explicit and tacit expertise which is represented by their traditional working methods.

[2]Michael Polanyi has emphasised the importance of "personal knowing" which "operates by an expansion of our person into a subsidiary awareness of particulars" (Polanyi & Prosch, 1975, p. 44). This personal knowledge is an "integral part of all knowledge"—including scientific knowledge. Thus scientific knowledge has its tacit dimension in just the same way as any other area of expertise.

Normal science is characterised by the fact that practitioners typically do not question the basis of the subject: "it is precisely the abandonment of critical discourse that marks the transition to a science" (Kuhn, 1981, p. 110). The only questions that are asked are those allowed by the rules of the "paradigm". This has led many commentators to point out how narrow the vision of a "normal" scientist is.

The user of an accounting package appears to be in a very similar position to the normal scientist. Undertaking a spreadsheet cost analysis or the plotting of a histogram are puzzles within a given framework. The difference here is that these accounting puzzles are not solved within the "corpus of accepted scientific knowledge", they are solved in the context of the given software and documentation. These puzzles may not be trivial: even producing a histogram with a spreadsheet may require considerable ingenuity if the spreadsheet does not incorporate the appropriate built-in routines (Snell, 1991).[3]

Packages, like scientific paradigms, may be powerful, but the power is often at the expense of a rigidity which makes them difficult to adapt to new circumstances. The only data and format that the user is permitted to enter and the only analyses which can be performed are those allowed by the package. The package will control the style of reasoning and problem solving that users can engage in. The development of a package also seems to follow a similar pattern to the development of a scientific paradigm: minor changes are made but always in such a way as not to challenge the basic structure of the package.

The term "paradigm" can be used on a number of different levels. Spreadsheets in general, and their users and uses, constitute a "spreadsheet paradigm". At a lower level, Lotus 123 and Microsoft Excel (two spreadsheet packages) may be regarded as defining two more specific paradigms. The situation is complicated by the fact that different groups of users may interact with the same package in very different ways: these then clearly have to be regarded as separate paradigms (see the section on actual package-based paradigms below).

It would also be possible to talk of a general, high level paradigm encompassing all computer packages. This could then be contrasted with the pre-computer paradigm. In this paper, for reasons of clarity, we will

[3]Spreadsheets have built in routines for standard types of diagram. Unfortunately a histogram, although a relatively common type of statistical diagram, is not on the menu of many of the older spreadsheets. Snell shows how to build a histogram from the commands for drawing lines between specified points and other routines which are on the menu. This turns out to be a fairly complex puzzle. More recent spreadsheets—for example Microsoft Excel—do have a menu option for histograms so users do not have to solve this puzzle. However, there are inevitably other omissions in these newer packages which lead to further puzzles for users to tackle.

not use the word paradigm in this sense. Instead, we will refer to the shift from a pre-package "cognitive framework" to one which depends on computer packages.

Expected Package-based Paradigms

It is convenient to start with *expected* package-based paradigms—where the system works as expected by the designers of the package.

Accounting packages of all types both incorporate expertise and also assume some expertise on the part of the user. The incorporation of expertise is most obvious with expert systems but is also true of other types of package. For example, spreadsheets have the formulae for calculating net present values (NPV) built in, and payroll packages have the "expertise" to calculate net pay.

The nature of the expertise assumed of the user is a more subtle issue. First, the user must understand what the package does and what it can be used for (Wood, 1989). Second, the user is assumed to be familiar with concepts and terminology, processes and regulations incorporated in the package. For example, terms such as "cash", "credit", "debit" and "net present value" have technical meanings and refer to underlying concepts of which users need to be aware. Clearly, different packages make very different presuppositions about the expertise of their users, and incorporate expertise of very different types and levels.

In addition to explicit expertise involved in using a package (which may for example be expressed as a formula or an account of the meaning of a concept), there is also a tacit dimension to the process of using a package fruitfully.

The Tacit Dimension of Package-based Paradigms

There is a strong argument that any human cognitive process involves a tacit component (Polanyi & Prosch, 1975). This is likely to be as true of package-based paradigms as of any other paradigm. There are two (at least) important general respects in which the tacit component of a package-based paradigm may differ from that of other paradigms. The first concerns users' access to contextual information, and the second concerns the style of reasoning imposed or required by a package.

Manual accounting records consist (ideally) of a series of explicit files which may be accessed easily and operated upon in ergonomically efficient ways. The manual general ledger can be opened without the need of an electronic box or a password. Pages can be read, annotated and compared to several other records quickly and easily. Such a record is easy to access and operate upon in a way which becomes second nature for skilled users. The computerised general ledger, by contrast, appears as a vertically presented keyhole view of an accounting record. This ledger must be accessed via a defined sequence of keystrokes. Only one page is visible at a time. The pages are smaller than the manual equivalent. They cannot be manipulated, annotated, or compared to other records as easily as their manual counterparts. Automating a process by means of an accounting package may mean that users are no longer able to look at details such as the context of particular data items, and to observe the process by which the data is manipulated. This means they cannot keep an adequate intuitive eye on the process because they have no easy access to the kind of peripheral information that they need and so have little choice but to follow the dictates of the package.[4]

Clearly, the package user will also have tacit cues—perhaps a general understanding (or a misunderstanding) of how the package works, and perhaps an implicit assumption that "the computer is always right". The point is simply that these tacit cues are likely to be very different from their equivalents in a manual system.

The other important aspect of tacit awareness concerns the style of reasoning required to use the package. It is not enough to know what the package does and what "net present value" means, it is also necessary to understand how to tell the package to calculate a net present value, how to input the data, whether repeat calculations can be performed on similar sets of data, how to experiment with a package, how to decode help facilities and manuals, and so on. This may be explicitly described in the manuals, but is of little practical benefit unless the user has a "subsidiary awareness" of the appropriate tactics at the appropriate time. The user needs a "feel" for how the package works.

At a general level, use of a package involves what we might call *menu thinking*: the only choice the user has is that of choosing the appropriate option from a menu of possibilities (or a set of icons).[5] This is an obvious constraint in that it is difficult to go outside the universe of possibilities defined by the package.[6] Menus are normally organised hierarchically which means that users have the (often difficult) task of finding a given

[4]These problems may not be resolved by merely printing out computerised accounting records because printing, binding and storage facilities may be unavailable or awkward to use. However, developments in technology of the kind suggested by Cahill (1993) may help.

[5]This is also true of packages which require users to enter formulae or commands because these must be composed from the list of possibilities allowed by the package. Some packages —e.g., the statistical package, SPSS PC+ (SPSS, 1990)—even use a menu interface to allow the user to write commands and formulae.

[6]Since so many spreadsheets now use visual basic for writing macros, it is possible for a user to go outside the universe of possibilities defined by the package, but it is very difficult unless the user is quite proficient in visual basic.

option in the hierarchy. These hierarchies may impose a much more rigid framework on package users than other cognitive frameworks—which can be reorganised in the user's mind.

The inclusion of suspense accounts[7] is an example of how the framework imposed by an accounting package can be made less rigid. However, to make good use of it, users need to be aware of the suspense accounts facility, of what use it is and how to use it, and to know where to find it in the menu structure.

Spreadsheets provide an interesting example of a package encouraging a specific mode of reasoning. This is a "brute force" style which is a distinctly different approach from the conventional arithmetical one. The following example is typical of many simple spreadsheet models set up by students.

The problem is to find out how many units need to be sold to cover a fixed cost of £10,000 if the contribution per unit is £1600. The "brute force" spreadsheet method would be to set up a table:

Number of units	Contribution (£)
1	1600
2	3200
3	4800
4	6400
5	8000
6	9600
7	11,200

and so on. It is then obvious that the answer is 7 units.

The alternative, of course, is simply to divide £10,000 by £1600. The point here is that the style of thinking encouraged by spreadsheets leads people to the above solution, often without noticing that more efficient methods of problem solving may be available. The students may be so deeply immersed in the "spreadsheet paradigm" that they do not notice the alternative solution by division.[8] Other types of package, needless to say, may encourage different styles of thinking—but these different styles are likely to differ in fundamental ways from those facilitated by the pre-computer paradigms.

[7]Many accounting packages have "suspense accounts" to hold information whose final destination cannot yet be determined. This allows transaction processing to continue even when the data is incomplete or contains errors, with the problematic aspect of the data recorded in a suspense account to be dealt with later.

[8]Most modern spreadsheets incorporate a routine for solving equations by an iterative procedure (i.e., trial and error). This is an immensely powerful facility which, in this example, provides, in effect, an automation of the strategy used by the students, but which in more complex situations provides a means of solving problems which are impossible or impractical to solve algebraically. The brute force approach of the "spreadsheet paradigm" may sometimes be more powerful than the alternatives.

Actual Package-based Paradigms

The above discussion assumes that packages are used in the way expected by their designers. However, this is far too simplistic to be acceptable as the general rule. Software packages sometimes fail to deliver the anticipated benefits because of difficulties, for example, in eliciting user requirements or in implementation. The cultural context has a strong influence on the way any computer system is used (Robey & Azevedo, 1994). This is commonly accepted and there are numerous examples which we will not review here in detail.[9] Our focus here is on the difficulties which may arise from a cognitive (as opposed to organisational) perspective.

One important issue centres around the possibility that the answers produced by a package may be wrong, or they may be right numerically speaking, but misinterpreted by users. In a recent survey, "'at least 25%' of spreadsheet models contained unrecognised errors" (Cragg & King, 1993). Given the number of spreadsheet models in use, this represents a substantial number of potential erroneous conclusions. Users are not the only party at fault: there was a period when the Sage accounts package's automatic accruals and prepayments routine put in double entries which were the exact opposite of the correct ones.

More generally, because packages will typically implement mathematical routines or accounting processes or will interpret accounting terminology in the appropriate way—all without explicit assistance from the user, there is always the possibility that the users will fail to understand correctly what the answers mean or what the package is doing.[10] This means that the actual operation of the computer-based paradigm may be very different from its expected operation. It is difficult to gauge the extent of this type of mismatch between practice and designer's intentions because, for obvious reasons, the users may not be keen to advertise this mismatch or may not even realise that there is a mismatch.

The following (real) examples illustrate some of these possibilities. The first example is a straightforward business one, the second concerns education and training, and the third comes from the academic environment.

[9]The basic point is that the way information technology is used depends on "socially constructed meanings" which implies that "technology's social consequences are largely indeterminate because of the variety of meanings that technology can assume" (Robey & Azevedo, 1994, p. 23). The same software package may be used in one way in one cultural context, and in another, quite different way, in another context. In one context it may "preserve institutionalised practices" and in another it may be "a catalyst for change".

[10]This implies that such understanding is an all or nothing matter. This is often unrealistic: there may be varying degrees of adequacy of the user's understanding.

A brewery's system for routing deliveries to pubs and other retail outlets. This was an off-the-shelf system which had been adapted to the needs of the particular brewery. It was used to produce a daily route for the delivery lorries to follow. In practice the route it suggested always needed amending by the route planner (a person, not a computer package) because of various incidental factors of which the package failed to take account (e.g., road works, constraints on the times of deliveries to particular places, and so on). The route planner often took so long with these amendments that it may indeed have been quicker to have planned the route from scratch and made no use of the package. However, the package continued to be used because of pressure from higher in the organisation which decreed that the package had to be used. It is also worth noting that nobody involved in using the software had any idea how the package chose the "best" route or even the criteria it used. We would perhaps expect that the people using the package would not be familiar with the details of the algorithms used, but they also had no idea of the objective underlying the algorithm—was the route chosen, for example, to minimise distance, time, cost, or what? Even more strangely, this was a question which they appeared not to have thought of asking.

A package for implementing statistical quality control (SQC) techniques used with an in-house training course on SQC. The training course in question was an ambitious affair: more than 1000 employees did the course over a five year period. Despite this the package used was a very crude one: it was written in BASIC, started its life as a GCSE[11] project, and had none of the facilities and convenience to which its users would have been accustomed in the other applications they used (e.g., it did not allow users to transfer data from other sources, or even between routines in the one package; it had no on-line help; the facilities for editing data and the graphs and results produced were very inflexible). Despite this it was very convenient for the course organisers because it "led" students to do what was expected. The initial menu had nine options—corresponding to the nine techniques covered on the course, and the submenus covered just those elaborations covered on the two day course, but no more. For example, one of the options was a "variable sample *p* chart", which led on to the two further options of "adjusted" and "unadjusted" (in that order). Users were thus encouraged to "adjust" their charts; the adjustment in question was designed for the particular circumstance faced in this organisation—it produced a chart which looked tidy (the lines were straight) but was extremely difficult to interpret rigorously. In practice, of

[11]The GCSE—"General Certificate of Secondary Education"—is the public examination taken by most children in the UK just before they finish their compulsory schooling at the age of 16.

course, very few users did interpret it rigorously; they just accepted it at face value although this interpretation was very dubious.[12]

Statistical packages used by academic researchers. These include dedicated statistical packages (e.g., SPSS) as well as database and spreadsheets which can perform many of the same functions. They are used by researchers to analyse their results statistically and cite the appropriate "*p* values", "ANOVA tables" or whatever else is considered relevant to the research. The packages are relatively easy to use in the sense that simple commands will produce a large range of tables and statistics—some of which may be obvious and elementary but others of which may be mathematically extremely complex. Many researchers using these packages have very little idea of the rationale behind the techniques embodied in these packages so misinterpretations and misapplications are very common. Altman and Bland (1991) provide a survey of these problems as they occur in medical research—they claim that "the oft-quoted figure of 50% of papers containing statistical errors seems about right" (p. 228). It seems very likely that the situation in other academic areas—such as management, education and social sciences—is broadly similar.

There is another aspect of this problem. The statistical packages themselves may be—and sometimes are—in error. Altman and Bland (1991) review research which highlights some of these errors: which may be that the answer produced is wrong because of errors in the mathematical algorithms used, or it may be that the package is misleading regarding the naming and interpretation of techniques and results.

These examples illustrate some of the complexity of actual package-based paradigms. The brewery system had little to do with the "rational" cost minimising approach which was doubtless the designers' intention—the staff involved appeared to have little idea of what the package was intended for. They appeared to regard it as an arbitrary set of rules. The SQC package was designed to support the course and so was the basis of a deliberately designed "paradigm". The actual paradigm here was close to the designers' intentions, although these intentions were different from the standard SQC paradigm (because of the strange, and mathematically

[12]The difficulty with straightening the lines on a graph is that it means that the vertical scale will be different at different points on the horizontal scale, which in practice means that whatever is written on the vertical scale will be wrong at most points on the horizontal scale. If, for example, the vertical scale seems to indicate that the "proportion defective" in a particular sample is 5%, this is unlikely to be correct—the true value might be 6, 7 or 3%. This seemed to worry the users less than might be expected: this was perhaps partly due to a reluctance to check details and partly to an implicit assumption that they should not expect to make complete sense of these statistical diagrams.

dubious, adjustments to some of the charts). In the case of the statistical packages, the discrepancies between the expected and actual use are mainly due to users' difficulties in understanding the theory incorporated in the packages.

Progress through Package-based Paradigms

Package users tend to assume, inevitably, that the package represents progress, and that later versions are better than earlier ones—just as proponents of a scientific paradigm assume that their paradigm is better than any alternatives. From a wider perspective, the evidence from actual practice (such as the examples above) makes this claim much less plausible as a general truth.

Even in terms of the intentions of the package designers, the assumption of automatic progress may, on occasions, be suspect. Packages may continue to reinforce existing accounting paradigms when these are no longer appropriate. Computer accounting systems incorporate double entry bookkeeping even though this is no longer necessary as a proof of arithmetical accuracy. New ways of using accounting information are inhibited by the rigidity of this double entry representation (e.g., Sorter, 1969). Double entry, and the trial balance, perform certain control functions in manual systems. To a large extent, these controls have not been replaced in the computer environment.

While there are undoubtedly continuities between package-based and non package-based paradigms, and important differences between different packages, the extent of the change brought about by computer packages justifies the assertion that packages enable a new mode of thinking. Furthermore, this new mode of thinking has the potential to be more efficient and more powerful than the conventional mode of thinking.

If, for example, a package performs 50% of the thinking[13] and the human partner the other 50%, the human partner now only needs 50% of the expertise needed before the package was available to achieve the same end. Alternatively, with the same effort, twice as much output can be produced. It could be argued that the process of knowing is being made easier, or being made more powerful because more is now possible with the same effort from the human partner. In addition, the existence of a package to undertake some of the "thinking" means that the nature of the

[13]In rough terms—it would clearly be difficult to define "thinking" sufficiently clearly to make this statement rigorous. There is also the point that the user may need additional computer skills. 50% for the package would probably be on the low side in most applications—which strengthens the argument here.

knowledge or expertise required of the user must differ in important respects from the pre-package situation.

THE EDUCATIONAL PERSPECTIVE: PACKAGES AND KNOWLEDGE AND LEARNING

Computer packages are an important part of the accounting environment. They clearly have increased very substantially the ease (from the human point of view), power and efficiency of information processing. For these reasons alone, they have an inescapable role to play in the education of accountants.

All packages presuppose knowledge and skills on the part of their users. Education aims to develop such knowledge and skills and so may—among other things—be a tool to assist in the use of packages and reduce some of the discrepancies between actual and expected package-based paradigms. For example, the problems with the statistical packages reviewed above could, in principle, be reduced by suitable education for users.

However, we have argued that these packages represent a substantial shift in cognitive framework. This has important implications for education: if the "type" of knowledge has changed it is very likely that old assumptions about the content and delivery of syllabuses will no longer be appropriate. In the subsections below we outline three broad areas in which educationalists need to make choices about their treatment of package-based paradigms.

It is likely to be the tacit dimension of computer packages that is of particular importance for the educational process, because it is this aspect which is least likely to be discussed explicitly in manuals or training courses. An awareness of the styles of thinking encouraged by packages, and of the different nature of the tacit cues available to package users (as opposed to users of manual methods) is likely to be advantageous for students.

The rhetoric of business suggests that the latest computer package is always an improvement. There is always the danger that education will reinforce this assumption instead of questioning it. Ijiri (1983) talks of the problem of "the erroneous impression of objectivity and precision that the computer is bound to create on the nature of the accounting process." Clearly one role of education is to develop a critical perspective on assumptions such as this.

Even when the degree of progress may be problematic, the fact that the paradigm has changed is often very clear. We argued above that expected package-based paradigms involved expectations about the knowledge and skills of users. Students, as potential users, need to be introduced to the appropriate range of knowledge and skills. Software may bring with it changes in the sensible content of syllabuses which may go far beyond the actual use of the packages.[14]

The three subsections below highlight three important areas of choice concerning the way computer packages are treated in education. These choices are not simple ones: they will depend on the detailed circumstances of each situation. In the short term, they will be constrained by the assumptions and motivations of individual students and teachers. If, for example, a student's perception is that good marks can be obtained merely by learning the syntax and menu structure of a package, and the student's motivation is geared towards obtaining high marks, then that student's effort is likely to be directed towards learning the package in this sense—regardless of any further considerations. In the longer term, the choices made in these three areas may alter some of the basic assumptions of the educational process.

Choice 1: Learning a Package or Learning about Packages

We can distinguish two senses in which students may meet a computer package in their studies of accounting and related areas.

1. Students can *learn to use* a specific package. Gallagher et al. (1989) describe how such knowledge may be included in the educational environment. This is puzzle solving within the paradigm defined by the package. The difficulty, from the students' perspective, is that they may not be able to see beyond the package they are using to consider the benefits of other packages, or perhaps the benefits of using a package at all; they may be trapped in the paradigm defined by the package and not even notice possibilities which cannot be encompassed by their particular package.
2. Students can *learn about* a package or a group of packages. This is described by Seddon (1987) and Er and Ng (1989) as learning about accounting information systems. Students might learn to use a particular package as an illustration, but this would be incidental to the main aim of learning about the given type of software. They might also learn about the style of the cognitive framework imposed by the package and about user problems. This has the advantage over learning to use a single package by enabling students to develop

[14]For example, Gallagher et al. (1989) indicate that packages can be used to teach students to develop strategic decision making skills made possible by "what if" analyses permitted by spreadsheet models. Sensitivity analysis has become important as a technique primarily because computers make it viable, and packages such as spreadsheets are built on the assumption that users will do such analyses. In contrast, bookkeeping has decreased in importance because computers have reduced it to a mundane, mechanical skill.

a wider and more flexible perspective. It is also possible to learn general strategies about how to use packages—about using the Help facilities, about using trial and error tactics to see how a package works, and so on. Students are generally expected to pick these points up themselves, but there may be a strong case for making some of these metalevel principles explicit.

These considerations underline the virtues of an educational tradition which is firmly on a "meta" level and regards packages in general terms rather than being trapped within the paradigm defined by a particular package or group of packages. On the other hand, of course, students do need to use packages, and they do need training in puzzle solving with the paradigm defined by a particular package. This is particularly true of packages which are used very widely—such as spreadsheets and word processors; it is clearly important that these are used efficiently. "Learning to use", and "learning about", are both relevant to education.[15]

Choice 2: Treating a Package as a Black Box or a Transparent Box

Perhaps the most obviously important point about computer packages is that they will do much of the information processing for their human users. This is both an opportunity—in terms of reduced demands on the user and increased power of the whole system—and a problem. The problem arises if the user does not have an adequate understanding of what the package does. The education process should have a crucial role here in so far as it aims to influence the knowledge and expertise of the human partner.

Packages can be described as *black boxes* where they are understood by their human users only in terms of inputs and outputs. In contrast to this, the traditional educational attitude is that potential users should only use a package if they understand everything inside it—i.e., all the mathematical and other processing methods implemented by the package. Therefore students are often taught the "manual" method first, and only then the use of a computer package, which is viewed as a simple automation of the manual procedures. We will call this the *transparent box* approach.

[15]There may be confusion between these two senses in which students can "learn" a package. An examination paper which requires students to memorise the sequence of key strokes necessary to make a package perform a specified analysis would seem an unfair test of "learning about" a package; on the other hand as a test of "learning to use" the package the examination would be entirely reasonable. Obviously, misunderstandings can arise if different parties make different assumptions about the purpose of a course.

The *black box* approach has the enormous advantage that the package can be used as a *substitute* for the traditional understanding of manual methods. This may save users a considerable amount of time and may be inevitable where users do not have the expertise to look into the black box. On the other hand the transparent box approach entails users' learning the manual methods *and* how to use the package. Thus the technology has the effect of *increasing* the amount that users must learn, although it does enable the exploration of more complex scenarios than would be possible without the computer.

The usual educational rhetoric comes down firmly on the transparent box side. However, this is often little more than rhetoric. Few students understand everything they are supposed to understand, and there are often unexplained steps in explanations of traditional procedures (e.g., the origin of statistical tables is rarely mentioned). Furthermore, the expansion in the size of syllabuses makes the transparent box approach increasingly unrealistic. Some of the opportunities afforded by information technology to reduce the amount of information students are expected to store in their heads must be taken.

This is particularly the case in respect to the more mathematical aspects of the syllabus. Linear regression provides a good example. The "manual" method, as explained in any textbook on elementary statistics, is arithmetically complex and very unlikely to be illuminating to non-mathematicians in the sense of clarifying how the method "works". (The rationale for the formulae is based on the differential calculus.) On the other hand, a computer implementation of linear regression incorporating a graph showing the data and the regression line (as can be set up on any spreadsheet) makes it immediately obvious that the regression line is, in some sense, a "best fit" straight line. The student who uses the computer implementation of regression as a black box, without going into details of the algorithms the computer is using, is probably in as good a position as the student who knows the formulae (Wood, 1992).

Similar comments apply to the brewery routing system, the statistical process control system and academics using statistical packages outlined above. In each case what the users need to have is an understanding of the basic concepts and rationale behind the algorithms implemented by the computer, not an understanding of the algorithms themselves. They do not need to see inside the black box, but they do need to have a useful interpretation of what goes in and what comes out of the box.

The exact nature of this "useful interpretation" is a vital—and probably under-researched—question. The potential danger is that users will misinterpret or misuse the output; a "useful" interpretation should clearly prevent this. However, a recommendation for all packages to be treated as transparent boxes is not a feasible response. For example, the statistical techniques implemented by statistical packages are mostly far too complex (and are based on too advanced mathematics) for academic researchers to follow in detail.

On the other hand, the black box principle is not always appropriate. The concept of information flows provides a very different example. In a manual system the concept of information flows is not explicit but is implicit in the traditional method of T-accounts, trial balance and accounts preparation. A computer system "hides" these information flows, so it becomes necessary to include their understanding as a specific learning outcome. The software system is thus treated as a transparent box, and furthermore, students are encouraged to look carefully inside it.

Choice 3: Packages or Education

Real world package users may learn to use a package "on the job". They are, in effect, using the package in place of attending a course on the topic area. Instead of attending a course on, for example, project management, they learn to use one of the many packages available for project management. This package then guides them through the various techniques and procedures of project management, and they pick up the meaning of concepts such as "critical path" from the manuals, the help screens and by experimenting with the package. This could be described as a just-in-time (JIT) education system, in contrast to the traditional practice of teaching people just-in-case (JIC) they need what they have learned. The potential advantages of the JIT approach are that it reduces unnecessary learning and enables a flexible response to new situations.

On the other hand the manuals and computer-based tutorials on which package users depend may not be adequate; people may not use them or if they do they may not understand them. In short, unassisted JIT learning may not work.

There is also the very serious danger that the chosen package may impose an inappropriate or unnecessarily restrictive paradigm. Project management software packages, for example, tend to emphasise project timing and resource analysis at the expense of softer "people" issues.

Education may have a role to play here in helping people develop the skills of learning to use packages and their associated cognitive paradigms, and learning to "learn about" them—as and when they need to. This is, in effect, applying the principle that "learning to learn" is a more powerful strategy that simple learning. The difference with package-based paradigms in this context is that the learning may be much easier (because of on-line help facilities and so on) and the results may be much more powerful than in the case of paradigms not based on computer packages. This suggests that the relative advantage of "learning to learn" over mere learning may become progressively greater as packages become friendlier and more powerful.

CONCLUSIONS

Computer packages are becoming progressively more widespread and influential in accounting and accounting education. This has a number of important implications for accounting in general and accounting education in particular:

- Performing an accounting task with a computer package is a substantially different process from performing a similar task without a package. The collaboration between a package and its human users can be regarded as a new cognitive paradigm in a very similar way to that in which, according to Kuhn, a scientific revolution launches a new scientific paradigm. In general, these new, computer-based paradigms are likely to differ in important respects from their predecessors.
- Like any other paradigms, these package-based paradigms are likely to have important tacit components—for example a package may encourage particular styles of reasoning and discourage other styles without users being fully aware of this. It is also necessary to distinguish between expected package-based paradigms and the actual ones. The actual paradigm based on a package may be very different —often much less "rational"—than the expected paradigm.
- Computer packages are often assumed to represent definite progress. The main reason is that the processing power they incorporate reduces the amount of work that human users need to do, and the expertise they need to possess, to achieve a given end; or the package may render possible tasks which would not be possible without the package. However, these advantages are sometimes not as clear cut as might be assumed. Packages do, however, undoubtedly involve important changes in the cognitive paradigm—regardless of whether these changes are viewed as progress.
- From an educational perspective, it is clearly necessary to take account of these differences between package-based paradigms and their predecessors. There are important choices to be made in three areas:
    1. The sense in which students learn (about) a package: whether this is learning to use the package or learning about packages on a "meta" level. The latter is of particular importance, although the former also has a vital role to play.
    2. The extent to which users need to learn how the package works and the extent to which it is possible to treat it as a "black box". The arguments in favour of the latter approach may be stronger than is often realised, although there is a very real danger of users misunderstanding the techniques implemented by a package and using it inappropriately.

3. The extent to which packages can serve as a substitute for education. If a convenient package exists for an area of expertise, is it then necessary to include this area of expertise in students' education?

These are all crucial issues which may not even be noticed by an attitude to education which treats the content of a syllabus as given and uncontroversial and which disregards the fundamentally new opportunities and constraints offered by computer technology. The assumption that learning computer packages is a straightforward process which is necessary simply to provide access to greater computational power is far too simplistic. Computers change what is worth knowing, as well as the manner in which it is known.

REFERENCES

Altman, D. G., & Bland, M. J. (1991). Improving doctors' understanding of statistics. *Journal of Retail Statistics Society, A 1542*, 223–267.

Cahill, P. True, fair, virtually real. *Accountancy, August (1993)*, 52–53.

Cragg, P. B., & King, M. (1993). Spreadsheet modelling abuse: an opportunity for OR. *Journal of Operational Research Society, 448*, 743–752.

Er, M. C., & Ng, A. C. (1989). The use of computers in accountancy courses: a new perspective. *Accounting and Business Research, 19*, 319–326.

Gallagher, J., Hoskin, R., & Capettini, R. (1989). Instructional applications of personal computers and spreadsheet software in accounting: a taxonomy of objectives and methods. *British Accounting Review, 21*, 23–41.

Ijiri, Y. (1983). New dimensions in accounting education: computers and algorithms. In *Issues in Accounting Education* (pp. 168–173). American Accounting Association.

Kuhn, T. S. (1970). *The Structure of Scientific Revolutions* (2nd edn). Chicago: University of Chicago Press.

Kuhn, T. S. (1981). The sciences as puzzle-solving traditions. In S. Brown, J. Fauvel, and R. Finnegan (Eds), *Conceptions of Inquiry*, pp. 107–113. London: Methuen.

Polanyi, M., & Prosch, H. (1975). *Meaning*. Chicago: University of Chicago Press.

Robey, D., & Azevedo, A. (1994). Cultural analysis of the organisational consequences of information technology. *Accounting, Management and Information Technologies, 41*, 23–38.

Seddon, P. (1987). Computing in the undergraduate accounting curriculum: three distinct goals. *British Accounting Review, 19*, 267–276.

Snell, M. (1991). Drawing histograms using LOTUS. *Computers in Higher Education Economics Review, 14*, 12–19.

Sorter, G. H. (1969). An events approach to basic accounting theory. *Accounting Review, 44*, 12–19.

SPSS (1990). *SPSSPC+, version 4.01*. SPSS Inc.

Wood, M. (1989). Expert systems, expert tutors and training in elementary statistics. In N. Shadbolt (Ed.) *Research and Development in Expert Systems, VI*, pp. 195–206. Cambridge University Press.

Wood, M. (1992). Using spreadsheets to make statistics easier for novices. *Computers and Education, 193*, 229–235.