CHAPTER THREE

Planning and ill-defined problems

Thomas C. Ormerod

Introduction

The distinction between well-defined and ill-defined problems has its origins in the specification of components of a problem space (cf. Hayes, 1978), that is, the space of possible move sequences given the context in which the problem is set and the information-processing limitations of the problem-solver. A well-defined problem is one in which the start-state of the problem, its goal-state, the available operators (i.e. methods that can be applied to make moves from the start-state towards the goal-state) and the constraints upon operator selection (i.e. the rules that define legal moves) are known in advance. Solving well-defined problems can be viewed as a task of navigating from the start-state to the goal-state by applying operators at appropriate times to shift from one problem state to another under given constraints. Planning involves the evaluation of moves in advance of their selection in an attempt to discover a sequence of one or more moves that optimises the route from start-state to goal-state. Heuristics, such as hill-climbing and means-ends analysis, are central to explanations of human performance with well-defined problems, such as the Tower of Hanoi and Missionaries and Cannibals puzzles (e.g. Simon & Reed, 1976; Anderson, 1993). Under these heuristics, the problem-solver plans moves by assessing the difference between the current state and the goal-state and then selecting sequences of operators or establishing that minimise this difference.

Ill-defined problems present a dilemma for planning: how can one plan the route towards a solution if one knows so little about the path ahead, especially when one does not know the final destination or goal state. Puzzles such as the nine-dot and radiation problems (illustrated in Figures 1 and 2, respectively) are considered ill-defined because components of their problem spaces are not fully specified. In particular, the problem descriptions lack a statement of a concrete and visualisable goal-state. Heuristics that evaluate the progress made towards a goal-state are not obviously applicable to solving ill-defined problems, because it is difficult to describe a test for the final state that could be used in evaluating progress (VanLehn, 1989). However, faced with an ill-defined problem, individuals can and do plan. In fact, planning is central to both failure and success at solving ill-defined problems. Moreover, planning lies at the heart of commonalities among expert problem-solvers.

---Insert Figures 1 and 2 about here---

In the remainder of this chapter, we distinguish between local planning, where choices are made between alternative moves from any particular problem state, and global planning, where decisions are made about how to structure problem-solving activity as a whole. We also distinguish between the application of pre-compiled plans retrieved from prior knowledge, and on-line planning consisting of the assembly and evaluation of move sequences in real time, either at the commencement of problem-solving or during the execution of problem moves. We begin by focussing upon small-scale ill-defined puzzles, before turning our attention to large-scale and realistic problem-solving activities.

Planning In Puzzle-Solving

The nine-dot problem (Figure 1) exemplifies a small-scale ill-defined problem. Its start-state is clear - an array of unconnected dots - as is the only available operator - to draw straight lines. The constraints upon what moves may be legally made by applying the operator are, in principle, known in advance. However, the goal-state is under-specified: indeed, if the goal-state were known in advance, there would be no problem to solve. As ill-defined problems go, the nine-dot problem is reasonably well-defined. Arguably, Duncker’s (1945) radiation problem (Figure 2) is much less well-defined. The start-state is specified but difficult to visualise, the operators that might be applied are apparently limitless, some constraints are present in the problem statement but are embedded in a complex description while others depend upon prior knowledge and beliefs, and the goal-state is described only in abstract terms. As we shall see, the planning activities that people undertake in attempting to solve the nine-dot and radiation problems explain why they are so difficult (typically fewer than 10% of people solve either problem within a ten-minute period). However, it is only through planning that the problems can be solved at all.

Using prior problems as solution plans

In the absence of concrete and visualisable goal-states, it has generally been assumed that individuals do not use hill-climbing or means-ends analysis, but instead use alternative heuristics to guide move selection that do not require complete goal information. The heuristic on which the vast majority of the literature on ill-defined problems focuses is structural analogy (e.g. Gick & Holyoak, 1980; Anderson, 1993). Although not usually conceived of in such terms, analogical problem-solving can be characterised as a plan-based heuristic. In drawing analogies, one effectively uses the solution to a source problem as a plan for solving a target problem. The analogical re use of pre-compiled plans is related to the concept of total-order planning (see Davies, chapter 2 this volume), but differs in being based upon the retrieval of existing plans rather than the construction of new plans prior to commencing move attempts.

Gick and Holyoak (1980, 1983) demonstrated the potential for analogical problem-solving as a heuristic for solving ill-defined problems. They presented participants with source problems that had the same underlying solution concept (splitting and convergence) as Duncker's radiation problem but different superficial descriptions (e.g. a story about a general trying to attack a well-defended fortress). In a number of experiments, they demonstrated that solutions to the radiation problem could be facilitated by presenting analogical source problems and solutions. Their findings led them to conclude that analogy lies at the core of human learning. They proposed that, through multiple analogical problem-solving episodes, individuals induce abstract schematic representations of problems and their solutions that can be retrieved and applied as solution plans when structurally similar problems are presented. In a similar vein, Anderson (1993) has suggested that structural analogy is the key heuristic that leads to the compilation of new procedures for action. More recently, Thompson, Gentner, and Lowenstein (2000) have argued that the same kinds of analogical heuristic mediate skilled problem-solving in practical domains such as managerial negotiation.

In almost all the studies that demonstrate successful analogical transfer of a conceptual solution to a superficially different target problem, participants are either given a hint that the target and source problems are related, or are otherwise placed in situations where the requirement to analogise is unavoidable. Evidence for spontaneous analogical transfer based on more than superficial similarities is sparse. Moreover, there is plenty of evidence suggesting that individuals are strongly influenced by surface similarities among source and target problems (e.g. Holyoak & Koh, 1987), and that the effects of surface similarities among problems is often to lead participants into drawing superficial analogies that do not deliver a conceptually relevant solution. Thus, there are grounds for suspecting that analogy might not be the default heuristic for tackling ill-defined problems.

Recent evidence collected by Chris Bearman, a PhD student in our laboratory, suggests that encouraging individuals to make use of analogies can actively detract from their subsequent problem-solving performance. In one experiment, he reversed the order of problems used by Gick and Holyoak, giving the radiation problem as the source and a number of variants including the fortress problem as targets. When participants were given a hint to use the source solution analogically, they actually were worse at solving the target than participants who received the target alone. One point that is generally overlooked in the literature is that the radiation problem is much harder to solve than the variants (e.g. the fortress problem) that are used as analogues. It appears, then, that the encouragement to analogise led participants to use an analogical mapping strategy rather than tackling the relatively simple target problem in a more direct fashion (e.g. by trial-and-error or by means-ends analysis), which would have been more effective. Analogy, then, may actually be a strategy of last resort when a problem is too difficult to be tackled on its own by conventional means.

On-line planning and puzzle-solving

There are situations where individuals encounter completely novel problems for which no analogical problem and solution information may be known. The nine-dot problem presents such an example. Until recently, accounts of human performance have generally proposed that problem-solvers impose additional and inappropriate constraints that preclude the discovery of a solution. Once these inappropriate constraints are removed, so the story goes, the problem then becomes easy to solve. The traditional Gestalt account is that the nine-dot array is unavoidably processed perceptually as a square whose boundaries should not be violated. Weisberg and Alba (1981), amongst others, point out that providing instructions to include lines that go beyond the boundary of the square does not reliably lead to solution. Instead, Weisberg and Alba propose that individuals impose an inappropriate constraint that lines should extend only between dots, a constraint derived from prior experience of dot-to-dot drawing. Curiously, they fail to notice that their critique of the Gestalt position applies equally to them.

Whatever the success of these alternative accounts, the key point is that planning behaviour has not been considered as offering an account of human performance. Indeed, some authors (e.g. Anderson, 2000) adopt a definition of insight problems that is based upon peoples' inability to plan when solving them. For example, Metcalfe and Weibe (1987) found that measures of participants' feeling of warmth (i.e. measures of anticipated closeness to solution) strongly predicted imminent solution of non-insight puzzles (such as the Tower of Hanoi), but not insight puzzles (such as the nine-dot problem). These results suggest that participants are unable to monitor their progress during insight problem-solving, an essential pre-requisite of planning behaviour.

We have recently proposed an account of human performance on the nine-dot problem that invokes precisely the same kinds of hill-climbing strategy used for planning moves in non-insight problems such as the Towers of Hanoi (MacGregor, Ormerod, & Chronicle, 2001). Although individuals do not have a concrete and visualizable goal-state against which to monitor progress, the nine-dot problem statement describes some of the goal-state properties: namely, that each dot must be cancelled and that there are four lines available to do so. These properties are enough for individuals to derive hypotheses about what might constitute 'locally rational' moves, that is, moves that appear to make progress in improving the current state against the hypothesised goal properties (for similar accounts of locally-rational move selection, see Chater & Oaksford, 1999; Simon & Reed, 1976). So, individuals endeavour to maximise the number of dots cancelled by each move they make (where a move can be one or more lines). More importantly, moves are evaluated under a criterion of satisfactory progress - in the nine-dot problem, each move must cancel the absolute number of dots given by the ratio of remaining dots divided by remaining lines.

According to our account, the standard nine-dot problem is so difficult, not because individuals impose inappropriate constraints on the lines they are prepared to sample, but because there are so many moves available that meet the criterion but that do not allow solutions to be found. Consider, for example, someone planning two lines at a time (that is, working at ‘two-lookahead’, a value that provided the best-fit for planning behaviour across the five experiments in MacGregor et al, 2001). The criterion for each line in a satisfactory two-line move is initially 9/4, so the move must cancel at least 4 dots. There are 24 different first moves that cancel five dots and 48 that cancel four dots. Assuming the selection of a five-dot first move, the criterion for each line in a satisfactory subsequent move is 4/2, a criterion satisfied by an available third line after all 24 moves. It is only when individuals plan their fourth and final line that they discover that they are unable to find one that meets the criterion.

We believe that the early experience of criterion failure is an essential prerequisite for the discovery of insightful moves. Individuals have to recognise that they are failing to make sufficient progress with the moves available within their current conceptualisation of the problem space, and must seek novel non-maximising moves. There are so many criterion-fitting moves that can be tried in the nine-dot problem (and whose failure must be remembered if they are not to be repeated), that individuals are unlikely to experience early criterion-failure. There is little incentive when attempting the standard version of the problem to include move attempts that extend beyond the array of dots, since such moves would not apparently make any more progress than the moves that lie within the array.

---Insert Figure 3 about here---

MacGregor et al (2001, experiment 5) presents a test of our account against theories of inappropriate constraint imposition. Figure 3 shows two versions of the problem in which participants were given the first line of the solution, and were told to cancel the remaining dots by drawing three further lines starting from an end of the given line. All other accounts should predict greater facilitation with the version where the first line extends beyond the array of dots, since it violates the perceived boundary and includes a non-dot end point. Our account predicted the opposite result. Consider again, an individual working at two-lookahead. The criterion for the first move in both versions is 4 (6/3 followed by 4/2), so the first move must cancel four dots. There are two available moves that meet this criterion in the 'line-outside' version, whereas there are none in the 'line within' version. Thus, we predicted that, because participants would encounter criterion-failure earlier in the 'line within' version, participants would be more likely to seek alternative moves including those that extended beyond the dot array, and would therefore be more likely to solve. The results confirmed this prediction, with 45% solving the 'line outside' version and 65% solving the 'line within' version.

The role of planning in insight puzzle-solving is not limited to move selection. Although criterion failure is necessary for the subsequent discovery of insightful solutions, it is not sufficient, since there are, in principle, an infinite number of alternative, non-maximising moves that participants might try once criterion failure has occurred. We suggest that, while individuals relax the requirement for moves to maximise progress once criterion failure has occurred, they continue to plan alternative moves by assessing the progress that they might make. Moreover, they retain novel move attempts that fail first time round for further exploration only if the move attempt makes progress towards satisfying the criterion relative to any previous move. Also shown in Figure 3 are examples of failed attempts from the 'line outside' and 'line within' versions that occurred in our data. Both embody the conceptual 'insights' to draw lines outside the dot array and to turn on non-dot points. However, only the move attempt shown below the 'line within' version was a strong predictor of success on the next attempt. Participants who produced the attempt shown below the 'line outside' version were more likely to return to drawing move attempts within the dot array on their subsequent move than they were to retain the lines that they drew outside the dot array. According to our theory, they failed to capitalise on the conceptual promise of their move attempt because it made no progress relative to previous attempts, whereas the failed attempt associated with the 'line within' version cancels one more dot than any three line attempt that remains within the dot array.

In summary, it appears that individuals, when solving ill-defined puzzles, plan their move attempts by evaluating each move against a hypothesised criterion of satisfactory progress. If they fail to discover satisfactory moves, they search for emergent non-maximising alternatives, and they are likely to retain these for further exploration if they make progress: in effect, they plan to reuse failed attempts that show promise. We have recently implemented our theory as a fully specified computational model (Ormerod, Chronicle, & MacGregor, in preparation), which implements the planning behaviours described here as a search for moves guided by a register of promising states, that is, previous failed states that have not been fully exhausted and are prioritised according to the extent to which they made progress over other attempts. As well as testing our theory with the classic nine-dot problem, we have generalised the theory to a novel insight puzzle, the eight-coin problem, which requires the discovery of moves in three dimensions (Ormerod, MacGregor, & Chronicle, 2002). The on-line local planning that we believe individuals engage in when they tackle such problems is a variant of the partial-order planning approach that high-ability problem-solvers appear to use in tackling well-defined problems (see Davies, chapter 2, this volume).

Plans, Planning and Expert Skill

The problems described in the previous section are small in scale, and the consequent planning demands are relatively localised. Where the study of problem-solving really becomes interesting and of relevance to explaining human activity is with large-scale realistic problems. Planning in large problem spaces can be computationally expensive. Consider the task of planning a sequence of moves at the start of a game of chess, using an exhaustive planning strategy in which you evaluate the outcomes of all possible moves. There are 20 different first moves. In reply to any one of these, the opposing player would choose a move from among 20 alternatives. For each of those opposing player's moves, you could choose a move from between 19 and 31 alternatives, depending on what the preceding moves had been. By the time you had planned and evaluated a second move exhaustively, you would have had to consider around 10 000 different board configurations. Complex and large-scale ill-defined problems defy exhaustive planning, because of the combinatorial explosion of possible problem states. It seems possible, therefore, that a characteristic of domain experts that differentiates them from novices is the use of selective planning strategies. For most of the last half century, the assumption among cognitive psychologists has been that experts do not plan in real time: instead they retrieve and adapt pre-compiled plans from memory.

Is expertise just plan recall?

The roots of this assumption lie in pioneering research conducted by deGroot (1965), who challenged the folk hypothesis that the skill of grandmaster chess players is based upon an ability to plan longer sequences of moves in their heads than less skilled players. deGroot found that there was no difference in the length of the sequence of moves that grandmasters and less skilled players planned in response to presented board positions, nor was there a difference in the time taken to select moves. Nonetheless grandmasters were able to select better quality moves. Further studies demonstrated that there was no difference between skilled and less skilled chess players in terms of general memory abilities. Thus, deGroot concluded that chess expertise is based upon the learning and subsequent retrieval of stored knowledge about the most appropriate move to make given a specific game position. Chase and Simon (1973) confirmed this view, demonstrating that grandmasters are able to reconstruct realistic chess boards in fewer glances, placing more pieces with each glance, than less experienced players. Thus, expertise in chess appears to be based, according to these studies, on the perceptual recognition of a board position, and the retrieval of a plan that details the appropriate moves for the recognised position.

Similar plan-based accounts can be found in many areas of expertise research. For example, Soloway and Erhlich (1984) proposed that expert programmers retrieve, interleave and flesh-out plans in developing computer programs. Programming plans are kinds of general templates that capture generic functionality of program structures at different levels of abstraction. A recent incarnation of the programming plans approach is described by Rist (1995). Rist argues that there is no overall strategic planning or control in program design. Instead, programs are constructed through cue-based search and retrieval from memory (both internal and external) of plans that set up further cues to search upon until final implementation (i.e. until no further cue slots are left unfilled). As Rist states (p.558) “The only planning mechanism is the shift of attention from one focus to another in internal and external memory”. The key cue in Rist’s view is the ‘focal line’, that is, an item of code or pseudo-code that directly encodes the specific goal of the plan to be retrieved from memory (e.g. a plan to accumulate a running total would be cued by the focal line “count = count + 1”). Once a focal line is retrieved, according to Rist’s account, a process of focal expansion occurs in which the remainder of the relevant code is retrieved from memory.

The general notion of plans as abstract or semi-abstract templates is common to many theories of expertise. Indeed, the absence of the word ‘plan’ from the subject index of current textbooks can be accounted for by the fact that plan-based accounts tend to be described under different jargon such as schemas (e.g. Neisser, 1976), scripts (e.g. Schank & Abelson, 1972), frames (Minsky, 1980), and Memory Organisation Packets (Schank, 1982). It is, of course, obvious that experts possess more domain knowledge than novices, and it seems likely that expert knowledge is stored in an abstract fashion that would make it amenable to plan-like adaptation as new task requirements arise. However, there is evidence to suggest that the retrieval of pre-compiled plans provides only a partial account of expert planning behaviour in ill-defined problem-solving domains. For example, Holding and Reynolds (1982) found that, when experts and novices were presented with random board positions (i.e. layouts that were not part of a realistic game and therefore could not be part of an expert's prior knowledge of chess positions), the experts were able to make a better subsequent move. In other words, it appears that experts were using some kind of on-line planning to select the best move given the current unrecognisable (and therefore not recallable) board configuration.

We have conducted research that challenges the sufficiency of a plan-retrieval account of programming expertise (Ormerod & Ball, 1993). We used verbal protocol analysis to examine how experts in the Prolog language (i.e. programmers with more than 10 years of Prolog programming experience) constructed a solution to the relatively complex program coding task (to count and calculate temporal flow statistics from a vehicle sensing device). In particular, we examined the order in which solutions were constructed. The key finding was that, while all of our nine expert programmers produced essentially the same solution in terms of its underlying structure, the order in which solutions were produced was different for every participant. Had their programming been guided by the kinds of plan structures proposed by Soloway and by Rist, then we would have expected to see focal lines emerge first, followed by template-like structures for handling the procedural control elements of the program, followed by the details. Instead, the order in which programmers produced their solutions appeared to be more under the control of their current inferential goal. Each of our experts differed somewhat in terms of their backgrounds, some coming from formal specification traditions, others being jobbing programmers. This led them to emphasise different aspects of the programming task (e.g. efficiency, coding aesthetics, functionality) at different times. They all possessed the relevant programming knowledge (e.g. about the use of variables to collect intermediate results that are passed to the final output in the stopping case of the program), but the order in which this relevant knowledge was sampled was seemingly determined on-line rather than as pre-compiled execution orders.

Plans, planning and the conundrum of creative expertise

Perhaps the most challenging kinds of ill-defined problem-solving in which to explore planning is in creative design domains, where simple re-use of previous solutions cannot meet the necessary task condition of creativity. Design is perhaps the ultimate domain in which to explore human-problem-solving, a fact recognised by Simon (1981). The main issue is how problem-solvers can control the process of seeking multi-design solutions in an infinite problem space, when a principle task requirement is that designers do not simply re use previous design ideas. In the previous chapter, Davies outlined the means-ends analysis heuristic, a process by which a complex goal that cannot be achieved through a single operator application is decomposed into smaller subgoals, to a point where operators can be applied. Complex design problems require decomposition, and it is the process of managing decomposition while holding onto the earliest solution ideas that, we argue, lies at the heart of planning in expert problem-solving.

---Insert Figure 4 about here---

There are two so-called structured control strategies for decomposing complex design problems, breadth-first and depth-first decomposition, which are illustrated in Figure 4. In following a breadth-first decomposition approach, the problem-solver decomposes the overall goal into subgoals, and then decomposes each of these subgoals in turn before trying to further decompose any one subgoal. Breadth-first decomposition has been proposed as the prescriptively optimal approach to planning in design, since it minimises commitment to specific design solutions until the whole of a design problem has been explored (e.g. Hoare, 1978). However, breadth-first design is computationally costly, since the problem-solver must maintain a mental or external register of all the design subgoals and their inter-relations until the final stages of design, when all subgoals have been decomposed to a level where operators may be applied in their solution. In following a depth-first decomposition approach, the problem-solver decomposes a goal into subgoals, then focuses upon these one at a time, decomposing each one as far as necessary before operator application before returning to tackle the next subgoal. The depth-first decomposition approach underlies the means-ends analysis heuristic that appears to underlie the solution of novel problems (e.g. Newell & Simon, 1972). The advantage of depth-first design is in minimising the amount of unresolved subgoals that need to be remembered as design problem-solving proceeds. As well as minimising computational load, a depth-first approach also allows for the early testing of emerging design components and enables designers to demonstrate tangible progress. The disadvantage is that early design decisions may prematurely commit the problem-solver to solution approaches that do not work for later subgoals.

Empirical studies of design planning have typically demonstrated that breadth-first design is associated with expert performance, while depth-first design is associated with novice performance (e.g. Adelson & Soloway, 1985; Jeffries, Turner, Polson & Atwood, 1981). Davies (1990) has further shown that expert designers mitigate the computational load associated with breadth-first decomposition by relying much more than novices on externalisation of emerging design work. This shifts the computational burden from the mental resources of the problem-solver to the external task environment.

In a study of expert Prolog and C programmers designing solutions to a more complex version of the programming task shown in Figure 4, we found that the experts, regardless of the language in which they were working, adopted a subtle variant of problem-decomposition, which we term children-first decomposition (Ormerod, Ball & Lang, 2003). In following this decomposition approach (which is also illustrated in Figure 4), designers effectively mixed breadth-first and depth-first approaches, by decomposing across all the subgoals of the main goal before focusing upon the first of these subgoals. We argue that a key component of the expertise of experienced software designers is in knowing how to manage the competing needs to maximise visible progress and early design testing while minimising premature commitment. Planning by children-first decomposition provides a structured way for experts to achieve this balance that works well in relatively constrained design domains such as programming.

The idea that designers plan the order of their problem-solving activities by applying structured control strategies has not gone uncriticised. The most radical alternative to structured decomposition is so-called opportunistic planning (e.g. Hayes-Roth & Hayes-Roth, 1979). Opportunistic accounts of design problem-solving have been proposed by a number of researchers, including Guindon (1990) and Visser (1993). In both of these accounts, opportunism is characterised as a positive attribute for design, in that it provides a source for creative idea generation.

It should be noted, however, that Guindon characterises as opportunistic any design activity that does not follow a strictly breadth-first decomposition approach, while Visser makes the same contrast against a strictly depth-first approach (effectively contrasting opportunism against the implementation of a retrieved plan that is executed in depth-first fashion). We have argued that there are flaws in the analyses of both Guindon and Visser, that call into question their observations of opportunistic planning (Ball & Ormerod, 1995). Instead, we argue, the design protocols they report that do not fit a single pure decomposition strategy actually reflect strategic switches between breadth-first and depth-first decomposition. Expert designers know that breadth-first design has advantages, but they also know when it is appropriate to switch to depth-first design, for example when they want to explore a potential solution to a specific design subgoal that they believe to be particularly complex or critical to the overall design. Thus, we believe that there is an important distinction to be made between opportunistic planning, that consists of genuinely serendipitous changes in goal focus (and abandonment of previous plans), and on-line planning, where the next moves may be determined either serendipitously or (we believe more commonly in expert design) by the pursuit of one or more global control strategies.

Our studies of expert programmers (Ormerod, Ball, & Lang, 2003) also provide evidence for local planning decisions (i.e. given a current level of focus and two or more goals to choose from at that level, the factors that determine the programmer's selection). We categorised all the transitions across programming activities that did not fit a children-first decomposition strategy (18% of all transitions), according to the stated or inferred reasons for switching to a new focus. Categories of transitions accounting for more than 10% of the structure-divergent transitions were as follows: to a node whose parent has not been coded but which has been recognised-30%, jumping back to a structured approach after a divergence-15%, capitalising on an analogy-12%, and debugging-12%. Of these categories, only the 'analogy' category seems opportunistic in the sense that they were serendipitous opportunities that participants capitalised upon.

Software design and programming are highly skilled tasks, but one might question the extent to which they are highly creative. Indeed, the commonality among solutions produced by experts in the study by Ormerod and Ball (1993) suggests that creativity, at least in the scale of task discussed above, is limited. A recent study of expertise in the design of educational tasks enabled us to explore design problem-solving in a creative domain (Ormerod, Fritz, & Ridgway, 1999). Indeed, creativity is one of the primary metrics on which newly-designed tasks are judged, since they must be original enough to secure copyright for the author and publisher. We set out to address two key questions in relation to planning. First, do task designers use the same kinds of problem decomposition strategy seen in other design domains or is their planning better characterised as opportunistic? Second, do task design specialists differ from non-specialists in terms of the planning strategies that they employ?

We studied experts designing a novel educational task for teaching English as a second language (ESL) to meet the requirements of a brief that included information on target audience, level, duration, topic, and so forth. People who solve such problems professionally, typically the authors of ESL textbooks, may undertake months of problem-solving activity between coming up with initial ideas and finally putting their solution, a finished task, down in print. Educational task design nicely illustrates the complexities and scale of real problem-solving activities. Professional task design involves many different groups of problem-solvers, including authors, publishers, teachers and target audiences. There are unlimited constraints upon task design: tasks must be short enough to fit a taught curriculum but not trivially so, they must not disenfranchise any ethnic or other group, they must not be gender-specific, they must allow all learners to engage in the task activity while offering a challenge to faster learners, they must be creative and original yet not weird or incomprehensible, and so on. Moreover, there is no single best outcome.

In our study, videotapes were recorded of experienced designers and teachers (our specialist and non-specialist groups, respectively) undertaking the design of novel ESL tasks. We coded the resulting protocols according to cognitive acts (e.g. generate, evaluate, understand) and focus of activity (e.g. design brief, prior experience, analogy, emerging design, implementation). The study showed that designers and teachers differ in the tasks they produce and also in the distribution of cognitive acts across time. Each of the specialists came up with a radically different task, yet all of their solutions to the task design brief were of far higher quality than the solutions provided by teachers. Inspection of specialists' protocols revealed four phases: idea generation, design description, expansion, and implementation. Teachers operated more on exemplar retrieval than design. Previous research suggested that designers might be less systematic in creative than in constrained domains, but precisely the opposite emerged. Surprisingly, there was little evidence (less than 3% of focus transitions) of opportunism by either designers or teachers (defined as a switch from one focus of activity to another in a way that is inconsistent with either breadth-first or depth-first decomposition). Instead, specialist designers worked from depth-first decomposition in the first phase to breadth-first decomposition in later phases, whereas teachers worked from breadth to depth across phases. This reverses the typical expert/novice difference in control strategies used to manage problem-solving, and provides a novel demonstration of strategy changes over time. In this domain, it makes sense for designers to begin working in a depth-first mode, since it allows them to rapidly explore and reject early ideas (a "fail fast, fail often" strategy). Once an idea has survived early testing, they switch to breadth-first decomposition, since it produces an orderly emergent design. The importance of these results for the current chapter is as follows: the only commonality that we found among expert task designers was in their approach to planning.

Conclusions

In this chapter we have examined the role played by planning in solving so-called ill-defined problems. We have examined how plans can play a key role in solving ill defined problems, in the form of expert or domain-specific knowledge of appropriate steps to execute, and as strategic search of the problem-space for best value moves under a progress-monitoring criterion. The effects of such planning activities can be seen both locally, in deciding what the next appropriate move might be, and globally, in deciding how to schedule the pursuit of a complex problem that decomposes in to many smaller problems. Much of the work from our own laboratory has focused upon the role of strategic planning skills, in contrast to domain knowledge-focused accounts of problem-solving found elsewhere. Domain knowledge and strategic knowledge are both essential components of skilled problem-solving, and are probably ultimately inseparable. This chapter is intended to re-emphasise the importance of strategic knowledge rather than replace domain-knowledge theories.

Given the generally accepted view that planning in advance of action - to “look before you leap” - is good practice, one might expect that planning would play a major role in current accounts of human problem-solving. The study of planning has certainly been central to research into machine learning and artificial intelligence (e.g. Nilsson, 1982). Yet, the words ‘plan’ and ‘planning’ do not occur in the indexes of a sample of recent textbooks on cognitive psychology (e.g. Anderson 2000; Eysenk & Keane, 2000; Groome, 1999; Reisberg, 1999; Solso, 2001). Planning has largely been overlooked in accounts of how humans tackle ill-defined problems such as insight puzzles, and plays a role that is at best secondary to memory-based accounts of expert skill in large-scale realistic problem-solving.

One explanation for the relative absence of planning research is as follows: Planning is a risky cognitive activity. First, we have seen how planning can be computationally expensive. Second, we have seen how plans can be misleading or inflexible. Finally, planning can be unnecessary: decisions about what move to choose at any particular point do not necessarily require the evaluation in advance of a sequence of moves. Under this view, it is perhaps surprising to argue that such a potentially costly, error-prone and wasteful cognitive activity plays an essential and positive role in human problem-solving. In this chapter, we have seen how individuals are able to minimise the risks of planning and how they are able to plan in the absence of complete problem information. We have explored some of the conditions under which planning has positive outcomes in tackling ill-defined problems, as well as others where it can be detrimental to problem-solving performance.

We have added another dimension, that of global versus local planning, to the distinctions between initial and concurrent planning outlined by Davies in the previous chapter, and the distinction between total and partial order planning proposed by Ratterman et al (2001). Initial versus concurrent planning refers to the position of planning activity; total versus partial order planning refers to the completeness of planning activity; and global versus local planning refers to the scope of planning activity. Plans retrieved from memory may be total-order (e.g. Soloway & Erhlich's, 1984, programming plans) or partial-order (e.g. Rist's, 1995, focal expansion of programming plans). Similarly, plans constructed on-line can be total-order (e.g. as in the third stage of expert task design, as observed by Ormerod, et al, 1999) or partial-order (e.g. as in attempts to solve the nine-dot problem as observed by MacGregor, et al, 2001). In common with the way individuals appear to tackle well-defined problems, we suggest that the same kinds of goal-directed planning strategy are brought to bear with ill-defined problems. However, the view of Rattterman et al (2001), that opportunistic planning (which they, wrongly in our view, equate with on-line planning) will be more in evidence with ill-defined problem-solving, does not seem to be borne out by the evidence (though ironically, the task that Rattterman et al (2001) use to study planning behaviour is, under our account of problem definition, a well-defined problem). Indeed, there is a case for saying that, when the problem definition does not structure the activities that problem-solvers can undertake, the need for structured planning increases. We suggest that people adopt structured planning approaches at both global and local levels. Moreover, the planning strategies employed at particular phases of problem-solving provide a point of consistency among otherwise diverse expert behaviours in creative problem-solving domains.

References

Adelson, B. & Soloway, E. (1985). The role of domain experience in software design. IEEE Transactions on Software Engineering SE-1.,

Anderson, J. R. (1993). Rules of the mind. New York, LEA.

Anderson, J. R. (2000). Cognitive psychology and its implications, 5th Edn. New York, Worth.

Ball, L. J., & Ormerod, T. C. (1995). Structured and opportunistic processing in design: A critical discussion. International Journal of Human-Computer Studies, 43, 131-151.

Chase, W. G., & Simon, H. A. (1973). Perception in chess. Cognitive Psychology, 4, 55-81.

Chater, N., & Oaksford, M. (1999). Ten years of the rational analysis of cognition.

Trends in Cognitive Sciences, 3, 57-65.

Davies, S. P. (1990). The nature and development of programming plans. International Journal of Man-Machine Studies, 32, 461-481.

DeGroot, A. D. (1965). Thought and choice in chess. The Hague, Mouton.

Duncker, K. (1945). "On problem solving." Psychological monographs, 58, 1-113.

Eysenck, M. W., & Keane, M. T. (2000). Cognitive psychology: A student's handbook, 4th edition. London, LEA.

Gick, M. L., & Holyoak, K. J. (1980). Analogical problem solving. Cognitive Psychology, 12, 306-355.

Gick, M. L., & Holyoak, K. J. (1983). Schema induction and analogical transfer. Cognitive Psychology, 15, 1-38.

Groome, D. (1999). An introduction to cognitive psychology: processes and disorders. London, Psychology Press.

Guindon, R. (1990). Designing the design process: Exploiting opportunistic thoughts. Human-Computer Interaction, 5, 305-344.

Hayes, J. R. (1978). Cognitive Psychology: Thinking and creating. Homewood, Ill.: Dorsey Press.

Hayes-Roth, B., & Hayes-Roth, F. (1979). A cognitive model of planning. Cognitive Science, 3, 275-310.

Hoare, C. A. R. (1978). Communicating sequential processes. Communications of the ACM, 21, 666-677.

Holding, D. H., & Reynolds, J. R. (1982). Recall or evaluation of chess positions as determinants of chess skill. Memory and Cognition, 10, 237-242.

Holyoak, K. J, & Koh, K. (1987). Surface and structural similarity in analogical transfer. Memory and Cognition, 15, 332-340.

Jeffries, R., Turner, A. A, Polson, P. G., & Atwood, M. E. (1981). The processes involved in designing software. J.R. Anderson (Ed.), Cognitive skills and their acquisition, Hillsdale, New Jersey, Lawrence Erlbaum Associates.

MacGregor, J. N., Ormerod, T. C., & Chronicle, E. P. (2001). Information-processing and insight: A process model of performance on the nine-dot and related problems. Journal of Experimental Psychology: Learning, Memory and Cognition, 27, 176-201.

Metcalfe, J., & Weibe, D. (1987). Intuition in insight and non-insight problem-solving. Memory and Cognition, 15, 238-246.

Minsky, M. (1980). K-lines: A theory of memory. Cognitive Science, 4, 117-130.

Neisser, U. (1976). Cognition and reality. San Fransisco, Freeman.

Newell, A., & Simon, H. A. (1972). Human problem solving. Englewood Cliffs, N.J., Prentice-Hall.

Nilsson, N. J. (1982). Principles of Artificial Intelligence. New York, Springer-Verlag.

Ormerod, T. C., & Ball, L. J. (1993). Does programming knowledge or design strategy determine shifts of focus in Prolog programming? Empirical studies of programmers 5. C. R. Cook, J.C.Scholtz and J.C.Spohrer. Palo Alto CA, Ablex.

Ormerod, T. C., Ball, L. J., & Lang S. (2003). Global and local control processes of programmers. Unpublished ms. Lancaster University, UK.

Ormerod, T.C., Fritz, C.O., & Ridgway, J. (1999). From deep to superficial categorisation with increasing expertise. Proc. 21st conference of the Cognitive Science Society, Vancouver, August 1999. pp.502-506.

Ormerod, T. C., MacGregor, J. N., & Chronicle, E. P. (2002). Dynamics and Constraints in Insight Problem Solving. Journal of Experimental Psychology Learning, Memory, and Cognition, 28, 791-799

Ratterman, M.J., Spector, L., Grafman, J., Levin, H., & Harward, H. (2001). Partial and total-order planning: evidence from normal and prefrontally damaged populations. Cognitive Science, 25, 941-975.

Reisberg, D. (1999). Cognition. New York, Norton.

Rist, R. (1995). Program structure and design. Cognitive Science, 19, 507-562.

Simon, H. A., & Reed, S. K (1976). Modelling strategy shifts in a problem solving task. Cognitive Psychology, 8, 86-97.

Schank, R. C., & Abelson, R.P. (1972). Conceptual dependancy: A theory of natural language understanding. Cognitive Psychology, 3, 552-631.

Schank, R. C. (1982). Dynamic Memory. New York, Cambridge University Press.

Scheerer, M. (1963). Problem-solving. Scientific American, 208, 118-128.

Simon, H. A. (1981). The sciences of the artificial, 2nd edition. Cambridge MA, MIT Press.

Soloway, E., & Erlich, K. (1984). Empirical studies of programming knowledge. IEEE Transactions on Software Engineering, SE-10, 595-609.

Solso, R. L. (2001). Cognitive Psychology, 6th Edn. Boston, Allyn & Bacon.

Thompson, L., Gentner, & Lowenstein, J. (2000). Avoiding missed opportunities in managerial life: Analogical training more powerful than individual case training. Organizational Behavior and Human Decision Processes, 82, 60-75.

VanLehn, K. (1989). Problem solving and cognitive skill acquisition. In M. I. Posner, (Ed.) Foundations of cognitive science. Cambridge, Mass.: MIT Press.

Visser, W. (1994). Organisation of design activities: Opportunistic, with hierarchical episodes. Interacting with Computers, 6, 235-274.

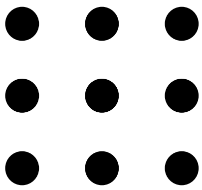
Weisberg, R. W., & Alba, J. W. (1981). An examination of the alleged role of "fixation" in the solution of several "insight" problems. Journal of Experimental Psychology: General, 110, 169-192.

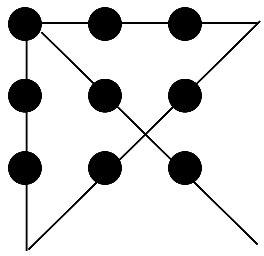
Figure 1. The nine-dot problem and its solution. The task is to cancel each of the dots by drawing four straight lines without retracing a line or lifting the pencil from the paper.

Figure 2. Duncker's (1945) radiation problem

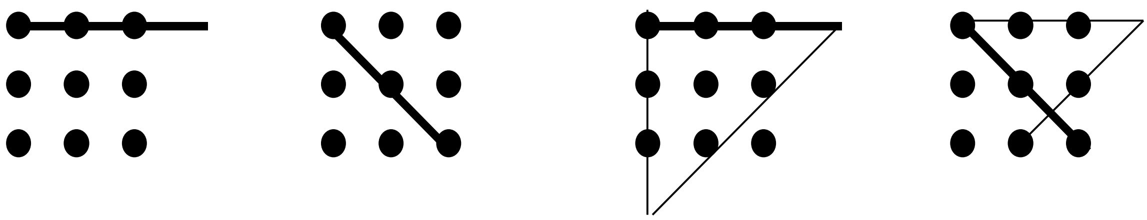
Figure 3. The 'line outside' (a) and 'line within' (b) versions of the nine-dot problem tested by MacGregor, et al (2001). Diagrams (c) and (d) show common failed attempts of participants with the 'line outside' and 'line within' versions, respectively (both attempts start at the top left dot).

Figure 4. Decomposition of a hypothetical design hierarchy using three different control strategies (number order shows the order in which goals are visited under each strategy).

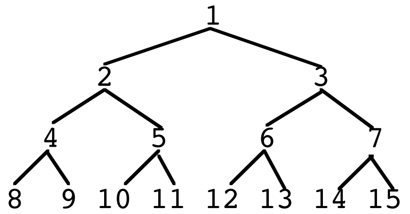




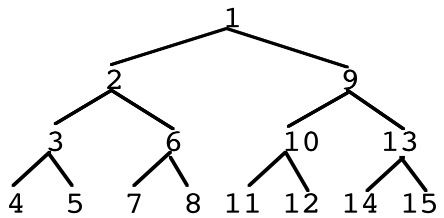
Suppose you are a doctor faced with a patient who has a malignant tumour in his or her stomach. It is impossible to operate on the patient, but unless the tumour is destroyed the patient will die. There is a special type of ray that can be used to destroy the tumour, as long as the rays reach the tumour with sufficient intensity. However, at the necessary intensity, the healthy tissue that the rays pass through will also be destroyed and the patient will die. At lower intensities, the rays are harmless but they will not affect the tumour either. What procedure might the doctor employ to destroy the tumour with the rays, at the same time avoiding destroying any healthy tissue?



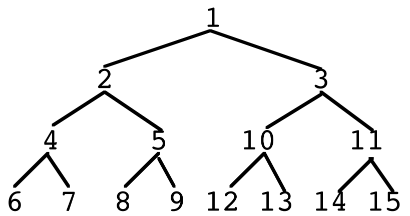
(a) (b) (c) (d)



a). Goal recognition order with a breadth-first strategy



b). Goal recognition order with a depth-first strategy



c). Goal recognition order with a children-first strategy