Theoretical Models of the Role of Visualisation in Learning Formal Reasoning

Martin Oliver, James Aczel

Abstract:

Although there is empirical evidence that visualisation tools can help students to learn formal subjects such as logic, and although particular strategies and conceptual difficulties have been identified, it has so far proved difficult to provide a general model of learning in this context that accounts for these findings in a systematic way. In this paper, four attempts at explaining the relative difficulty of formal concepts and the role of visualisation in this learning process are presented. These explanations draw on several existing theories, including Vygotsky's Zone of Proximal Development, Green's Cognitive Dimensions, the Popper-Campbell model of conjectural learning, and cognitive complexity.

The paper concludes with a comparison of the utility and applicability of the different models. It is also accompanied by a reflexive commentary[0] (see footnotes) that examines the ways in which theory has been used within these arguments, and which attempts to relate these uses to the wider context of learning technology research.

Commentaries:

All JIME articles are published with links to a commentaries area, which includes part of the article's original review debate. Readers are invited to make use of this resource, and to add their own commentaries. The authors, reviewers, and anyone else who has 'subscribed' to this article via the website will receive e-mail copies of your postings.

[0]   See page 29: *A commentary on the use of theory in the analysis of the Jape study*

Martin Oliver, *Department of Education and Professional Development, University College London, 1-19 Torrington Place, London, WC1E 6BT, 020 7679 1905, martin.oliver@ucl.ac.uk*

James Aczel, *Institute for Educational Technology, Open University, Milton Keynes, MK7 6AA, UK, J.C.Aczel@open.ac.uk*

1   Introduction

Studies of students studying logic have demonstrated that using visual representations can both speed and facilitate learning. However, although these studies have identified specific problems and difficulties encountered by students during this process, few have attempted to explain either these or the role of visualisation in terms of theories of learning.

In this paper, a series of theoretical models will be used to try and provide just such an account of the role of visualisation in logic learning[1]. The analysis is based upon research into the ways in which students use the software tool Jape (Just Another Proof Editor; Bornat & Sufrin, 1996; available at http://users.comlab.ox.ac.uk/bernard.sufrin/jape.html). These analyses attempt to explain how the development of formal reasoning skills relates to the students' learning of logical concepts, why certain logical rules are more complicated than others and how visualisation provides support for students as they learn these topics. The paper will conclude with a discussion of the strengths and limitations of each approach.

2   The development of strategies for formal reasoning

Formal reasoning is an abstract subject that many students find difficult and demotivating (Cheng *et al.*, 1986). Moreover, it is a topic that may require years of instruction, and even then, may not support transfer, even to closely related domains (van der Pal, 1996). Research to date has helped to map out and to explain some of the key difficulties facing students learning formal reasoning. Fung & O'Shea (1994), for example, identified a number of problems facing learners, including a lack of familiarity with formal notation, an inability to break problems into manageable components, a lack of formula manipulation skills and an inability to extract general principles from specific cases. Oliver (1997) has carried out further research into a particular refinement of classical first-order logic called Modal logic. This work showed that negative propositions, double negatives, longer strings of operators and the use of abstract notation exacerbate students' problems, and can result in students acquiring the skills needed to manipulate logical expressions without fully understanding them.

However, one criticism of these findings is that they concentrate on specific learning objectives or obstacles. They relate purely to demonstrable outcomes of learning, rather than explaining the process by which these concepts or abilities are acquired. As a way of overcoming these limitations, Aczel *et al.* (1999a) have begun to explore the processes whereby students acquire such logical concepts.

This research involved a series of studies with 170 computer science undergraduates taking an introductory course in propositional and predicate logic. The students were expected to translate

[1]   See page 29: *A commentary on the use of theory in the analysis of the Jape study*

natural language statements into a given formal representation, to evaluate formal representations semantically, and to prove conjectures using formal rules. *Jape* was provided to support the learning process. The software is a proof tool that allows students to manipulate proofs using a mouse; when users apply a rule to a line of the proof, the software calculates the consequences. *Jape* can be configured to use a variety of logics, to present proofs in different ways and to allow various user actions. However, this research involved a particular implementation, called *ItL Jape*, which has been configured with the style of logic used on the course and pre-loaded with around 70 conjectures.

Three overlapping studies were conducted. In order to identify those features of the software that support learning, it was decided to compare students' proving behaviour on paper and in *ItL Jape*. This was done both as part of the course ("The Observational Study") and in structured interviews ("The Reflection Study"). Data was also collected on students' backgrounds, their usage of *Jape*, and their success in the course ("The Measurement Study").

The use of control groups was not possible - on both ethical and pragmatic grounds - and so the Measurement Study focused on data that would enable an exploration of whether different student groups were more likely to use *Jape*, and of whether students who made more use of *Jape* did better in the course. The main data collection methods for the study were written tests, questionnaires, and automatic logs of program usage. Full details of the instruments, samples, and assumptions for the data collection are provided in Aczel (2000).

In the Observational Study, the work of students was observed over the whole course, following a naturalistic approach (Guba & Lincoln, 1981) in order to understand how *ItL Jape* fitted into the learning context. The bulk of this observation took place in eleven weekly small-group "workshops", although the teaching also included twice-weekly hour-long lectures, comprehensive course notes and tutorial classes. The workshops provided an opportunity to observe students actively engaged with the subject matter, and to talk informally to them about their understandings and difficulties. Four volunteer students were videotaped using *Jape* to assist them in constructing proofs during each of the workshops.

The main aim of the Reflection Study was to test the findings of the Observational Study by putting students in situations similar to those in which important incidents had occurred, seeing if the incidents were replicated, and, if so, obtaining students' interpretation of the incident. Such interventions were not part of the Observational Study because they would have disturbed the natural flow of the students' work and directed attention to aspects of the situation that might not otherwise be noticed. In the Reflection Study, therefore, ten students were videotaped using the program in task-based interviews rather than being observed in a naturalistic setting. The interviews took place some five months after the end of the logic course, and shortly before

the end of year exam. Students were given a choice of five topics that they wished to revise, and open questions were used to investigate their conceptions as they worked through activities. Students initially tackled five to ten conjectures or partially-completed proofs on paper, and then the same conjectures were tackled using *Jape*. The intention was that where the paper attempts had been successful, interface issues would be highlighted when *Jape* was used; and where paper attempts had been invalid or had stalled, the role of the software in enabling progress would be clarified. The 11 interviews were audio-taped (around 8 hours of paper-based work) and video-taped (about 12 hours of work both on paper and on *Jape*) and usually lasted around 90 minutes. 10 students were involved - 6 as individuals and 2 sets of pairs. The students were paid for participation, but the value of the session as a means of revising for the imminent exam was also given as a possible incentive. One student took part in 4 sessions, and so was able to cover almost the whole range of topics.

The studies, particularly the Reflection Study, led to the development of a set of descriptive strategies that categorise and explain the patterns of behaviour observed when students attempted to solve logic problems. These strategies, which are inherently pragmatic, can form efficient heuristics for completing proofs.

Although this has extended our understanding of the process of learning logic, the account to date remains entirely empirical and descriptive. In the following sections, attempts are made to use accounts of learning and development to explain the patterns of behaviour and the claims of students.

3   Using the Zone of Proximal Development to model logic learning

As discussed above, research findings show that students have problems understanding and applying formal methods, that generalising and transferring this understanding may take years, but that supportive software and visualisation tools can facilitate this process. One possible interpretation of these findings is that learning logic can act as a precursor to the development of formal reasoning skills, and that the problems facing students can be mediated or scaffolded by providing appropriate support.

This situation seems consonant with the model of the Zone of Proximal Development (ZPD) proposed by Vygotsky (1978). Briefly, the ZPD links the developmental process in children to a social account of learning. Central to the model of the ZPD is the premise that learners will be able to solve more abstract or advanced problems than would otherwise be possible when their learning process is supported or scaffolded by a more able peer. The 'zone' denotes the range of activities that the child can perform when supported, but is unable to complete

unassisted; this indicates their potential for development, rather than simply their current ability. In due course, however, it is possible for the learner to 'internalise' the kind of interactions that create this zone, thus developing their ability.

The ZPD has been used to justify and explain a number of developments in the area of computer-assisted learning. Its use varies widely, from providing a central thesis upon which the work is founded (e.g. Luckin, 1997) to acting as a passing reference used to endorse collaborative approaches to learning in higher education (as discussed by Crook, 1991).[9] One interesting feature of this use of the ZPD is the way in which the computer has come to be viewed as the more able peer. Although this interpretation of the ZPD seems rather different from its original use, it has been used to explain the effect of supportive software (see, e.g., Crook, 1991; Howe & Tolmie, 1999).

Taking this particular interpretation of the ZPD – rather than just the general model for the long-term development of logical ability noted above – may offer insights into specific observations made during the research. For example,

> *For many students, using ItL Jape allowed them to consider many more examples than would otherwise be possible using pencil-and-paper (because the program takes on the task of drawing the proof) and it also guarantees that inadequate proof attempts and incorrect rule applications were immediately challenged.*
>
> *(Aczel et al, 1999b)*

In this particular context, the use of the ZPD seems justified; the expertise in representation and the active challenges certainly imply that the software is acting in the role of a more able peer, providing active and tailored feedback to the learner which will eventually be internalised, thus supporting learning.

Such an interpretation must be made with caution, however, since it makes an important change to the type of situation described by Vygotsky.[8] In the original formulation of the ZPD, the learner was assisted by a more able peer; here, they are assisted by a system. Need this system be computer based? Given the role of visualisation in this context, it would be reasonable to ask whether a system of representation might also be able to provide support.

Drawing on Ainsworth's taxonomy of functions of multiple representations (1999), one use of a representation is to constrain the interpretation of a second. In this context, the use of visual cues (such as an ellipsis to denote missing lines of reasoning, or characters to indicate as-yet-unknown propositions) could be interpreted as part of a system (of representation) that provides support for the learner. In effect, these cues provide a layer of representations that supports the

---

[8, 9]   See page 31: *A commentary on the use of theory in the analysis of the Jape study*

learners' interpretation of logical notation. ("Secondary notation", in Green's (1989) terminology.) In this sense, the cues could be interpreted as a 'more able peer' that supports the learner in making sense of proofs. Importantly, this support would be provided irrespective of whether the representation was embodied in a software application or drawn on paper.

However, although these cues serve a pedagogic role, it would be hard to interpret their role as being 'collaboration' with a more able peer. Although it would be possible to compare a student's ability with and without a particular representation, the problem arises from the notion of this representation providing 'feedback', even though it is providing what might easily be interpreted as advice. Whilst the *creator* of this particular representation is, in effect, providing feedback from a 'more able' position, they will be doing so passively or indirectly. In such a situation, it seems inappropriate to draw on a model that is social and, essentially, interactional in nature.

This discussion makes it possible to judge whether it is appropriate to use the ZPD as an explanatory framework for students learning with *Jape*. The main criterion is that the system should demonstrate a degree of active agency, and should not just passively communicate advice from some 'more able' person. The use of 'degree' in this criterion suggests that there may well be debate over whether specific cases of system use justify recourse to the ZPD. In this case, however, it seems to offer a framework that can be used to interpret the way in which students work ('collaborate') with *Jape*, although using it to explain the specific role of representations in this process would be inappropriate.

In addition to these conceptual issues about the appropriateness of the model, there are a number of pragmatic problems that limit the usefulness of the ZPD. Although the ZPD seems promising as a general explanation of the research findings concerning use of the software, a finer-grained analysis identifies further difficulties. Splitting the learning process into increasingly abstract steps or layers is not straightforward. As noted above, the strategies that were observed evolved pragmatically, and gave little insight into any conceptual or cognitive development that might accompany their use. This makes it particularly difficult to envisage any notion of 'direction' in learning these topics, which raises the question of whether it is feasible to map out a zone against which progress might be interpreted. (Vygotsky suggested (1962) that this might not be an unusual problem.)

At a simple level, however, the observational study suggests that students learn in the following steps:

- concept acquisition (lectures and texts)
- use of software (labs); paper & pencil problem solving
- trial and error (as evidenced by quotes such as, "he doesn't know what he's doing... he's proving them but he doesn't know what's going on.")

• development of ad hoc heuristics for problem solving
• abandonment of heuristics in favour of more general strategies

This model is essentially chronological, although it also suggests a move from surface learning (of facts or routines) towards a deeper conceptual understanding of the topics. However, this implied model of conceptual development does not seem to hold in practice. Students may make a leap of understanding when initially presented with the concepts in a lecture, whilst others may develop concepts through trial and error, only later referring these back to content delivered in formal teaching settings.

Moreover, although this model is less general than simply saying that studying logic allows the development of logical thinking, it still says nothing about the process of learning specific topics, concepts or rules. These steps are neither strictly linear nor independent, and learners may well be at different stages with respect to different rules. If the ZPD is to be used to model learning, it becomes important to have some sort of 'map' that enables comparisons of current and assisted ability to be made. In other words, it becomes important to have some structured form of representing ability (specifically, the ability to use and understand rules and concepts) in this domain. Three different types of relationships between rules are possible:

1. The learning processes for different topics are sequential. In this case, the learning process for each rule can be viewed as a series of steps within one overarching 'map' that describes the entire process of learning logic using *Jape*. (Figure 1)

| Rule 1 | Rule 2 | Rule 3 | Rule 4 |

Figure 1: sequential rule learning

2. The learning processes are independent. In this case, the 'map' is fragmented, and the ZPD can only be used to model the acquisition of each rule separately. (Figure 2)
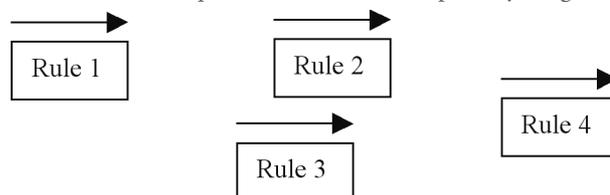
| Rule 1 | Rule 2 |
| Rule 3 | Rule 4 |

Figure 2: independent rule learning

3. The learning processes are mutually dependent. In this case, the ZPD no longer maps linear processes, but a multi-dimensional one. This means that the overall learning process for logic is inherently complex, with each step in the learning process being influenced by what the students already understand. (Figure 3)
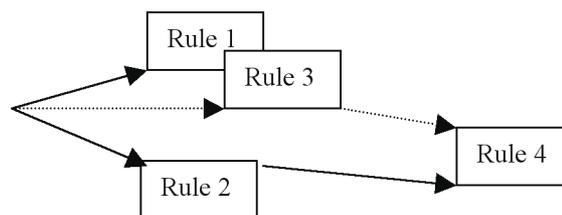

Figure 3: inter-dependent rule learning

The first possibility can be discounted, not least because the Observational Study revealed that preliminary sessions introduce a number of rules simultaneously, and this precedes the workshops with *Jape* where strategies for rule application are developed. Similarly, the second possibility can be discounted if the students make any kind of analogical inferences from one rule to another, or if the strategies for use of any rule draw on knowledge of others. Evidence for this was provided by the Reflection Study:

> *Some of these rule-specific strategies help students to choose between rules; for example, «if there is a choice between vE forwards and vI backwards, try vE first». The student Kusi described this as the "precedence" of vE over vI.*
>
> *(Aczel et al., 1999b)*

Moreover, several rules are introduced through derivation from others, such as the definition of A$\lor$B ("either A or B is true") as ¬(¬A $\land$ ¬B) ("it is not the case that neither A nor B is true").

Having eliminated the first two options (Figures 1 and 2), it can be concluded that the process of learning first order logic follows the third model, requiring an account of the inter-relationships between rules to be drawn up before the notion of 'difficulty' in logic learning can be properly explained. The implication of this is that as well as lacking a meaningful measure of the relative 'distance' between developmental steps, it is also impossible to elaborate what the 'shape' of this space is. It may be possible for future work, perhaps using a methodology such as grounded theorising (Strauss, 1987), which focuses on understanding the complex inter-relationships between concepts as encountered and used in real contexts, to develop a more useful map. Until this happens, however, the usefulness of the ZPD as a model for learning logic will be limited to general conclusions about logical development. Lacking a clear model of the

process of learning to reason formally, the ZPD cannot be used to explain students' progress. It can, of course, be used to explain differences in students' ability when working alone or in collaboration, but only in terms of the current general, poorly articulated model of the domain.

4   Using cognitive dimensions as a basis for comparing the difficulty of rules

Green's cognitive dimensions (1989) characterise the way in which information artefacts structure and represent information, and are intended to support the software design process. These address important aspects of systems' usability, such as the ease with which changes can be made (viscosity), the degree to which links between components are made explicit (hidden dependencies), how long the user can postpone strategic decisions until all implications are known (premature commitment), and so on. These descriptors are used to analyse 'ideal' (hypothetical) uses of a system, with the intention of understanding the pay-offs between different design options.

An obvious extension of this concept is to use the dimensions to describe the relative usability of specific components of systems; indeed, Kadoda *et al.* (1999) have followed this approach in order to identify the features of theorem proving software that support learnability. In this case, cognitive dimensions will be applied to the process through which rules are applied to proofs within the context of *Jape*. Such an analysis may provide some explanation of the relative difficulty of using different rules – one of the issues identified when considering the ZPD above. Table 1 shows how some of the rules implemented in *Jape* compare in terms of a relevant subset of cognitive dimensions. Note that because no standard metric has been devised for the dimensions, these comparisons should be interpreted as being based on a relative ranking.

Table 1: Examples of differences in rule applications in Jape

Rule	Explanation	Hidden dependencies	Abstraction level	Premature commitment	Viscocity
→E forwards	Turns A, A → B into B	None	Low	None	Moderate
→I backwards	Given A → B, a section of proof will be added (before A → B) that requires B to be deduced from A	Moderate	High	Low	Low
∨E forwards	If C follows from A∨B, then user must show that C follows from A and C follows from B	Low	Moderate	Low	High
∨I (R) backwards	Can be applied to justify B∨A if A already features in the proof. (B does not need to feature elsewhere.)	Moderate	Moderate	High	Low

Evidence to support the fact that students encountered problems with premature commitment and abstraction, for example, can be found in the analysis of the Reflective Studies.

> *It seemed to take many students some time to realise that unexpected results were often attributable to a line not being selected before a rule is applied. Also, several students suggested that they were confused at certain points in particular conjectures about whether, when they selected a rule, it would be applied "forwards or backwards."*
>
> *(Aczel et al., 1999b)*

This problem is well illustrated by comments from one of the students, referred to as Caroline. The first concerns the problem of guessing from the interface whether or not a rule would be applied forwards or backwards (premature commitment):

> *It's working opposite to the way I would. It's working backwards.*

Similarly, a second quote concerns whether applying the rule, $\wedge\neg E(L)$, *selected* or *removed* the left-hand-side of the formula:

> *Which one does it keep and which one does it chuck away?*

This highlights some of the advantages and disadvantages that arise from the use of a software tool. On the one hand, *Jape's* use of visualisation introduces additional premature commitment (by assuming which line rules apply to and forcing students to guess whether the rule is an introduction or elimination, rather than forward or backwards). On the other, it introduces secondary notation (such as the ellipsis and justifications of moves, and also in the form of feedback on errors) and reduces viscosity by allowing moves to be 'undone' in order to make changes. However, as noted in Table 1, not all rules are equally easy to undo, with some (such as $\vee$ E forwards) requiring additional information to be given in order to restore the previous step.

As noted above, one of the outputs from the study was a set of strategies that students adopted in order to complete proofs. These strategies are 'meta-moves'; they help the student to decide which rule they should be applying next. Because these strategies are essentially complexes of the above rules, it is proposed that in the same way that rules can be analysed in *Jape*, the strategies used to determine which rule to apply can also be analysed using cognitive dimensions. Table 2 illustrates this for a selection of the strategies developed from the Reflection Study.

Strategy	Hidden dependencies	Abstraction level	Premature commitment
«Break up implications in the conclusion»	Low	Low	Low
«Look for the 'main symbol' in the most complicated line. Find the same symbol in the list of rules. Try one of the matching rules. Undo the rule if the display doesn't look right... and try another»	Low	Low	Moderate
«Make an assumption»	Low	Moderate	High

Table 2: An analysis of strategies for proof construction using cognitive dimensions

Having demonstrated that this mapping can be carried out, the next question to be asked is, does this help construct a model of learning? One implication is that it is possible to predict that certain strategies – i.e. those that are consistently 'low' in the table above – will be easier to use successfully. Those that are learnt early on but which have fairly complex implications (such as making an assumption) are likely to lead to difficulties.

The way in which these strategies are used to apply increasingly complex logical rules (as indicated by consistently high ratings in the third to fifth columns of Table 1 – a high rating in the sixth reflects the difficulty of undoing steps, e.g. to correct mistakes, and so may not be relevant in certain contexts) may provide an insight into the learning process, allowing predictions to be made about the versatility of certain strategies and the relative difficulty posed by specific proofs.

Finally, this mapping process provides an explanation of a role for visualisation in learning logic:

> *Strategy development seemed most successful when the visible effects of an action were not only sufficient to allow students to make an informed decision about the utility of the action, but were also subtle enough to place the onus of strategy-development on the student.*
> *(Aczel et. al., 1999c)*

Cognitive dimensions provide a way of estimating the 'size' of the scaffolding that Jape offers, albeit one limited by the same provisos about relative rankings and combining categories made for the analysis of applying rules. The implication of this is that if we can gauge how much support a student needs, it may be possible to adjust Jape to provide just enough scaffolding for problems to be pedagogically challenging rather than too easy or inappropriately complex.

Returning to the point made above, working with *Jape* introduces both advantages and disadvantages. It is reasonable to ask whether the advantages outweigh the disadvantages. A comparison can be made between the application of rules either with or without the support provided by various elements of *Jape*, particularly in terms of the use of visual cues such as secondary notation. (Table 3)

Table 3: A comparison showing the 'size' of scaffolding provided by Jape's visual clues

Feature	Effect
Justification provided on right hand side of line of proof	Secondary notation increased, abstraction decreased, hidden dependencies decreased
'Placeholder' (e.g. _P) introduced when rules are applied without reference to a particular line	Secondary notation increased (probably unhelpfully), abstraction increased, hidden dependencies increased
Error messages (reports that could be interpreted by the developers)	Secondary notation increased (unhelpfully), abstraction increased, hidden dependencies increased
Explanatory messages (intended to advise the learner)	Secondary notation increased, abstraction may decrease, hidden dependencies may decrease
Ellipsis to indicate missing steps or justifications	Secondary notation increased, abstraction decreased
Use of menus divided into 'Introduction' and 'Elimination' rules (rather than 'forwards' and 'backwards' rules)	Abstraction increased, hidden dependencies increased

What this highlights is that, for most of these features, the situation is improved: either the relative difficulty is decreased, or extra support (e.g. from secondary notation) is provided. However, it should be noted that the list of features in table 3 is indicative, not exhaustive; as such, this kind of analysis will be limited to identifying areas of strength and weakness, rather than providing any absolute judgement about the 'good-ness' of the tool.

In summary, then, cognitive dimensions provide a useful vocabulary for describing relative complexity and degrees of support. Although the lack of a metric for these dimensions means that the analysis relies on rank ordering of options, this still provides some insight into the difficulty of concepts and strategies, and the type of support provided by Jape's use of visualisation.

5   Popper-Campbell Psychology

Another model that may help in accounting for the pattern of students' engagement with formal reasoning is based on the work of Karl Popper. Developments of this model (Aczel, 1998) suggest that learning can be analysed in terms of the trial-and-improvement of psychological entities called "strategic theories" in response to problems of special interest to the individual – "concerns". [10]

Popper's critique of the "bucket theory of mind", his insistence that what can be learned is heavily dependent on the individual's prior theories "of persons, places, things, linguistic usages, social conventions, and so on" (Popper, 1963) and his view that "knowledge" in the public sense comes about through complex intersubjective processes, are clearly in resonance with the work of Vygotsky. Yet it should be pointed out that what is described here is very much a *psychological* rather than a *sociological* perspective. In contrast to some interpretations of Vygotsky, it is assumed that it is the individual – and this individual's interactions with the worlds of physical objects, ideas, and people, *mediated* by language, social forces, culture and history – that is the focus of study, rather than language, social forces, culture and history themselves.

Rather than learning consisting in the passive, steady, repetitive accumulation of information, there are active processes of decoding and sifting, in which existing theories are modified by creative, conjectural, discontinuous trial-and-error-elimination. Campbell (1960) describes a mechanism called "Blind-Variation-and-Selective-Retention" (BVSR) for such imaginative processes, in which, by analogy with evolution by natural selection, there is "a mechanism for introducing variation", "a consistent selection process", and "a mechanism for preserving and reproducing the selected variations". Campbell also suggests that mechanisms shortcutting BVSR were themselves created by BVSR.

It is important to note that a key feature of this psychological perspective is that creative theory-formation processes do not occur in isolation, but in response to the selection pressures afforded by problems of special interest to the individual – a "concern". Concerns would include desires, motivations and fears. By attempting to address one's concerns, new strategic theories are constructed from old, and these may in turn generate new concerns. In the research described here, the main concern would be to prove conjectures.

One subtle aspect of Popperian psychology is that action, context and theory are intertwined. Students have myriads of complicated, contextual and implicit theories (taken as constructions of reality), created from a wide variety of experiences and concerns. There is often a strategic nature to these theories in that they solve problems.

[10]   See page 31: *A commentary on the use of theory in the analysis of the Jape study*

For example, students tended to interpret *Jape*'s representation of an *incomplete* step as indicative of an *incorrect* step. One student said, "When _A and _B [Jape's representation of as-yet-unknown propositions] came up you knew you were on the wrong track". This interpretation constitutes a constructed expectation or *theory* about what constitutes progress in reasoning, but within Popperian psychology it can also be seen as an elementary *strategy* for proving conjectures - «If you get _A and _B, undo the step».

Conversely, an action, strategy, plan, heuristic, procedure or process can be considered as a theory, in that it incorporates expectations about what is. Hence we refer to *strategic theories*. For example, the strategy, «Break up implications in the conclusion», can be seen as a simplistic reference to a theory about reasoning in Natural Deduction that propositions with an implication in the conclusion are soluble by first applying the rule $\rightarrow$ I.[11]

In short, then, the learning of Natural Deduction using *Jape* would consist of the trial-and-improvement of proof strategies. Some of the more or less readily identifiable strategies that students might be using to help them construct proofs are *rule-specific*, such as «If there is an arrow as the principal operator in the conclusion, break up the conclusion using $\rightarrow$ I.». Some of these rule-specific strategies help students to choose between rules; for example, «If there is a choice between $\vee$E forwards and $\vee$I backwards, try $\vee$E first.». The student Kusi described this as the "precedence" of $\vee$E over $\vee$I. In contrast to rule-specific strategies, there appear to be strategies that might be called *global strategies*; for example «When reasoning forwards, check if the lines produced are useful in obtaining the conclusion.»; «When reasoning backwards, check if the lines produced are provable from the premises.»; and «The principle operator in a line is the only operator that determines the legal rules applicable to that line.».

It could be argued that the proof strategies represent no more than an *ad hoc* collection of "rules of thumb" - purely *mechanical* responses to a limited set of straightforward syntactical inputs - that demonstrate little of the deep understanding that an experienced logician might have, and show little regard for the circumstances in which they might fail. However, Popperian psychology would emphasise that these strategies incorporate expectations about what a proof should look like, about why a particular rule might be applicable in certain circumstances, and about what might or might not be provable; and *these expectations constitute nascent knowledge.*

In the remainder of this section, we consider what this Popperian psychology may have to tell us about the role of the computer, about why the software seems to be able to help some students more than others, and about why certain logical rules seem to be harder than others.

---

[11]   See page 32: *A commentary on the use of theory in the analysis of the Jape study*

The role of the computer

The introduction of the computer has a profound impact on proof strategies. For example, it appears that novices' attention on paper is unfocussed – there are no universal strategies for deciding which line to attend to. But with the computer, students readily construct the strategy «Only try to prove the line directly below the ellipsis [the three dots that *Jape* uses to indicate missing lines]».

Another example is of students who have been following a paper-based strategy akin to «To create the line 'P∧Q', write it down, find P (on line *x*, say), find Q (on line *y*, say), and then write down the justification∧I, *x, y*» constructing, when using the computer, a strategy akin to «To create 'P∧Q', find a more complex line containing it, and find some rules that break the line down». The concern has changed from, "How do I justify this line I've written down?", to, "Which rule do I apply to which line to generate this line?" For example, the student Caroline is heard to ask, "How do you get the 'ands' to come into it?"

Two fallback strategies were noticed that are unavailable on paper. It was rare to see students using «Click on one of lines; keep on trying rules; try a different line», but when more sophisticated strategies failed, quite a few students fell back on the "symbol-matching strategy":

«Look for the 'main symbol' in the most complicated line. Find the same symbol in the list of rules. Try one of the matching rules. Undo the rule if the display doesn't look right (criteria for which might include any large, unexpected increases in the length of the proof, the number of boxes, the number of gaps in the proof, the appearance of unfamiliar symbols); and try another.»

In fact most of the students who were observed to be fixated on reasoning forwards on paper soon discovered (using this symbol-matching strategy) that the inefficient rule "make an assumption" was unnecessary and that «apply ➤I backwards» generated assumptions automatically.

One student drew attention to the almost mindless character of symbol-matching strategy: "He doesn't know what he's doing. … He's proving them but he doesn't know what's going on." Yet these were very much strategies of last resort – on the whole students seemed motivated to develop ever more efficient proof strategies, even though these require debugging and greater care in application.[7]The value of the software to the students could therefore be characterised as allowing students to consider many more examples than would be possible using pencil-and-paper, and to debug proof strategies that depend on incorrect conceptions of individual rules and inadequate conceptions of the requirements for a complete proof.

[7]   See page 31: *A commentary on the use of theory in the analysis of the Jape study*

Why the software seems to be help some students more than others

What follows is a particular argument based on the Popperian model, described in order to illustrate the style of analysis – other arguments are possible within this model. The analysis has been simplified quite a bit for the purpose of illustration. (A more detailed explanation of this argument, with examples of individual students' activity and talk, is available in Aczel, 2000.)

When considering why the software helps some students more than others, four groups of users can be considered, delineated with respect to their prior knowledge of the rules:

> 1. Those who know the name of the rule they want to apply, but are not necessarily aware of what the precise effects of the rule might be.

> 2. Those who know how they want the transformed proof to look, but are less sure about the name of the rule that achieves this transformation.

> 3. Those who have a limited grasp of the rules and are trying to work out from the output of the program how they can be used.

> 4. Those who have never met the rules before.

Group 1 and Group 2 students may already have some experience of tackling paper proofs before using *Jape*; Group 3 and Group 4 students have not. In fact, Group 4 students are not target *Jape* users at all.

Group 1 students - the nominalists - know the name of the rule, but are not necessarily aware of what the effects of the rule might be. While they may sometimes be surprised by the effects of applying a rule, such surprise is not automatically to be taken as indicative of error; only if further reasonable moves are blocked would error be suspected. The strategy for choosing the rule would be the assumed culprit.

On the other hand, Group 2 students - the causationists - know what they expect to see, and would suspect error if they did not see it. The name of the rule is of secondary importance. Typically, when such students have constructed the proof on paper, they ask themselves as they write down the justifications, "What is the rule that describes the step I've just carried out here?", rather than (say) writing down the justifications and trying to remember how the relevant rule works. Group 2 students therefore have a difficulty with *Jape* in that they are more comfortable carrying out transformations on a proof *without* being forced to use a named transformation. Their main difficulties centre on getting the interface to produce the effects they

want to see, and finding a way of remembering which rules correspond to the desired visual effects.

Note that this classification of students into Group 1 if they know the name of the rule they want to apply, and Group 2 if they know the step they want to apply, may not be applicable across all rules; but these user groups can serve as broad categories that might help in interpreting specific instances of student behaviour.

Each of these groups face a different strategic-theory problem situation; these will be outlined in turn for Groups 1-3 (Group 4 not being a target group of users). For example, at each stage of a paper proof, a Group 1 student:

> (i) chooses a rule to implement (using what might be called a "rule-choice strategy"),
> (ii) implements the rule (using a "rule-implementation strategy"); and
> (iii) justifies new lines (using a "justification strategy"):
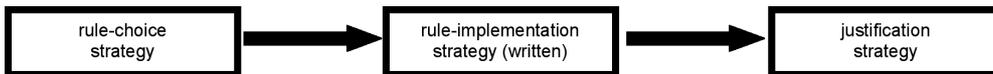

Figure 4: Paper proof – Group 1 students

The next question to consider is how this situation differs when using *Jape*. *Jape* takes care of the justification strategy, and makes the rule-implementation strategy much easier. It also provides feedback that can be used in an assessment of whether the new proof constitutes a movement in the direction of proof completion (what might be called a "progress-assessment strategy"), so as to debug their rule-choice strategy:
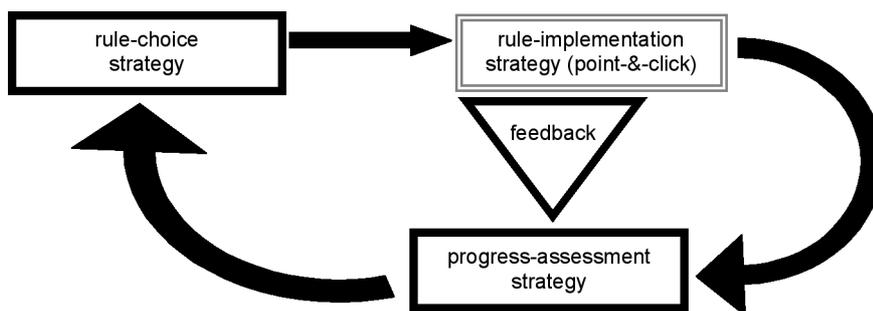

Figure 5: Jape proof – Group 1 students

By contrast, for each stage of a paper proof, a Group 2 student:

> (i) chooses a *step* to implement (using a "step-choice strategy" - note that we refer here to a "step" rather than a "rule", because a Group 2 student's strategy for constructing proofs does not initially require them to know the name of the rule); (ii) implements the step (using a "step-implementation strategy"); (iii) finds the rule corresponding to the step (using a "step-name strategy"); and so (iv) justifies new lines (using a justification strategy, as for Group 1 students).
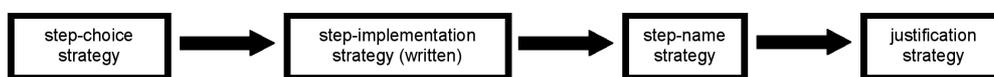

Figure 6: Paper proof – Group 2 students

Again, the situation differs when using *Jape*. *Jape* provides feedback that can be used in a progress-assessment strategy. The program also takes care of the justification strategy, but it does not allow the user to implement a step without selecting the corresponding rule. Hence with *Jape*, the step-name strategy has to be used *before* using the step-implementation strategy, unlike their approach for paper-based proofs.
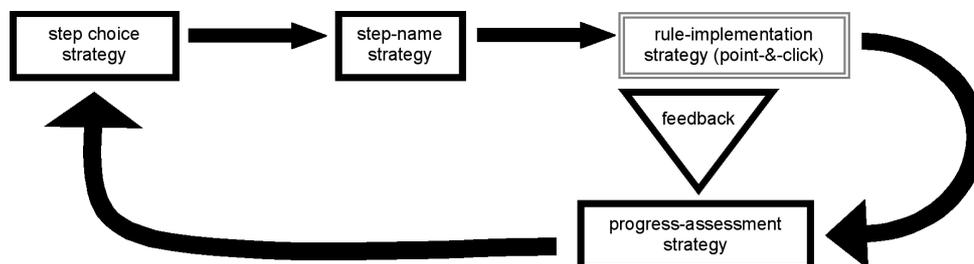

Figure 7: Jape proof – Group 2 students

So even though they have good progress-assessment strategies (they know what they expect to see) Group 2 students may find it difficult to improve their step-choice strategies using Jape because their step-name strategies are undeveloped. Their typical fallback strategy on paper «When all else fails, assume something» is particularly unhelpful in *Jape*. Consequently, Group 2 students appear to gain least from the software.

For each stage of a paper proof, Group 3 students will make best (but idiosyncratic) use of their limited knowledge to progress. Since they receive no feedback except through comparison with lecture notes and comments from tutors, it is difficult to predict how their strategies improve, or even whether they will end up as Group 1 or Group 2 students.

However, when Group 3 students start using *Jape*, this idiosyncratic process can be dramatically transformed. In order to progress, they must select a rule and use the feedback to determine if it was a good choice. Hence it is likely that *if* they are successful in learning from ItL *Jape* they will turn into Group 1 students rather than Group 2 students. But this transformation crucially depends on them having a viable embryonic rule-choice strategy (such as the symbol-matching strategy), and this can place strain on the memory because there will often be at least four possible rules to check (introduction and elimination rules for a premise and a conclusion), and sometimes more (if there are multiple premises or the possibility of proof by contradiction). So it is particularly important for Group 3 students to be systematic, to avoid any additional complexity, and to recapitulate soon afterwards in another form what they have learned. In the short-term, these students have the potential to gain most from *Jape*.

The above analysis can go some way to explain observed student behaviour in individual case studies, suggests potential gaps in students' knowledge that may remain (rule-implementation and justification), and produces testable hypotheses about the impact of changing aspects of the software on different user groups.

Why certain logical rules seem to be harder than others

Some rules appear to be harder than others: conjectures involving implication and conjunction tended to be seen by students as the easiest conjectures to prove; the Disjunction topic (in conjectures which could also involve implications and conjunctions) was next; Negation and Quantifiers were held in about equal dread. This perception is matched by measures of the time spent per proof attempt, and by success in written tests.

Unlike the cognitive dimensions model (earlier), or the cognitive load model (below), it has to be admitted that the Popperian model has no ready-made explanations for this. Nevertheless, it is possible that measures could be developed that attempt to capture (in something akin to the cognitive load model) the complexity of the proof strategies that students are typically using for a particular rule. It is also possible that the development of different proof strategies could be compared with respect to their reliance on a number of basic proof strategies that are more "basic" in some sense (a criterion something akin to "abstraction level"), or with respect to their reliance on properties of the proof and the interface that are more or less implicit (a criterion akin to "hidden dependencies"), or with respect to their reliance on an order of actions that is relatively inflexible (a criterion akin to "premature commitment").

To summarise, the weakness of the Popperian model is that it does not provide off-the-peg measures or dimensions. However, the strength of the model is that, by emphasising that new strategic theories are constructed from old and in response to an individual's concerns, attention

is immediately drawn – as with the Vygotskian approach – to the student's prior knowledge and to the precise sequence of learning activities. For example, it turns out that the perceived and measured order of difficulty of the topics matches the order in which the rules were introduced in the lectures, the order in which the rules were practised on paper, and the order in which the conjectures were presented in the software. This "order of difficulty" may very well depend on some absolute properties of the rules themselves, but the strong possibility that it is dependent too on the student's prior experiences should not be forgotten.

6   Cognitive Complexity

The fourth (and final) approach to explaining learning that will be considered in this paper is that of cognitive complexity. This approach focuses on problem solving activities, taking into account factors such as prior knowledge and 'cognitive load'. The idea of cognitive load rests on the assumption that people's capacity to process information is limited (Sweller, 1988); in other words, that the more a learner tries to hold 'in their head' at any point, the harder their learning will become. (This idea clearly draws upon the notion that there are analogies between the human mind and computers.) It is also assumed that some activities entail a higher cognitive load than others, so that (for example) integrating information from multiple sources is harder than studying a worked-out example (Sweller, 1989).

Within the context of this study, the relatively constrained and well-defined nature of the domain means that it is possible to apply the idea of cognitive load to the rules used during proof construction. For example, it could be argued that the rules ➤ E and ➤ I, when applied on paper, make similar demands on working memory:

➤ **E**	From A and A ➤ B we can conclude B		➤ **I**	If you assume A, and prove B, we can conclude A ➤ B

For ➤ E, for example, one might hypothesise (in the manner of Sweller, 1988) the following demands: "hold A in memory", "hold A ➤ B in memory", "apply this rule to deduce B". For ➤ I, the demands might be represented as, "hold A in memory", "hold B in memory", "apply this rule to deduce A➤ B".

So all other things being equal, one would expect students to fare equally well in implementing these rules.

However, the software changes the situation. $\rightarrow$ E still requires the recognition of the situation "A, A $\rightarrow$ B" in the premises, and the expectation that "B" will result from applying the rule. But in addition, the software requires a selection; the student has to choose the selection from, the "A" in the premise, the "A" in "A $\rightarrow$ B", "A $\rightarrow$ B", the "B" in "A $\rightarrow$ B", and the "B" in the conclusion. In fact the selection should be "A $\rightarrow$ B"; but either failing to make a selection or selecting the "B" in the conclusion create confusing results. The implication of this is that the cognitive load associated with $\rightarrow$ E goes up using the software.

With $\rightarrow$ I, on the other hand, the cognitive load is dramatically reduced. Not only is it not necessary to decide what to assume, and there is no problematic ambiguity about selection, but the consequences of applying the rule (in terms of both logical form and layout) are automatically handled by the computer. The only task is to recognise the form "A $\rightarrow$ B" in a conclusion, and apply the rule to it. The consequences do not need to be worked out by the student, because this particular rule-application is nearly always productive.

From roughly equal parity on paper, the cognitive complexity of the $\rightarrow$ E rule can be as much as five times that of the $\rightarrow$ I rule (based on measures such as those in Sweller, 1988) when learning is supported by the software. The empirical results indeed suggest that $\rightarrow$ I is much more readily used – and used successfully – at the computer than on paper.

Some important caveats

Yet although this analysis produces an accurate prediction of the increase in accessibility of the $\rightarrow$ I rule, some weighty assumptions and simplifications have been made.

For example, it is assumed that each of the demands described above are of the same size, yet our representation of the paper-based version of the $\rightarrow$ E rule as "hold A in memory", "hold A $\rightarrow$ B in memory" and "apply this rule to deduce B" could be rather an oversimplification. Perhaps a better representation is "identify A as a premise", "identify A $\rightarrow$ B" as a premise", "conclude B, justified by the rule $\rightarrow$ E". But this still leaves many questions unanswered: What are the demands of "identifying a premise"? How much harder is it to identify "A $\rightarrow$ B" than "A"? Moreover, if we are considering how students construct a proof of conclusions from premises rather than whether students can validate a given proof, we have to ask - what formal rules (e.g. Braine, 1978), cognitive schema (e.g. Chi *et al*, 1982) or mental models (Johnson-Laird, 1983) motivated having B as a conclusion in the first place?

Meanwhile, the representation of $\rightarrow$ I is even more questionable. What motivates the making of assumptions? Some students would appear to make assumptions as a last resort, when no other forward moves were possible. Other students would use the existence of conclusions of the

form "A ➤ B" to trigger ➤ I. Whether "A" is guessed or calculated must surely have a large impact on the memory demands of this rule. Moreover, does representing assumption scope by drawing boxes place demands on cognitive processing capacity? What is the effect of having to leave a space of indeterminate size for lines to link A to B?

Clearly, there are several fundamental problems that can be identified with this approach. Leaving aside the assumption that the mind-computer analogy is justified, it is clear that the notion of weighting is based on a fairly arbitrary model of the process of proof construction, with no indication about the level of granularity that this model should be constructed at. Additionally, whilst the analogy remains internally consistent for analysing the way in which given information is processed, it has little to offer as a way of explaining the spontaneous creation of additional information, for example in the form of new assumptions. Moreover, the theory has little to say about the modality in which information is presented, something which clearly does affect learning (Ainsworth, 1999). As such, its contribution to understanding the role of visualisation in the process of learning formal reasoning is inherently limited.

Nevertheless, this model does have some empirical evidence from cognitive psychology; and in the *Jape* research, the time spent on each proof correlates with the textual length of the statement of the conjecture, which is a rudimentary measure of complexity for simple proofs. The approach also helps to explain why students had difficulties with rules involving examining multiple cases (e.g. " ∨ E forwards"), incomplete steps (e.g. " ➤ E backwards") or ambiguities about left and right (e.g. " ∧ E forwards"). It is also possible that when students attempted the Negation and Quantifier topics using the software, some abandoned the process of properly implementing the rules with high cognitive loads in favour of a much less demanding click-and-see trial-and-error approach. However, it must be concluded that this model has not produced convincing accounts of the actual processes underlying students' reasoning, particularly in the more complex proofs, and especially not those involving negation. Nor does it succeed in explaining students' reluctance to use justifications, structural aspects of proofs, and semantic considerations as ways to reduce cognitive load in sometimes quite dramatic ways.

Despite the apparent existence of a "comprehensive theory in psychology to explain all the main varieties of deduction" (Johnson-Laird & Byrne, 1991), it is far from clear how students are reasoning in this formal context. Rips (1994) points out that the study of reasoning within experimental psychology has focused on rather specialised informal deduction tasks. In particular, the typical approach is to ask subjects to evaluate the validity of simple inferences or to draw conclusions from given premises. Yet the task that is faced by novice logicians is rather different – it is to construct a valid formal proof from given conclusions to given premises. It is at this point that the problems of cognitive complexity, which was originally developed as a way of explaining the benefits of learning from worked-out examples, become clear.

7   Conclusions

The problem identified at the outset of this paper involved re-visiting accounts of students learning logic using *Jape* in an attempt to make these explicable and meaningful. In order to do this, a number of theoretical interpretations of the situation were considered.[6] What, then, can be concluded, both about the situation being analysed and about the utility of these theories?

The first conclusion to draw is that, to a greater or lesser degree, each theory considered offered a useful but partial perspective on the situation. This is, perhaps, unsurprising; each originated from a different set of values and concerns, and thus makes sense of the phenomena in a way that emphasises certain aspects at the expense of others.[12]

The notion of the ZPD, for example, has provided a useful way of reconsidering the way in which the students learn with *Jape*. One of the things that this theory emphasises is that the important word in the preceding sentence is 'with', which might have been replaced by 'from'. Whilst the ZPD cannot adequately explain the fine-grained detail of the learning process, preventing it from becoming a detailed *model* of this situation, the notion of working with a more able peer and learning by internalising these social interactions recurs throughout the paper. In the section on Cognitive Dimensions, for example, the role of secondary notation and other cues as a way of providing feedback on actions is explored. Similarly, in the section on Popper-Campbell psychology, the process of revising strategies rests, at least in part, on the way in which students learn from what *Jape* does, and by working with the software to try out different approaches that, were they working on paper, would be beyond their means (either conceptually, if *Jape* is used in a trial-and-error manner, or in terms of number, given that *Jape* allows more examples to be considered) to attempt. Thus whilst the ZPD cannot, in itself, explain all of the phenomena observed in the studies, it does provide a way of framing the subsequent interpretations and taking a broader view of the learning process.

With the Popperian model, unlike the Vygotskian approach, learning mechanisms are examined in terms of the explicit analytical tools that are the hypothesised strategic theories and concerns. These qualitative tools enable detailed conjectures to be made about how a different instructional sequence, a different implementation of the propositional calculus, or changes to the interface might affect learning outcomes. For example, the analysis of users with differing prior subject knowledge can go some way to explain the relative perceived difficulty of the topics. In particular, that analysis would suggest that for Group 3 students, features of the interface that contribute to the systematic testing of embryonic rule-choice and progress-assessment strategies

---

[6]   See page 31: *A commentary on the use of theory in the analysis of the Jape study*

[12]   See page 33: *A commentary on the use of theory in the analysis of the Jape study*

would be appreciated; features that were irrelevant to this testing would be ignored; and features that inhibited this testing would be problematic. Examples of the latter features are significantly more common in the Quantifiers topic than in the other topics; such features also appear in the Disjunction topic to a lesser extent. In the case of the Negation topic, the fallback symbol-matching rule-choice strategy is inadequate to allow students to develop two particular strategies that are key to proving these conjectures. Thus by concentrating on the students' internal process of meaning-making, this particular theory allows predictions to be made about how specific changes will alter what students learn. As such, this specific internal interpretation of the situation complements the general, social interpretation offered by the ZPD.

The idea of Cognitive Dimensions changes the focus yet again. It, too, could be construed as a social model; as with the ZPD (as used in this context), it focuses on the interactions between user and system. However, whereas the ZPD proved difficult to use at fine-grained levels of analysis, these dimensions *only* work at this level. Whilst problems remain, particularly in terms of the subjective and relative way in which they are measured, the dimensions do offer a way of explaining and theorising about why certain rules or strategies were difficult to use, either on paper or with *Jape*. It also provided direct insights into the role of visualisation in this process, something that the other theories touch on only tangentially, if at all. By comparing the ways that learners interpret and interact with visual representations (as opposed to syntactic ones), it provides an insight into which features of *Jape* support and which hinder learning – insights which are of wider relevance to researchers working in the area of visual reasoning. In doing so, it also overcomes the problems of only supporting relativistic measures of the dimensions. Thus this approach has direct and immediate benefits, for example in terms of explaining why the use of the symbol "_P" as a placeholder hindered learning, whilst use of the ellipsis supported it. It also provides the prediction that rules such as "➤E forwards" should be easy to use in *Jape*, whilst " I(R) backwards" is likely to be difficult, allowing teachers to direct their attention towards supporting specific topics. In addition, this offers a way of anticipating the impact that changes in the software design, and in particular, of the way in which visual representations can be incorporated into software in order to support learning.

Finally, it must be concluded that although the notion of cognitive complexity appears, at first impression, to be well-suited to explaining learning in this context, it actually has little to offer. There is certainly a resonance between the notion of cognitive load and some of the measurements inherent in the use of cognitive dimensions, but the approach fails to move beyond relativistic measures to anything more objective, and lacks the versatility of considering different aspects of the process of interacting with the system.

To conclude, these separate analyses allow a much richer explanation to be offered of the situation under study. The introduction of a software tool, *Jape*, that uses visualisation

techniques to support formal reasoning changed learning in a number of identifiable and explicable ways. Firstly, it made certain proof-solving approaches easier, and others harder, as a result of the way that the interface was structured and information was represented. Specifically, it helped in ways such as cueing students in to issues such as missing lines that might otherwise have been missed, whilst hindering through the introduction of notation that misled either by making the notation more abstract or by appearing overwhelming in length or complexity. The way that students used the software – and the extent to which it supported or inhibited their learning – depended in part on the way that they understood proof construction. Those who knew the names of rules (but not what its precise effects might be) were able to learn effectively from *Jape* by trying these rules out, and 'undoing' them if they were unhelpful. Those who knew what they were trying to achieve, but did not know exactly what this 'move' was called, had problems, since the interface relied on using names to apply rules. Finally, those who had a limited grasp of the rules and were attempting to learn from their exploration of the software either had real problems or else turned into 'Group 1' students (i.e. those who knew the names but not necessarily the effects).

With all of these students, the learning process involved building up patterns of interaction that 'worked'; these strategies were developed and refined on the basis of whether or not they were useful in completing proofs. Moreover, it was possible to explain why some of these strategies were harder to use than others, with approaches such as, «Break up implications in the conclusion» being a safe, easily understood tactic, whilst the strategy «Make an assumption» involved guessing ahead about how the assumption might be used later in the proof and also added new, complicating information into the proof structure. Drawing all this together, it seems that the way in which students developed these strategies and thus learnt about formal reasoning involved working with *Jape* to test different approaches and try out unfamiliar rules, gradually internalising the feedback that resulted from their interactions with the system.

At the outset, it was possible to say what happened when students started using *Jape* to learn formal reasoning. Through the analysis outlined above, it has proved possible to develop these fragmented facts into an explanation which, although doubtless still incomplete and worthy of further elaboration, nonetheless provides a meaningful and credible account of how this learning took place.

Acknowledgements

[4]   See page 30: *A commentary on the use of theory in the analysis of the Jape study*

8   References

Aczel et. al. (1999) *Notes for the ICCE/INTERACT papers.* Internal project document.[2]

Aczel, J. C. (1998) *Learning Equations using a Computerised Balance Model: A Popperian Approach to Learning Symbolic Algebra.* Unpublished DPhil thesis, University of Oxford.

Aczel, J. C. (2000) "The Evaluation of a Computer Program for Learning Logic: The Role of Students' Formal Reasoning Strategies in Visualising Proofs", CALRG report, The Open University

Aczel, J. C., Fung, P., Bornat, R., Oliver, M., O'Shea, T, & Sufrin, B. (1999b) Using computers to learn logic: undergraduates' experiences. *Proceedings of the 7th International Conference on Computers in Education,* Amsterdam.

Aczel, J. C., Fung, P., Bornat, R., Oliver, M., O'Shea, T, & Sufrin, B. (1999a) Computer Science Undergraduates Learning Logic Using a Proof Editor: Work in Progress. *Proceedings of the Psychology of Programming Interest Group conference.*

Aczel, J., Fung, P., Bornat, R., Oliver, M, O'Shea, T. & Sufrin, B. (1999c) Influences of Software Design on Formal Reasoning. In Brewster, S., Cawsey, A. & Cockton, G. (Eds.) *Proceedings of IFIP TC.13 International Conference on Human-Computer Interaction INTERACT '99,* 2, 3-4. ISBN 1-902505-19-0. British Computer Society: Swindon.

Ainsworth, S. (1999) The functions of multiple representations. *Computers & Education,* **33**, 131-152.

Bornat, R. & Sufrin, B. (1996) *Animating the formal proof at the surface: the Jape proof calculator.* Technical Report, Department of Computer Science, Queen Mary & Westfield College, University of London. ftp://ftp.dcs.qmw.ac.uk/jape/papers/

Braine, M. D. S. (1978) On the relation between the natural logic of reasoning and standard logic. *Psychological Review*, **85**, 1-21

Campbell, D. T. (1960) Blind Variation and Selective Retention in Creative Thought as in Other Knowledge Processes. *Psychological Review*, **67** (6), 380-400.

Cheng, P., Holyoak, K., Nisbett, R. & Oliver, L. (1986) Pragmatic versus Syntactic approaches to training deductive reasoning. *Cognitive Psychology*, **18**, 293-328.

---

[2]   See page 30: *A commentary on the use of theory in the analysis of the Jape study*

Chi, M. T. H., Glaser, R. & Rees, E. (1982) Expertise in problem solving. In Sternberg, R. (Ed.) *Advances in the psychology of human intelligence*, 7-75. Hillsdale, NJ: Erlbaum

Crook, C. (1991) The Zone of Proximal Development: Implications for Evaluation. *Computers and Education.* **17** (1), 81-91.

Fung, P. & O'Shea, T. (1994) *Using software tools to learn formal reasoning: a first assessment.* CITE report no. 168, Open University.

Green, T. (1989) Cognitive Dimensions of Notations. In Winder, R. & Sutcliffe, A. (Eds) *People and Computers V,* 443-460. Cambridge University Press.

Guba, E.G. & Lincoln, Y.S (1989) *Fourth Generation Evaluation.* Newbury Park, CA: Sage Publications.

Howe, C. & Tolmie, A. (1999) Collaborative learning in science. In Littleton, K. & Light, P. (Eds), *Learning with Computers*. London: Routledge.

Johnson-Laird, P. N. & Byrne, R. M. J. (1991) *Deduction*. Hove, East Sussex, UK: Erlbaum

Johnson-Laird, P. N. (1983) *Mental models: towards a cognitive science of language, inference and consciousness.* Cambridge: Cambridge University Press

Kadoda, G., Stone, R. & Diaper, D. (1999) Desirable features of educational theorem provers – a Cognitive Dimensions viewpoint. *Proceedings of Psychology of Programming Interest Group conference, 1999.*

Luckin, R. (1997) *Ecolab: Explorations in the Zone of Proximal Development.* Unpublished PhD Thesis, University of Sussex CSRP No.386.

Oliver, M. (1997) *Visualisation and manipulation tools for Modal logic*. Unpublished PhD thesis, Open University.[3]

Popper, K. (1963) *Conjectures and Refutations*. New York: Harper and Row.

Rips, L. J. (1994) *The Psychology of Proof: Deductive Reasoning in Human Thinking.* Cambridge Massachusetts: MIT Press.

----

[3]   See page 30: *A commentary on the use of theory in the analysis of the Jape study*

Strauss, A. (1987) *Qualitative Analysis for Social Scientists.* Cambridge: Cambridge University Press.

Sweller, J. (1988) Cognitive load during problem solving: effects on learning. *Cognitive Science*,**12**, 257-285.

Sweller, J. (1989) Cognitive technology: some procedures for facilitating learning and problem solving in Mathematics and Science. *Journal of Educational Psychology,* **81** (4), 457-466.

van der Pal, J. (1996) The role of interactive graphics in learning to reason abstractly. Proceedings of the IEE Computing and Control division's *Thinking with Diagrams* colloquium. Digest no. 96/010, pp.14/1-14/3. London: Institute for Electronic Engineers.

Vygotsky, L. (1962) *Thought and Language.* Cambridge, USA: MIT Press.

Vygotsky, L. (1978) *Mind in Society: the Development of Higher Psychological Processes.* London: Harvard University Press.