Technology-supported environments for learning through cognitive conflict

Anne McDougall
The University of Melbourne
email: a.mcdougall@unimelb.edu.au

This paper examines ways in which the idea of cognitive conflict is used to facilitate learning, looking at the design and use of learning environments for this purpose. Drawing on previous work in science education and educational computing, three approaches to the design of learning environments utilizing cognitive conflict are introduced. These approaches are described as confrontational, guiding and explanatory, based on the level of the designer's concern with learners' pre-existing understanding, the extent of modification to the learner's conceptual structures intended by the designer, and the directness of steering the learner to the desired understanding. The examples used to illustrate the three approaches are taken from science education, specifically software for learning about Newtonian physics; it is contended however that the argument of the paper applies more broadly, to learning environments for many curriculum areas for school levels and in higher education.

Foreword

I was fortunate to enjoy a friendship of twenty years and a writing collaboration of some fifteen years with David Squires. I very much miss our productive and enjoyable discussions and arguments, his well informed and original contributions to our work, and, since our collaboration was from countries on opposite sides of the world, his frequent email messages. He was a most valued colleague.

Just before he went into hospital in November 2000, David sent me a draft outline for a chapter on software environments for learning through cognitive conflict, for a new book on which we were working at that time. I have developed this paper from a part of David's material.

Introduction

Piaget used the term disequilibration to describe the process of an individual's encountering a new experience that generates a contradiction with the individual's existing cognitive structures (see for example Piaget, 1977), and argued that cognitive development or learning occurs as the individual attempts to resolve this cognitive conflict, a process he referred to as accommodation. He proposed three possible types of accommodation: the individual might ignore the contradiction; the individual might hold two theories simultaneously, dealing with the contradiction by applying one theory in some specific cases, and the other in others; or the individual might construct a new, more encompassing notion that explains and resolves the prior contradiction. Achieving this last type of accommodation, that is learning, is the purpose of teaching and of educational software design in the present context.

Cognitive conflict and learning

Cognitive conflict in science education literature

The idea of cognitive conflict has been central in much recent work on learning and teaching, particularly in the area of science education. Terms such as cognitive dissonance, learner inconsistencies and divergent events are also used to describe this idea (see, for example, Driver, 1983; Fensham and Kass, 1988).

There is well established evidence in the science education literature that significant numbers of students have concepts and understandings of natural phenomena inconsistent with those of contemporary science. This has been found to be the case at all school levels, and for all of the phenomena and concepts studied (Fensham and Kass, 1988). These student views are referred to in the literature as misconceptions, alternative frameworks, or children's science. As a result of these findings there is currently great interest among science educators in the issue of conceptual change through the use of cognitive conflict.

If teaching is to facilitate conceptual change for students, it is important to appreciate and understand learners' existing views and understandings. Driver states that assimilation depends not only on the environment, but on the learner's existing cognitive structure (Driver, 1983: 53). Consistent with Piaget's argument outlined earlier, she argues that if there is no dissonance between an experience and a learner's cognitive structure then the information is assimilated without any change in the structure, and if the dissonance is too great then assimilation will not take place at all.

The challenge for teachers, and for educational software designers, is thus to present to learners a situation provoking an optimal amount of cognitive conflict, while appreciating that the optimal amount may vary widely among individual learners. Acknowledging the difficulty of this, Fensham and Kass note a significant gap between experimental studies on the use of cognitive conflict as a facilitator of student learning and classroom implementation of the research findings (Fensham and Kass, 1988).

Cognitive conflict and theory building

Some other writers, a leader among whom is Seymour Papert, regard the issue of cognitive conflict with a somewhat different emphasis from that in the body of work just described. Papert considers cognitive conflict to be part of the vital process of theory building in learning. He assigns rather less importance to the efficient achievement of 'correct' concepts, valuing the unorthodox theories and explanations developed by learners. He emphasizes the importance for learning of this process of building and modification of theories.

Papert argues that the unorthodox theories of young children are not deficiencies or cognitive gaps, but that they serve as a way of flexing cognitive muscles, or developing and working through the necessary skills needed for more orthodox theorizing. He asserts that educators distort Piaget's message by seeing his contribution as revealing that children hold false beliefs, which they, the educators, must overcome (Papert, 1980).

This leads to the development of learning environments that use cognitive conflict somewhat less directly than implied by the science education work outlined earlier. It suggests the creation of open-ended learning environments in which students might explore concepts and ideas, developing (possibly 'wrong' or transitional) theories and testing these. Papert uses the word *microworlds* to describe such learning environments.

Microworlds (Papert, 1980; Squires and McDougall, 1986) are self-contained environments, simple restricted worlds 'in which certain questions are relevant and others are not' (Papert, 1980: 117). Learners explore the properties of a chosen microworld in a completely open-ended fashion, developing and testing theories about the environment as they do so. Papert advocates the construction of many such microworlds, each with its own set of assumptions and constraints; a microworld designer can lead a learner to new understandings by careful control of these assumptions and constraints. In later work (see for example Harel and Papert, 1991; Harel, 1991) Papert and his colleagues advocate the construction by learners of microworlds of their own; however, this paper is concerned only with the development for teaching purposes of environments designed specifically to promote cognitive conflict in learners.

Cognitive conflict and intuitive knowledge

Andrea diSessa, who worked for some years with Papert, assigns even greater importance to learners' pre-existing cognitive structures. He discusses these in terms of intuitive knowledge, referring to the many fragmentary small structures that a learner gleans from experience in the world as *phenomenological primitives* or *p-prims* (diSessa, 2000).

DiSessa describes cognitive conflict in the following way. People have hundreds if not thousands of p-prims, and when they make a judgement of reasonableness or unreasonableness they are frequently summoning all the relevant p-prims and deciding which one, or which collection of a few, best matches the situation. Then what happens in the situation is reasonable if it matches the chosen p-prim or p-prims and surprising if it does not.

Like Papert, diSessa is critical of what he sees as educators' under-valuing of learners' intuitive knowledge, arguing that learners are adjusting their p-prims to new experiences all the time and that it is not difficult to make small changes of this kind.

Approaches to the design of learning environments utilizing cognitive conflict

In the light of these views, it is appropriate to consider implications for the design of learning experiences using cognitive conflict. For purposes of design, the three approaches just outlined might be classified in the following way. The first, from the science education literature, might be called a *confrontational* approach; in such environments learners are confronted by the scientifically 'correct' concepts and laws in order that the puzzlement or cognitive dissonance thus generated will lead to modification of the learners' cognitive structures to match conventional scientific understanding. The designer may or may not have a detailed understanding of the range of views brought to the situation by learners. The design of the environment aims, through clear presentation of the concept or idea to be understood, to change as directly as possible any misconceptions or alternative frameworks previously held by learners.

The second approach, based on the ideas of Papert outlined above, might be thought of as a strategy in which the designer *guides* the learner to build theories and to modify these in ways more nearly approaching scientifically accepted ones. The learning process here will probably be more gradual, involving the development and modification of a series of increasingly 'correct' theories and understandings.

The third approach, based on diSessa's work, might be called *explanatory*. This strategy requires that the designer have some considerable appreciation of learners' prior understandings. It involves less actual change in concepts as such, but rather a deepening of the learner's understanding of the ideas under consideration. It is most appropriately used when the learner's conceptual structures are not far from the accepted 'correct' ones.

No one of these approaches is intrinsically superior to the others; each will be valuable in some situations. They are delineated and examined in some detail here in order to emphasize the importance of thinking about exactly what is required when learning environments are designed to utilize cognitive conflict for learning. The three approaches will be illustrated with examples in the next section.

Examples of learning environments utilizing cognitive conflict

This section describes three learning environments to illustrate the three approaches outlined above. These examples are designed to utilize cognitive conflict to assist learners' understanding of concepts in Newtonian physics. The first, Dynaturtle, is chosen to illustrate the confrontational approach; the second, a sequence of Newtonian microworlds, illustrates the guiding approach; the third, a ball-toss microworld, provides an example of the explanatory approach.

The confrontational approach: Dynaturtle

Although the work of diSessa has been identified with the explanatory approach to the utilization of cognitive conflict, an environment designed early in his work will serve well here to illustrate the confrontational approach. This software environment, called the Dynaturtle (Papert, 1980; diSessa, 1982), is a microworld developed to enable exploration of a formal law of physics, Newton's first law of motion. The original version was written in Logo, though it is not difficult to construct the environment in any of several contemporary programming languages; the code provided in Logo by Abelson (Abelson, 1982: 121–5) is most easily rewritten in Logo's contemporary descendant, MicroWorlds.

Newton's first law states that a body in motion will, if left alone, continue to move forever at a constant speed and in a straight line. This contradicts common experience, in which motion appears to result from the exerting of a force and removal of the force is associated with cessation of the motion. The theory conflicts sharply with everyday observation.

The dynaturtle is a modified version of the standard Logo turtle; in addition to the two geometric components, position and heading, associated with the standard turtle, the state of a dynaturtle is further specified by two velocity components. A set of dynaturtle procedures, and suggested activities for working with these are given in a then-current version of Logo by Abelson (1982: 121–5). Abelson introduces the microworld as follows:

> A dynamic turtle or dynaturtle behaves as though it were a rocket ship in outer space. To make it move you have to give it a kick by 'firing a rocket'. It then keeps moving in the same direction until you give it another kick. When you change its direction, it does not move in the new direction until you give it a new kick. Its new motion is a combination of the old motion and the motion caused by the new kick. (Abelson, 1982: 121)

Initially the turtle will stay at the centre of the screen, facing upwards. Three dynamic commands can be given:

- typing R causes the turtle to turn right 30 degrees;
- typing L causes the turtle to turn left 30 degrees;
- typing K gives the turtle a 'kick' in the direction it is heading.

Abelson suggests some activities for exploring the dynaturtle's behaviour: changing the direction of its motion, attempting to make it move horizontally across the screen or to make it go faster or slower without changing its direction, to make it stop, and to make it stop at a pre-marked position.

The guiding approach: sequential Newtonian microworlds

Papert suggests the development of a series of sequential microworlds, to lead a learner toward understanding of Newtonian physics (Papert, 1980). These microworlds would contain turtles whose behaviours are characterized and constrained in different ways, and Papert argues that such microworlds might be used to assist a learner familiar with the standard Logo (or more recently MicroWorlds) geometry turtle to develop understanding of Newtonian physics.

The geometry turtle has its state specified by its position and its heading. Essentially it responds to two state change operators, the FORWARD (number) command which changes its position and the RIGHT (number) command which changes its heading. In Newtonian physics there is only one state change operator – force, which changes momentum. So a Newtonian turtle should accept only one kind of command – one that changes its momentum. Papert proposes the development of microworlds that lie between that of the geometry turtle and that of a Newtonian turtle. Thus a velocity turtle would have its state specified by its position and its velocity, with a single state change operator SETVELOCITY (number). An acceleration turtle would similarly have its state specified by its position and its velocity, with a state change operator that changes the velocity by a specified amount, such as SETACCELERATION (number).

Papert argues that experience with the sequence of turtles – geometry turtle, velocity turtle, acceleration turtle, Newtonian turtle – would constitute a path into Newtonian physics that builds gradually and systematically, in a clear and transparent way, on what a learner already knows at each stage of the process.

The explanatory approach: vector representation of tossing a ball

In more recent work (diSessa, 2000), diSessa presents a microworld for uniform motion, similar in some ways to the dynaturtle procedures described earlier. The dynaturtle was developed in Logo; the newer 'tick model' version (so called because it shows what happens at each 'tick of the clock') is developed in the Boxer programming environment.

DiSessa shows the effect of running the acceleration tick model program with a large value for the initial velocity and a (constant) negative acceleration – as would represent tossing a ball into the air. The object moves upward while slowing down, comes to a peak where it appears to stop for an instant, and then falls downward. The actual motion is familiar from everyday experience. However, as the program runs, the constant negative acceleration and the changing values of the velocity are shown dynamically by vectors of lengths that change (or not) appropriately as the motion progresses. Thus an explanation for the motion, which is not easily arrived at intuitively, is drawn to the user's attention and clearly illustrated.

> Most people, and almost all students, find this uniformity amazing. Few are prepared to believe that going up works in exactly the same way as falling. Even more, essentially everyone believes the peak of the toss is very special, expressed in some sort of balancing (say, balancing of the upward toss force with downward gravity). In contrast, the top of a toss is really an innocuous happenstance, that adding [negative] *a* to *v* will drive *v* through its zero length position. (diSessa, 2000: 37)

Here the motion matches experience; what needs changing is many learners' understandings of the explanation for the motion. The program describes a process that is the same at every instant; the only thing that changes is the value of the velocity. This is designed to facilitate learners' modifying their previous intuitive ideas to understand that the force acting is constant throughout.

Issues in the design of technology-supported learning environments utilizing cognitive conflict

The design and use of learning environments utilizing cognitive conflict to promote learning necessarily involve consideration of three processes: concept representation, conflict recognition and conflict resolution. Of these, the process most critical for design is concept representation. The effectiveness of the learning environment in facilitating the learner's recognition and resolution of cognitive conflict depends largely on effective concept representation in the environment.

This section will consider these processes, focusing particularly on concept representation. A framework provided by the ideas of the confrontational, guiding and explanatory approaches to design will be used to structure the analysis, and features of the software environments described in the previous section will be used to exemplify issues discussed. Although educational software examples have been presented, the discussion so far has not otherwise been particularly related to technology-supported learning environments. I shall look briefly now at some matters relating specifically to the use of technology-supported environments in the present context.

Concept representation

In the confrontational approach the 'correct' or target structure must be represented in the environment. For example, in the Dynaturtle software the Logo procedures governing the behaviour of the dynaturtle provided a formalism for representing the 'target' concept of Newtonian motion. Typically in this approach the designer will make assumptions about the cognitive structures likely to be brought to the situation by learners but will not attempt to represent these directly in the environment. In the Dynaturtle case the designer's conjectures about likely learner expectations for the motion of the object, based on everyday non-Newtonian experience, are evident in the suggested activities for learners. However there is no attempt to represent directly in the software any prior understandings or expectations that learners might bring to the experience.

The guiding approach requires that the designer have a good appreciation of the conceptual structures likely to be brought to the situation by learners, as this approach requires the planning of experiences that will help learners to build on their existing conceptual structures to develop more 'appropriate' ones. Explicit representation of these structures in the environment may not be necessary. However if the developmental sequence is to be effective, the designer must apprehend not only learners' initial conceptual structures but also changes to these that are likely to result from experiences within the environment. In Papert's sequential Newtonian microworlds, the concepts overtly represented are some relatively abstract ideas, appreciation of which should facilitate learners' understanding of Newtonian physics. These ideas include the significance of an object's attributes, use of these attributes to specify the object's state, and the idea of state change operators. Concepts such as these may already be at least partially grasped by learners. The characteristics of the geometry, velocity, acceleration and Newtonian turtles represent these concepts directly, in a sequence of increasing sophistication.

From the point of view of design, the explanatory approach is perhaps the most subtle of the three described. In comparison with the previous approaches, this one requires the designer to have a clear understanding of the conceptual structures of the learner and, further, a reasonable idea of the learner's underlying rationale for these structures. In the ball toss example, the designer is confident that learners can make correct predictions about the motion based on their current conceptual structures, but he anticipates that these structures contain weaknesses that are revealed only when reasons or explanations for the predictions are investigated. Compare this with the Dynaturtle example, where the designer could be confident simply that the learner would predict the motion incorrectly. In the ball toss example the designer, expecting learners to make correct predictions for the nature of the motion, represents in the software the underlying physical explanation for the predicted motion – an explanation that he knows from study of learners' 'intuitive knowledge' of physics is widely unexpected. As well as illustrating the actual motion through animation, he uses the Boxer vectors to represent and illustrate the constant negative acceleration and the ball's changing velocity, the essential explanatory concepts to be understood in association with the motion in this situation.

Through increasing the options for concept representation, the use of technology has enhanced considerably the potential for design and development of learning environments utilizing cognitive conflict. An exhaustive discussion of the possibilities here is beyond the scope of the present paper. However it is appropriate to mention the ability of these environments to provide multiple formalisms for representation of concepts for students, including linguistic, graphical, animated, mathematical, programming code and many other representational modes. Papert argues, for example, that before the advent of the computer, there were only very poor materials available for the construction and representation of a Newtonian 'world' (Papert, 1980). Further, diSessa emphasizes the importance of direct experience in learning, arguing that the use of computers can 'move us dramatically away from the near-zero involvement of direct experience that is the hallmark of learning science only by reading books or even by talking with teachers and peers' (diSessa, 2000: 99).

Conflict recognition and resolution

For effective utilization of cognitive conflict in a learning environment, the designer must ensure that the consequences of the differences between 'target' and learner conceptual structures are evident and understandable by the learner. For learners, apprehension and articulation of a puzzling or unexpected situation is a major step towards resolving it. In environments designed with the confrontational approach the conflict is generally immediate and very obvious. In the guided approach where the designer expects gradual, probably small changes in the learners' cognitive structures, the recognition of conflict may not be as striking.

Using the explanatory approach the designer must take great care to ensure that the recognition of conflict with previously held understandings in fact actually happens. This might mean directing the learner's attention quite deliberately to the aspects of the environment designed for this, perhaps through suggested activities to be undertaken in the environment or by some other suitable design strategy. To illustrate this, consider again the ball toss environment. Since the motion of the ball is not surprising to most learners, attention must be directed to the explanatory part of the software, the Boxer vectors representing the acceleration and the velocity of the ball, to facilitate recognition of any conflict with learners' previous understanding of the ball toss situation.

Some aspects of technology-supported learning environments can be used to enhance the processes of conflict recognition and resolution. For example, interactivity in well designed technology-based learning environments can provide feedback to learners about the appropriateness of their assumptions and predictions as they work in the environment, and technology might be used to sponsor collaboration to help conflict resolution.

Conclusion

Based on work in science education and educational computing, this paper argues against considering the strategy of cognitive conflict for learning simplistically, describing three approaches to the design of learning environments utilizing cognitive conflict. These approaches are referred to as confrontational, guiding and explanatory, and differences among them are discussed in terms of the relative importance of learners' pre-existing concepts and the directness of, and the designer's control, of learners' pathways to improved understanding.

The processes of concept representation, conflict recognition and conflict resolution are discussed, and the most important of these from a design perspective, concept representation, is examined in some detail in the context of the three approaches. Finally the paper considers the power of technology-supported learning environments to enable, particularly through increased options for concept representation, the design and development of learning environments that utilize cognitive conflict in the more complex and subtle ways described in the three approaches delineated in the paper.

Although much of the recent work on cognitive conflict has been undertaken in science education, the argument presented in the paper should not be confined to learning in science. Papert points out that 'Piaget has shown that children hold false theories as a necessary part of the process of learning to think' (Papert, 1980: 132–3). It is contended that the argument made here has application well beyond science education, in learning in a wide variety of subject areas and contexts.

References

Abelson, H. (1982), *Logo for the Apple II,* Peterborough, NH: Byte/McGraw-Hill.

DiSessa, A. (1982), 'Unlearning Aristotelian physics: a study of knowledge-based learning', *Cognitive Science*, 6, 1–36.

DiSessa, A. (2000), *Changing Minds: Computers, Learning, and Literacy*, Cambridge, MA: MIT Press.

Driver, R. (1983), *The Pupil as Scientist?*, Milton Keynes: Open University Press.

Fensham, P. and Kass, H. (1988), 'Inconsistent or discrepant events in science instruction', *Studies in Science Education*, 15, 1–16.

Harel, I. (1991), *Children Designers*, Norwood, NJ: Ablex Publishing Corporation.

Harel, I. and Papert, S. (eds.) (1991), *Constructionism*, Norwood, NJ: Ablex Publishing Corporation.

Papert, S. (1980), *Mindstorms: Children, Computers and Powerful Ideas,* Brighton: Harvester.

Piaget, J. (1977), *The Development of Thought: Equilibration of Cognitive Structures*, New York: Viking.

Squires, D. and McDougall, A. (1986), 'Computer-based microworlds – a definition to aid design', *Computers and Education*, 10 (3), 375–8.