# *TNT*: Technological Thinking, with No Technology

Ian Frank, Malcolm Field
Mirai University, Japan

Track: ICT in curriculum and pedagogical transformation

**Abstract**

*How can technological thinking be taught without using technology? We ask this question because of the potential offered by myriad good answers. Most importantly, if significant aspects of technological thinking can be taught without direct recourse to technology, the options for reinforcing such thinking throughout a curriculum can be greatly increased. In fact, simply having a clear idea of what technological thinking* is *can already make a significant step towards this goal. A further important application is situations, such as developing economies or relatively underfunded education systems, where resources for acquiring technology are inadequate. This paper expands on the approach we have called TNT (Technological Thinking, with No Technology). We survey existing ideas to initiate a concrete list of teachable skills, show how teaching can be based on these skills, and discuss how the TNT mindset is linked to creativity.*

## Introduction

*It is important for those who do not usually doubt to have doubts, and it is necessary for those who have doubts to resolve them.*
— Zhu Zhi

We have been experimenting with new educational practices that draw on a diversity of sources, including artificial intelligence, communication, Japanese culture and robotics (Frank & Field 2006, Frank & Field 2007). In particular, we have used the notion of *koto* to underpin a target of 'interaction' between people and an environment with a philosophical understanding of how the world can be apprehended, rather than with any particular theory of learning. This enables our teaching to share aspects of numerous theoretical approaches, including Dewey's emphasis on the social, Vygotsky's social formation of the mind, and Problem-Based Learning. Our work has led us to the original research challenge: 'How can technological thinking be taught without technology?'

‘Technological Thinking, with No Technology’ may seem self-contradictory (note that the acronym $TNT$ is self-prescriptive: the first ‘T’ of ‘Technological’ is dropped). But the tension of reconciling paradoxes can be creative. This paper suggests that $TNT$ can unleash significant potential, for both students and teachers alike.

We begin by summarising our overall $TNT$ philosophy, making a case for how the approach can provide fundamental support for technological thinking as widely as possible across a curriculum. We especially highlight how $TNT$ has potential within education systems that have relatively limited resources. We then draw on the existing literature to seed a list of candidate skills key to successful technological thinking. We give examples of how ideas from this list can be implemented in real activities and also expand on the benefits of adopting a $TNT$ mindset, showing how novel ideas can be a result of the approach.

## Background

As investment in educational technology continues to increase, so does the need for more systematic research. Traditional studies have been interested in discovering the ‘successes’ of innovations, producing a ‘list’ of winners and losers (Zhao, Pugh, Sheldon & Byers, 2002). However, many early findings are approaching obsolescence (for example, results on the benefits of drill-and-practice type software, instructional television, or Web-based closed-exercises) due to the expanded opportunities for interaction offered by Information and Communication Technologies (ICTs). Nevertheless, we can still stop to consider the question of whether ICTs may be receiving more credit than they really deserve.

There are two elements that we feel are commonly overlooked: firstly, the interface between the technology and the educational institution (Zhao, et al 2002); secondly, the influence of culture (educational, societal, local and global) on the interaction between the learner, the teacher, the institution, the educational philosophies of the society, and the previous learning experiences of the students (Field, 2005).

Researchers from different countries (Levy, 1997, Cuban, 2001) have already challenged some of the claims made about ICTs in education. For instance, although technology can provide undoubted opportunities, (Jungck, 1987) has discussed how uninformed development and implementation can lead to adverse effects. Also, others such as (Sakamoto 1992, Warschauer 1999) have shown how interaction between pedagogical practice and ICTs can obstruct students’ development of critical enquiry and analytical skills, or their understanding of learning or cross-cultural appreciation.

We would argue that one reason explaining the lack of comprehensive evidence for the benefits of the ‘new approaches through technology’ is that we are still engaging in linear structures; that is, within distinct academic disciplines, rather than blending across-disciplines (refer also (Brody 2007)). Although it was not our original intent, *TNT* actually represents some form of blending approach, since it can unify varied curriculum areas through the targeting of common goals (as we discuss in the following section). The 'no technology' nature of *TNT* also allows us to sidestep—at least initially—the direct concerns over the pedagogical use of ICTs in the classroom.

Nevertheless, we expect *TNT* to feed back into teaching using technology, and we further anticipate *TNT* helping to understand what can be achieved *before* the discussions on ICTs become relevant. That is, what conditions can we create in early education that will make learning through ICTs more effective when it is actually introduced? Are there examples of Good Practice (GP)? And, how can we use the fundamental elements in culture or society that affect peoples' learning and use of technology?

According to Mioduser (1998), technological thinking requires a repertoire of technological primitives that entail four categories: rudiments, mental models, method and meta-knowledge. We have come to believe that this repertoire—especially the important final three categories—can be explicitly addressed even before any engagement with ICTs; that is, as an *a priori* experience. In previous work, we have developed workshops that create interactive experiences just with everyday materials (more information on these *koto-tsukuri* workshops can be found at `koto-tsukuri.org`). It was the breadth of subjects that we found could be addressed in this way that led us to us consider how even technological thinking could be introduced to students with just the simplest of tools.

## The *TNT* Approach

Figure 1 depicts the overall concept for the development of *TNT*. The primary goal is a formalisation of technological thinking primitives, with accompanying teaching practices and workshop activities. To this end, reading and literature survey will support concrete activities in three areas: technological education, across-curriculum support, and dedicated workshops:

- **Development of materials for technological education**. Framework for these materials will come from two primary sources. First, literature on technology, education, and thinking in general. Second, the monitoring and examination of difficulties experienced by learners both in technological subjects (*e.g.*, computer programming) and other subjects (*e.g.*, those which may use ICTs as a tool, even incidentally).
- **Reinforcement of technological thinking in classes other than those directly related to technology**. We believe it is possible to strengthen technical university education through the use of explicit unifying goals. While it is clear that classes making use of technology can reinforce appropriate modes of thinking, *TNT* represents the challenge of achieving this across the curriculum.

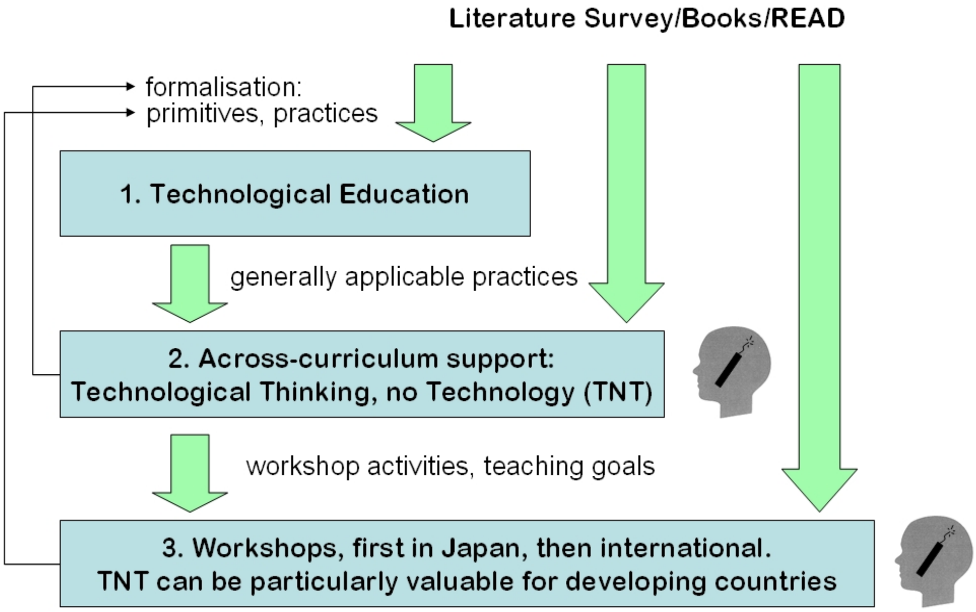


Figure 1: An overview of TNT

- **Application through local and international workshops** (since we are based in Japan, these workshops will start from this base). A 'proof of concept' of the *TNT* philosophy will be the successful development and execution of dedicated workshops that directly target technological thinking, but themselves require no technology. Significantly, such workshops have a unique potential for addressing technological thinking in less privileged environments, where access to resources is at a premium. Further, we suspect that in more plentiful environments technology is often pressed into service without addressing the understanding of required thinking, thus limiting effective use, as well as pedagogy. Through activities requiring just everyday objects, *TNT* has the potential to address technological thinking at early ages, before explicit exposure to technology has started to fix mindsets.

We envisage that *TNT* processes will be useful for administrators, educators, researchers and policy makers alike. Especially, there is little work on across-borders on technological thinking, and our background of an Asian context has naturally led us to confront differences in cultures.

One issue closely connected with *TNT* when viewed from a cultural perspective is multi-literacy. While we appreciate the need for multi-literate skills, we have concerns whether ICTs should be seen as the primary reason for this need, or whether multi-literacy can be viewed primarily as resulting from the implementation of ICTs into a curriculum. A brief review of the history of cultural practices shows that multi-literacy was an essential part of many educational systems long before ICTs. For instance, Omolewa (2007:595) has pointed out ‘Although Africans do not have the same and equal educational experience in traditional ways of knowing, it would not be out of place to describe the basic characteristic of traditional education in Africa as that which is intimately integrated with the social, cultural, political, occupational, artistic, religious, and recreational life of the people. It is usually stored in people’s memories and activities and is expressed in stories, songs, folklore, proverbs, dances, myths, cultural values, beliefs, rituals, community laws, local language and taxonomy, agricultural practices, equipment...’ Similar characterisations could be made of traditional Australian indigenous and New Zealand Maori cultural ways of learning.

$TNT$ also occupies an unusual position in the debate on *transfer*. According to Perkins and Salomon (1992), transfer occurs when learning in one context or with one set of materials impacts on performance in another context or with other related materials. Clearly, $TNT$ encourages transfer in this sense, since, as shown in our figure, it targets cross-disciplinary support of unified goals. An important issue in transfer is that the context of learning (exercise books, tests, simple streamlined tasks) often differs markedly from the ultimate contexts of application (in the home, on the job, within complex tasks). In contrast, as we shall discuss in the section on $TNT$ and creativity, a $TNT$ mindset leads to the use of everyday objects and activities to illustrate technological ideas. Thus, we are encouraged to facilitate the *complementary* transfer from the everyday to the technological, and have a natural perspective on the reflexivity of transfer (we recommend (Billing 2007) for an excellent summary of the transfer literature).

In this initial $TNT$ paper we first focus on the foundational work of defining a list of explicitly addressable skills. Then we provide some examples of how concepts from this list can be taught without technology, and finally examine the $TNT$ mindset itself. Note that in tandem, we have developed a 90 minute $TNT$ workshop that demonstrates the teachability of these ideas without recourse to technology, described in a separate, short paper in this proceedings.

## Technological Thinking, v0.1

There are many perspectives on technology, and on technological thinking. We provide a foundation based on our own backgrounds in ICT, Artificial Intelligence, and in the teaching of undergraduate programming. Since an expansive treatment would require a paper or more to itself, we have condensed the presentation here in order to allow the remainder of the paper to convey the overall pedagogical arguments.

Of the four categories proposed by Mioduser (rudiments, mental models, method and meta-knowledge) it is the first that seems most problematic for $TNT$. For instance, it must be easier to teach the basic operation of a text editing package while using the software itself (although some may argue whether this statement applies to Microsoft Word). We will concentrate on the final three categories, which have much of the character of general problem-solving skills. This allows us to proceed with the following list:

- **Remember four problem-solving steps:** 1) Understand the problem, 2) Devise a plan, 3) Carry out the plan, and 4) Look back at the solution. These four steps actually come from (Polya 1945) in the field of mathematics. But, as the book's title of *How To Solve It* suggests, the treatment is very broad in scope, tracing the science of 'heuristic' back through Descartes' *Rules for the Direction of the Mind*, and Pappus of Alexandria (300BC). Of particular value is Polya's emphasis on the final step of 'looking back', which we ourselves often find that students are unused to practising systematically.

- **Understand the 'Develop-test cycle'**. Relatively, the least developed area in Polya's book is the third step of 'Carry out the plan'. In technological thinking on the other hand, the development cycle has been much studied as an inductive process of discovery about the environment, and the testing of new models.

- **Use algorithmic thinking**. Characterised by Kramer (2002) as 'a methodical, meticulous, almost maniacal dedication to process flow integrity and to 100.00000% redundantly, superfluous, total, complete accuracy.' One way of approaching the development cycle. Kramer also calls algorithmic thinking a 'mental discipline that coalesces after years of repetition of operations called education and application.'

- **Is it correct?** A generalisation of the 'test' portion of develop-test. Polya already provides examples and techniques for questioning a result that has been produced. For instance, what conditions should be tested? Ben-Ari (1999) uses the explosion of the Ariane 5 rocket as an example to illustrate computer science concepts: 'Even a single test with a *representative input* would have uncovered the problem.'

- **Structure the problem**. In different fields, structuring can take on specific meanings. For instance, in programming, it is a way of *guaranteeing* the correctness of a result. Edward Dijkstra, in *Notes on Structured Programming*, says: 'When... the programmer's task [is] to produce a correct program [and] also to demonstrate its correctness in a convincing manner, ...the object he has to produce must be usefully structured.'

- **Use black box thinking**. A testing technique where the innards of the thing being tested are assumed to be unknown and inaccessible: it is a black box. By analogy, focussing instead on the internals is 'white box' thinking, and a mixture of the two approaches is 'grey box'.

- **Destroy the prototype and start again**. Even if a produced solution is correct, it may be poor. Very often it is better to redefine the prototype without the limitations and constraints that have been created within the original.
- **Is it elegant?** One reason for the poor quality of a solution. The desire for elegant solutions features widely in mathematics and in technology. For instance, Donald Knuth has said in the field of computer science: 'Something is elegant if it is spare, memorable and pleasingly symmetrical.' (Platoni 2006)
- **Be literate**. One aspect of literacy in technology is the documenting of progress so that others (and the originator) can understand and re-use ideas at a later date. *Literate programming* is another of Donald Knuth's contributions to computer science: 'programs should be written so that people, not just computers, can understand them... An ideal program, [Knuth] says, can be read by the fireside, like good prose.' (Platoni 2006)
- **Remember the social**. Just as technology should be developed for humans to understand, so it should be remembered that technology will also be *used* by humans. Reeves & Nass (1996) have shown that people react in a surprisingly social way to technology, for instance prefering a computer whose character matches their own.
- **What is the pattern that connects?** The phrase 'pattern that connects' is due to Bateson (1979). It has deep resonances in much of science, and could actually be used as an umbrella term for much of the above list. One extra item that it further allows us to address is historical perspective. This aspect of the 'pattern that connects' is clearly teachable without actually using technology.

The brevity of this embryonic list should actually have one benefit: the ideas are general enough to suggest to teachers of any subject how they can be connected with class activities. For illustration, we give below two examples that address the apparently most challenging application: that of teaching *technology* with no technology.

## Example 1: The Turing Test (A Paper Black Box)

The Turing Test was proposed in 1950 by the British mathematician Alan Turing as a way to understand machine intelligence. Modern versions involve a computer-mediated 'chat' session with a partner that is either a computer or a human: correctly guessing the identity is the test.

This procedure can be carried out in a classroom, and is a good demonstration of 'black box' questioning. The first author has in the past organised (for a university open day) an event with ten simultaneous chat sessions.

Students rotate round each of the ten computers, and then their guesses are compared. Although student feedback on the event was extremely positive, the setup work is not trivial, requiring at least two rooms with at least ten computers in each, operators for each of the 'unseen' computers, and download and setup of numerous automated 'chatbots'.

Only some time after the event, while considering $TNT$, did it become clear that a similar lesson could be imparted in a much simpler way: use a set of large, paper cards. The face of each card has a dialogue, and the reverse reveals whether the dialogue is between human and human or human and computer. The students simply move around all cards in turn, and their answers are all tallied before turning the cards one by one. In this scenario, interaction with actual computers is clearly absent, but the students may even *increase* their explicit thinking about the task by discussing with each other the reasons for their decisions. The teacher can also debrief with 'What questions would *you* have asked?'.

### Example 2: The Butterfly Effect (Is it Correct?)

Edward Lorenz's naming of the butterfly effect provides an excellent historical lesson. As described by Roger Von Oech (2002, Page 38), in 1960 Lorenz had developed a simple computer model for predicting the weather:

> One day, Lorenz needed to recheck the results of a long calculation. He decided to take a short-cut, and entered the same data he had used previously but rounded it to the nearest one-thousandth rather than to the nearest one millionth (for example, .506 instead of .506127). He thought this would have little impact on the overall result — perhaps no more than one tenth of one percent. When he looked at his printout patterns, however, he was amazed to discover they were significantly different from the first run. He soon realised that even an infinitesimal change in the numbers reflecting wind, temperature, or pressure conditions would be magnified [...] and the end result would be greatly altered.

Even exposure to this story alone—given students with some explicit awareness of the higher level notions such as those we have brought together here—can lead to learning. For an activity that can directly enable students to experience their own tendency to question too little, we can recommend a revealing experiment we encountered in (Bateson 1979). Give the students a simple sequence of numbers, such as $2, 4, 6, 8$, and tell them they may ask as many questions as they like before guessing the secret rule behind the sequence. Questions must be in the form of a new sequence of numbers: the teacher replies 'yes' if the new sequence obeys the secret rule, and 'no otherwise'. You might like to imagine how you would guess before stating the rule. Many students will be satisfied with a small number of questions before answering ‘each number is bigger by two’, or maybe ‘four numbers that increase by two’. Few will check their intuition to the extent of finding the actual rule: ‘Any sequence of four non-negative, increasing numbers’.

## $TNT$ and Creativity

The quote from Zhu Zhi that we used to start this paper in fact has a continuation: ‘If our doubt is great, our progress will be significant; if our doubt is small, our progress will be insignificant’. $TNT$ is in some sense the maximal introduction of doubt into the use of technology. In terms of progress, we believe $TNT$ to be an example of what Polya called the ‘inventor’s paradox’, which he states as ‘The more ambitious plan may have more chances of success’.

We have found it beneficial to remember $TNT$ in many situations, not only pedagogical. For, it is true of any tool or object that it can either constrain thinking or free thinking, as shown in Figure 2.

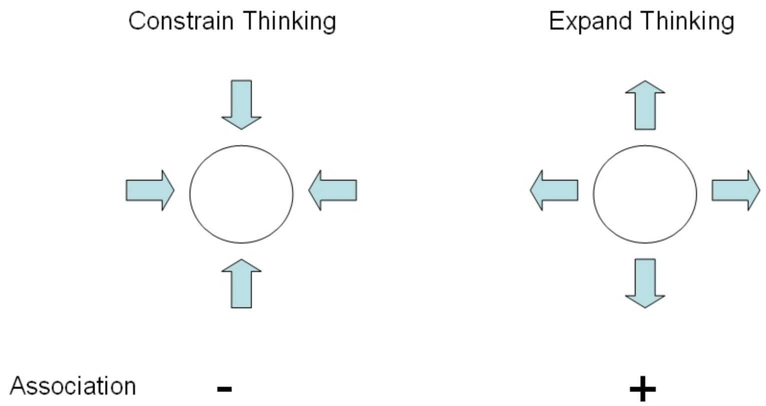


Figure 2: How a tool interacts with thinking, first impression

The obvious association of a constraint on thinking is that this is negative, and conversely, that more freedom in thinking is positive. However, as shown in Figure 3, there are secondary possibilities: constraints on thinking can be productive, and freedom to think can be negative.

We have only limited space here to expand this deep area, but we can at least give two examples. The notion of constraints being positive can be traced back at least 2,500 years, to Heraclitus’ aphorism ‘That which opposes produces a benefit.’ A more up-to-date re-statement can be found in the Henry Lime speech from the 1949 film *The Third Man*:

> ‘In Italy for thirty years under the Borgias, they had warfare, terror, murder and bloodshed, but they produced Michelangelo, Leonardo da Vinci, and the Renaissance. In Switzerland they

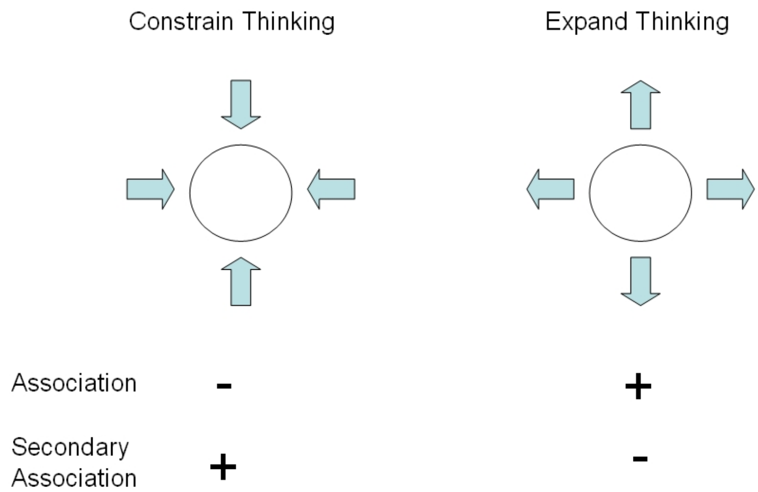


Figure 3: Secondary possibilities for a tool interacting with thinking

> had brotherly love and five hundred years' of democracy and peace, and what did that produce? The cuckoo clock.'

This quote also gives an (admittedly light-hearted) example of freedom being counter-productive. For more authoratative evidence relevant to the theme of this paper we can refer to a recent New Scientist article (Buchanan 2008), which carried the following statement from a researcher on the psychology of interaction with computational devices: 'We often see that automated thinking tools tend to block people's capacity to see or know the broader context of the problem they face.' As an example of this 'sticky' mindset, Buchanan's article describes research on the use of the Mathematica software package by physics graduates. Although Mathematica can solve equations that by hand might take days, the article warns that 'this comes at a cost'. Buchanan gives a good characterisation of the research results: 'Using Mathematica for physics involves two stages: choosing a strategy for solving the problem, and then implementing that strategy by typing in a few lines of computer code. Although the second stage can require formidable mathematical ability, it is the first that trains a student in physics. The researchers found that Mathematica encourages students to focus on the second, programming stage, at the expense of the first'. That is, the technology encourages the students to focus on the computational aspects of the problem, while suppressing the connection with the actual physics.

Ideally, for any tool or object, we would like to maximise the '+' and minimise the '-'. However, this maximisation can be difficult from within the framework of the object itself. One way to gain a new perspective is to confront the inventor's paradox and imagine the more ambitious problem of addressing the same goal with more limited (at least different) tools.

What can ideally result is a competing way of achieving similar goals, perhaps cheaper, or more interesting, or addressing different aspects, or maybe more easily incorporated into a wide variety of situations, or easier to use due to less necessity for preparation or initial practice or training.

A secondary benefit may be a new realisation of how the original problem can be addressed better without changing tools. This can happen as the result of gaining new understanding of how to maximise the positives in the original configuration, or of how the negatives can be minimised. To give just one example, after years of teaching the work management principles of David Allen's *Getting Things Done* to students using software packages, the first author tried instead to run a class using the Hipster PDA (hPDA). This tool was popularised by Merlin Mann (see `wiki.43folders.com/index.php/Hipster_PDA`), and was originally intended as a tongue-in-cheek response to the increasing complexity of handhled PDAs. Since the hPDA consists of just some index cards held together with a bulldog clip, the learning curve for the class was extremely shallow, and the target workflow concepts far easier to teach. Subseqeuently, a research student has begun incorporating the lessons learned from this class back into a piece of software that emphasises the analogy with a paper-based system.

Since technology plays an increasing part in education, we especially want to maximise the positives in its use. The primary *TNT* question of 'How could this be done without technology' encourages this by bringing a (sometimes radically) different context to a problem.

In 'The Question Concerning Technology', Martin Heidegger's thesis was that the use of technology trains humans to think like machines. He wished instead to prepare us for a 'free relationship' with technology. We believe that *TNT* can provide one escape from the trap of technology mastering man instead of man mastering technology.

## Conclusions

We have presented *TNT*, Technological Thinking with No Technology. Faced with such a title, a common reaction may have been one of doubt. If we have convinced some readers of the benefits of transferring this doubt to technology itself, our job is done.

## References

Bateson, G. (1979). Mind and Nature: A Necessary Unity. Bantam Books. Some chapters also available online at www.oikos.org/mind&nature.htm.

Ben-Ari, M. (1999). 'The Bug That Destroyed a Rocket', *Journal of Computer Science Education*, 13:2, 15–16.

Billing, D. (2007). 'Teaching for transfer of core/key skills in higher education:

Cognitive skills', *Journal of Higher Education*, 53: 483–516.

Brody, W.R. (2007). 'College Goes Global', *Foreign Affairs*, March/April.

Colwell, B. (2005). 'Engineers, Programmers, and Black Boxes', *IEEE Computer*, March, 8–11.

Cuban, L. (2001). *Oversold and Underused: Computers in the Classroom*, Harvard University Press, Cambridge:Massachusetts.

Field, M.H. (2005). 'Student-led Multimedia CD development: Does IT make a difference?' *CALICO 2005*, May17-21, Michigan, USA.

Frank, I. and Field, M. (2007). 'Learning From the Koto Wedge: The Sphere and the Shawl', *International Journal of Learning*. 14(9): 83–92.

Frank, I. and Field, M. (2006). 'Lessons from Indigenous Thought—Working with koto and robotics', *Australian Computers in Education Conference, 2006*. Available from `ianlab.ddo.jp/koto-tsukuri/LessonsfromIndigeneousThought_koto.doc`

Jungck, S. (1987). 'Computer literacy in practice: Curricula, contradictions, and context'. In G. Spindler and L. Spindler (Eds.), *Interpretive ethnography in education: At home and abroad*. Hillsdale: New Jersey, Lawrence Erlbaum.

Kramer, D. (2002). *Algorithms Should Mind Your Business*, downloaded from `www.outsourcing-russia.com/docs/?doc=680`.

Levy, M. (1997). *Computer-Assisted Language learning: Context and Conceptualization*, Oxford, England, Clarendon Press.

Mioduser, D. (1998). 'Framework for the study of the cognitive nature and architecture of technological problem solving', *Journal of Technology Education and Design*, 8(2): 167–184.

von Oech, R. (2002). *Expect the Unexpected, or You Won't Find It*, Berret Koehler Publishers, San Francisco.

Omolewa, M. (2007). 'Traditional African Modes of Education: Their relevance in the modern world', *International Review of Education*, 53: 593–612.

Perkins, D.N. and Salomon, G. (1992). *Transfer of Learning, International Encyclopaedia of Education*, (2ndEd). Pergamon Press, Oxford.

Platoni, K. (2006). 'Love at First Byte', *Stanford Alumni Magazine*, online at `www.stanfordalumni.org/news/magazine/2006/mayjun/features/knuth.html`.

Polya, G. (1945). *How To Solve It.* (Expanded Princeton Science Library Edition), Princeton University Press.

Reeves, B. and Nass, C. (1996). *The Media Equation*, CSLI Publications.

Sakamoto, T. (1992) 'Impact of informatics on school education systems: National strategies for the introduction of Informatics into schools - Nonsystematic but still systematic', *Education and Computing*, 8: 129–135.

Warschauer, M. (1996). 'Comparing face-to-face and electronic discussion in the second language classroom', *CALICO Journal* 13(2): 7–26.

Zhao, Y., Pugh, K., Sheldon, S., and Byers, J. (2002). 'Conditions for Classroom Technology Innovations', *Teachers College Record*, 104:3, April, 482—515