FROM SOCIAL ALGORITHM TO PEDAGOGICAL APPLICATION: SOME IMPLICATIONS FOR EDUCATIONAL SOFTWARE

Dee Pratt
Durban University of Technology
Durban, KwaZulu-Natal
South Africa
deep@dit.ac.za

ABSTRACT

Educational software geared to literacy development is too often based on what computer programs can do rather than on any deep-level consideration of the social process involved or how social processes are learned. As social psychology suggests that young people learn social behaviour by means of social algorithms, it is suggested that designers of educational software should consider identifying the algorithm involved as a basis for effective program design. Apart from resulting in versatile courseware, following this process means that the program structure itself can be made to reinforce the algorithm to be learned. This innovative educational design process is demonstrated by showing how a writing tutor program was designed around a composing algorithm underpinned by a deep structure of communicative functions. The resulting application could then be used flexibly in a variety of different educational contexts because the commonalities and variables in composing had been established.

KEY WORDS

Modelling, courseware, composition, design, algorithms.

1. Introduction

A problem one frequently encounters with educational software geared at developing literacy is that applications are often based on what computers can offer the learner, [1] rather than a consideration of intrinsic features of the process being learned, the nature of learning in a given case, or the role of the software in facilitating learning (i.e. what aspects of instruction can be transferred from the teacher to the application). When considering the nature of learning involved, it must be borne in mind that the learning of social processes does not necessarily involve the same type of learning as that involved in the natural sciences, and even the acquisition of social skills does not always occur in the same way.[2] Because educational goals and methods are often not considered first, we frequently find skills-based software which ignores educational research and focuses on the textual manipulations which computer programs do so well. Smith's comment about the use of technology in reading in the 1970s is as relevant now as then: "Sophisticated electronic gadgetry is often paired with naive ideas about how learning takes place and about the subject being taught". [3] Fairly recently our English Department was offered a demonstration of skills-based reading software involving what Smith calls "automated reflexes to scores on a multitude of arbitrary tests" as well as endless manipulation of discrete letters and words, as if reading were not a complex cognitive process involving deep structure learning. [4] More recently I was invited to view a presentation on "writing" software which on closer inspection comprised mainly skills-based *language* lessons combined with human tutoring via email. The package was to be purchased by the institution, but the cost of individual tutoring was to be borne by the students. The tutoring costs were well beyond the means of our students, 70% of whom are ESL learners from educationally disadvantaged backgrounds. Hiring local tutors who were au fait with the educational context and specific student needs would in fact have been a more effective and affordable option, and would have been covered by the cost of the software alone. The problem which is explored in this paper, then, is precisely *how* the courseware designer establishes intrinsic features of the process being learned, the nature of learning involved, and the role of the software in facilitating learning. The designer also needs to take cognisance of the context in which the courseware is to be used.

These issues will be explored in connection with doctoral research on academic writing. This involved modelling composing processes, drawing on data obtained from video protocol analyses [5][6][7] over the last fifteen years, and culminating in a theoretical model of communicative functions and an empirical model of composing. [8] The two models comprise the social mechanism involved in communication in written mode, the formal aspect of the mechanism being the theoretical model of functions, and the applied aspect the empirical model. The modelling process, based on Franck's seminal work on modelling, [9] was carried out to provide a rationale for a pedagogical model of composing developed in masters research. The pedagogical model was used as the basis for educational software in the form of a writing tutor program, and a working prototype was produced. The pedagogical model is based around a social algorithm [10] of the type described by Blunt Bugental (see [2]). How the algorithm was formulated, how it was represented as a pedagogical model, and how this was translated into courseware design will be examined more closely in the rest of this account. It is hoped that this paper will make some contribution towards effective courseware design, particularly where the learning of complex social processes is involved.

2. The Innovative Design Process

It must be emphasised that the modelling approach used here is a very different from the more usual cycle of developing composition software, which tends to start with the translation of teaching approaches into computer programs, followed by a period of testing out and modifying the resulting software, as exemplified in Rowley & Meyer's *Computer Tutor for Writers*. [11] The current-traditional rhetoric approach has resulted in programs based on text-analysis, such as *Easy Writer* [12], with genre-based approaches forming the basis for more socially-aware text analysis programs, such as *Writing Safari*. [13] The conferencing method (i.e. teacher response throughout composing) is the basis for Costanzo's *Story Tutor* [14], which is, however, limited to one genre, the short story, while heuristic or invention approaches have led to programs such as *Writer's Helper* [15]. There are also organiser-type programs such as *Writing tutor* [16] and *The Literary Machine*, [17] which are based more on the organising capacity of computer programs (e.g. as in spell checkers or databases) than on approaches to teaching composition.

Process-based writing tutors come the closest to the software developed in this account, with *Maestro Writing Process Tutor* [18], *Writing Process Workshop* [19] and *Writing Tutor* [20] superficially resembling the writing tutor programme (WTP) prototype described below. However, the resemblance is limited to the fact that all of these are premised on putative stages of composing (some derived from existing theories e.g. cognitive theory, others being ad hoc constructs). The significant differences are, firstly, that the other programs involve teaching programmes based on composing processes rather than providing help based around learner composing, and secondly, that none of these distinguish between the extra- and intra-systemic aspects of composing. The latter omission means that social aspects of composing, in particular, differing local academic requirements, cannot be catered for by the software. Process-based approaches have on the whole not proved successful in the teaching of academic writing for precisely this reason. While the omission can be rectified by situating the software in the social context of a composition teaching programme, this tends to become part of the software, making the program unnecessarily cumbersome and limiting its versatility. The WTP describes composing and offers help (as accessed through a help menu) rather than being an instructional programme per se: this means that it can be adapted to suit a wide variety of formal and informal learning contexts. Distinguishing between the extra and intra-systemic aspects of composing has made it possible to program an input option allowing the software to be customised to suit different contexts, genres and learners, thus making it suitable for facilitating expertise in very different academic contexts. For these reasons, the WTP is a much more versatile option than any of the above. This versatility would not have been possible without carrying out the rigorous modelling process to arrive at the essence of the social process *before* programming commenced. The resulting prototype software, will, of course, be tested out in a variety of contexts, and modified if and as necessary.

3. The Composing Algorithm

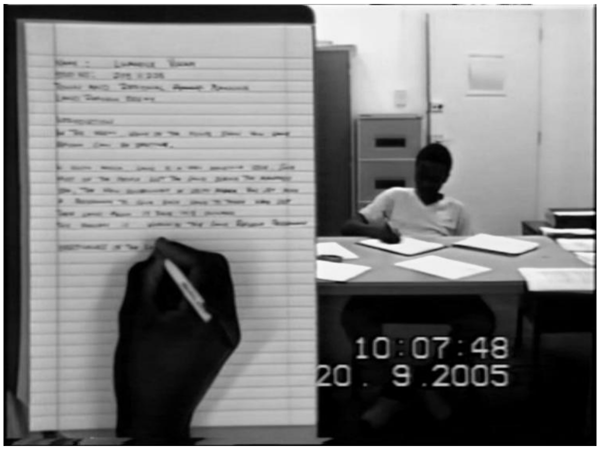


Figure 1 Single Frame from a Video Playback

The "essence" of the social process of writing, a composing algorithm, was established as follows. Over fifty video protocols (see Figure 1) confirmed accounts of composing in the process-based literature, which had suggested that composing is a complex series of recursive stages. [21] Graph profiles of writers could be drawn up to show how they focused on different stages of composing at different times, shown in Figure 2. The graph profiles showed the writer's focus on various stages per minute while composing, as reconstructed by the writer and researcher from the video playback and written texts. Yet even process-based researchers claimed that composing was too complex and idiosyncratic to categorise. [22][23] Moreover, postmodern approaches to composition, which dominated the field subsequent to process-based research, emphasised the social aspect of writing, focusing almost entirely on social aspects of *text*, however. [24] Using Franck's modelling process within a critical realist approach offered solutions to both of these problems, explaining both the patterning underpinning the surface complexity of specific instances of composing as well as explaining the surface variations.

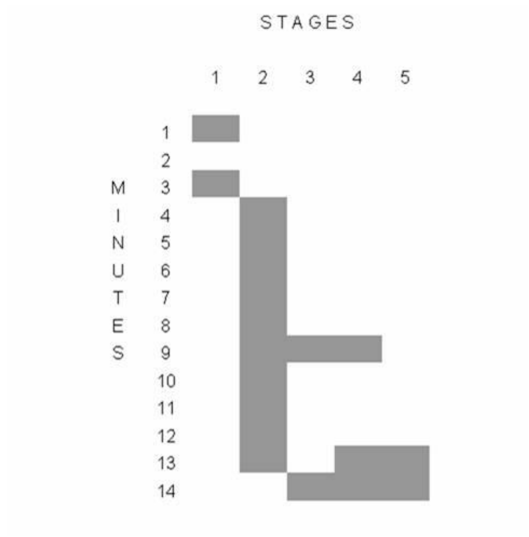


Figure 2 Graph of Composing Profile

The theoretical model underpinning composing comprises a system of five functions constituting felicity conditions for communication, namely the contextual, ideational, interactive, social and reflexive functions. [8]

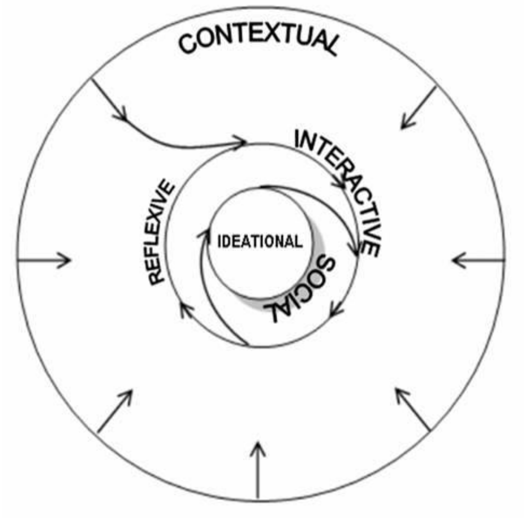


Figure 3 Model of Communicative Functions

As shown in Figure 3, for communication to take place, it must be contextualised, which then drives the interaction, producing ideational content: this in turn must be made socially appropriate. The reflexive function regulates the interaction in the nature of a feedback loop. The composing system shown in the empirical model (Figure 4) can then be viewed as the social mechanism carrying out communication in written mode, where the material form of production [25] combined with temporal and spatial distancing results in the complex series of recursive stages observed by process-based researchers. Internal variation can be seen to result from the fact that the system is infinitely open-ended and flexible. In addition, the output of each stage affects the performance of subsequent stages. However, it comprises a set series of functions which need to be performed to achieve a goal, which means that it constitutes an algorithm. External variation is the result of input into the composing system, which can be seen to affect each stage of composing. It is the combination of internal and external variation which makes composing so difficult to categorise, and led even process-based researchers to declare that composing was too complex and idiosyncratic to describe. Moreover, process-based research was criticised by adherents of subsequent approaches as being naïve, passé and even reprehensible because of its omission to deal with social issues, notably, unequal relations of power. As communication is a social process, any model of composing must account for the operation of social aspects. The model of composing in Figure 4 shows the intra-systemic operation of social aspects as the "Editing" process, and the extra-systemic operation of social aspects (i.e. contingent factors) as "input" into the whole system which then permeates all stages of composing.

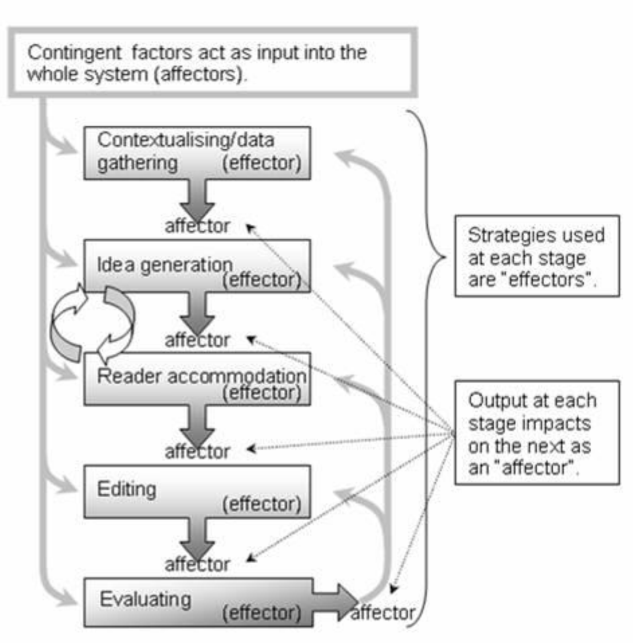
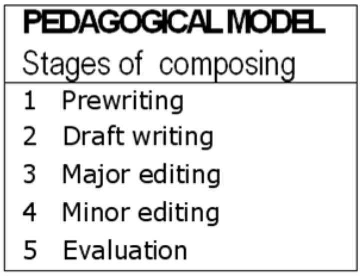

Contingent factors act as input into the whole system (affectors).
Contextualising/data gathering (effector)
affector
Idea generation (effector)
affector
Reader accommodation (effector)
affector
Editing (effector)
affector
Evaluating (effector)
affector
Strategies used at each stage are "effectors".
Output at each stage impacts on the next as an "affector".

Figure 4 The Composing System in the Empirical Model

4. Formulating a Pedagogical Model

The pedagogical model which formed the basis for the design of the writing tutor program is based on the assumption that young people learn to engage in social life by acquiring social algorithms or typical patterns. According to Blunt Bugental, three levels are involved. Young humans (1) are *biologically* designed to solve recurring problems flexibly, (2) come to represent their experiences *cognitively* in organised ways, and (3) ultimately learn to *reflect on* and manage their environment (see [1] p 188). These observations explain why providing learners with social algorithms (in accessible form, that is) can constitute a powerful mechanism for learning (what I have termed a “conceptual mechanism”). Firstly, if learners are biologically pre-disposed to solve problems flexibly, exposure to an algorithm provides a possible solution for which there is already a *innate predisposition.* Next, if there is a tendency in young learners to organise experiences cognitively, a social algorithm not only models social processes cognitively, but also models the acquisition of such processes, (i.e. models learning how to learn). Finally, consciousness of a social algorithm facilitates reflection on and control of not only the environment but the social process itself. In this sense, providing learners with social algorithms not only empowers them in carrying out social processes but offers them more scope for transforming social structures because *they are more aware of the inherent patterns on which these are constructed.*



PEDAGOGICAL MODEL

Stages of composing

1 Prewriting
2 Draft writing
3 Major editing
4 Minor editing
5 Evaluation

Figure 5 The “Core” of the Pedagogical Model

One could, of course, base instruction around the algorithm, keeping control with the teacher and not the learner, as is the general tendency with very young learners. The pedagogical model (Figure 5) was initially formulated so that I could respond appropriately (i.e. according to learner needs) to student drafts in conferencing sessions. In my first session with a student (first-year B. Comm.), I explained the model to him, and he then told me what he needed to do next: thereafter I gave the model to the students and let them decide what kind of feedback (or further action) was required. The model has been used as the basis for composition courses, and for intensive one-on-one coaching where psychological stress and time pressure required swift intervention rather than spending time explaining the model. Thus, while its main use is in fact andragogical (i.e. for independent adult-type learning), it can be used within structured or scaffolded learning programmes with younger learners, with the true meaning of “pedagogical” (i.e. child-centred learning), although is it commonly used (as here) for any application used in teaching.

The “core” of the pedagogical model is given in Figure 5. In the version given to learners, each stage was shown to have a underlying function, and was accompanied by practical advice for each stage, as follows:

Prewriting:
Consider purpose and reader, gather data, let it mull round. *(Underlying focus* - **DATA GATHERING)**

Draft writing:
Suggest structures or outlines, jot down ideas or fragments, write larger pieces. *(Underlying focus* - **IDEA GENERATION)**

Major editing:
Reread and structure for reader, order, add, delete (go back to 2 if necessary). *(Underlying focus* - **IDEA ORGANISATION/STRUCTURING)**

Minor editing and polishing:
Check for correctness, check format and minor editing conventions. *(Underlying focus* - **EDITING)**

Evaluation:
Assessment (by writer and others) in terms of purpose. *(underlying focus* **– EVALUATING)**

The term “recursive” appeared on the right hand side of the model. Younger learners were given a simplified version of the pedagogical model and told to repeat the stages as many time as needed until they were satisfied with the result.

The pedagogical model is underpinned by the system of communicative functions: Prewriting is the stage where composing is contextualised, Draft writing, where ideational content is generated, Major editing, where interaction with the intended reader mainly takes place, Minor editing where the message is made socially correct, and Evaluation, where feedback on performance is given. While there were complementary models of inner dialogues, and an account of “writer’s block”, both related to the stages, these were usually explained orally to the learner (when needed) with reference to the pedagogical model, otherwise the diagram would have become too cumbersome (a handout of 4-5 five pages was sometimes used in class situations.)

5. From Pedagogical Model to Courseware

3.1 Using Algorithms as a "Meta-approach" to Design

At this stage I would like to sum up the approach to courseware design adopted here and suggest how it might be seen as innovative. Courseware designers on the whole appear to consider how a subject is *taught* (or learned, to be fair), usually in a formal context (e.g. school). In this approach the focus is rather on how the social process of composing *takes place*. This was done by formulating a model of the phenomenon of communication in written mode. Once one has discovered the algorithm underpinning a process, the instructional programme can take as many forms as there are different teaching approaches. However, bearing in mind that communication of the social algorithm itself can be an effective intervention for learners, as well as giving the teacher a metacognitive view informing instruction, really versatile courseware could be developed by basing it around the *algorithm* rather than by basing it on any given *teaching approach*. Granted that this is itself a teaching approach, but it is one which lends itself to infinite permutations and variations: this is because it is a "meta-approach" within which specific idiosyncratic learning/teaching strategies can be used as preferred. It is an approach which starts outside the classroom by looking at the nature of social phenomena and how they are learned in real life. How they are dealt with in actual learning programmes (including informal learning) can then be handled within the range of disciplinary requirements, learner needs, teacher preferences and institutional constraints.

3.2 The Scope and Purpose of the Writing Tutor Program

The term "tutor" is often used very widely in computer mediated learning, sometimes to refer to text lessons with no interactive element. The types of writing tutor (or tutor-type) applications currently available include (1) conferencing-type tutors, (2) tutors based on text-analysis, (3) tutors based on heuristics or invention strategies, (4) text or revision tools, (5) organisers and (6) process-based tutors. There are also special "training" type word processors (usually designed for younger writers) which have not been included in the above list.

The prototype writing tutor program (WTP) described here fits into the category of process-based tutor, and was intended for use as a flexible learning tool within as many different educational contexts as possible (including informal contexts). While geared to learners from Grade 8 to first year tertiary level, it could be used by younger learners and adult writers of all ages, provided that they possess basic computer literacy. It is in fact a type of reusable learning object, [26] and was designed primarily to take my place in teaching, tutoring or coaching students so that the basics could be covered without my having to repeat them time and time again to different groups of learners. The WTP, then, was an attempt to replicate myself as teacher/tutor, so that I could have "quality time" following up specific composing issues (both problems and successes) with learners and not keep repeating the basics. The WTP was therefore designed to accommodate those of my own activities which a computer program could replicate, namely:

- to model composing processes for users;
- to model reader-responses, so that learners could internalise the inner dialogues which experienced writers have been observed to use;
- to allow input from the user which would reflect the social context (including academic requirements);
- to give generalised advice and guidance throughout composing;
- to give specific feedback on user-identified issues and problems;
- to be such as requiring to be consulted only when/where necessary;
- to be open-ended in allowing users to focus on their specific learning/remediation needs;
- to have interactive elements, such as input from the user in various sections, or feedback on performance;
- to remain unobtrusively on the screen to assist with composing as/when needed;
- to present to the user as a friend and helper;
- to prompt what have been observed by teachers and researchers to be effective composing practices; and
- to initiate a reflective dialogue on work-in-progress which could eventually be internalised by the user.

3.3 The Design of the Writing Tutor Program

The program design fits the general principle that young learners are guided in social behaviour by use of social algorithms, as the program is based around the composing algorithm (see Figure 4). It fits the scope and purpose given above. These design features are related, as it would not have been possible to design a program as flexible as the WTP without the initial formulation of the algorithm. The program took the form of a floating help menu to be consulted as/when needed, particularly while learners were composing (see Figure 6). The format of a help menu lends itself to use as/when needed, and in particular, suggests recursive activity without this having to be stated overtly. The core of the program is the composing algorithm, expressed in terms which are easily accessible to learners and self-explanatory: "preparing to write, writing rough drafts, revising for your reader, editing and proofreading, and evaluating your writing".

The main menu can be contracted to the composing algorithm by mouse-clicking on HELP WITH COMPOSING. The program itself is self-explanatory ("Program overview"), and does not require a manual for use. Various submenus lead off the main menu, giving general advice and tips for each stages of composing, as well as various interactive routines or checklists. The “Help with writer’s block” and “Inner dialogues” sections off the main menu can be treated as separate lessons/routines or accessed from submenus linked to stages of composing. “About composing” explains how the program is built around the stages of composing, and “Composing on computer” is a similar (text) lesson. “Assess your composing” is a self-test directly related to the characteristics of experienced writers as summarised in the composing algorithm. The “Readings database” and “Working notes” are included as practical reinforcements of the habits of experienced writers (as mirrored in the algorithm), and not just because computer programs “can do” databases and word processor documents. Expressing the pedagogical model as a help menu with subsections instead of a static diagram has meant that much more supporting material could be included in ways which are directly linked to the composing algorithm, re-emphasising the basic pattern at all levels.

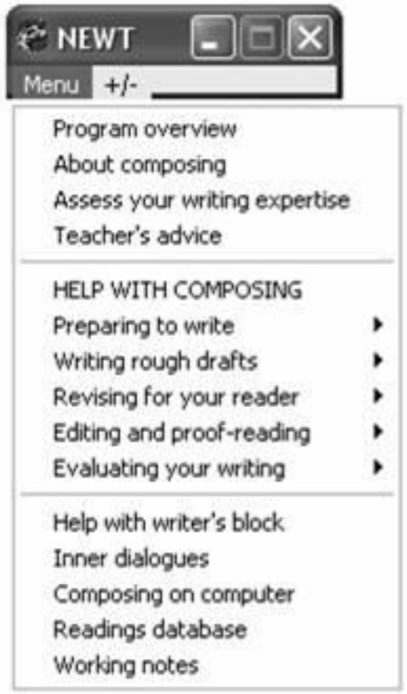


Figure 6 The Main Menu of the Writing Tutor Program

However, the most important feature of program design, the inclusion of social input which impacts on the performance of all stages, is contained in the “Teacher’s advice” menu item, named thus because the teacher usually mediates social requirements for learners. This item prompts learners to enquire about local social conditions (mostly, but not always, academic requirements) and to input these in the relevant sections. The “advice” can then be displayed on the screen at the various stages of composing, reinforcing the idea that social input impacts on the performance of each stage of composing. Learners are thus not only reinforcing the conceptual imprint of the algorithm each time they click back and forth on various composing stages in the HELP WITH COMPOSING section: they are also re-enacting (at a simpler level) the social dynamic outlined in Figure 4, where “input into the system” impacts on all stages of the composing algorithm. Without prior identification of the algorithm, social requirements could not be conceptualised and dealt with in an orderly, systematic manner by learners. Teacher requirements would then come across as a mass of ad hoc precepts advice without priority or rationale, and the role of social requirements in further defining each stage of composing for the learner would not be recognised. Academic requirements are all too often expressed (sometimes on specially printed assignment folders) as an undifferentiated list of orthographic conventions. If learners have the option of inputting these in the “Editing and proof-reading” section, they can at least avoid the form of writer’s block caused by over-focusing on orthographic correctness in the earlier stages of composing.

6. Conclusion

What I have attempted to do in this paper is to demonstrate how educational software based around social algorithms has distinct advantages over programs based on what computer programs can do or even on specific learning approaches. The latter are inevitably based on beliefs and values about teaching and learning rather than the “essence” of the actual social process involved, and inevitably exclude important aspects of the process or deal with them in ad hoc ways. Identification of the given algorithm makes it easier to base the software on the social process, rather than ad hoc teacher advice. Moreover, computer programming itself is based on algorithms, and programmers will find it easier to translate educational software into program form when the central algorithm is clearly identified by the designer. If the structure of the program echoes the structure of the algorithm to be acquired by the learner, it is far more likely to reinforce learning than arbitrary animations and add-on features, which have actually been found to divert learners’ attention from learning materials rather than reinforce learning.[27] The most important gain, however, is that, if a modelling process is used which clearly identifies the commonalities and variables in a given social process, extremely versatile courseware can be designed which is not limited to specific socio-cultural settings. This is because local input can then be factored into the program design, making customising a user function, and sensitising learners to the fact that knowledge is a social construct. The efficacy of such an approach still remains to be tested out with learner writers in different contexts: this is scheduled to take place at a multicultural university of technology, starting in the second semester of 2006. While the university is situated in South Africa, the writing tutor program will be made available internationally as an Internet download, which will make it possible to gauge the effectiveness of its use in a variety of very different contexts worldwide.

References

[1] B.T. Hughes, Balancing enthusiasm with skepticism: training writing teachers in computer-aided instruction. *Computers and Composition 7*(1), 1989, 65-78.

[2] D. Blunt Bugental, Acquisition of the algorithms of social life: a domain-based approach. *Psychological Bulletin 126* (2), 2000, 187 -219.

[3] F. Smith, *Psycholinguistics and reading* (New York, Holt, Rinehart and Winston, 1973).

[4] F. Smith, *Understanding reading: a psycholinguistic analysis of learning to read, 2nd edn.* (New York, Holt, Rinehart and Winston, 1971).

[5] C.S. Jones, Composing in a second language: a process study, *Proc. 16th Annual TESOL Convention*, Honolulu, May 1982.

[6] A. Matsuhashi, Explorations in the real-time production of written discourse, in N. Nystrand (ed.) *What writers know: the language, process, and structure of written discourse* (New York: Academic Press, 1982).

[7] S. Pianko, A description of the composing processes of college freshman writers, *Research in the Teaching of English 13*(1), 1979, 5-22.

[8] D.D. Pratt, Modelling writing as the basis for a writing tutor computer program, *Proc. Ed-Media World Conference on Educational Multimedia, Hypermedia & Telecommunication*, Montreal, Canada, 27 June-2 July 2005, 786-791.

[9] R. Franck, *The explanatory power of models: bridging the gap between empirical and theoretical research in the social sciences* (Norwell, MA: Kluwer Academic Publishers, 2002).

[10] D.D. Pratt, Social mechanism and software design: the use of a stochastic social-process algorithm in the design of a writing tutor program, *Proc. 3rd International Conference on Computer Science and its Applications (ICCSA-2005),* San Diego, California, USA, June 27-30 2005, 249-254.

[11] K. Rowley & N. Meyer, The effects of a computer tutor for writers on student writing achievement. *Journal of Educational Computing Research 29*(2), 2003,169-187.

[12] Software for Students, *Easy Writer*, n.d. [online], available: http://www.softwareforstudents.com (Accessed 20 May 2006).

[13] K. Farrow, D. Power & P. Freebody, Computer-assisted writing development for deaf students: "Writing Safari", *Proc. Australian and New Zealand Conference for Educators of Deaf Students, Quality and Choice*, Sydney, Australia, 1994 [online], available: http://www.cltr.uq.edu.au/oncall/farrow91.html (Accessed 31 August 2004

[14] W. Costanzo, The English teacher as programmer, *Computers and Composition 4*(3), 1987, 65-76.

[15] E. McDaniel, E. Bibliography of text-analysis and writing-instruction software, *JAC 7*, 1987 [online], available: http://jac.gsu.edu/jac/7/Articles/15.htm (Accessed 2 March 2003).

[16] SelectSoft Publishing, *Writing Tutor*, 2006 [online], available: http://support.selectsoft.com/manuals/writing_tutor.htm (Accessed 30 June 2006).

[17] Literary Machine Group, *The Literary Machine*, n.d. [online], available: http://www.sommestad.com/LM _1_1 .htm (Accessed 1 October 2004).

[18] K. Steuck, FOCUS: Intelligent tutoring systems, *Distance Education Systemwide Interactive Newsletter*, 2004 [online], available: http://www.uwex.edu/disted /desien/2000/0003/focus.htm (Accessed 21 October 2004)

[19] Siboney Learning Group, *Writing Process Workshop*, 2002 [online], available: http://www.ea-software.com/ language5.shtml (Accessed 2 October 2004).

[20] TutorTek.com (Command Technologies) *Writing Tutor*, 2002 [online], available: http://www.tutortek.com /docs/Time1p.PDF (Accessed 1 June 2006).

[21] M. Hairston, The winds of change: Thomas Kuhn and the revolution in the teaching of writing, *College Composition and Communication 33*(1), 1982, 76-88.

[22] A. Raimes, What unskilled ESL students do as they write: a classroom study of composing, *TESOL Quarterly 19*(2),1985, 229-256.

[23] R. Spack, Invention strategies and the ESL college composition student, *TESOL Quarterly 18*(4), 1984, 649-670.

[24] R.M. Coe, Teaching writing: the process approach, humanism, and the context of "crisis", in S. de Castell, A. Luke & K. Egan (eds.) *Literacy, society and schooling* (Cambridge: Cambridge University Press, 1986).

[25] T. Nellhaus, Subtexts: or, the materiality of discourse - an alternative view, Paper delivered at *Rethinking Marxism Conference*, Amherst MA, 7 December 1996.

[26] S. Downes, Learning objects: resources for distance education worldwide, *International Review of Research in Open and Distance Learning*, July, 2001.

[27] Daily News, Computers can harm reading, *Independent Newspapers*, Greyville, Durban, 10 January 2006.