Do students apply constructivist principles in designing computer-supported learning environments?

Philippe Dessus, Erica de Vries

PHILIPPE DESSUS & ERICA DE VRIES

DO STUDENTS APPLY CONSTRUCTIVIST PRINCIPLES IN DESIGNING COMPUTER-SUPPORTED LEARNING ENVIRONMENTS?

Abstract. In this paper, we examine three interrelated issues: designing for teaching instructional design, the feasibility of constructivist ID methods and the multifaceted evaluation of computer-supported learning environments. After a brief review of the available literature, we present our own course which mixes traditional ID methods with the Crossley and Green's practical guide for developing computer lessonware that can be considered a constructivist ID method before the term existed. We critically examined portfolios and computer programs produced by university students enrolled in this course focussing on three main constructivist principles: the authenticity of learner experiences, the construction and manipulation of external representations, and communication and participation. In particular, we aimed at identifying and interpreting major mismatches concerning these principles in student productions. In the discussion, we put forward some preliminary explanations for students' difficulties in applying constructivist principles in the design of computer-supported learning environments.

1. INTRODUCTION

Paradoxically, instructional design (ID) methods have hardly ever been applied to themselves in order to provide the foundations for *teaching* the design of instruction. For example, no hierarchical task analysis (HTA) has been performed on the task of doing one as a preliminary step in designing a course for teaching HTA (Patrick, Gregov, & Halliday, 2000). Nor do teachers of top edge ID methods appear to actually use them while planning their own teaching (Young, Reiser, & Dick, 1998, but still Shambaugh & Magliaro, 2001). In sum, whereas instructional designers should thoroughly analyse domain knowledge and tasks in constructing teaching sequences, little is known about what instructional designers themselves actually do (Kirschner, Carr, van Merriënboer, & Sloep, 2002; Pieters & Bergman, 1995; Rowland, 1992, 1993). Are these all simply cases of shoemaker's children going barefoot? By way of defence, the situation seems to be complicated and involves several levels of prescription: how to design a course for teaching instructional design? And how do learning theories relate to the design of learning environments? Moreover, the design of computer-supported learning environments adds another level: designers' and teachers' viewpoints on the role of computers in education.

Trying not to fall into the trap of an infinite regression, i.e. looking for constructivist ways to teach constructivist ID models for developing constructivist learning environments, the current paper addresses the question of how to teach the design of computer-supported learning environments. Potential answers have to take into account the large variety of both learning environments and methods to design them. In the following, we first oppose traditional versus constructivist computer-supported learning and review their close relations with ID. We then present our own course based on Crossley and Green's (1985/1990) practical guide which can be considered a constructivist ID method before the term existed. Finally, we critically examine our course through the student portfolios and computer programs produced in the last two years.

2. ID METHODS FOR COMPUTER-SUPPORTED LEARNING

Computer-supported learning environments nowadays range from classical tutorial and drill and practice programs, to modern state of the art environments for active exploration, representation and communication. Likewise, a variety of ID methods supporting their implementation co-exist.

2.1. Traditional versus Constructivist ID Methods

Traditional educational computer programs rely heavily on the cognitivist paradigm: information is delivered through media for processing and storage by students (Alessi & Trollip, 1991). Domain knowledge itself is thought to exist independently of a "knower" and domain experts need to be probed so as to uncover domain concepts, rules and principles. The expert knowledge then needs to be adapted for educational purposes. Thus, traditional ID focuses on methods for analyzing knowledge and skills in order to find optimal ways for presenting information and exercising rule application.

Most of modern learning environments adhere to constructivist and situationist perspectives on learning largely defined by three principles. First, they should provide opportunities for authentic tasks and learning experiences in context (e.g., CTGV, 1996). Second, they should afford the manipulation and construction of external representations (e.g., Ainsworth, 1999; Reimann, 1999). Third, they should facilitate communication and discussion for knowledge building (e.g., Scardamalia, Bereiter, & Lamon, 1994). More generally, knowledge and learning are considered to be tied to real life context through individual as well as collaborative construction of knowledge. Constructivist and situationist notions constitute the foundations of numerous computer programs such as hypertext and hypermedia, simulations, micro-worlds and computer-supported collaborative learning environments. In contrast to the highly structured traditional ID methods, a number of general principles guide the design of this large diversity of environments, such as mentioned in the literature (Jonassen, 1994; Kirschner, Carr, van Merriënboer, & Sloep, 2002; Lave & Wenger, 1991):

— Apply a holistic systemic design model that considers instructional factors such as learner, task, and setting, in increasing detail throughout the design process.
— Do not expect to capture content in a task analysis. Design rich learning experiences where learners can pick up information on their own. Focus on knowledge construction, not reproduction.
— Consider multiple perspectives and representations, avoid oversimplification, and encourage the learner to exercise responsibility.
— Choose instructional strategies with authentic problems in collaborative, meaningful learning environments close to the performance context.

— Engage learners into meaningful activities in which they have the opportunity to expose, explain, and discuss or critic the main arguments of a given problem.

In fact, constructivist ID methods are considered an oxymoron (Petraglia, 1998), since learners themselves are supposed to be the designers of their own learning experiences. It therefore might seem incoherent to deliberately plan and structure a constructivist learning environment. But, referring to Simon’s (1996) distinction between the sciences of the natural (i.e., natural phenomena or objects) and the sciences of the artificial (i.e., human-made or designed objects), human-arranged learning environments actually can be considered non-natural or prescriptive artefacts whatever their designers’ stance on learning.

2.2. ID Method, Learning Conditions and Outcomes

A widespread explicitly held assumption in the field is that different learning outcomes require different learning conditions (Jonassen, 1997). Another assumption, although more implicit, is that creating different learning conditions requires different ID methods. Jonassen (1997), for example, adopts the classical distinction between well-structured and ill-structured problems in order to elaborate appropriate ID models. Not surprisingly, the proposed model for well-structured problems relies on information processing theories of learning, whereas the one for ill-structured problems is based on constructivist and situationist approaches. The latter model instantiates some of the general principles mentioned above, e.g., creating situations in which learners need to deal with multiple problem definitions, multiple constraints, and alternative opinions, positions and perspectives. Given the fact that instructional design itself can be described as ill-structured problem solving, the model for designing instruction for teaching instructional design clearly should be a constructivist one. However, it is not clear what a constructivist course on creating constructivist learning environments really should look like, let alone how to establish whether students enrolled in such a course would produce traditional or genuine constructivist environments.

In this paper, we use examples from our own teaching experience for exemplifying some of the issues at stake. In fact, our own course is not an exception to the rule: i.e., no ID method was applied in its design. It contains elements of constructivist and traditional ID methods, the choice of which merely being a matter of availability. We will describe the course and analyse student productions according to the three principles mentioned above (authenticity, representation, and communication). In doing so, we provide food for thought on the following questions: How to evaluate student portfolios and computer programs? How to introduce constructivist principles when teaching the design of computer-supported learning environments? Is their expression sufficient to promote their application by students? Should students themselves be immerged in a constructivist learning environment for learning to design one? Are there any side effects emerging from our course rationale, i.e., to what extent can deviations from design principles in the students productions be explained by characteristics of the course?

3. A MIXED ID COURSE FOR COMPUTER-SUPPORTED LEARNING

This section describes the setting, design methods, and the roles of students and teachers in the instructional design course taught at the University of Grenoble for more than 10 years. The course is part of the third year curriculum in Educational Science (bachelor level). The first semester contains a compulsory course on computer-supported learning environments presenting a typology of educational computer programs (de Vries, 2003) as well as an introduction to the authoring environment Toolbook (Asymetrix). The second semester then proposes an optional course on the design of computer-supported learning environments. The course is given by two teachers in parallel (two weekly sessions of two hours each during twelve weeks). One teacher (the first author) introduces some theoretical notions on instructional design as well as some case-based exercises. The other teacher (the second author) introduces the Crossley and Green's (1985) practical guide for designing computer lessonware (henceforth C&G) in a few lectures, followed by a number of hands-on sessions in which students design and develop a computer program using Toolbook. The instructions tell students to design and implement an educational computer program and write out a portfolio describing the main steps in the design process, as well as some prescriptions of use for teachers.

3.1. The Design Methods

Once students have chosen the subject-matter content (domain), a number of different design methods are used in order to provide tools that might be appropriate for different students. A number of two hour sessions introduce more traditional ID methods:

— *Writing instructional objectives*: Students are asked to specify clear objectives (following Mager's, 1962 rationale) about the content to be learned by the future users of their program.
— *Task analysis*: Students are asked to perform a hierarchical task analysis for a better understanding of the relationships between learner and subject matter.
— *Instructional design*: Students are given the task to plan an instructional sequence using either the classic tylerian's specification of educational objectives, or the Heller and Martin's (1995), in which content is successively reformulated in different formats (text; text and audio; text, audio and image).

The more constructivist C&G method aims at building an open learning environment. The openness is introduced by way of the market metaphor: the designer's job is to propose a market place providing intellectual foods and allowing different routes. The learner constructs his or her own learner experience depending on the choices made. The method heavily relies on the experience of teachers as designers of appropriate learning environments. The rationale (see Table 1) is to design learning environments in which learners can experience things that are difficult to accomplish with traditional means (activities that are costly, dangerous, time-consuming, repetitive, or that suppose large individual differences). A number of recommendations are in fact reformulations of the three constructivist principles mentioned earlier. First, the environment has to bring about authentic activities pertaining to every day life and constituting an occasion for placing the learner in a context different from a classical one that requires reading and answering questions. Second, the learner has to exercise responsibility and tasks involving manipulation or construction of multiple representations (rather than just interpreting them) should be privileged. Third, the computer program has to be seen as part of the whole learner environment: the learner might be encouraged to consult other sources, i.e., the teacher, libraries, domain experts. Whereas ID methods generally specify the roles of learners and media, the C&G method considers the role of the teacher as well. In designing the setting, students need to consider the issue of participation by other actors (learners, teacher) before, during and after using the computer program.

Table 1. The Crossley and Green (1985/1990) method (after the French translation)

1. Choose a subject	5. Define typical states of the key display
2. Establish global design: learner experience, viewpoint (place, role, time), and table of responsibilities (learner, computer, teacher)	6. Design table of user commands 7. Design table of conditions (for accepting commands) 8. Design table of computer reactions
3. Design the market place (circulation diagram)	9. Design secondary screens 10. Getting additional information
4. Design the key display where the learner spends most of the time	11. Rules and logic 12. Write teachers' guide

3.2. The Students as Learners-Designers

The level of proficiency of our students in using Toolbook is low but suffices for the purposes of the course. Most of the students are preparing for an examination for entering the teacher training institute. A major drawback of this situation is the students' own learning context: they are asked to design an adequate context for a future learner, yet they themselves do not have sufficient teaching experience for doing so, e.g., for matching content and learner, media and instructional method, and so on. Moreover, they do not have sufficient background knowledge of institutional viewpoints, e.g., on the desirability of constructivist-oriented teaching.

3.3. Defining our Role as the Teachers of the Course

As teachers, we provide students with ID methods, the authoring environment for computer programs, and advice. We refer to the C&G method as a planning and regulation aid to guide student activities. Furthermore, our guidance is mainly based on teaching experience, e.g., dissuade students to look for visual material (or they will waste time searching for photos on the internet) and on the C&G approach, e.g., focus on other than traditional (reading, exercising) learner activities. Finally, we assess the students' portfolios and computer programs at the end of the course. Note that at the time of teaching, we did not plan to inspect student productions in the light of constructivist principles.

4. EVALUATING STUDENT PRODUCTIONS

This section describes the analysis of student productions.

4.1. The Analysis Grid and Procedure

The three principles were operationalised as shown in Table 2 into categories that more or less followed the prescriptions. Each student production, i.e. portfolio and computer program, could be attributed to one of the categories for each principle (categories fulfilling the C&G prescriptions are in bold face). Both authors first independently examined and categorised the portfolios and computer programs. Cases of disagreement between raters were subsequently discussed and resolved.

Table 2. Analysis grid for constructivist computer-supported learning environments

Category	*Description*
1. Authenticity	
Tutorial	Traditional school context, content and/or exercises
Game	Artificial context, tasks and rewards soliciting motivation
Authentic	Real life context, authentic task highly related to content
2. Representations	
Interpret	Exposes content through multimedia sequences
Move symbols	Move text, images, numbers in predefined structures
Manipulate	Situation or variable manipulation modifying the setting
Construct	Construction of structures of representations
3. Participation and communication	
No interaction	No interaction of any form is planned in order to build knowledge about domain content
Interaction between teacher and learner	Planned interactions between learners and teacher for encouraging discussion and knowledge building
Interaction between learners	Planned interactions between learners for encouraging discussion and knowledge building

4.2. Overview of the Results

Thirty-seven student portfolios and computer programs produced in the last two years (22 in 2002 and 15 in 2003) were inspected using the threefold framework. 38 percent of the environments (14 out of 37) follow the C&G prescriptions of principles 1 and 2 to some extent (Table 3). At first glance, authentic environments seldom are only interpretation-based and tutorials rarely provide for construction-type activities, but the results show some notable exceptions. For example, five tutorials allow manipulation and one authentic environment allows neither manipulation nor construction. These cases provide an indication for considering the two principles as separate features of constructivist learning environments.

Principle 3 is considered to be relatively independent from principle 1 and 2. More than half of the student productions (23) do not plan any interaction between learners or between teacher and learners. Moreover, only 11 productions mention teacher-learner interactions, but these mostly pertain to a "please come and fix" procedure in which the teacher's role is restricted to resolving learner difficulties. The three remaining productions mention some kind of learner-learner interaction.

Table 3. Frequency of student productions for principles 1 and 2

Principle 1	Principle 2 Representations				
Authenticity	Interpret	Move symbols	Manipulate	Construct	Total
Tutorial	7	1	5	0	13
Game	2	5	2	0	9
Authentic	1	0	**11**	**3**	15
Total	10	6	18	3	37

An in-depth analysis of the portfolios and computer programs could provide more information on the nature of students' difficulties in implementing constructivist principles when designing the instructional sequence. In order to highlight these, we focus on apparent mismatches between prescribed design principles and students' actual design decisions. The following subparagraphs present detailed examples of student productions for each principle. In the ensuing discussion, we then give some preliminary explanations for student difficulties in terms of both student misinterpretations of design principles and the course rationale itself.

4.3. Principle 1. Authenticity

Roll little bowl is a tutorial in which learners place themselves on the terrestrial globe and determine the ongoing season depending on their longitude. Although an authentic context could easily be imagined for this task, *Roll little bowl* presents only arbitrary questions about seasons and dates unrelated to real life or even to each other. Two other tutorials about the digestive system (*The Digestion* and *Balanced Food*) also failed to embed tasks and questions into yet feasible authentic contexts.

The design of *educational* games requires blending a special mixture of content and leisure into the learner context and student productions show marked difficulties to determine the appropriate balance between them. *The Caves* is a computer program about musical notation. The learner is situated in prehistoric times and visits three caves, one for learning white piano keys, one for black keys, and one combining both. By playing the right key, the learner earns prehistoric bones that allow him to return home. This choice for playing a caveman in order to learn about musical notation on a simulated piano seems rather anachronistic. Moreover, we can speak of inadequate task decomposition: the three exercises for white, black, and white and black keys bear no relation to the reality of playing the piano. Another example of a peculiar combination of subject-matter and context are *Dinosaurs' World* where the user is a palaeontologist, represented as a little dinosaur in a labyrinth, being sent to prehistoric times in order learn about the life of dinosaurs in English. The portfolio states several learning objectives: be exposed to English early in life, learn about animal life, learn to use the computer and become autonomous. The main instructional method however consists of question answering.

Thus, even in games, answering questions is the main mode of interaction but with varying strength in context—subject-matter relations. In *Heading for Maths*, the learner has to designate rectangles in boats and rank order boats with different numbers on them. Although the context involves mathematics, the boats can be seen as mere packaging—i.e., the context bears no relation to the problem posed to the learner. Games also propose specific rewards intended to increase learner motivation but often not specifically related to the content. Examples are *Treasure Hunt*—the learner wins a sword if he or she answers correctly to English vocabulary questions, *Ultima Europa*—the learner collects European monuments by guessing European countries with the help of clues, and *The Animals*—the learner is a detective investigating with the help of a companion dog and getting a diploma involves correctly identifying English animal names in different locations (clinic, zoo, …).

Finally, as an authentic environment, *Lake Annecy* proposes a genuine tourist office situation: as a hostess, the learner has to compose a journey in Annecy and its surroundings for different types of tourists (retired couple, youngsters…). *Lake Annecy* offers a number of choices in cultural, gastronomic, and sportive activities. Whereas the students initially aimed at building a kind of data base with information about Annecy and its surroundings, they did change the global design in order to allow for what can be characterized as an authentic activity. However, the portfolio states they designed the program for evaluating and selecting candidates applying for a job in a tourist office. The portfolio contains no mention of a constructivist rationale, but recommends the use of the program in a traditional test situation.

4.4. Principle 2. Interpreting, Manipulating and Constructing Representations

The principle recommends exploiting the computer for manipulating and constructing representations, but more than a quarter of the programs (10) propose activities that involve mere interpretation of information in different representational formats (text, pictures, sound, clip-arts). Moreover, students make extensive use of pictures mainly for rendering the computer program more attractive. For example, in *Logicode* (a program that teaches road signs) the learner is first exposed to some rules concerning road circulation then he or she has to choose the right answer among three by pointing a checkbox below each image or proposition.

Computer-based simulations propose situations in which learners manipulate the values of some variables and observe the effects. Student-created simulations often use some table that dictates outcomes as a result of values on the input variables. An example is *Simergy* in which the learner chooses between different energy, waste and transportation measures and then observes the immediate and delayed (20 years) effect on the city and country side. Another example is *The Apprentice Chemist's Lab* that presents subject-matter content, exercises, and simulations. The latter involves mixing different liquids (lime water, soda, barium chloride, etc.) and observing the colour of the resulting mixture (blue, transparent, etc.). Still, these simulations show effects of manipulations by means of pictures rather than more complex graphical representations (tables, graphs and so on).

Very few computer programs provide for real construction-type environments, these seem to be hard to conceptualise for students. *Youth Budget* is a program for learning to manage a budget. Students based the design on the French secondary school curriculum. The main justifications for the choice of subject-matter are to render budgeting more practical and interactive and to take over tedious calculations. The key display shows columns for receipts and expenses, as well as different items, e.g., clothing, cigarettes, etc. Learners can individually construct their budget and then see whether it balances; a notebook allows for comparing different trials and for discussing them with a teacher. In their portfolio, students themselves stress the authenticity of the proposed activities. They also conclude on what they themselves gained from the course regarding computer programming with Toolbook, pedagogical aspects, the C&G method, and their own collaboration.

4.5. Principle 3. Participation and Communication

Only three of the portfolios actually mention a learner-learner interaction, the first in situ (*Simergy*), and the others, paradoxically not among learners in the same classroom but through e-mail among distant learners (cf. *Roll Little Bowl*). Thus, students mainly plan one-to-one interactions between the learner and the computer rather than interactions between two learners and one computer.

Few portfolios and programs mention forms of teacher-learner interaction. *Road Safety* is a computer program devised to teach the meaning of different road signs, as well as the application of courtesy rules when driving cars. The environment respects some major constructivist principles like authenticity: the learner moves freely within the environment and has to respect the different road signs. However, the teacher's mere task is either to intervene when a difficulty occurs or to introduce theoretical notions before or after the use of the program. This way of considering the teacher's role is commonly encountered in portfolios; students seem to think of knowledge as being conveyed through a conduit, where only one transmitter (e.g., computer, teacher) can be used at once. Such a conception of the teacher ignores the essential role of discussing, criticising or arguing about the content. This can also be observed in how students personify the computer: "the computer pays attention to learners". An extreme case consists of pointing out that the teacher can be threatened with the use of the computer, because "knowledge replaces the computer" (*sic*).

5. DISCUSSION

On the basis of our analysis, we now elaborate on the main issues raised in the introduction. First, we rated student productions regarding the degree to which they complied with constructivist principles. The analysis grid contained several distinct categories that were straightforward in their application. Moreover, they allowed evaluation of the computer programs on the appropriateness regarding several components, i.e., instructional method, subject-matter content, type of learner and the use of media, simultaneously (Tergan, 1998).

Second, did our students adhere to and follow the three constructivist principles? The *authenticity principle* seems to be the most difficult to implement, the majority of computer programs were tutorial or game-centred, even though developed using the C&G method. Several explanations can be put forward. First, students, even when understanding constructivist principles, may stick to their initial intention or decision to design a game or tutorial. Second, some of the proposed exercises, such as formulating educational goals and hierarchical task analysis, may be advantageous for developing an authentic context for some students, but counterproductive for others by reinforcing traditional ways of conceptualising an instructional sequence. The *representation principle* was relatively well applied, more than half of the computer programs involved manipulation or construction activities. Although this may be attributed to the availability of tools in Toolbook (e.g., drag and drop, inserting images), students appeared to be convinced of the relevance of exploiting the computer for manipulation and construction (more than of the relevance of authenticity). Concerning the *communication principle*, the students did not seem to consider knowledge building as the product of discussion or argumentation within a community. Moreover, although possible, communication tools are not easy to develop with Toolbook. Nevertheless, as shown in their portfolios, students seem to think of knowledge as individually acquired, with the help of the teacher only when encountering difficulties. This can be related to a folk psychology view in which learning is merely conceived as transfer of information (Bereiter & Scardamalia, 1996).

Third, whereas it appears that the mere expression of constructivist principles is not sufficient for their application by students, one might argue that their definition is more problematic than revealed by our analysis. As Petraglia (1998) pointed out, constructivist ID principles often lead to misinterpretations because they are translated into *preauthentic* prescriptions, that is, rules-of-thumb that do not take into account epistemological dimensions. Briefly, such epistemological accounts argue that “the world is not understood independently of our experiences, and that, therefore, authenticity can be neither predetermined nor preordained” (Petraglia, 1998, p. 58). Thus, the learner’s environment has to be considered from the learner’s point of view and past experiences, rather than per se. For example, Petraglia points out that the task of balancing a cheque book might be authentic from the point of view of adults, but not for children. As a result, one could argue that asserting constructivist principles for design is misleading since a constructivist program might be used in a non-constructivist way (e.g., exercising with LOGO) and vice versa. On the other hand, it might go too far for student-designers to understand the distinction between their own designers' rationale, teachers' prescriptions, and the learners' actual use before even starting to design an educational computer program.

Fourth, our course rationale could be responsible for some of the students' misinterpretations as it is a mixture of behaviorist, cognitivist, and constructivist principles. Are these contradictory prescriptions? Regarding the choice of subject-matter, whereas the C&G method relies on teachers' experience, our students generally are neither teachers nor domain experts. On the other hand, domain knowledge, although essential for C&G, does not strictly pertain to constructivist's rhetoric, but to cognitivist's. In any case, some of the weak designs can be attributed to the choice of subject-matter content in combination with instructional method, e.g., tutorials for exercising maths problems. Furthermore, the use of hierarchical task analysis is debatable since it advocates structuring the world for teaching purposes. Conversely, a high-quality task-decomposition may also effectively guide the design of a highly interactive environment. In fact, expert knowledge might be ineffective since tacit procedural knowledge cannot be transferred to instructional design, but the lack of domain knowledge leads to mere traditional ways of using the computer in education. Finally, a drawback of the course may be its weak consideration for participation and communication. Since C&G separately considers the roles of the computer, learner, and pupils, it is difficult for the students to take into account their multiple interactions, more particularly in knowledge-building discussions.

Finally, can we speak of effective and enjoyable learning environments? Both effectiveness and enjoyability are tacit concerns for our students as shown in the number of exercisers and games (22 computer programs). Authenticity, construction, and communication seem to be less appealing to them (15 computer programs). A discussion of our analysis grid with some examples of student productions early on in the course might develop students' awareness of constructivist principles and their role in designing computer programs for learning. Future work should address both the relationships between them and their value as guiding principles in teaching the design of computer-supported learning environments.

6. ACKNOWLEDGEMENTS

We thank Benoît Lemaire and Emmanuel Sylvestre for providing valuable comments of an earlier version of this paper. This paper was written during a sabbatical of the first author at TECFA, University of Geneva, Switzerland.

AFFILIATIONS

Laboratory for Educational Sciences, Pierre-Mendès-France University and IUFM, Grenoble, BP 47, 38040 Grenoble cedex 9, France, Philippe.Dessus@upmf-grenoble.fr, Erica.deVries@upmf-grenoble.fr

REFERENCES

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

Alessi, S. M. & Trollip, S. R. (1991). *Computer-Based Instruction: Methods and Development*. Englewood Cliffs, NJ: Prentice-Hall.

Bereiter, C., & Scardamalia, M. (1996). Rethinking learning. In D. Olson & N. Torrance (Eds.), *The Handbook of Education and Human Development* (pp. 485-513). Oxford: Blackwell.

Crossley, K., & Green, L. (1985). *Designing Computer Lessonware, a Practical Guide for Teachers* (fr. trans. 1990). Toronto: Crossley and Green.

CTGV [Cognition and Technology Group at Vanderbilt] (1996). Looking at technology in context: a framework for understanding technology and education research. In D. C. Berliner & R. C. Calfee (Eds.), *Handbook of Educational Psychology* (pp. 807-840). New York: McMillan.

De Vries, E. (2003). Educational technology and multimedia from a cognitive perspective: Knowledge from inside the computer, onto the screen, and into our heads? In H. van Oostendorp (Ed.), *Cognition in a Digital World* (pp. 155-174). Mahwah, NJ: Lawrence Erlbaum Associates.

Heller, R. S., & Martin, C. D. (1995). A media taxonomy. *IEEE Multimedia, 2*(4), 36-45.

Jonassen, D. H. (1994, April). Thinking technology. Towards a constructivist design model. *Educational Technology*, *34*(4), 34-37.

Jonassen, D. H. (1997). Instructional design models for well-structured and ill-structured problem-solving learning outcomes. *Educational Technology Research and Development, 45*(1), 65-94.

Kirschner, P., Carr, C., van Merriënboer, J., & Sloep, P. (2002). How expert designers design. *Performance Improvement Quarterly*, *15*(4), 86-104.

Lave, J., & Wenger, E. (1991). *Situated Learning: Legitimate Peripheral Participation*. Cambridge: Cambridge University Press.

Mager, R. F. (1962). *Preparing Instructional Objectives* (fr. trans. 1972). Palo Alto: Fearon.

Patrick, J., Gregov, A., & Halliday, P. (2000). Analysing and training task analysis. *Instructional Science, 28*, 51-79.

Petraglia, J. (1998). The real world on a short leach: the (mis)application of constructivism to the design of educational technology. *Educational Technology Research and Development, 46*(3), 53-65.

Pieters, J. M. & Bergman, R. (1995). The empirical basis of designing instruction. *Performance Improvement Quarterly, 8*, 118-129.

Reimann, P. (1999). Commentary: The role of external representations in distributed problem solving. *Learning and Instruction, 9*, 411-418.

Rowland, G. (1992). What do instructional designers actually do? An investigation of expert practice. *Performance Improvement Quarterly, 5*, 65-86.

Rowland, G. (1993). Designing and instructional design. *Educational Technology, Research and Development*, *41*, 79-91.

Scardamalia, M., Bereiter, C., & Lamon, M. (1994). The CSILE project: Trying to bring the classroom into World 3. In K. McGilly (Ed.), *Classroom Lessons: Integrating Cognitive Theory* (pp. 201-228). Cambridge, MA: MIT Press.

Shambaugh, N., & Magliaro, S. (2001). A reflective model for teaching instructional design. *Educational Technology, Research and Development*, *49*, 69-92.

Simon, H. A. (1996). *The Sciences of the Artificial* (fr. trans. 2004). Cambridge, MA: MIT Press.

Tergan, S.-O. (1998). Checklists for the evaluation of educational software: Critical review and prospects. *Innovations in Education and Training International, 35*(1), 9-20.

Young, A. C., Reiser, R. A., & Dick, W. (1998). Do superior teachers employ systematic instructional planning procedures? A descriptive study. *Educational Technology Research & Development, 46*(2), 65-78.