DOCUMENT RESUME

ED 374 775 IR 016 815

AUTHOR Nieveen, Nienke; And Others

TITLE Exploration of Computer Assisted Curriculum Development.

PUB DATE Apr 94

NOTE 17p.; Paper presented at the Annual Meeting of the American Educational Research Association (New Orleans, LA, April 4-8, 1994).

PUB TYPE Reports - Evaluative/Feasibility (142) -- Speeches/Conference Papers (150)

EDRS PRICE MF01/PC01 Plus Postage.

DESCRIPTORS *Computer Assisted Instruction; Cooperative Programs; *Curriculum Development; Educational Media; Educational Technology; Foreign Countries; Higher Education; Research Needs

IDENTIFIERS Performance Support Systems; University of Twente (Netherlands)

ABSTRACT

This paper contains a report about a collaborative project of the University of Twente (Netherlands) Faculty of Educational Science and Technology, Department of Curriculum and the Dutch National Institute for Curriculum Development (SLO). The project explores the potentially supportive role of the computer in improving the quality and efficiency of curriculum development activities. This report describes the early steps and findings of this long term project. A brief introduction on the context of curriculum development by the SLO is presented. The functions and characteristics of the system are then outlined, based on an analysis of available systems and on recent trends in electronic performance support systems. The project's development and evaluation approach is described, and the current version of the prototype is examined. Finally, recommendations are presented for further research. (Contains 27 references.) (JLB)

Exploration of Computer Assisted Curriculum Development

Paper presented at AERA meeting, April 1994, New Orleans

Nienke Nieveen

Jan van den Akker

Tjeerd Plomp

University of Twente

Correspondence address:
University of Twente, Faculty of Educational Science and Technology,
Department of Curriculum, P.O. Box 217, 7500 AE Enschede, The Netherlands

Exploration of Computer Assisted Curriculum Development

1. Introduction

In this paper we will report about a collaborative project of the University of Twente (Faculty of Educational Science and Technology, Department of Curriculum) and the Dutch National Institute for Curriculum Development (acronym SLO), in which we explore the potentially supportive role of the computer in improving the quality and efficiency of curriculum development activities.

The project has been preceded by a preliminary study that included:

- an analysis of five existing (prototypes of) computer assisted curriculum/instructional development systems;
- an orientation on computer support systems (especially 'electronic performance support systems' [EPSS]) in other task environments;
- design of a first version of a prototype (in Hypercard) for the component of formative evaluation in a hypothetical computer assisted curriculum development system.

This preliminary study (Nieveen, 1993; Nieveen, van den Akker & Plomp, 1993) resulted in a set of recommendations for the design (functions, structure, content, lay-out) of such a system and for the methodology of its further development.

In an ongoing R&D project these findings are specified and elaborated by designing a prototype (with complete and tested specifications) for the SLO context. The initial focus is still on supporting the component of formative evaluation within the framework of a more comprehensive model of curriculum development. Along an iterative design and evaluation approach successive versions of this prototype will be developed in co-operation with a small group of potential users. In a later stage the prototype will be tested on its practicality and effectiveness for a large and diverse target group of professionals in curriculum development. Moreover, it is foreseen to gradually broaden the system by including several related components of curriculum development.

This paper will describe the early steps and findings of this long term project. First, a brief introduction on the context of curriculum development by the SLO will be offered. Second, the envisaged functions and characteristics of the system will be outlined, partly based on an analysis of available systems and on recent trends in EPSS's. Third, the project's development and evaluation approach will be described. Afterwards, we sketch the current version of the prototype, including the comments on it from the group of user-participants. Finally, we present some conclusions for further steps.

2. Context of the SLO

The organization of curriculum development in The Netherlands reflects the general educational policy of balancing between central regulations and decentral autonomy. The SLO publishes curriculum plans and materials (at both national, school, and classroom level) that have only an exemplary, not obligatory, status. Everyone is free to use these products and to adapt them to their own needs and situation. This curriculum policy aims at interaction between generic curriculum develoment by a center (SLO) and site-specific activities by schools and teachers. Moreover, various intermediairy organizations (especially educational publishers and teacher training institutes) play an important role.

The SLO (in operation since 1976; staff size: about 150 curriculum developers) carries out a large number of curriculum development projects for (subjects in) elementary, secondary, vocational, and adult education. In recent years, concerns grew about the lack of reliable and systematic information about the development strategies that are actually practiced in the projects. This was seen as an obstacle to learn from previous experiences and to improve the professional repertoire and instruments of the developers.

For that reason, a series of retrospective case studies on the development practices of a representative sample of 18 projects have been conducted. One of the main conclusions of the study was that the overall development approach can be characterized as (overly) intuitive and unstructured. Curricular decisions were predominantly inspired by ideas (visions), but insufficiently adjusted on the basis of evaluative information about their effects. Systematic iteration with gradual improvement of successive product versions, based on systematic formative evaluation, and anticipation on implementation problems was uncommon. Besides, there was a lot of confusion about basic concepts, target groups and product functions.

The results of the study (van den Akker, Boersma & Nies, 1990; van den Akker & Boersma, 1993) stimulated to a debate about the possibilities of increasing the rationality of curriculum development. The reactions sometimes tended to narrow down to a simplistic controversy between 'creative' and 'technocratic' approaches. Eventually there was little disagreement that a combination of creativity, common sense, and systematic procedures is necessary for improving the product quality and process efficiency.

Of course, it is impossible to rationalize the entire process of generic curriculum development, which is usually characterized by an overwhelming variety of actors, ideas, interests, and contexts. However, we expect that several components (notably: context and needs analysis, and formative evaluation; see van den Akker, 1992) can benefit from more systematic procedures and instruments. Moreover, the efficiency of the overall process can be increased by a stronger standardization of the development routes and the procedures for decision making in projects (Boersma, 1992).

One of these issues has got institutional priority for professional improvement, that is the domain of formative evaluation, because that seems so critical for 'quality assurance' of the developmental activities. A start has been made to devise a set of practical guidelines and instruments for planning and conducting formative evaluation (Keursten & Nies, 1993).

This booklet is intended to support the curriculum developers to integrate formative evaluation activities in their flow of activities. Although the first reactions by SLO developers on this kind of information was rather positive, there was also some reluctance to the rather static nature of the written guidelines and examples. It was felt that a more flexible and interactive kind of support would increase the surplus value of the procedural advices and instruments. Therefore the initiative arose to explore the possibility of designing an electronic support version.

3. Computer Support Systems for Curriculum Developers?

3.1 Analysis of available computer support systems for curriculum development

To explore in general the possibilities of supporting curriculum development activities by a computer system, five existing computer supported curriculum development systems in the Netherlands, Germany and the United States were analyzed (Nieveen, 1993):

- CEDID - Computer Ergänztes Didaktisches Design
  [Computer Assisted Instructional Design]
  Georg-August-Universität Göttingen, Göttingen, Germany;
- COCOS - Computer Ondersteunde Cursus Ontwerp Systeem
  [Computer Assisted Course Design System]
  Anderson Consulting-ECC, Enschede, the Netherlands;
- IDD Advisor - Instructional Design and Development Advisor
  University of Colorado (Division of Instructional Technology) in Denver, USA
- ID Expert - Instructional Development Expert;
  Utah State University (Department of Instructional Technology) in Logan, USA
- ID Library - Instructional Development Library
  University of Georgia (Department of Instructional Technology) in Athens, USA.

Most of the analyzed computer support systems were still in a prototypical stage and the developers had not yet collected systematic experiences with it in practice. Only evaluation results of ID Library were known; this system got positive evaluation results.

When looking at the curriculum levels the support systems aim at, it appears that in most of the systems the support is restricted to the development of instructional packages; more comprehensive levels of curriculum planning were only dealt with in CEDID. Some of them support the entire development process until final delivery (ID Expert, COCOS), others aim at supporting only parts of the instructional development process. For example, IDD Advisor supports only the selection of instructional strategies and the media selection for a specific instructional process. In CEDID, however, the development of a curriculum on course level is seen as a condition for developing an instructional package. CEDID also provides support for this meso level.

Looking at the kind of support the systems provide, it becomes clear that two of them (COCOS and ID Library) provide support in structuring and systematizing the instructional development process. These systems also provide information on different aspects of instructional development. CEDID combines this kind of support with some advice. IDD Advisor and ID Expert only provide advice on the development process. It is apparent that non of the support systems is developed as a learning system. However, by providing information and advice, there is a great chance that instructional developers will learn from the system.

The developers of the different curriculum support systems assume several advantages of providing support with a computer:

- It can encourage a more structured curriculum development approach (ID Library).
- It can encourage the internal consistency of design decisions (CEDID, ID Expert).
- All kinds of examples can be stored in a kind of library. Later on these examples can be revised to fit another situation (ID Library, COCOS).
- When an employee changes jobs, often a lot of knowledge will leave with him. By storing the knowledge into a computer system, the knowledge can be saved (CEDID).
- Less experienced developers can use the support system in order to the get familiar with the development process of an organization (CEDID, COCOS).

3.2 How do these systems relate to recent thinking about EPSS?

With the idea of Electronic Performance Support Systems a fresh light is shed on the possibilities of support on the workplace. Where formerly different job aids (paper-based as well as inteligent computer-based job aids) and courses were separately provided to employees, now with an EPSS integrated *information, advice* and *learning opportunities* are provided, to improve the performance of the user (Raybould, 1990; Gery, 1991).

Gery (1991) assumes that an advantage of an EPSS is, that with such a system the employee will learn to perform his task more efficiently than in a traditional training situation. Another advantage is that an EPSS can provide advice, information and learning 'just in time'. The employee doesn't need to remember all issues related to his work, but he can consult the EPSS on the issue at the time he really needs it (Raybould, 1990; Gery, 1991; Geber, 1991; Horn, 1989). This can lead to less information load. By integrating all support in a single system a user can quickly find the required advice, information and learning opportunities, without being detracted.

Most of the analyzed support systems for curriculum development provide support which is not integrated like in an EPSS. Especially, explicit learning opportunities are lacking in all support systems. However, there is a great chance that a user will learn by using the system. Integrating different kinds of support in a single system seems to be useful. The support could even be extended to support of cooperative work. Collis and Verwijs (1994, p. 9) categorize various kinds of computer support in the term 'hybrid EPSS', which is "an electronic system, through which, via a common front-end, the user can interact with the system to obtain various types of local or distributed help and resources for individual or group-oriented activities related to learning, problem oriented thinking, and collaboration".

3.3 Conclusion: CASCADE

Based on the analysis of computer support systems for curriculum development and on the recent ideas of EPSSs, it was concluded that an computer support system, especially developed for the SLO context, could fit the special needs of curriculum developers of the SLO. The system is called CASCADE: Computer ASsisted Curriculum Analysis, Design and Evaluation.

4. Intended Functions and Characteristics of CASCADE

Based upon earlier research findings and group discussions, several assumptions have been made about the functions of CASCADE. First, the computer system should support the improvement of the quality of curriculum products. This refers to the questions like: Are the various curricular elements (like goals, content, instructional activities, etc .) correct and adequately related to each other?; Are the products workable for the target group; Are the products effective? Second, the computer system should support the efficiency of the curriculum development process.

With respect to the curriculum development process CASCADE will focus on the middle part of a model that represents the major stages of that process.

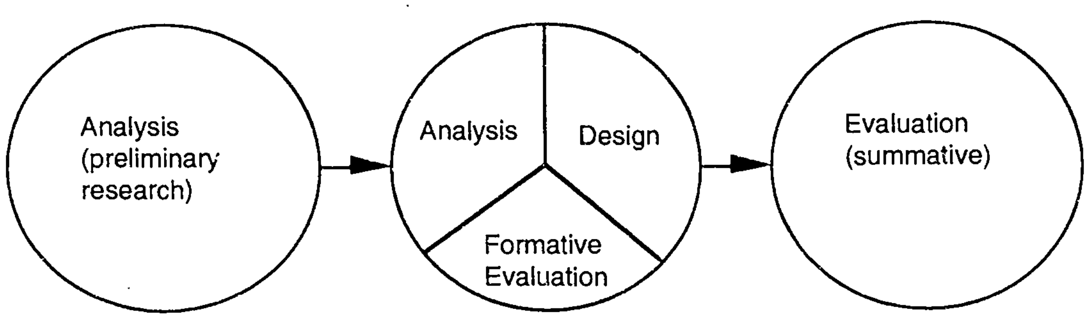

Figure 1: Curriculum development model in CASCADE

Based upon the analysis of problems, needs and context, tentative ideas for a new curriculum will iteratively evolve to a final version. To make this 'evolution' of the curriculum possible, evaluative feedback from different sources is necessary. Therefore, in the model it is stressed that formative evaluation is a continuous process in the development process. Looking at possible needs users might have, CASCADE supports with two types of 'just-in-time knowledge': an advice and an information system. Learning opportunities will not be provided explicitly by CASCADE.

	Kinds of support			
	Advice	Information		
		Examples	Explanation	External sources
Contents of support	Heuristic advice about procedures for: Analyzing Designing Evaluating new curricula	Instruments Procedures Curriculum documents Instructional materials ...	Concepts	Background information

Table 1: Kinds of support in CASCADE

The first kind of support is 'just-in-time advice'. CASCADE gives heuristic advice about procedures of analyzing, designing and evaluating during different development stages of the curriculum. While providing the advice, the support system will not force the user to follow the advices. The user maintains control on the development activities.

Next to this kind of advice CASCADE provides 'just-in-time information'. The system makes available explanations of certain concepts; it provides examples, which can be useful during analyzing, design and evaluation activities. Also it might be useful when the system would provide access to all kinds of external available information sources.

Of course, it is important to pay attention to design aspects of the user interface. The user interface "must serve as the vehicle that enables communication between the system and any user of the system" (McGraw, 1992, p. 4). An user interface has several attributes, like learnability, satisfaction, memorability and error rate (cf. Nielsen, 1993). With respect to these aspects,

- CASCADE should be easy to learn to use (learnability);
- CASCADE should be pleasant in use (satisfaction);
- it should be easy to remember the way of using CASCADE (memorability);
- the error rate of CASCADE should be low (error free); when the user makes an error, it should be easy to recover it.

5. Development Approach for CASCADE

The development approach taken in this project can be characterized by the terms iterative and participatory development.

Iterative development

When developing a new system, it is hard for both the developers and the target group to make clear on beforehand what exactly the specifications for the system should be. In such cases the requirements of the design can come from an iterative succession of experiences with real work in the field (cf. Goodrum, Dorsey & Schwen, 1993b; Schneiderman, 1992; Tessmer, 1994). Smith (1991) refers to this kind of development approach with the term 'evolutionary prototyping'. By providing concrete prototypes a better common ground for the developer of the support system and the target group can be achieved, than by providing some abstract specifications. Such an early prototype can be discussed by the developer and persons of the target group. Based on these discussions the specifications can be refined, which will also lead to revision of the early prototype. In this way the computer system will evolve during the development process towards final delivery. By reviewing functional versions of the product early in the design stage, formative evaluation activities are made more a part of the front-end analysis.

Another important characteristic of iterative development approaches is the idea of

"think BIG, start small".

By first trying to develop a small part of the final system, one keeps the development process manageable and one can learn from failures and apply successes when developing the next parts. Gery (1991, p. 208) warns us as follows: "Some of the biggest failures I have seen result from the developing team biting off more than it can chew, while it is simultaneously learning how to develop EPSS and creating the initial EPSS infrastructure."

Participatory approach

When applying an iterative development approach, the participation of the target group is of paramount importance. Monk, Wright, Haber and Davenport (1993, p. 5) stress that "involving users in design does not mean having endless meetings where some abstract specifications is discussed with union representatives or management. [It] requires access to people typical of those who will actually use the system, not their representatives or management." Goodrum, Dorsey and Schwen (1993a) agree on this by pointing out that "as collaborators their [the participants] responsibilities extend beyond their traditional role where one might interview them about their needs, have them sign-off on a list of specifications, or have them test the product after it has been designed and developed. In participatory design, users are rightfully active participants in the creative process of design" (p. 14). Schneiderman

(1992, p. 474) sums up the following advantages of participatory development strategies:

- more user involvement brings more accurate information about tasks;
- opportunity to argue over design decisions;
- potential for increased user commitment and ownership of the final system.

6. Design of first prototype focused on supporting formative evaluation activities

With the above principles in mind, a start was made to develop a first prototype of CASCADE. The prototype is developed with HyperCard, being an excellent tool for iterative development (cf. Schneiderman, 1992)

To narrow down the project's field of interest, it was decided that the prototype would focus on the formative evaluation part of the proposed development process. As explained in the first section of the paper, formative evaluation activities can be seen as a condition for developing high quality curricula, but are not adequately incorporated in the SLO projects.

In the first prototype only guidelines are given for the formative evaluation of lesson materials.

	Kinds of support			
	Advice	Information		
		Examples	Explanation	Other sources
Contents of support	Heuristic advice about: Evaluation question; Method selection; (Performing activities)	(Instruments Procedures)	Concepts	

( ) support between brackets will be available in next prototypes

Table 2: Kinds of support in the first prototype of CASCADE

The user can get advice on two aspects. The first aspect relates to the question "What should I evaluate formatively?" The user can specify his formative evaluation question by selecting different options:

a. the object of the formative evaluation (lesson materials on the micro level or curriculum documents on the meso or macro level);
b. the uncertainties with respect to the object (correctness, internal consistency, acceptability,

practicality or effectiveness);

c. the development stage of the object (design specification, tentative development, detailed development of part of the materials, detailed development of materials, final delivery).

The second point of advice refers to the question "What formative evaluation method should I use in my situation?" Based upon the selected kind of question, CASCADE selects possible formative evaluation methods. A selection can be made between four kinds of methods:

- **Screening**: the first design can be screened with some checklists on several attributes of the instructional materials, like instructional design attributes, language attributes and subject matter attributes (Monk, Wright, Haber & Davenport, 1993; Thiagarajan, 1991; McAlpine & Weston, 1994). For example McAlpine and Weston (1994) provide such checklists of attributes of instructional materials. A part of the checklist for presentation attributes looks like this:

Presentation Attributes

1. Space:
   - Is ample space provided where written answers are elicited?
   - Is a consistent method used for allocating space between headings, sub-headings, paragraphs, words, and lines?

2. Typeface:
   - Is a legible typeface used?
   - Is upper case type used only for initial letters and proper nouns since lower case facilitates reading?

3. Titles, headings and sub-headings:
   - Do they clarify and guide?
   - Are they as short as possible?

4. Graphics, illustrations, visuals:
   - Are these elements supportive of content and accomplish somethimg that the narrative cannot?
   - Are they appropriate for the intended audience?

5. ...

Table 3: Part of checklist of attributes of instructional materials (McAlpine & Weston, 1994)

- **Expert Appraisal**: depending on the uncertainties of the developer, experts are asked to review the design (for example: subject matter experts, instructional design experts and teachers) (Tessmer, 1993; Thiagarajan, 1991).
- **Micro Evaluation**: informal tryouts are carried out with persons of the target group.The evaluator can more/less interfere with the learning process, and the evaluation can be carried out under a situation which is more/less the same compared to practice

(Thiagarajan, 1991; Tessmer, 1993; Keursten & Nies, 1993);

- **Field Testing**: the design will be evaluated under the same circumstances as in practice. (Tessmer, 1993; Keursten & Nies, 1993).

The following method selections are possible:

		Developmental stages				
		design specification	tentative development	detailed development of part of materials	detailed development of materials	final delivery
Uncertainties	Correctness	s e	s e	s e	s e	s e
	Internal consistency	s e	s e	s e	s e	s e
	Acceptability	e	e m	e m	m f	f
	Practicality	e	e m	e m	m f	f
	Effectiveness		m	m	m f	f

s = screening
e = expert appraisal
m = micro evaluation
f = field test

Table 4: Selection table for formative evaluation methods

A next prototype will provide a third kind of advice (between brackets in table 2) related to the question "How should I conduct formative evaluation activities?" The prototype will give guidelines on how to carry out the activities for every method.

Next to these kinds of advice, CASCADE provides information about formative evaluation. This information is referring to the question "What is meant by certain concepts of formative evaluation?" At all moments the user of CASCADE can ask for explanation of concepts. When the user has finished the explanation, he will return to the place he left. In a next version also the following question: "Can I have some examples of instruments and procedures?" will be answered, by including a selected set of examples.

With respect to the user interface of CASCADE, the following guidelines are taken into account. To improve the learnability, satisfaction and memorability system should be very consistent in operation (McGraw, 1992; Grabinger, 1993). This means that for example the use of functional areas and commands, the style and use of specific words are worked out the same throughout CASCADE. Next to being consistent, the screen design should be well-organized.

The following guidelines are being followed (Rivlin, Lewis & Davies-Cooper, 1990):

- clear screen areas;
- limited information and text on each screen;
- correct text, without errors;
- avoid 'scrolling' of text.

The ease of use of the system can also be influenced by the kind of navigation (Keursten, 1994; Schneiderman, 1992). With respect to this aspect the following guidelines are being followed:

- at every moment it should be clear to the user what he is expected to do and what options he has;
- at every moment the user can look up information about the navigation;
- after selecting an option it should be made clear to the user, what option he selected.

The ease of use will also be improved by using understandable language. With respect to making errors it is remarked that the system should be error free. If a user makes an error it should be easy to recover it.

7. Comments on first prototype

7.1 Try-out

To find out if the assumed functions and characteristics of the prototype meet the needs of the target group the first prototype of CASCADE was tried out. As pointed out earlier in this paper, to meet this goal participation of persons of the target group is very important. Therefore a 'user group' was recruited. The group consists of five SLO developers and is in some sense typical of the target group. The persons in the group differ on the following aspects:

- department of the SLO they are working (2 primary education; 3 secondary education);
- experience with formative evaluation (2 relatively much; 3 relatively little);
- experience with HyperCard-like applications (2 relatively much; 3 relatively little).

For the formative evaluation of the prototype the method 'cooperative evaluation' has been used, which can be seen as a method for participatory design (Monk, Wright, Haber & Davenport, 1993). By using this method users work through a set of tasks with the system. These tasks must be representative of the work that the product will support. Based on the problems while using the system and the user's comments the prototype can be revised.

In the case of the prototype of CASCADE the following tasks were selected:

- Task Sheet -

A. Find some background information on CASCADE.
B. Find some information on the navigation of CASCADE.
C. Find out information on how to prepare the formative evaluation method for some lesson materials you are developing:
   1. Define your formative evaluation question.
   2. Select a formative evaluation method.
   3. Find out how the evaluation activity can be prepared.
D. Find some explanation on the concept of 'formative evaluation'.
E. The same task as C, but for another question.

Figure 2.: Task sheet for the formative evaluation of first prototype of CASCADE

Task A and B have nothing to do with the actual goal of the prototype of CASCADE (to support the formative evaluation activities) but they can be important when someone would like to carry out formative evaluation activities with CASCADE.

Each user worked for about half an hour with CASCADE to perform the tasks. While using the system the users were encouraged by the evaluator to think aloud. In this way everything they said and did was recorded. When the user had finished the set of tasks, there was a debriefing. They were asked what they liked or disliked of the prototype. With respect to the functions of CASCADE they were asked whether the support of CASCADE fits their needs and what they think of the idea to add examples to CASCADE. With respect to the user interface, they were asked what they think of the ease of use of CASCADE.

7.2 Comments

With respect to the functions all users agreed that there is a need for the kind of advice and information CASCADE provides on the topic of formative evaluation. So, the user group received the basic idea of the prototype positively. The most important comment they made was that there is no merit yet compared to a handbook. According to the users this will occur, when:

- instruments are available, which can be revised for the user's situation and checklists for how to carry out a certain method are offered;
- all input could be saved in a file, so that it can be used later on;
- when input in one part of the system has consequences in the rest of the system (for example when the user is working for the target group 'learners of first grade of secondary school, everywhere in the system the term 'target group' should be replaced with 'learners of first grade of secondary school').

Next to this comment, it was said that the uncertainties should be more differentiated, so that the user can make a better fit between the question he has and the support of the system.

Some problems occurred during the use of the prototype. There were some navigational problems, due to a lack of explanation in the system. Also, the system not always was consistent in providing explanations. One of the general comments upon the practicality was that one should be able to go quickly through the system. This could probably be improved by allowing cross references. Finally the users thought that the ease to find explanation on an important or new concept would probably be improved, when it would be possible to get explanation by just 'clicking' on the concept in the screen, instead of going to the explanation menu.

8. Epilogue

In this paper we tried to make explicit our preliminary thoughts about a computer support system for curriculum developers of the National Institute of Curriculum Development in the Netherlands. We also tried to give an impression of our efforts to specify these ideas in an early prototype.

The development and testing of the prototype helps us in clarifying our aims and assumptions and in identifying the strong and the weak parts of it. Based on this we can revise the initial specifications. Obviously, we have to refine our thinking about the way we can improve the merit of computer based support. We need to elaborate and refine the contents of the support system and we need to look for ways of improving the usability of the system. Also we need to look more carefully for methods of testing the prototypes.

In the next coming future we will try to tackle these 'problems', in order to let the computer support system for curriculum developers evolve.

References

Boersma, K. (1992). *Improving the efficiency of curriculum development of the National Institute for Curriculum Development (SLO).* Paper presented at the European Conference on Educational Research, June 22-25, Enschede, The Netherlands.

Collis, B.A., & Verwijs, C. (1994). *A human approach to Electronic Performance and Learning Support Systems: Hybrid EPSSs.* (draft) Enschede: University of Twente.

Geber, B. (1991a). Help! The rise of performance support systems. *Training, 28*(12), 23-29.

Gery, G.J. (1991). *Electronic performance support systems: How and why to remake the workplace through strategic application of technology.* Boston, MA: Weingarten Publications.

Goodrum, L.T., Dorsey, D.A., & Schwen, T.M. (1993a). Defining and building an enriched learning and information environment. *Educational Technology, 11*, 10-20.

Goodrum, L.T., Dorsey, D.A., & Schwen, T.M. (1993b). Just-in-time knowledge performance support: A test of concept. *Educational Technology, 11*, 21-29.

Grabinger, R.S. (1993). Computer screen designs: Viewer judgments. *Educational Technology Research & Development, 41*(2), 35-73.

Horn, R. (1989). The economic advantages of knowledge support systems. *Performance Improvement Quarterly, 2*(3), 72-86.

Kessels, J.W.M. (1993). *Towards design standards for curriculum consistency in corporate education.* Enschede: University of Twente.

Keursten, P. (1994). *Courseware-ontwikkeling met het oog op implementatie: De docent centraal* [Courseware development from an implementation perspective: A central role for the teacher]. Enschede: University of Twente.

Keursten, P. & Nies A. (1993). *Evalueren van lesmateriaal* [Evaluation of instructional materials]. Enschede: SLO.

McAlpine, L., & Weston, C. (1994). The attributes of instructional materials. *Performance Improvement Quarterly, 7*(1), 19-30.

McGraw, K. (1992). *Designing and evaluating user interfaces for knowledge-based systems.* New York: Ellis Horwood.

Monk, A., Wright, P., Haber, J., & Davenport, L. (1993). *Improving your human-computer interface: A practical technique.* New York: Prentice Hall.

Nielsen, J. (1993). *Usability engineering.* Boston: Academic Press.

Nieveen, N.M. (1993). *Computerondersteunde curriculumontwikkeling: Een verkennende studie.* [Computer supported curriculum development: An exploratory study] Enschede: University of Twente.

Nieveen, N.M., van den Akker, J.J.H., & Plomp, Tj. (1993). *Computerondersteunde curriculumontwikkeling: Een verkennende studie* [Computer supported curriculum development: An exploratory study]. Paper presented at the Onderwijsresearchdagen in Maastricht, May 26-28.

Raybould, B. (1990). Solving human performance problems with computers. A case-study:

Building an electronic performance support system. *Performance & Instruction, 29*(10), 4-14.

Rivlin, C., Lewis, R., & Davies-Cooper, R. (Eds.) (1990). *Guidelines for screen design.* Oxford: Blackwell Scientific Publications.

Schneiderman, B. (1992). *Designing the user interface: Strategies for effective human-computer interaction.* Reading (MA): Addison-Wesley.

Smith, M.F. (1991). *Software prototyping: Adopton, practice and management.* London: McGraw-Hill.

Tessmer, M. (1993). *Planning and conducting formative evaluations: Improving the quality of education and training.* London: Kogan Page.

Tessmer, M. (1994). Formative evaluation alternatives. *Performance Improvement Quarterly, 7*(1), 3-18.

Thiagarajan, S. (1991). Formative evaluation in performance technology. *Performance Improvement Quarterly, 4*(2), 22-34.

van den Akker, J. (1992). *How to optimize the quality of curriculum materials?* Paper presented at the European Conference of Educational Research, June 22-25, Enschede, The Netherlands.

van den Akker, J. & Boersma, K. (1993). *Improving the effectiveness of curriculum development.* Paper presented at the AERA annual meeting, April 12-16, Atlanta (USA).

van den Akker, J., Boersma, K., & Nies, A. (1990). *Ontwikkelstrategieën in SLO-projecten.* [Development strategies in SLO projects]. Enschede: SLO.