DOCUMENT RESUME

ED 276 406 IR 012 401

AUTHOR	Moonen, Jef; Schoenmaker, Jan
TITLE	Production Techniques for Computer-Based Learning Material.
INSTITUTION	Twente Univ. of Technology, Enschede (Netherlands).
REPORT NO	COI-86-611
PUB DATE	Apr 86
NOTE	15p.; Paper presented at the Annual Meeting of the American Educational Research Association (67th, San Francisco, CA, April 16-20, 1986). A publication of the Centrum voor Onderwijs en Informatietechnologie.
PUB TYPE	Reports - Descriptive (141) -- Speeches/Conference Papers (150)
EDRS PRICE	MF01/PC01 Plus Postage.
DESCRIPTORS	*Computer Assisted Instruction; *Computer Software; Foreign Countries; Government Role; *Instructional Design; *Instructional Development; *Production Techniques; Teacher Education
IDENTIFIERS	*Netherlands

ABSTRACT

Experiences in the development of educational software in the Netherlands have included the use of individual and team approaches, the determination of software content and how it should be presented, and the organization of the entire development process, from experimental programs to prototype to final product. Because educational software is a relatively new phenomenon in education itself, it is not easy to determine in advance the actual desires of the potential user; current production strategies are based on either an individual approach, a team approach, or a project approach. None of the educational sectors has reached the stage at which it would be profitable to publish the materials developed. Guidance tools, project management and system development methodologies, and the organization of education into macro, meso and micro levels have been developed to support software development. Working out problems has led to splitting the development process into four distinct phases: (1) describing the purpose of the program; (2) describing the functional aspects of different subsystems; (3) determining detailed pedagogical design; and (4) designing the system by which the encoding takes place. In order to stimulate the introduction of computers in education, the Dutch government has established the 5-year "Informatics Stimulation Plan"; three development centers, each one specifically aimed at a particular sector of education; 28 regional centers for education and information technology; and a clearinghouse for educational software. The need for teacher training programs to include courses illustrating the use of computers as educational aids is noted. (DJR)

coi centrum voor onderwijs en informatietechnologie

PRODUCTION TECHNIQUES FOR COMPUTER-BASED LEARNING MATERIAL

postbus 217
7500 AE Enschede
telefoon 053-892190

centrum voor
onderwijs en
informatietechnologie

Ref.: COI.86.611
Date: March 1986

PRODUCTION TECHNIQUES FOR COMPUTER-BASED LEARNING MATERIAL

Jef Moonen
Jan Schoenmaker
Centre for Education and Information Technology

Twente University of Technology
P.O. Box 217
7500 AE Enschede
The Netherlands

Paper presented at the annual AERA meeting, San Francisco, April 16-20, 1986

Production techniques for computer-based learning material

Jef Moonen, Jan Schoenmaker

1. Experiences

Experiences in the development of educational software in the Netherlands can be summarized as follows.

a) A great number of projects have been initiated from a research point of view. As a result a product-oriented procedure could hardly hold ground.
b) Approaches which are supported by one individual, mostly have to incorporate more people after a while, or fail as a result of the frustations of insufficient recognition, insufficient compensation for investments, or insufficient perspectives with regard to the work which had been carried out.
c) In a team-oriented approach, it appears that the communication between teams of teachers and teams of programmers forms a large obstacle. Both groups speak different 'languages' and do not readily understand the problematics of the other.
d) Within a team of teachers, it is difficult to reach agreements in relation to a common pedagogical approach.
e) There is a considerable shortage of professional designers of educational software. Even experienced teachers have great difficulty in achieving originality in the pedagogical design. In general, however, it is considered that a wide experience in education is a necessary condition to be a good designer, but it is also apparent that this characteristic is not sufficient in itself.
f) Although it is conceived that the production of educational software has to be a multidisciplinary approach between specialists, one hardly takes into account experiences already known for years in the computing profession, and from what is known as essential in a project approach.
g) The direct development of educational software on target-machines is hindered, at least as far as the limited capacity of those target-machines (8-bits) is concerned. This restriction affects both the quality of the final product, the time investment necessary, and the possibilities to convert to other target-machines.
h) Discussions about using general programming languages, authoring languages or authoring systems are still going on.
i) Publishers need support from the government (financially, or through policy decisions) in order to reduce the risks they encounter when introducing their educational software to the commercial market. Because of the uncertainty about the profitability of these kinds of activities, it is questionable whether this support will become necessary on a temporary basis or to create support on a more permanent basis.
j) Educational software is just like any other educational material: it gets out of date. None of these materials will remain available for an indefinite period of time. It will be necessary to adapt or modify this material, forced by changing circumstances in curricula and new

technical developments. In the software industry, it is known that 30-60% of the total costs of the product comprise the 'maintenance' of programs.

2. Consequences

Neither a strict individual approach, nor a strictly separated procedure of teams of teachers and teams of programmers, offer the real solution. The individual approach is impossible to maintain if one aims at large-scale and professional production. The second approach has the disadvantage, as a result of lacking standards and concrete specification-forms, that the communication between both teams becomes extremely difficult.

Because educational software is a relatively new phenomenon in education itself, it is not easy to determine in advance, the actual desires of the potential user. When the completion of the final product is almost reached, the potential user then realizes more so, what is desired and what is not. In the stage of development, this means that moments of evaluation must be continuously taken into account, which again leads to the fact that the entire process becomes distinctly iterative, with as participants the developers (pedagogical designers, system designers and programmers) and the target-groups (teachers and pupils).
At this moment (at least in the Netherlands), production strategies are as follows:

	higher education	secondary education	primary education
individual approach	x	x	x
team approach	x	x	
project approach	x		
industrial stage			

In the individual approach, the whole project is run by one person who designs, developes, does the coding and evaluates. In the team approach, different teams are responsible for the pedagogical design and the system-design and coding. The project approach also makes use of different teams but the emphasis is layed upon on the use of professional project-management techniques.
In the industrial stage, the production of educational software has to become an activity which is profitable for the publishers (book publishers, software houses, computer manufacturers). As can be seen in the outline above, none of the educational sectors have yet achieved this stage.
As a strategy to reach this stage, one has to concentrate firstly on the project approach. This will set the scene for a successful industrial activity.

Besides this organizational problem, there is also a problem as regards content that will be of great influence on the development of educational software. From an educational point of view it is necessary to produce educational software that can be integrated into existing curricula. This can be achieved by presenting software as a content-oriented supplement to

different parts of textbooks. Another possibilty is the use of software as a tool which can operate throughout the entire curriculum (i.e. a wordprocessor). One must realize that it is common in the educational community to be able to choose between different textbooks. The same attitude will be the case in relation to educational software. These aspects directly concern each of the subject-areas, and may even include various sorts of applications of computers in education. Because of the high costs involved, one could question whether this kind of opportunity - to choose between different packages - will ever come up.
A solution to this problem is to make educational software easily adaptable to new situations, by creating possibilities for teachers/users to modify existing software. A number of strategies can then be followed:

- creating so-called 'semi-products'; these are programs with a fixed general structure and which have to be supplemented with relevant text, graphics, and illustrative material by the teacher;
- products which can be modified by teachers using specialized user-friendly software tools.

At the same time, this approach could offer a solution in situations in which used educational software has come out of date.

Solutions as mentioned above, have not yet been found. Meanwhile, developments are in a transitional phase, from 'grassroot activities' to industrial approaches. This is an extremely critical phase, because there is great pressure to accelerate this transition process. Nevertheless, it is essential that both teachers and pupils remain involved in the development process of educational software.

3. Methodologies of educatonal software development

As guiding tools for the support of software development computer science knows among others:

a) Project management methodologies: including planning, financial estimates, organization.
With the assistance of these methodologies, projects are divided into phases which are evaluated separately.
b) System development methodologies: these are methodologies that can be used as step by step means, from problem definition to the design of a programme. Methodologies are often specific for the application area.

In order to formulate methodologies for the development of educational software it is necessary to specify the term more clearly.

Educational software is a combined name for a broad scale of programmes used in education.
In order to gain a clear vision of the division relevant for the determination of specifications for software, our approach is based on the organization of education.
Distinctions are made for three levels of educational organization:

a) The macro level: the level of school organization that is of direct importance for the school management and administration personnel.
b) The meso level: the level of the teacher, whereby a distiction is made in management (management of learning materials and item banks), control (item scoring and analysis, choice of learning activities), decision support (advices with respect to the prescription of learning activities).

c) The micro level: the level of the educational learning process, whereby teacher, student and computer programme, in some way interact. A three part division based on these interactions is relevant for the software to be developed:
   c.1) The computer as a learning environment: here the programme is a tool for the student.
   c.2) The computer as a learning device: here the teacher has transferred a part of the teaching process to the programme. These are the more traditional CAI-programmes.
   c.3) The computer as teacher: here the learning material, instruction strategies, formative and summative evaluation elements are housed in one programme. This means that the student learns without direct interaction with the teacher.

4. Software specifications and tools

The educational applications on macro and meso levels are to be generated for a large part with methodologies and tools as they occur in general computer science.
Considering the strong procedural character, administrative applications in education - whereby database management systems are used - and applications as item analysis can be developed with the assistance of higher programming languages and methodologies from the technical, scientific or administrative automation.
As opposed to this, the micro level is very specific for education. The functions that are of importance here are related to the presentation of text and graphic examples, pattern recognition in respect to the analysis of (re)actions of the user and the determination of the "flow of control", i.e. the progress of a student in relation to the programme.

In the case of use of computers as a teacher, this progress is determined by the learning material, the student characteristics and instruction strategies. These strategies are chosen dependently of the available learning material and student characteristics. These kind of programmes have the most difficult structure, because there is no general agreement on the educational models needed. Moreover, the complexity is even greater when we demand that learner-controlled strategies must also be possible. In this case we are dealing with the domain of the intelligent CAL.

Much experience has already been made with respect to the computer as learning device, which has also resulted in the development of specialized tools: drivers, authoring systems and languages.
The lack in flexibility by making use of these tools, has had as a result that many developers prefer giving a well structured higher programming language as PASCAL, where in procedures can be developed that support the implementation of specific CAL functions.
Besides these technical considerations there are also two actual arguments that determine the choice of these last solutions:
a) The growing interest for the computer as learning environment. Learning environments do not usually fit into the structure of authoring systems.
b) The interest for the integration of different applications, for example, computer-managed instruction with computer-assisted instruction.

The discussion about the benefit of different CAL tools has continued for a long time. However, the question is whether the choice of a certain tool, in the sense of a coding tool, is the most important problem. If we look at system development methodologies we must establish that the coding level comprises of only one aspect in the whole development process. Defining the functional and system design and choosing between a linear approach of the development process or a more iterative approach, are two other, even more important, aspects in this process.

5 An ideal solution?

A workout of the previous leads to splitting up of development process into four phases.

Phase 1

In phase 1 is described for what purpose the programme should serve inside the new (organizational) situation. In relation to the educational aspect, an analysis is made of the function the programme will fill. The distinct value of the use of the computer has to be made clear in this phase. We describe these activities as curriculum analysis where, as well as the analysing of the learning material, organizational changes in relation to teacher-student-computer, are described. In the curriculum analysis, teachers, pedagogical and subject-area specialists, and in advisory sense, an educational technologist, are involved. The education technologist (or media specialist) fulfills a bridge function between education and computer science. His advice here is related to media selection.

Phase 2

A description of the variety of information structures, which processing processes take place: in short a description of the functional aspects of the different subsystems. Translated to CAL aspects: the kind of subsystems to be used such as, learning material modules, data based on students' performance, diagnostics, drawing of conclusions in relation to the flow of control, subsystems with instruction strategies. This can be termed a global pedagogical design. This design is made by the education technologist.

Phase 3

Here is recorded how the subsystems are realised. The interactive dialogues, characteristic for CAL, CAL are worked out here to screen designs and dialogue forms. The education technologist and system designer are involved here. This can be termed the detailed pedagogical design.

Phase 4

Here is described, for example in pseudo language, what the algorithms are, how data structures should be implemented, etc. This is the system design by which the coding takes place.

The motive for this approach is, on the one hand, that the development process can be made more efficient when at each phase there is a direct feedback to the aims (objectives) of that phase. On the other hand, the successive continuation of these four phases, does not create an optimum development strategy. Instead the developers must be given the opportunity to examine former phases before determining the next.

This is of special importance when applying in education. It is well known that the essentials of CAL lie in the interaction that is then made possible. On the other hand, the moulding of the man-machine interaction on paper is a roundabout process. In practise, we see that the teacher who has put his specifications for screen design and dialogue on paper, is often dissatisfied with what the programmer has made out of it. In a linear development process it will be difficult to correct this activity in phase 4 because phase 3 has already been determined. That is why, certainly when using computers in education on a micro level, development must take place according to an iterative process.

6. Development levels

As a consequence of what is said in paragraph 1 and 2, development of educational software can take place at three different levels.

Level 1: Development of experimental programs

The development of experimental programs occurs in laboratory-environments. That development is characterized by the use of computers in new kinds of applications.

These developments occur without direct feed-back to the potential target-group. The development generally comprises a research-oriented component, aimed at either the development tools used (hardware-facilities, programming languages), or aimed at learning or instruction-theories.

Level 2: Development of prototypes

This is the level of development which demands the most attention. Until present, most developments have taken place at this particular level. This occurred in either a strictly individual context, or in a more team-directed context. In most cases, attempts were made to commercialize the product after it had been obtained. This generally led to various additional difficulties with regard to conversion, and also extra costs. This made it less attractive for firms to start such a commercialization process.

Recently, it was thought that the development of educational software could be held in comparison with, for example, the development of administrative software. Here, there was the central concept that a complete specification of the product could be achieved on the basis of the requirements of the user, after which the software-producer could carry out the programming-work. This situation does not (yet) apply in education. It is not generally realized that the use of computers in education still shows an unstable pattern. It is for this reason that it is rather difficult for the final users to indicate, at the beginning, exactly what is required. There is also the fact that available equipment is continually evolving, so that many more possibilities - even in the field of didactics - become at one's disposal. All users of computers in education find themselves in the process of learning which continuously leads to new, modified views and demands.

The consequences of this are that the development process of an educational product must be set up in an iterative fashion. Within each phase, the possibilities of interaction between developers and representatives of the target-group are of essential importance. During a specific length of time, the opportunity to make amendments and modifications must be available. To stimulate the efficiency of this process, one must make use of advanced hardware and software tools.

It is quite obvious that these circumstances do not adapt well to a more industrial approach. The first step in the direction of a solution is therefore the separation of the development levels. The level described above - known as 'level 2' - must provide a tested prototype which is well documented and accompanied by detailed specifications. At this level 2, the four phases as were mentioned in paragraph 6 apply, followed by the evaluation of the prototype via pilot-projects.

The most significant problem at this level, is the smooth organization of this kind of co-operative activity.

Level 3: Development of consumer programmes

After the final specifications of the desired product have been established, the production of the software should be initiated in an economically effective manner.

The interaction with the target-group has already been completed in the previous stage and is no longer of significance. Because the entire specifications are now available, one can now concentrate on the search for the most efficient developing tools. Also, from the beginning of this activity, the various target-machines for which must be developed can be taken into account.

The level 3 development preferably takes place in an industrial environment. In such an environment, the correct atmosphere and experience is present, which is required in order to think and act in strictly economical terms. At the same time, a direct contact can be made with those firms which will introduce the end product to the market.

7. Approach in the Netherlands

As in many other countries, the Dutch government too, has taken the initiative of stimulating the introduction of computers in education. This policy commenced in 1981 with the foundation of the Advice Committee for Education and Information Technology (AOI) and of the Centre for Education and Information Technology (COI).
The first policy-recommendations by the government appeared in 1982, and provided for a five-year project (1983-87) in which all sectors of education were involved. Following discussions in parliament, this project was further developed in the so-called 'Informatics Stimulation Plan' (INSP), which was established in june 1984. Again, this INSP comprised a five-year plan (1984-88), directed towards education, research and the market-sector. The educational part of the project has a total budget of 290 million Dutch Guilders, and stands in relation to all sectors of education (i.e. from primary education through higher vocational education, but excluding university education). The objective of this project is to acquaint, within no more than 5-10 years, all pupils with the possibilities of the computer and to use the computer in education as a tool. This project must also contribute to the stimulation of the Dutch industry by providing well-skilled employees, the stimulation of the acceptance of the

new information technology in future generations, and to put education to use as a potential market for products such as hardware, software and related services.

The INSP is concentrated around five clusters of measures:

Cluster I: the construction of a national and regional infrastructure for the development and distribution of educational software;

Cluster II: carrying out projects for each school sector in education, in which, per sector, distinct priorities with regard to policies are established;

Cluster III: measures to be taken within the framework of the in-service training of teachers, and provision of information;

Cluster IV: measures within the framework of the pre-service training of teachers;

Cluster V: measures to be taken within the framework of educational research.

In addition to the INSP, a so-called NIVO-project (New Information Technologies in Secondary Education) was started in 1985. This project aims towards broad introduction of computers in the secundary educational sector. It's purpose is to provide hardware (at least 8 to 10 computers per school, 16-bits with MS-DOS) to every school in this sector by the end of 1988, to train (at least three) teachers per school and to provide every school with a basic software package. The NIVO is a co-operative project between the government, educational organisations and industry.

In the INSP the priorities have been established with regard to the inservice training of teachers, projects in vocational education and to the initiation of a national infrastructure for the benefit of software development and distribution.
Characteristic of the INSP is the co-ordinated approach by means of project-oriented policy within the Ministry of Education and Science, and an approach by means of project-management. The stipulation of the policy to be followed is carried out by the Project Staff Education and Information Technology (PSOI). This group has been affiliated to the Ministry. The project-management (i.e. the production of, or the initiaton of planned activities, advice, guidance, and evaluation) is carried out by the COI, at least as far as the construction of an infra-structure and the projects in special-, primary- and secondary education are concerned. For this objective, the COI has established and made provisions for project-management teams. Other external management teams have been assigned to the sector for vocational education, the training of teachers and the educational research.
Mostly significant for this article are the activities within the framework of cluster I.

As was mentioned before, the development of educational software can be carried out at three levels, with a major problem concerning the smooth organization of activities at the level 2. To solve this problem, some actions have been taken in the Netherlands:

- Development centres and development groups have been established.
- A regional support structure has been established.
- A clearinghouse for educational software has been established.

To support the level 3 development, a facilitary co-operation has been created in the context of the NIVO-project.

8. Development centres and development groups

The Dutch government has created three development centres, each one specially aiming at one particular sector of education. These development centres are affiliated to a number of teacher-training institutes. These centres have been entrusted with the (partial) task of developing educational software. Two main tasks can be distinguished:

- (involvement in) the development of software within the framework of projects from the annually established global plans of action in the INSP;
- the canalization and stimulation of good ideas for software development from the field of education.

The centres support an agreed methodological approach in software development, concentrated towards aspects such as: technical quality (use of certain (system) software and software tools), modularity, portability, documentation, maintenance. They can also provide support in the definition phase of a project.
The development centres therefore, play a central role within the framework of the 'level 2' development of educational software. Supplementary, they are given possibilities to be able to react flexibly to initiatives from the field of education, both by advising, offering programming assistance and so forth.
The project-manager of the sector in question, is responsible for the activities of the development centres, as far as the content-oriented aspects of projects is concerned. The methodological aspects are co-ordinated by the COI through the project-manager of cluster I.
In contrast, the development groups focus their activities towards the coding aspects of the development process. But they can also be asked, when necessary, to become involved in the design process. Development groups are founded and used as is needed. Mostly they are available within existing institutes.

9. Relations with other organizations

Apart from the importance of a common methodological approach, the organization of the interaction between the various groups and activities is essential.
Three kinds of activities play an important role in this respect:

- the activities of the regional centres for education and information technology;
- the activities of the clearinghouse for educational software;
- the activities of users associations.

The Regional Centres for Education and Information Technology (RSOI)

For the support and guidance of the introduction of computers in Dutch education, 28 RSOI's have been created. They are affiliated with teacher-training institutes and school-guidance institutes. The basic principle for this action was to create a near-school approach. Supplementary, there were two other considerations:

- to realize a regional concentration of the scarce expertise in this field;
- to concentrate a number of important aspects which are of fundamental

significance in the introduction of computers to education: in- and pre-service training facilities; co-ordination of information services about the use of computers in education; demonstration of educational software; initiation, guidance and support of projects; contributions to the development of educational software.
Three of these RSOI's are also appointed as development centres. Supplementary to the activities of these development centres, the RSOI's will have to create possibilities for teachers to come in touch with educational software in order to stimulate development of new ideas about the kind of educational software they want.

The Clearinghouse for Educational Software
To assist education in determining the quality of educational software, the government has started the INKON-project. INKON is the abbreviated form for Inventory, Assessment and Information distribution of educational software. The project is carried out bij the COI in co-operation with the CRL (Central Registration of Learning Aids) which is affiliated to the National Institute for Curriculum Development (SLO).
The CRL will chiefly direct itself to the registration and the objective description of educational software. The COI will concentrate its activities on the qualitative description and assessment of educational software.
Apart from being characterized by the provision of various services towards teachers, the INKON-project also has an additional value. It is expected that the project, in course of time, will increasingly evolve in research-activities, formulation and intensification of criteria-requirements which that educational software has to meet. This information can then be used by the developers of educational software and in the training of teachers. This results in a process by which the quality of the educational software - and the integration of that software in the curricula - can be structurally improved.

The Users Associations and their Special Interest Groups
There are many different users associations in the field of computers and education in the Netherlands. In the past, these organizations have played a very significant role as pioneers in the support and introduction of computers in education. With the introduction of the INSP, a number of these activities have been taken over by organizations supported by the government. However, this does not mean to say that these associations have now become superfluous. Sufficient tasks and activities have still remained to which they - and in particular their special interest groups - can contribute in the evaluation of software.
It is known that a valuable assessment is not quite complete without the experience of it's use in a classroom-situation. That data and information can particularly be collected by teachers who have gained experience in the use of computers. The special interest groups form the basis on which this element of the software-assessment can take place.

10. The Training of Developers

Developers of educational software demand contributions from both teachers and programmers, both in the role which fits their capabilities best of all. Until now, the training of teachers has hardly paid any attention to the use of computers as educational aids. This means that, in all trainingschemes for teachers in schools, courses must be organized which illustrate these specific aspects. There is also a great demand for project-managers who can fulfill the link-function between teams of teachers and programmers. These persons should be well informed of the educational-didactical-,technological- and program-technical aspects of development work. At the present, in the Netherlands there is only one training-scheme available for those wishing to prepare themselves for this particular specialism. These courses are provided by the Twente University of Technology, the same university to which the COI is affiliated.

11. The role of the Centre for Education and Information Technology (COI)

The COI plays a central role as co-ordinating institute in the methodological support for the development centres and groups. The COI also co-ordinates the activities of the regional centres and is responsible for the clearinghouse for educational software.
Within the COI, there is a development department which is entrusted with the production of experimental programs (=the level 1 development) and of prototype programs (=the level 2 development). The COI has also been appointed by the government as the Dutch national information point for education and information technology in the context of an international network of such centres, initiated by the EEC.

12. Postscript

At the end of the 1970's, the use of computer-based learning had been going on for more than ten years, but a definite breakthrough in education had not yet been achieved.
One could even say that computer-based learning was saved from extinction at that time by the introduction of the microcomputer. A new wave of enthusiasm arose, and the whole process of considering the use of computers in learning situations started all over again.
Now, 5-6 years later, one is confronted with the same problem: will there be a real breakthrough for computer-based learning, or not? The situation is again very critical. Some think that one needs more of the new technological wonders (videodisc, networking) to make it really happen. Reality is that a breakthrough will only be reached when it can be shown that computer-based learning results in an increase of productivity of our educational system. That increase in productivity can be achieved in two additional ways: achieving the same goals in education as before, but in less time, and/or, achieving new goals in education in the same period of time. Results show that in both circumstances, computer-based learning can help to achieve this objective.
However, the efficient use of computers in education depends entirely on the availability of good educational software. Until recently, such software has only partially been available. Much more software must, therefore, be made generally available. This means that the presently established forms, with regard to the development of such software, require careful revision. The end-product must be able to be produced in an efficient, industrially-oriented fashion.
As was said before, the organization of the entire development process - from experimental programs to prototype and final product - carried out coherently so that it is smoothly compatible, whereby each of the participants can influence activities at the right time and in an efficient manner, is the principle problem with which one is confronted.

References

Adcis, Journal of Computer-Based Instruction, Special Issue on Authoring Tools, volume 11, number 3, Summer 1984.

Fox, J., Rushby, N., Guidelines for Developing Educational Computer Programs, Computers and Education, volume 3, 1979, 35-41.

Gordon, R., Guidelines for the Design of Computer-Based Instructional Materials. Atlanta, Georgia : Academic Information Systems, 1985.

Kearsley, G., Microcomputer Software : Design and Development principles. Journal of Educational Computing Research, volume 1(2), 1985, 209-220.

Kontos, G., Instructional Computing : In search of better methods for the production of CAI lessons. Journal of Educational Technology Systems, volume 13, number 1, 1984-85, 3-14.

Le Corre, Y., Schwartz, J., Software and courseware for a multimedia educational environment. Organisation for Economic co-operation and development, Centre for Educational Research and Innovation, CERI/NT/84.05, Paris, 1984.

Magel, K., Software Engineering Principles for Courseware Development. AEDS Journal, volume 13, number 2, winter 1980, 144-155.

Merrill, D.M., Computer Aided Instructional Design. Dean Lecture, presented at the 26th ADCIS Conference, March 25-28, 1985, Philadelphia.

Moonen, J., Schoenmaker, J., Ontwikkeling van educatieve software in Nederland. Memo nr. 85.19 (in Dutch). Centre for Education and Information Technology, P.O. Box 217, 7500 AE, the Netherlands.

Moonen, J., The Dutch Informatics Stimulation Plan: contributions of the Centre for Education and Information Technology. Educational Technology, nov. 1985.

Roblyer, M.D., Instructional Design versus Authoring of Courseware : some crucial differences. AEDS Journal, volume 14, summer 1981, 173-181.

Walker, D.F., Hess, R.D., Instructional Software, Principles and Perspectives for Design and Use, Belmont, California : Wadsworth Publishing Company, 1984.

Watson, D.M., A model for the production of CAI material. Computers and Education, volume 7, number 3, 1983, 167-176.

Weinstock, H., Bork, A., Learning Physics and Mathematics via Computers. To be published by Springer, 1986.