DOCUMENT RESUME

ED 455 818 IR 020 774

AUTHOR	Nieveen, Nienke; Gustafson, Kent
TITLE	Computer-Based Tools To Support Curriculum Developers.
PUB DATE	2000-10-00
NOTE	12p.; In: Annual Proceedings of Selected Research and Development Papers Presented at the National Convention of the Association for Educational Communications and Technology (23rd, Denver, CO, October 25-28, 2000). Volumes 1-2; see IR 020 712.
PUB TYPE	Reports - Descriptive (141) -- Speeches/Meeting Papers (150)
EDRS PRICE	MF01/PC01 Plus Postage.
DESCRIPTORS	*Computer Assisted Instruction; Computer Oriented Programs; *Computer Uses in Education; *Curriculum Development; Educational Development; Educational Equipment; Educational Practices; *Educational Technology; Foreign Countries; Instructional Design; *Instructional Development; *Instructional Materials

ABSTRACT

Since the start of the early 1990s, an increasing number of people are interest in supporting the complex tasks of the curriculum development process with computer-based tools. "Curriculum development" refers to an intentional process or activity directed at (re)designing, developing, and implementing curricular interventions in schools, colleges, or corporate education. The term "curricular intervention" serves as a common denominator for curricular products, programs, materials (varying from teacher and student/trainee materials), procedures, scenarios, processes, and the like. A key issue and major challenge in curriculum development is how curricular interventions should be developed in order to achieve a satisfying balance between the ideals of a curriculum change and their realization in practice. This paper provides background information on the roots of computer-based tools for curriculum developers, provides an overview of those tools currently available in the United States and abroad (especially Australia and The Netherlands) and describes probable future trends. (Contains 19 references.) (Author/AEF)

Computer-Based Tools to Support Curriculum Developers

Nienke Nieveen
University of Twente, The Netherlands
Kent Gustafson
The University of Georgia

Since the start of the early 90's, an increasing number of people are interested in supporting the complex tasks of the curriculum development process with computer-based tools. 'Curriculum development' refers to an intentional process or activity directed at (re) designing, developing and implementing curricular interventions in schools, colleges, or corporate education. The term 'curricular intervention' serves as a common denominator for curricular products, programs, materials (varying from teacher and student/trainee materials), procedures, scenarios, processes, and the like. A key issue and major challenge in curriculum development is how curricular interventions should be developed in order to achieve a satisfying balance between the ideals of a curriculum change and their realization in practice. This paper provides background information on the roots of computer-based tools for curriculum developers, provides an overview of those tools currently available in the USA and abroad (especially Australia and The Netherlands) and describes probable future trends.

1. Roots of Computer-Based Tools for Curriculum Developers

Many designers make use of tools or job aids providing design support in their daily work. A job aid is a collection of several kinds of conceptual or procedural information (for instance: glossaries of terms, guidelines, decision tables, checklists) that supports work. Over the years, many of these job aids have been combined into handbooks for instructional designers. According to Rossett and Gautier-Downes (1991), job aids may have major advantages for their users, such as:

- they are available at the moment individuals feel a need for them;
- they increase the chance that an individual has up-to-date information to perform a task, especially in case of a very complex and infrequently performed activities;
- they prompt individuals through difficult processes and decisions.

Computer-based technologies have not only influenced the domain of job aids. They have also impacted other types of external support, such as communication and training. Today's computer and networking facilities can even integrate these types of performance support. Instead of separately providing different ways of support to individuals, an electronic performance support system (EPSS) provides integrated information, advice and learning opportunities to improve user performance (Gery, 1991; Raybould, 1995). EPSSs are given many names, such as performance support tools (Carr, 1992), (integrated) performance support systems (Geber, 1991), embedded performance support systems (McGraw, 1995). But regardless of the terminology used, they all refer to a computer-based system which provides integrated support in the format of any or all of the following: job aids (including conceptual and procedural information and advice), communication aids and learning opportunities to improve user performance.

With the increase of the number of computers used at work, a growing number of computer-based tools for designers and developers in education and training have been developed at various places around the world. More efficient development processes, more effective learning programs and increasingly competent designers are all potential benefits that make these tools attractive to many designers and their managers. In addition to these assumed advantages, some criticism may also be found in literature. Firstly, the potential supportive role of these tools should be carefully judged. For instance, as the consultation of an EPSS is usually largely self-directed, certain capacities of the individuals who use the EPSS are required: they need to know what they do not know; value a high degree of control and be able to evaluate the quality of the information, to name just a few characteristics. Not every individual possesses these cognitive as well as affective characteristics. Moreover, Clark (1992) suggests that in many dynamic work environments, individuals do not have the time to look for information in the job aids or learn from the CBT component of an EPSS. Developers of EPSSs and organizations who consider using these tools should take these potential problems into account.

2. Overview of Available Tools

In order to get an overview we examined the following available tools: GAIDA, QIPP EPSS, PLATO, MediaPlant, SimQuest, CASCADE-SEA, TeleTOP DST, Mercator, IDXelerator, AGD and GTE. In box 1 each tool is briefly introduced. For more information please refer to the references of each tool.

GAIDA: Guided Approach to Instructional Design Advising (GAIDA) offers on-line elaborated guidance for the application of Gagné's nine events of instruction (e.g. Gagné, Briggs & Wager, 1992) to the design of interactive courseware and other instructional materials. GAIDA was developed for novice instructional developers at the Air Force Research Laboratory (Gettman, McNelly & Muraida, 1999).

QIPP EPSS: This tool supports the application of a new development methodology (called Quality Information Products Process) for designing technical documentation at NCR. The system specifies phases and work activities of the instructional design process and provides job aids for each activity. Technical writers and instructional designers of NCR belong to its main target group (Jury & Reeves, 1999).

Plato Courseware Development Environment: Plato is an authoring tool to support the design and development of courseware including tutorials, simulations and constructivist learning environments. The systems can be used by non-programmers to author instructional activities by customizing objects that are copied from a library and assembled into completed multimedia components. For each phase of the process there are job aids accessible to all members of the design team (Preese & Foshay, 1999).

MediaPlant: This is a development environment that facilitates the production of complex cross platform learning environments. The development program is used to construct and test the learning environment, which is then distributed with the runtime program (Wright, Harper & Hedberg, 1999).

SimQuest: This is an authoring environment for creating learning environments that combines simulations with instructional support that helps learners in the process of discovery learning. An author (teacher) creates a learning environment by adapting building blocks selected from a library. The author gets support from an on-line help system, a wizard and an advice tool (de Jong, Limbach, Gellevij, Kuyper, Pieters & van Joolingen, 1999).

CASCADE-SEA: CASCADE-SEA (Computer Assisted Curriculum Analysis, Design and Evaluation for Science Education in Africa) aims to support curriculum development within the context of secondary level science and mathematics education in sub-Saharan Africa. One of its components is called "lesson builder". This component has been designed to help teachers make paper-based exemplary lesson materials. Based on input of the user, Lesson Builder's prompt the program to generate a draft (McKenney, 1999).

TeleTOP DST: The TeleTOP Decision Support Tool is a WWW-based environment that helps instructors become aware of technical possibilities for their courses and helps them to see how these could be integrated in an educationally useful way (Collis & de Boer, 1999).

Mercator: This system supports the design, production and delivery of course materials. On the one hand it helps to design and produce the material, and on the other hand it helps students to select specific materials and supports the actual delivery in a printed and/or electronic mode (Valcke, Kirschner & Bos; 1999).

IDXelerator: This authoring system automatically generates the instructional interactions required for the student to acquire a specific kind of knowledge or skill. The system has an author view (that supports the author) and a student view for delivery of the instruction (Merrill & Thompson, 1999).

AGD: Atelier de Génie Didactique provides a pedagogical design assistant to content experts in companies and university teachers in preparing lessons for distance learning settings (Paquette, Aubin & Crevier, 1994; Spector; 1999).

GTE: The Generic Tutoring Environment is focused on providing support for designing intelligent tutoring systems. The primary task of such systems is to integrate instructional knowledge in the system in a way that allows the system to adapt to learners just as expert teachers do (van Marcke, 1998; Spector, 1999).

Box 1. Brief description of computer-based tools to support curriculum developers

From the short overview in box 1 it becomes clear that available computer-based support tools for developers in the field of training and education can be classified in many ways. To analyze the tools in more detail we used a framework with the following set of attributes:

A. Type of output:

- Curriculum level (few lessons, product, course, collection of courses)
- Characteristics of results (target group, form, extensiveness)

B. Purpose and evidence of benefits:

- Purpose (transfer of knowledge and skills, improved task performance, organizational learning
- Evidence of claimed benefits (validity, practicality, effectiveness)

C. Type of development process supported and any underlying theory:

- Paradigm for engaging in education and training benefit (instrumental, communicative, pragmatic, artistic)
- Elements of systematic approach (analysis, design, development, implementation, evaluation)
- Underlying teaching/learning theory (behaviorism, cognitivism, constructivism)

D. Task support:

- Types of support (communication aids, job aids, training aids)
- Adaptability of support (outside the tool, inside the tool, inside networked tool, closed)

E. Intended user group:

- Expertise of user group (professional designer (ISD), subject matter expert, teacher, learner)
- Scope of intended user group (various organizations, specific organization)
- Computer experience (low, high)

The framework was used to examine the tools for developers mentioned in box 1 (see Table 1). The framework should be judged solely on its utility as a schema for examining and selecting from among tools and is not intended to be a scientifically valid taxonomy. Moreover, it should be noted that the analysis is based on limited information provided by the developers of the tools and is not based on personal experience with most of the tools. For an individual who wants to select a tool based on the information in the framework, technical issues such as needed operating system, software; and hardware would be critical to consider as well. The same is true for other issues such as time needed to learn the tool; time needed until a user starts to be productive with the tool; the costs of a tool; and its general availability.

		GAIDA	QIPP	PLATO	MediaPlant	SimQuest	CASCADE-SEA
A	Curriculum level	Product	Product	Product	Product	Product	Course and lesson
	Characteristics of results	- Learner-based - Site-specific - Paper-based/ Computer-based	- Learner-based - Site-specific - Paper-based	- Learner-based - Generic - Computer-based and Web-based	- Learner-based - Site-specific/ Generic - Computer-based	- Learner-based - Site-specific/ Generic - Computer-based	- Learner-based Teacher-based - Site-specific - Paper-based
B	Purpose of tool	- Better transfer - Performance improvement	- Better transfer - Performance improvement	- Performance improvement - Organizational learning	- Performance improvement	- Better transfer - Performance improvement	- Better transfer - Performance improvement - Organizational learning
	Evidence	Results available	Results available	Results available	IIA	IIA	IIA
C	Development paradigm	Instrumental	Pragmatic	Pragmatic	Pragmatic	Pragmatic	Pragmatic
	Elements of systematic approach	- Design	- Analysis - Design - Development - Implementation - Evaluation	- Analysis - Design - Development - Evaluation	- Design - Development	- Design - Development	- Analysis - Design - Development - Implementation - Evaluation
	Teaching/learning theory	Behavioristic	Cognitivistic	Cognitivistic	Constructivistic	Cognitivistic	Cognitivistic
D	Types of support	- Job aid: toolbox	- Job aid: toolbox, do-it-yourself kit	- Job aid: toolbox, do-it-yourself kit - Communication aids	- Job aids: do-it-yourself kit	- Job aids: do-it-yourself kit	- Job aid: toolbox, do-it-yourself kit
	Adaptability of support	Closed	Inside networked tool	Inside networked tool	Inside tool	Inside tool	Inside tool
E	Type of user group	- SME's	- Designers	- Team with range of roles	- Designers - Learners	- SME's - Teachers	- Teachers
	Scope	Various organizations	One organization	Various organizations	Various organizations	Various organizations	Various organizations
	Computer experience	Low	Medium	High	Medium	Medium	Low

Table 1 continues

		DST	**Mercator**	**IDXelerator**	**AGD**	**GTE**
A	Curriculum level	Course	Course	Product	Course	Course
	Characteristics of results	- Learner-based, Teacher-based - Site-specific - WWW-based	- Learner-based - Site-specific - Paper-/Computer-/ Web-based	- Learner-based - Site-specific - Computer-based	- Learner-based, Teacher-based - Site-specific - Computer-based	- Learner/based - Site-specific - Computer-based
B	Purpose of tool	- Performance improvement	- Performance improvement	- Performance improvement	- Better transfer - Performance improvement	- Performance improvement
	Evidence	Results available	Results available	IIA	IIA	N/A
C	Development paradigm	Pragmatic	Instrumental	Instrumental	Pragmatic	Instrumental
	Elements of systematic approach	- Design	- Design - Development	- Development	- Analysis - Design	- Analysis - Design - Development
	Teaching/learning theory	Cognitivistic	Cognitivistic	Behavioristic	Cognitive	Cognitivistic
D	Types of support	Job aid: automatic washing machine	Job aid: do-it-yourself kit	Job aids: automatic washing machine	Job aids: toolbox	Job aid: toolbox, automatic washing machine
	Adaptability of support	Closed	Inside tool	Closed	Closed	Closed
E	Type of user group	- Designers - Teachers	- Designers	- SME's	- SME's	- Teachers
	Scope	Specific organization	Various organizations	Various organizations	Various organizations	Various organizations
	Computer experience	Low	Medium	Medium	Medium	Medium

Note: IIA = Insufficient Information Available

Table 1. Conceptual framework

7

8

Here we provide a brief interpretation of the analysis.

A. Type of Output

In general, the use of most of the tools results in instructional products or courses that are computer-based and have learners as their main target group. Although it might be argued that many, if not all, of the tools could be used for creating many different forms of instruction, they do appear to lend themselves best to one or only a few forms of output. None of the systems seem to focus primarily on the development of interrelated collections of courses. Only a few of them may lead to paper-based materials (GAIDA, CASCADE-SEA and Mercator), and/or web-based materials (PLATO, DST, and Mercator) and only three tools support the development of teacher-based materials (CASCADE-SEA, DST, and AGD).

B. Purpose of the Tool

Generally speaking, all tools are designed with the expectation of improving the performance of developers of training and education. In describing the tools, some authors anticipate that their tool will lead to a better transfer of knowledge and skills to the actual task performance because it makes the rationale of the tool (and thus that of the design process) explicit (cf. QIPP, SimQuest, CASCADE-SEA, and AGD). Others expect that the use of their tool will lead to organizational learning, since the tool invites users to make newly acquired information available to their whole organization (PLATO and CASCADE-SEA). However, it should be noted that most of these claims appear to remain assumptions, since few data are available that demonstrate the actual benefits of these tools.

C. Type of Process Supported and Underlying Theory

Many tools that were analyzed make extensive use of a prototyping approach, which refers to a pragmatic paradigm. This is, with the exception of GAIDA, Mercator, IDXelerator and GTE: these tools seem to be based on a paradigm that follows a more linear completion of the instructional design process.

When looking closer at the underlying elements of the systematic approach to development of education and training it appears that two tools (QIPP and CASCADE-SEA) intend to support the designer during the entire process (from analysis through evaluation). All other tools support specific elements of the process, of which design and development get the most attention. When reviewing the tools with respect to the underlying teaching/learning theory, it appears that most tools are based (to various degrees) on a cognitivistic theory. Two tools seem to be based on a more behavioristic theory (GAIDA and IDXelerator) and one starts from constructivism (MediaPlant).

D. Task Support

All tools contain job aids to support users in their development activities. The metaphor of a toolbox and a do-it-yourself kit fits most tools. None of the systems that were analyzed have the ability to automate the entire instructional design process. In all cases, considerable human skills are needed to make effective instructional products and courses. It is noteworthy that none of the tools seem to include explicit learning facilities for designers who express a need for learning a specific design task. For novice designers, with the possible exception of GAIDA, the tools seem to count on an informal learning process of learning-by-doing or some form of external assistance.

E. Intended User Group

Generally speaking, it appears the designers of all tools started creating the tools with a specific organization in mind. In an overall view, the tools are intended for one or two of the following user groups: professional designers, subject matter experts, teachers and/or learners. This means that the support tools need to contain (parts of) team members' expertise that would have been needed in times when the tool was not available.

3. Trends in Computer Supported Curriculum Development

In looking ahead, there are several trends that will impact the form and substance of future computer-based tools for developing education and training.

Supporting a Constructivist Perspective on Learning

The increased influence of the constructivist perspective on learning is impacting the design of computer-based support tools. From this perspective, learning requires active construction rather than acquisition of knowledge by the learner. As a consequence, the teacher will increasingly become a designer of learning environments that support the construction of knowledge of the learners. Also, teachers or trainers increasingly fulfil design roles in the context of innovative projects, in which they participate, often emphasizing their own professional development.

Increasing Array of Tools

The first tools were almost all created to support only one or a few tasks related to the curriculum design process. Although there are now some tools that support many different tasks, none are completely adequate for all tasks on different types of projects. What we now see is expansion in several directions: there is an increase in the number of tools that attempt to integrate multiple tasks, however, at the same time the number of single purpose tools for highly specialized situations is also increasing. Continuing advances in computers, digital processing, and communication technology will all add to the demand for a complementary set of development tools and support new features related to future design efforts.

Supporting Teamwork

Team efforts are increasingly critical to large scale, complex projects, especially those that will result in technology-based instruction such as multimedia or web-based course. As a consequence, computer-based support tools may be extended with communication tools that facilitate collaboration. In addition, anticipation of how the curriculum intervention will be implemented is of growing significance during the design process.

Supporting Networks of Designers

As individual designers gain knowledge and skill in using the tools they can more readily share this knowledge with other members of the design community to prevent these insights and skills from being lost to others or not be otherwise leveraged in the organization. Based on today's database and networking technologies, effective computer-based infrastructures may be developed which makes knowledge sharing and knowledge management more possible.

For all of these reasons, we believe the future of computer-based support tools is very bright. The emergence and expansion of tool creation and use that we have witnessed over the last ten years will pale in comparison to what will happen during the next decade. We have no doubt that future tools will be as different from current ones as current desktop computers are from their predecessors of ten years ago. Continuing advances in computers, digital processing, and communication technology will both add to the demand for a complementary set of development tools and support new features we can only dream about today.

References

Carr, C. (1992). Performance support systems: The next step? Performance & Instruction, 31(2), 23-26.

Collis, B., Boer, W. de (1999). The TeleTOP decision support tool (DST). In J. Van den Akker, R.M. Branch, K. Gustafson, N. Nieveen, & Tj. Plomp. Design approaches and tools in education and training (pp. 235-248). Dordrecht: Kluwer.

Geber, B. (1991). 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.

Gettman, D., McNelly, T., & Muraida, D. (1999). The guided approach to instructional design advising (GAIDA): A case-based approach to developing instructional design expertise. In J. Van den Akker, R.M. Branch, K. Gustafson, N. Nieveen, & Tj. Plomp. Design approaches and tools in education and training (pp. 175-182). Dordrecht: Kluwer.

Jong, T. De, Limbach, R., Gellevij, M., Kuyper, M., Pieters, J., & Joolingen, W. van (1999). Cognitive tools to support the instructional design of simulation-based discovery learning environments: The SimQuest authoring system. In J. Van den Akker, R.M. Branch, K. Gustafson, N. Nieveen, & Tj. Plomp. Design approaches and tools in education and training (pp. 215-224). Dordrecht: Kluwer.

Jury, T. & Reeves, T. (1999) An EPSS for instructional design: NCR's quality information products process. In J. Van den Akker, R.M. Branch, K. Gustafson, N. Nieveen, & Tj. Plomp. Design approaches and tools in education and training (pp. 183-194). Dordrecht: Kluwer.

Marcke, K. Van (1998). GTE: An epistomological approach to instructional modeling. Instructional Science, 26(3/4), 147-191.

McGraw, K.L. (1995). Selecting appropriate functionality and technologies for EPSS. Innovations in Education and International Training, 32(1), 13-22.

McKenney, S. (1999). CASCADE-SEA: Computer assisted curriculum analysis, design and evaluation for science education in Africa. In J. Van den Akker, R.M. Branch, K. Gustafson, N. Nieveen, & Tj. Plomp. Design approaches and tools in education and training (pp. 225-234). Dordrecht: Kluwer.

Merrill, D. Thompson, B. (1999). The IDXelerator: Learning-centered instructional design. In J. Van den Akker, R.M. Branch, K. Gustafson, N. Nieveen, & Tj. Plomp. Design approaches and tools in education and training (pp. 265-278). Dordrecht: Kluwer.

Nieveen, N., & Gustafson, K. (1999). Characteristics of computer-based tools for education and training development: An introduction. In J. Van den Akker, R.M. Branch, K. Gustafson, N. Nieveen, & Tj. Plomp. Design approaches and tools in education and training (pp. 155-174). Dordrecht: Kluwer.

Paquette, G., Aubin, C., & Crevier, F. (1994). An intelligent support system for course design. Educational Technology, 34(9), 50-57.

Preese, F., & Foshay, Q. (1999). The PLATO courseware development environment. In J. Van den Akker, R.M. Branch, K. Gustafson, N. Nieveen, & Tj. Plomp. Design approaches and tools in education and training (pp. 195-204). Dordrecht: Kluwer.

Raybould, B. (1995). Performance support engineering: An emerging development methodology for enabling organizational learning. Performance Improvement Quarterly, 8(1), 7-22.

Rossett, A., & Gautier-Downes, J. (1991). A handbook of job aids. San Diego: Pfeiffer.

Spector, M. (1999). Intelligent support for instructional development: Approaches and limits. In J. Van den Akker, R.M. Branch, K. Gustafson, N. Nieveen, & Tj. Plomp. Design approaches and tools in education and training (pp. 279-290). Dordrecht: Kluwer.

Valcke, M., Kirschner, P, Bos, E. (1999). Enabling technologies to design, produce and exploit flexible, electronic learning materials. In J. Van den Akker, R.M. Branch, K. Gustafson, N. Nieveen, & Tj. Plomp. Design approaches and tools in education and training (pp. 249-264). Dordrecht: Kluwer.

Wright, R., Harper, B., & Hedberg, J. (1999). Visual support for authoring. In J. Van den Akker, R.M. Branch, K. Gustafson, N. Nieveen, & Tj. Plomp. Design approaches and tools in education and training (pp. 205-214). Dordrecht: Kluwer.

NOTICE

REPRODUCTION BASIS

☒ This document is covered by a signed "Reproduction Release (Blanket) form (on file within the ERIC system), encompassing all or classes of documents from its source organization and, therefore, does not require a "Specific Document" Release form.

☐ This document is Federally-funded, or carries its own permission to reproduce, or is otherwise in the public domain and, therefore, may be reproduced by ERIC without a signed Reproduction Release form (either "Specific Document" or "Blanket").

EFF-089 (9/97)