STORYIST
CONCEPT DEVELOPMENT TOOL FOR CBT- PROGRAMS

Wilfried Windmüller
DaimlerChrysler Aerospace Dornier GmbH, Training Systems,
88039 Friedrichshafen, Germany

Heinz-Bernd Lotz
German Federal Office for Military Technology and Procurement (BWB, FEI3)
56068 Koblenz, Germany

ABSTRACT

In this paper we describe a tool focusing on analysis, planning and design of CBT programs, called *STORYIST*. It is a tool for authoring conceptual ideas. The output is not a ready-to-run CBT program but a design from which users can get various multi-platform CBT programs. The process of designing a course usually involves different disciplines and people: problem-domain specialists, teachers, software engineers, psychologists, etc. *STORYIST* allows the co-operative discussion and design of projects. The overall structure of the CBT program can be improved version by version in a well documented way. Among the distinctive features of *STORYIST* are: learning goal- centered approach for the development of CBT programs; simultaneous definition of the problem domain, the content of the course and elements of course structure; evolutionary development of the detailed course program structure; reusability of design or design components; producing multi-platform-oriented CBT designs. Authoring with *STORYIST* actually means to achieve a step by step convergence of learning goals, learning content, and program course structure. The process of convergence passes through three conceptual phases: Conceptual Outline, Detailed Concept and Storyboard (Script). Finally, the author gets a detailed graphic description of the behaviour of the prospective CBT program and he can navigate by sequential or direct links through the whole storyboard. He is able to define a set of multimedia-oriented learning materials for embedding in the CBT programs and he can easily create layouts for learners graphical user interface. Graphical and textual representation and documentation of design components lead to increased performance of the whole course production and follow-up courses. Authors will be able to change the content and the structure easily, following their own ideas, needs, preferences, capabilities and experience.

AUTHORS BIOGRAPHY

Heinz-Bernd Lotz

born in 1954	
1981	graduation from Technical University in Darmstadt (GE) in electrical engineering/ automation
1981	start of Government Service at German Federal Office for Military Technology and Procurement (BWB)
1986 - 1987	Exchange Engineer in PMTRADE, Orlando, FL
1988 - 1997	Training Simulation branch
since 1997	CBT branch

Wilfried Windmüller

born in 1959	
1982	diploma from University of Armed Forces, Hamburg in mechanical engineering
1982 – 1989	Officer in German Armed Forces
1989	System Engineer in Dornier GmbH
1992	joining Education and Training department in Dornier GmbH
since 1993	Project Manager “Tutorial Systems”
since 1994	CBT-Author and Project Manager

STORYIST
CONCEPT DEVELOPMENT TOOL FOR CBT- PROGRAMS

Wilfried Windmüller
DaimlerChrysler Aerospace Dornier GmbH, Training Systems,
88039 Friedrichshafen, Germany

Heinz-Bernd Lotz
German Federal Office for Military Technology and Procurement (BWB, FEI3)
56068 Koblenz, Germany

OVERVIEW

The Federal Office for Defense Technology and Procurement (BWB) has a long-standing experience in CBT development. In 1993 a R&T project „Tutorial Systems“ was launched in order to research and develop didactical and methodological aspects of learning technology for the improvement and standardization of authoring activities and processes in CBT-projects. One of the outcomes of this project is a software package which supports the design of high-quality CBT-programs, corresponding to the guidelines of the BWB.

INTRODUCTION

The way from the idea to the completion of Computer Based Training (CBT) programs is long and the authors of such programs have to pass through several activities. These activities can be divided into three groups: analysis (choose a problem domain, teaching strategy, define global goals of the program etc.), planning (didactic design of the program, select learning materials, link the goals to the materials etc.) and authoring (producing ready-to-run computer programs, which encapsulate didactic structures, instructional strategies and learning materials). The first two, analysis and planning, are pedagogical oriented activities, while authoring points at the use of specialized authoring tools such as authoring languages or systems. The practice shows that, generally, there are two concepts of authoring tools. Following the first one, the authoring tools support all stages of instruction - analysis, planning, design, CBT creation, evaluation. According to the second conception, authoring tools are used mainly in the CBT generation process.
While you can find plenty of authoring tools and ready-to-run CBT programs on the market, very few programming tools are oriented toward the support of the first two mentioned activities: analysis and design. We even can observe a displacement in the description of commercially available authoring tools. Two or three years ago, it was said that authoring tools possessed multimedia capabilities. It meant that they allow users to integrate into a CBT program different kinds of information: text, graphics, video, sound, animation etc. Now, the things have turned: we read descriptions of (multipurpose) multimedia (authoring) tools which can be used for many, many things, including, in particular, creation of CBT programs. Thus the emphasis is put on producing materials and CBT programs; the initial analysis and planning, where the role of authors is significant, is to some extent neglected.
We would like to present *STORYIST* - a conception tool focusing on analysis, planning and design of CBT programs. It is a tool for authoring conceptual ideas, not learning materials. And the result is not a ready-to-run CBT program but a design from which you can get various (multi-platform) CBT programs.

DESIGN PRINCIPLES OF *STORYIST*

Each CBT program operates in a given context. By "context" we denote the set of specific problem-domain, target groups of learners defined by their learning characteristics, learning strategies, preferred by the authors. The tool set enables the author to design CBT programs in an evolutionary approach with the opportunity to define design elements from three different points of view (goals, content, course structure), but always preserving the consistency of the whole design.

- ***Learning goal-centered approach***

The "language" of learning goals is used as a means for creating context-dependable designs (of CBT programs). It gives authors a tool for mapping their mental visions, ideas, notions,

imagination into a coherent, hierarchically organized set of learning goals (see figure 1).

Figure 1. Learning goal description

- ***Simultaneous definition of learning goals, the content of the course and elements of course structure.***

While authors define learning goals and organize them in structures, they have the opportunity to attach structural course elements to these learning goals. In parallel topics and contents of a course can be edited and linked. These content objects are not simply numbered and counted. Authors can establish various links between them: hierarchical, partial, temporal, spatial, causal, activity, prerequisites, author-defined (see figure 2).

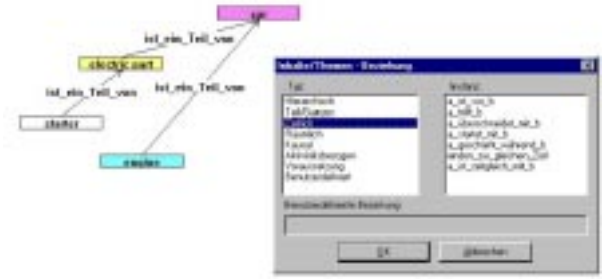

Figure 2. content objects and links

- ***Simultaneous development of the detailed course program structure.***

During the elaboration of detailed learning goals, authors may go to the next logical stage - this is the process of attaching learning steps, which are the detailization of structural elements like chapters (see figure 3), to learning goals and then convert learning steps into sets of pages. They are the smallest units, which contain learning materials (of any kind) to be presented to the learners.

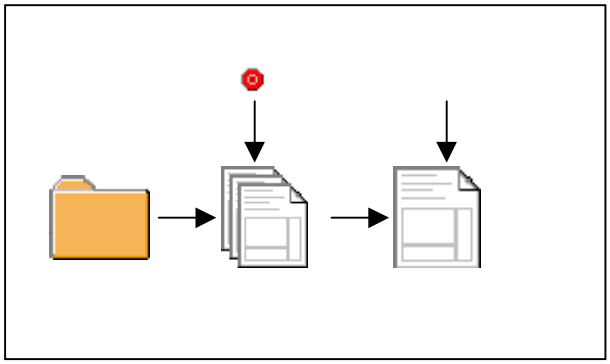

Figure 3. Elements of Program Structure

- ***Referencing content objects to learning material***

For each content object properties can be defined. One property allows to attach learning material to content objects, which can be a file reference or a media description.

- ***Reusability of design or design components and version management***.

Graphical and textual representation and documentation of design components lead to increased performance of the whole course production and follow-up courses. Authors will be able to change the content and the structure easily, following their own ideas, needs, preferences, capabilities and experience. Each design step can be saved as a new version. Additionally versions can be declared as „write-protected“.

LOGICAL STRUCTURE OF *STORYIST*

Each course design produced by the tool set consists of one or more Concepts. Each Concept represents specific authors' views and notions about the course being created.

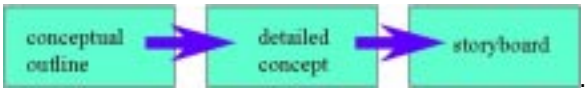

Figure 4. Concept of Conceptual Phases

The role of the Concept is threefold:

- each Concept is the place where authors can develop, modify, and refine learning goals and contents

- each Concept is the place where the learning goals developed by the authors are attached to the appropriate learning structures (chapters & pages);

- each Concept is a place where the contents are connected or have references to learning material.

- Following these roles, *STORYIST* offers the authors three conceptual phases (views) for any Concept and correspondingly, for the whole course design (see figure4).

Conceptual Outline

The first is Conceptual Outline. At this stage the authors can define the very global learning goals, contents, program structures and screen layouts.

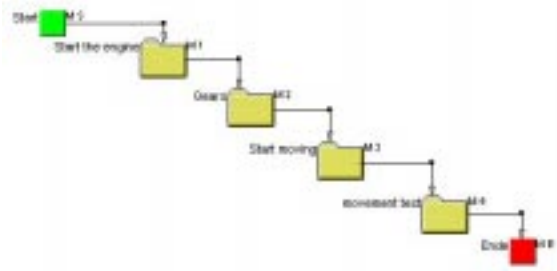

Figure 5. Conceptual Outline Program Structure
The basic structural elements are the *chapters (see figure 5)*. This allows authors to establish the global settings for the course design without going into unnecessary details.

Detailed Concept

The second phase is the Detailed Concept. Each Detailed Concept is a successor of a given Conceptual Outline. This ensures the hierarchical organization of learning goals. At this stage authors can go into unlimited depth of refinement and specification of goals inherited from the Conceptual Outline. Simultaneously, authors elaborate content structure and course structure up to the definition of *learning steps* (see figure 6) and make references to the learning material.

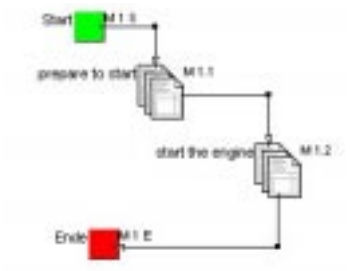

Figure 6. Learning Steps derived from chapters

Storyboard

The third phase is the Storyboard (Script). Reaching this stage, it is supposed that the development of learning goals, content structure and course structure has been defined to a level where pages (see figure 7*)* can be derived from learning steps.

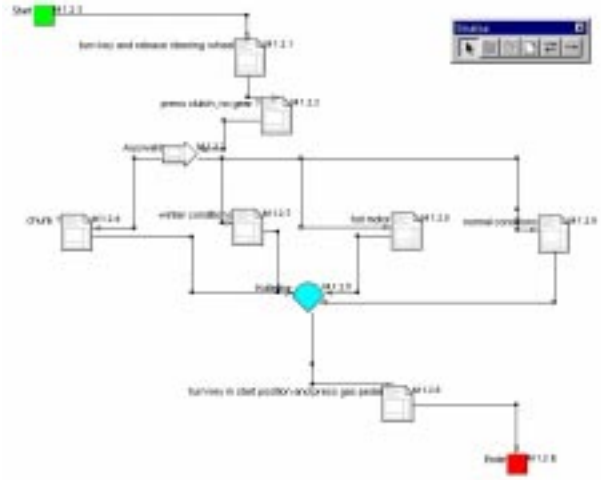

Figure 7. Pages derived from Learning Steps

The pages display the information to the learners and consist of presentational and interaction elements (see figure 8).

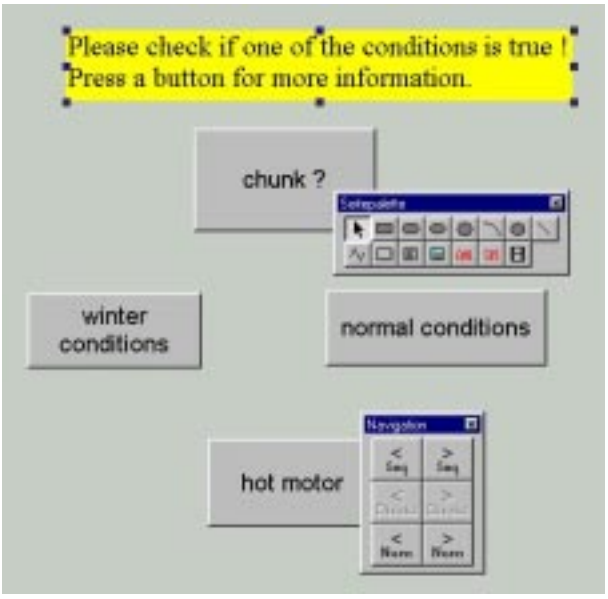

Figure 8. Page Editor and Navigation Tool

Pages can be linked in a sequential order but the author is also able to define direct links between pages which are in different chapters. This bears

the advantage that the author and the user see exactly how the learner will pass through the program. Since the author can link chapters to learning goals the hierarchy of chapters (and pages) reflect the hierarchy of learning goals. Moreover, the software detects conflicts between the right order of pages and the defined sequence of learning goals and steps.

Storyist Output

The *STORYIST* output consists of:

- A script: detailed, form-based, graphic prediction of the behaviour of the prospective CBT program. This is a set of forms, called for convenience pages, united in chapters and connected with links of transitions, thoroughly checked for consistency and completeness.

- A media library - a set of multimedia-oriented materials for embedding in the CBT programs. Some of them are ready-to-use and in this library there are references to them. For the others, there is a description what authors would expect. There are enough multimedia editing tools to fulfill the most strange authors imaginations.

- Interface collection (standards) - a completely specified graphical user interface: it is a detailed specification of the screen layouts, including menus, dialog boxes, help & info structures, multimedia objects.

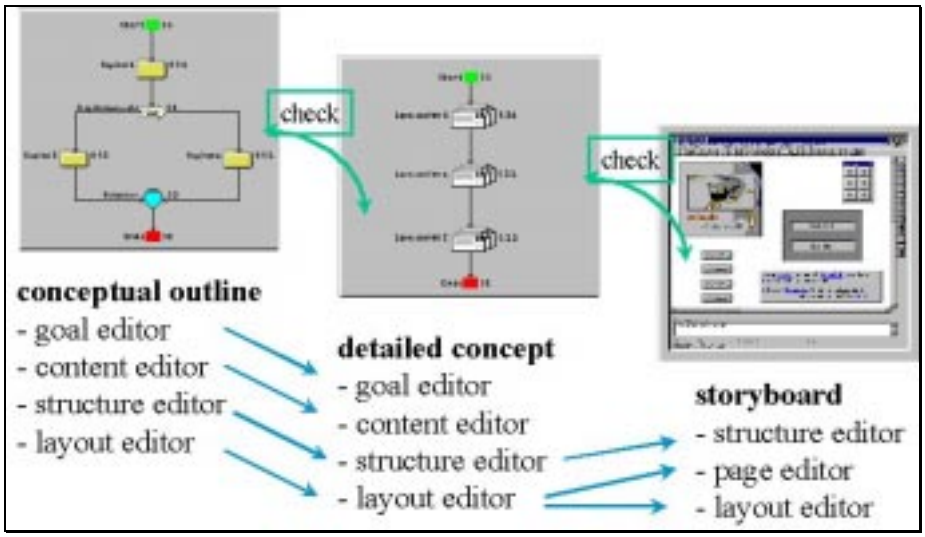

Figure 9. Editors for different Design Phases

STORYIST FUNCTIONALITY

The process of designing a course usually involves different disciplines and people: problem-domain specialists, teachers, software engineers, psychologists, etc. The tool set allows the co-operative discussion and design of projects. The overall structure of the CBT program could be improved version by version in a well documented way, and at the same time various learning materials can be selected and attached to the pages. Different user interfaces can be created, experimented and then easily changed not leaving the *STORYIST* environment.

Moreover, the system does not restrict authors to follow some models or requirements, set up in advance. It is fully open - open to the creativity, imagination and the professionalism of authors. The tool is intended to be an advisor. It is our opinion that human beings are the best teachers, better than the most powerful computers. So, the software tool has to advise and facilitate authors in doing their job - authoring. The software package provides several facilities for authors.

Editors

The five editors are supplementary and allow users to fully specify CBT programs without using a lot of different applications (see figure 9).

These are:

- ***GoalEditor*** for developing and manipulating learning goals;
- ***ContentEditor*** for exploring learning content;
- ***StructureEditor*** for building courseware structures
- ***PageEditor*** for creating and editing pages, sequencing pages, linking goals and contents, and defining multimedia materials used
- **LayoutEditor** for developing interface standards;

Services

This is not a tool but a set of services offered by *STORYIST*. During the whole process of course design, the services observe the work of authors, and where it is necessary proposes some actions:

- checking the course structures: for isolated elements, for correctly linked goals-steps-pages, and for logical non-contradiction (avoiding loops);
- automatically adding/removing branching constructions: select-page, chapter-select-page, collector, remedial sequences of pages;
- visualizing objects in accordance with the level at which the author works: Conceptual Outline, Detailed Concept or Storyboard (Script); this allows users to concentrate on the necessary piece of structure and to avoid unnecessary details;
- allowing users to navigate through very complex program structures

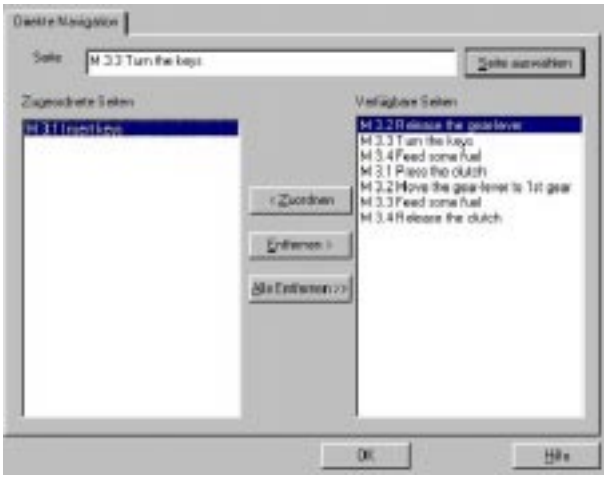

Figure 10. Definition of Navigation Paths

- preview of user interface elements (standards);
- visual screen layout construction.

Documentation

STORYIST has an extensive amount of printing options and provides additionally documentation facilities:
- options for scaling, or determining the level of detailization of the material to be printed;
- some macros like course name, author(s), date, time, version, page numbers.

Hardware and Software Requirements

Storyist is a 32-bit application which has been programmed with Microsoft C++ 5.0. The software runs on Microsoft Windows platforms 95, 98 and NT.

Standardisation

This software tool could support a standardisation of CBT design for:

- Better communication
- Better version management
- Better concistency of design
- Easier protopyping
- Easier maintainability
- Easier acceptance tests

CONCLUSION

We described the software system *STORYIST* as an authoring tool which focuses on analysis, planning and design of CBT programs. The main software feature is that this is a tool for authoring conceptual ideas, not learning materials. And the output is a design from which users can produce various multi-platform and multimedia-oriented CBT programs.

The tool set has been evaluated by experienced CBT-authors and has already been used to design CBT-program. A revision phase will follow until the end of 99. Beginning 2000 we plan to introduce *STORYIST* as a standard toolset for CBT-development to the Federal Office for Defense Technology and Procurement (BWB).

REFERENCES

D. Alistar, N. Edwards, S. Holland: Multimedia Interface Design in Education, Springer Verlag, Berlin, Heidelberg, 1992

E. de Corte, M.C. Linn, H. Mandl, L. Verschaffel: Computer Based Learning Environments and Problem Solving, Springer Verlag, Berlin, Heidelberg, 1992

S. Cunningham, R.J. Hubbold: Interactive Learning through Visualization, Springer Verlag, Berlin, Heidelberg, 1992

Patrick Dickson, Carrie Heester, Mark Rosenberg, Percicles Gomes, Itztik Artzi:“Exploratory Multi-media environments in: Adaptive learning Environments, Springer Verlag, Berlin, Heidelberg, p.155-173, 1992

Ton de Jong and Luigi Sarti: Design and Pro-duction of Multimedia and Simulation-based Lear-ning Material, Kluwer Academic Publishing, 1994

Harald Mispelkamp: Generic Tool for Courseware Authoring,in:Learning Technology in the European Communities, Proceedings of the DELTA Confe-rence on Research and Development, The Hague, Kluwer Academic Press 18-19 October 1990