

C. Alvegård (Ed.)

Computer Aided Learning and Instruction in Science and Engineering

Fourth International Conference, CALISCE '98
Göteborg, Sweden,
June 15 -17, 1998
Proceedings

MEDIT: A Web-based Environment for Advanced Pedagogical Issues	325
<i>O. Abou Khaled, N. Karacapilidis, Y. A. Rekik, C. Vanoirbeek</i>	
Remote Learning Over Broadband Networks. A Pilot Experience With ATM	334
<i>C. Engel, J.J. Casares</i>	
SUITES: A Learning Environment Based on a Computer Algebra System	342
<i>Dominique Lenne, Jean-Michel Gélis</i>	
Teaching Computational Science to Physics and Engineering Students	350
<i>Ian D. Johnston</i>	

Learning Strategies and Development of CAL Software

COLPO: An (Intelligent) Tutoring System for Teaching Linear Programming Modelling	356
<i>M. J. J. M. van de Ven, A. S. E. Bröring, T. J. Bedford, E. de Klerk</i>	
Comparing Two Authoring Approaches of Instructional Simulations: An Industrial Experimentation	363
<i>J. P. Pernin</i>	
A Dynamic Link Library of Explanation Functions	372
<i>Ute Trapp, Markus Lusti</i>	
Error Characterisation in Problem Solving Tasks	381
<i>J-B. Auriol, J-L. Dessalles</i>	
From the Learning Teacher to the Teaching Learner: A CAL Software Dedicated to a New Pedagogy for Higher Education, Application to Fundamental Computer Science.	390
<i>Béatrice Amerein Soltner, Claude Renaud, Michelle Proquin</i>	
ILESA: a Web-based Intelligent Learning Environment for the Simplex Algorithm	399
<i>J. M. López, E. Millán, J. L. Pérez-de-la-Cruz, F. Triguero</i>	
Towards a Library of Distributed Algorithms and Animations	407
<i>Marina Papatriantafilou, Philippos Tsigas</i>	
Providing an Adaptive Learning through the WEB Case of CAMELEON: Computer Aided MEdium for LEarning On Networks	411
<i>Mona Laroussi, Mohamed Benahmed</i>	
RATH – A Relational Adaptive Tutoring Hypertext WWW-Environment Based on Knowledge Space Theory	417
<i>Cord Hockemeyer, Theo Held, Dietrich Albert</i>	
Simulation Based Learning of Computer Architecture	424
<i>V. Shankararaman, J. M. Munoz, Elizabeth Banks</i>	
STORYIST: A Concept Development Tool	433
<i>Wilfried Windmüller, Roumen Radev, Nelly Maneva, Avram Eskenazi</i>	

Posters

Acquisition of Declarative and Procedural Knowledge in Engineering Education Through Hypermedia and Virtual Reality: An Experimental Study on Students' Learning of the Structure and Functioning of a Turning Lathe	441
<i>Alessandro Antonietti, Ernesto Imperio, Chiara Rasi, Marco Sacco</i>	

Artificial Intelligence and Knowledge Engineering Course in Distance Education Mode Delivery	445
<i>Bronius Tamulynas</i>	
Biomolecular Structures: An Internet Course in Biochemistry and Biophysics	448
<i>Örjan Hansson</i>	
Clarifying Ideas on Programming Languages and Fuzzy Control Theory by Using Mobile Robots	450
<i>Jelena Godjevac</i>	
Computer Assisted Study for one Discrete Dynamical System Experience of Teaching with CAL Software	454
<i>Nataly Ampilova</i>	
Creation and Use of an Digital Video Server to Enhance the Teaching and Learning Oral English for French Students	459
<i>Florent Pasquier</i>	
Finite Element Method Simulations of Electromagnetic Problems	460
<i>Dr. E. Peter Münger</i>	
Intelligent Development Tool for Adaptive Courseware on WWW	464
<i>T. Gavrilova, T. Chernigovskaya, A. Voinov, S. Udaltsov</i>	
Internet in Computer Aided Education in Mechanics	468
<i>Sorin Eugen Zaharia, Mihail Atanasiu, Ionut Andrei Damian</i>	
Piping Stress Analysis Using RaFlex® 7.1	472
<i>Carlos A. Graciano G.</i>	
Towards Personalised Web Based Instruction	476
<i>Claudia Engel</i>	
A Virtual Laboratory Unit 'POHL's Torsion Pendulum'	479
<i>Ulrich Harms, Gunther Kurz</i>	
Demonstrations	483
Cal Program for a Truss	485
<i>Andres Labe</i>	
A CAL Software Dedicated to Algorithmics: Let Us Point to the Pointer	485
<i>Béatrice Amerein Soltner, Claude Renaud, Michelle Proquin</i>	
Demonstration of CISP and ZAP – Two Pedagogically Designed Simulators for the EE Curriculum	487
<i>L. J. Peter Linnér</i>	
OASIS Operative Authoring tool for Scenarios and Interactive Simulations	493
<i>Viviane Gueraud, Gloria Cortes</i>	
STORYIST: A Concept Development Tool	495
<i>Wilfried Windmüller, Roumen Radev, Nelly Maneva, Avram Eskenazi</i>	
Authors	497
Keywords	500

- [DeNa94] DeNardo, A.M. & Pyzdrowski, A.S. 1994. A Study of the Effectiveness of Computer-Based Simulations in Teaching Computer Architecture, *Computers in the Schools*, 10(1-2), 125-140.
- [Frye91] Frye, D. 1991. Cognitive Strategies, Learning, and Educational Software. In L. Birnbaum (Ed.), *The International Conference on the Learning Sciences: Proceedings of the 1991 Conference. USA: Association for the Advancement of Computing in Education*.
- [Hend94] Henderson, W.D. 1994. Animated Models for Teaching Aspects of Computer Systems Organization, *IEEE Transactions on Education*, 37(3), 247-256.
- [Holy93] Holyer, A. 1993. Methods for Evaluating User Interfaces. *Cognitive Science Research Paper No. 301*. School of Cognitive and Computing Sciences, University of Sussex at Brighton.
- [Law85] Lawler, R. 1985. *Computer Experience and Cognitive Development*. New York: John Wiley & Sons.
- [Spens90] Spensley, F., Elsom-Cook, M., Byerley, P., Brooks, P., Federici, M. & Scaroni, C. 1990. Using Multiple Teaching Strategies in an ITS. In C. Frasson & G. Gauthier (Eds.), *Intelligent Tutoring Systems: At the Crossroads of Artificial Intelligence and Education*. New Jersey: Ablex Publishing Corporation.

STORYIST: A Concept Development Tool

Wilfried Windmüller

Daimler-Benz Aerospace Dornier GmbH, Training Systems, 88030 Friedrichshafen, Germany

Roumen Radev, Nelly Maneva, Avram Eskenazi

Software Engineering Dept., Institute of Mathematics and Informatics, Bulgarian Academy of Sciences, "Acad. G Bonchev" Str. bl.8, 1113 Sofia, Bulgaria tel. ++ 359 2 713 28 64

rradev@banmatpc.math.acad.bg

Abstract

In this paper we describe STORYIST - a tool focusing on analysis, planning and design of CBT programs. STORYIST is a tool for authoring conceptual ideas, not learning materials. And the output is not a ready-to-run CBT program but a design from which users can get various multi-platform CBT programs.

Among the distinctive features of STORYIST are: learning goals-centered approach for the development of CBT programs; simultaneous definition of the problem domain, the content of the course and elements of course structure; simultaneous development of the detailed course program structure; stepwise development of course designs; reusability of design or design components; producing multi-platform-oriented CBT designs.

Authoring with STORYIST actually means gradual 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 authors get a detailed graphic description of the behaviour of the prospective CBT program, a set of multimedia-oriented learning materials for embedding in the CBT programs and completely specified graphical user interface.

Keywords: Authoring systems, Learning goals, Concept modeling, Courseware design

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 conceptions 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 has 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 introduce STORYIST - a conception tool focusing on analysis, planning and design of CBT programs. STORYIST 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 (Background)

Each CBT program operates in a given context. Following [1], by "context" we denote the set of specific problem-domain, targeted groups of learners defined by their learning characteristics, learning strategies, preferred by the authors. STORYIST 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.

- (1) STORYIST supports a *learning goals-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. These goals are not simply numbered and counted. Authors can establish various links between them: hierarchical, partial, temporal, spatial, causal, activity, prerequisites, author-defined. Also, authors can set up a "pedagogical" order: which learning goals are prerequisites for others.
- (2) *Simultaneous definition of the problem domain* (in terms of learning goals), *the content of the course and elements of course structure*. While authors devise learning goals and organize them in structures, they have the opportunity to attach content and structural course elements to these learning goals. The link between them is enough flexible and allows the content to be altered at any time.
- (3) *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, to learning goals and then convert learning steps into sets of (so-called) pages. They are the smallest units, which contain learning materials (of any kind) to be presented to the learners.
- (4) *Stepwise development of coursedesigns*. Using STORYIST, authors begin with the development of sets of learning goals and simultaneously they can outline contents and course structure. Then they can refine these things, going into deeper details, following possible contexts.
- (5) *Reusability of design or design components*, thus increasing the productivity (of the course production). Authoring with STORYIST means gradual convergence of learning goals, learning content, and (program) course structure. Authors have to be able to change the content and the structure easily, following their own ideas, needs, preferences, capabilities, experience.

- (6) *Producing multi-platform-oriented CBT designs*. We regard STORYIST as an environment, where the authors can specify the hardware and software requirements for the course designs.

Logical Structure of Storyist

Each course design produced by STORYIST consists of one or more Concepts. Each Concept represents specific authors' views and notions about the course being created. The role of the Concept is threefold:

- (1) each Concept is the place where authors can develop, modify, and refine learning goals;
- (2) each Concept is the place where the learning goals developed by the authors are attached to the appropriate learning structures (chapters & pages);
- (3) each Concept is a place where the learning structures are connected or have references to learning materials (of multimedia type).

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

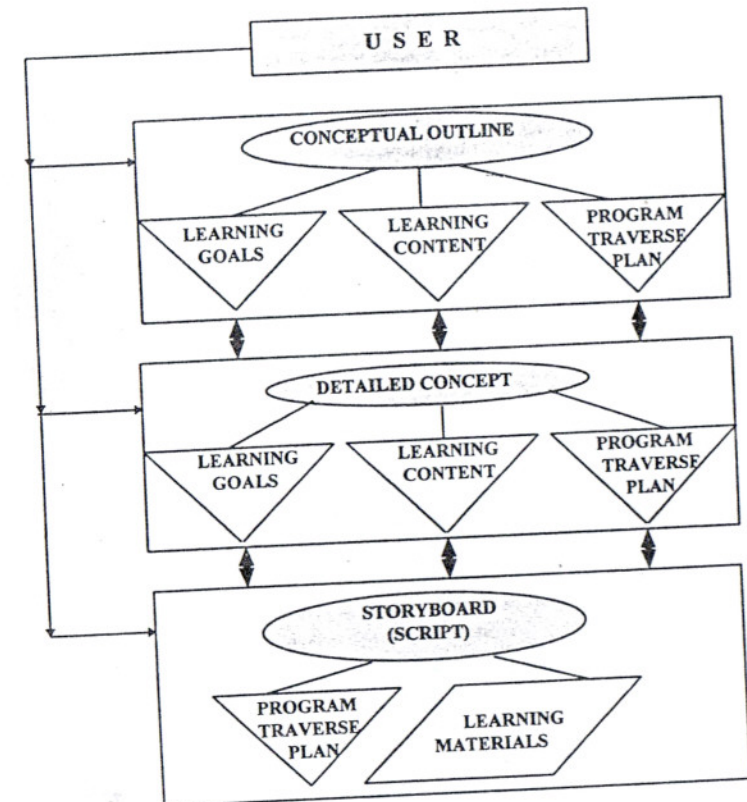


Fig.1. STORYIST logical structure

The first is Conceptual Outline. At this stage the authors can define the very global learning goals, contents, program structures and screen layouts. The basic structural elements are the chapters. This allows authors to establish the global settings for the course design without going into unnecessary details.

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 concretization of goals inherited from the Conceptual Outline. Simultaneously, authors elaborate content structure and course structure up to the definition of learning steps and making references to the learning material.

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 can be derived from learning steps. The pages display the information to the learners and consist of presentational and interaction elements. Pages can be linked in a sequential order but the author is also able to define direct links or hyper 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 and learning steps to sub-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.

The STORYIST output consists of:

- (1) *A script*: detailed, form-based, graphic prescription 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.
- (2) *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. This media library is independent from the script, and can be an uniform library of such a type. Thus, we accomplish reusability of learning materials.
- (3) *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.

Storyist Functionality

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 could be improved version by version in a well documented way, and in 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. STORYIST is intended to be an advisor. It is our opinion that human beings are the best teachers, better than the most powerful computers. So, STORYIST has to advise and facilitate authors in doing their job - authoring.

STORYIST provides several facilities for authors.

Editors.

STORYIST incorporates five editors. They are supplementary and allow users to fully specify CBT programs without using a lot of different applications. These are:

- GoalEditor - for developing and manipulating learning goals;

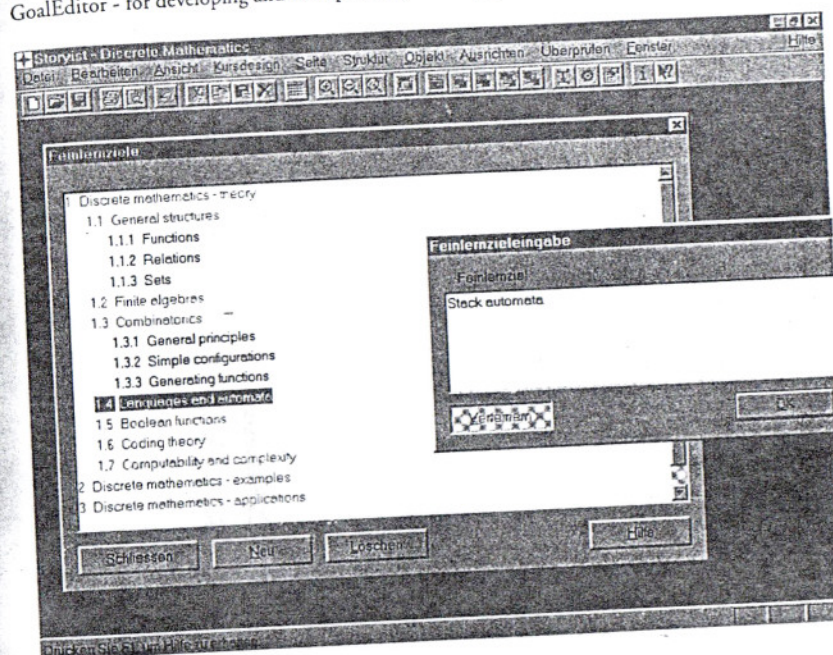


Fig.2. STORYIST Learning Goals Editor

- ContentEditor - for exploring learning content; - StructureEditor - for building courseware structures;

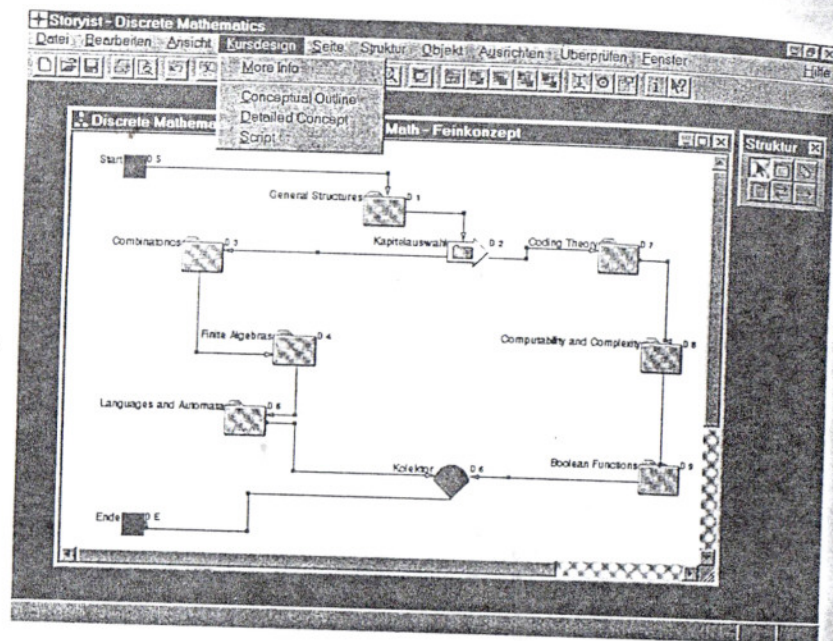


Fig.3. STORYIST StructureEditor

- PageEditor - for creating and editing pages, sequencing pages, linking goals and contents, and defining multimedia materials used;

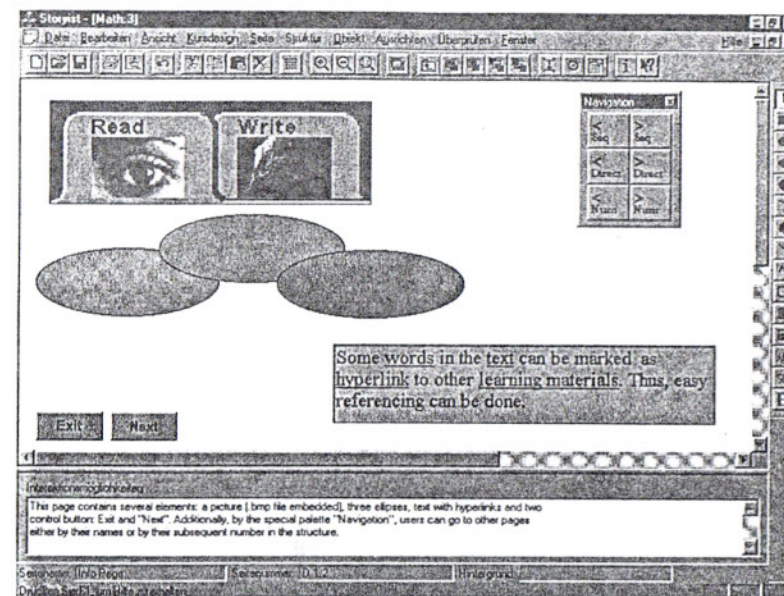


Fig.4. Storyist PageEditor

- LayoutEditor - for developing interface standards;

Services.

This is not a tool but a set of services offered by STORYIST. During the whole process of creating a course design, STORYIST observes the work of authors, does not interrupt them 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 structures by means of scrolling, zooming, mapping, immediate positioning (you type in the name of the object you want to see and the system automatically bring this object (approximately) to the center of the screen);
- optimizing links (for not-crossing);
- preview of user interface elements (standards);
- visual screen layout construction.

Documentation.

STORYIST has print options and provides additionally documenting options:

- facilities 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.

Adaptation.

Let us first consider the "off-line adaptation" - at authoring time. The Conceptual Outline and to a great extent the Detailed Concept can be regarded as courseware templates - and authors in Storyboard (Script) phase can make them more specific (author-oriented). As we stated in the beginning, each course design consists of one or more Concepts and each Concept is a different version of the course design. In other words, the course design consists of many sub-designs. In fact, each sub-design is an adaptation of the course design to peculiar authoring needs.

The question of on-line (run-time) adaptation is left for our future developments. It will be based on the specified system requirements, learners' characteristics, context-dependable segmentation of Storyboard (Script) structures, interface to other programming tools.

Conclusion

We described the software system STORYIST as an authoring tool which focuses on analysis, planning and design of CBT programs. The main STORYIST 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.

STORYIST is in its experimental phase and the German version is nearly completed. Experimental designs have been created in the following topics: "Working with Storyist", "Refrigeratory techniques", "Principles of exploring the coastal line".

Our future developments involve moving to commercially available DBMS. This would lead to increased productivity, better documentation, reusability of components, and which is more important - reusability (or sharing) of learning goals and teaching structures, adaptation to various pedagogical needs, automatization of the last authoring stage - programming.

We also plan to move from scripts and media library to Didactic Component Library in which learning goals, program structures and relevant learning materials will be combined into units.

References

1. R. Moyse, R.Jonson, J.Sime, Learning Functions for Author Modeling, Proceeding of AI-ED 93 (World Conference on Artificial Intelligence in Education), Edinburgh, Scotland, 23-27 August, 1993, pp. 74-81 (ed. by P.Brna, St.Ohlsson, H.Pain)

This work was partly supported by the Ministry of Education of Bulgaria under contract No I-406.