Wenyin Liu
Yuanchun Shi
Qing Li (Eds.)

LNCS 3143

Advances in Web-Based Learning – ICWL 2004

Third International Conference
Beijing, China, August 2004
Proceedings

Springer

Wenyin Liu   Yuanchun Shi
Qing Li (Eds.)

Advances in Web-Based Learning – ICWL 2004

Third International Conference
Beijing, China, August 8-11, 2004
Proceedings

Springer

Volume Editors

Wenyin Liu
City University of Hong Kong, Department of Computer Science
Kowloon, Hong Kong
E-mail: csliuwy@cityu.edu.hk

Yuanchun Shi
Tsinghua University, Department of Computer Science and Technology
Beijing 100084, China
E-mail: shiyc@tsinghu.edu.cn

Qing Li
City University of Hong Kong
Deptartment of Computer Engineering and Information Technology
Kowloon, Hong Kong
E-mail: itqli@cityu.edu.hk

Library of Congress Control Number: 2004109597

CR Subject Classification (1998): H.4, H.3, I.2.6, H.5, K.3, D.2, I.2

ISSN 0302-9743
ISBN 3-540-22542-0 Springer Berlin Heidelberg New York

Springer is a part of Springer Science+Business Media

springeronline.com

Printed in Germany

Typesetting: Camera-ready by author, data conversion by Olgun Computergrafik
Printed on acid-free paper SPIN: 11019794 06/3142 5 4 3 2 1 0

Lecture Notes in Computer Science 3143

Preface

With the rapid development of Web-based learning and new concepts like virtual classrooms, virtual laboratories and virtual universities, many issues need to be addressed. On the technical side, there is a need for effective technology for deployment of Web-based education. On the learning side, the cyber mode of learning is very different from classroom-based learning. How can instructional development cope with this new style of learning? On the management side, the establishment of the cyber university imposes very different requirements for the set-up. Does industry-university partnership provide a solution to addressing the technological and management issues? Why do we need to standardize e-learning and what can we do already? As with many other new developments, more research is needed to establish the concepts and best practice for Web-based learning.

ICWL 2004, the 3rd International Conference on Web-Based Learning, was held at the Tsinghua University (Beijing, China) from August 8th to 11th, 2004, as a continued attempt to address many of the above-mentioned issues. Following the great successes of ICWL 2002 (Hong Kong) and ICWL 2003 (Australia), ICWL 2004 aimed at presenting new progress in the technical, pedagogical, as well as management issues of Web-based learning. The conference featured a comprehensive program, including a tutorial session, a keynote talk, a main track for regular paper presentations, and an industrial track. We received 120 papers and accepted only 58 of them in the main track for both oral and poster presentations.

A conference like this can only succeed with an excellent team effort. We would like to acknowledge the great contribution from our program committee members and paper reviewers who helped review submitted papers, select high-quality papers, and provide valuable comments for the authors. We are particularly indebted to Dr. Liu Wenyin, the publication chair as well as the tutorial chair, who did much more than what the roles asked him to. Special thanks go to Guo Ling and Xiang Xin, who did wonderful jobs as the conference secretary and Web master, respectively. Sponsorships from Tsinghua University, City University of Hong Kong, CELTSC, and the Hong Kong Web Society are gratefully acknowledged.

May 2004

Maria Orlowska
Dennis McLeod
Xiaoming Li
Qing Li
Shi Yuanchun
Rynson Lau
Tosiyasu L. Kunii

Organization

ICWL 2004 was organized by Tsinghua University (Beijing, China) in conjunction with the Hong Kong Web Society.

General Co-chairs

Maria Orlowska	University of Queensland, Australia
Dennis McLeod	University of Southern California, USA
Xiaoming Li	Peking University, China
Qing Li	Hong Kong Web Society, HKSAR, China

Program Committee Co-chairs

Shi Yuanchun	Tsinghua University, Beijing, China
Rynson Lau	City University of Hong Kong, HKSAR, China
Tosiyasu L. Kunii	Kanazawa Institute of Technology, Japan

Steering Committee Liaison

Joseph S.P. Fong	City University of Hong Kong, HKSAR, China

Conference Organization

Organizing Chair

Lizhu Zhou — Tsinghua University, Beijing, China

Industrial Stream

Ronnie Cheung — Hong Kong Web Society, HKSAR, China

Treasurer

Jianhua Feng — Tsinghua University, Beijing, China

Publication

Liu Wenyin — City University of Hong Kong, HKSAR, China

Web Master

Xin Xiang — Tsinghua University, Beijing, China

Local Arrangements

Li Zheng — Tsinghua University, Beijing, China

Publicity

Jianwei Zhang — Tsinghua University, Beijing, China

Tutorial Management

Liu Wenyin — City University of Hong Kong, HKSAR, China

Conference Secretary

Ling Guo — Tsinghua University, Beijing, China

Program Committee Members

Howard Beck	University of Florida, USA
Stephane Bressan	National University of Singapore, Singapore
Wentong Cai	Nanyang Technological University, Singapore
Jiannong Cao	Hong Kong Polytechnic Univ. HKSAR, China
Shi-Kuo Chang	University of Pittsburgh, USA
Elizabeth Change	Curtin University, Australia
Jo Coldwell	Deakin University, Australia
Tharam Dillon	Sydney University of Technology, Australia
Andrzej Goscinski	Deakin University, Australia
John Hughes	University of Ulster, UK
Ali Hurson	Pennsylvania State University, USA
Horace H.S. Ip	City University of Hong Kong, HKSAR, China
Weijia Jia	City University of Hong Kong, HKSAR, China
Qun Jin	Waseda University, Japan
Feiyu Kang	Tsinghua University, Beijing, China
C.-T. King	National Tsinghua University, Taiwan, China
Reggie Kwan	Open University of Hong Kong, HKSAR, China
Tosiyasu L. Kunii	Kanazawa Institute of Technology, Japan
Rynson W.H. Lau	City University of Hong Kong, HKSAR, China
Sue Legg	University of Florida, USA
Clement Leung	Victoria University of Technology, Australia
Guojie Li	Institute of Computing Technology, China
Keqin Li	State University of New York at New Paltz, USA
Qing Li	City University of Hong Kong, HKSAR, China
Xiaoming Li	Peking University, China
Liu Wenyin	City University of Hong Kong, HKSAR, China
Dennis McLeod	University of Southern California, USA
Christine Morin	IRISA/Université de Rennes, France
John Mylopoulus	University of Toronto, Canada
Maria Orlowska	University of Queensland, Australia
Raul Rojas	Free University of Berlin, Germany
Geoff Romeo	Monash University, Australia
Shi Yuanchun	Tsinghua University, Beijing, China
Timothy Shih	Tamkang University, Taiwan, China
Chengzheng Sun	Griffith University, Australia
Changjie Tang	Sichuan University, China
Y.-M. Teo	National University of Singapore, Singapore
Ulrich Thiel	GMD-IPSI, Darmstadt, Germany
A Min Tjoa	Vienna University of Technology, Austria
Zhiwei Xu	Institute of Computing Technology, China
Zongkai Yang	Huazhong Sci. & Tech. University, China
Jianwei Zhang	Tsinghua University, Beijing, China

S.Q. Zheng	University of Texas at Dallas, USA
Lizhu Zhou	Tsinghua University, Beijing, China
Wanlei Zhou	Deakin University, Australia
Zhiting Zhu	Huadong Normal University, China
Yueting Zhuang	Zhejiang University, China

Additional Reviewers

In addition to all members of the Program Committee, the following people were also involved in reviewing the papers:

Enyi Chen	Quanbin Chen	Jianhua Feng	Ling Guo	Jinyu Li
Peifeng Xiang	Xin Xiang	Xin Xiao	Degan Zhang	Nan Zhang

Table of Contents

e-Learning Platforms and Tools

Learning Resource Deployment, Organization and Management

Practice and Experience Sharing

e-Learning Standards

Pedagogical Issues

Learning Algorithms with an Electronic Chalkboard over the Web

Margarita Esponda Argüero and Raúl Rojas

Institut für Informatik, Freie Universität Berlin, Takustr. 9, 14195 Berlin
{esponda,rojas}@inf.fu-berlin.de

Abstract. This paper describes a system for the animation of algorithms on an electronic chalkboard. The instructor teaching an algorithm can enter data directly through a drawing – the algorithm then makes use of this data, for example numbers, or the image of a graph. The drawing becomes alive. The result is a more natural way of teaching and starting algorithmic animations. The paper also describes how to couple a sign and handwriting recognition engine with the animation system. The lecturer can then write programs using her own handwriting, and the programs runs. All animations can be enriched with sound and explanations from the lecturer and can be posted to the Web.

1 Motivation

Algorithmic animations are becoming popular for teaching computer science [1]. There are now many visualizations of algorithmic animations available on-line. Most are used by students, as part of a course, and have been developed by university staff.

We have recently developed an algorithmic animation system called Chalk Animator as an extension for the E-Chalk system. Conventional systems for algorithmic animation handle only computer generated images as building blocks in an animation. Perfect rectangles, circles, arrows, etc. are drawn by the user using a graphical editor or are generated automatically by the system. In this paper we consider a more radical alternative: the generation of algorithmic animations starting from sketches drawn by the user on an electronic blackboard, which is both the presentation tool and the user interface for the lecturer. Not only is this approach time-effective for the lecturer, but also the "look and feel" of animated sketches is very different from computer generated graphics. Sketch animation resembles best the kind of teaching done using a traditional chalkboard. The animations. Once completed, can be see and heard through the Web.

2 The e-Chalk System

The main idea of the E-Chalk system is to provide the functionality of the traditional chalkboard using a large contact sensitive computer screen, but enhanced with all the capabilities of a digital system [7,2,3,4,5]. An electronic blackboard should be as easy to use as a traditional one. The only interface to the electronic board should be a stylus, instead of a piece of chalk.

W. Liu et al. (Eds.): ICWL 2004, LNCS 3143, pp. 1–10, 2004.

When an E-Chalk session is started, the server computer starts storing and sending three streams: the board events, the audio channel, and an optional video channel. The three streams can be accessed from a Web page, by starting the E-Chalk client, which is a collection of three client Applets, one for each stream. The streams are synchronized by the audio time stamp. Therefore, an E-Chalk session can be recorded completely, but the quality of the reproduction of the board on the client side is much higher, since the board is repainted with the full resolution of the computer screen. Fig. 1 below shows schematically a teaching scenario: a lecturer teaches to a live audience; the E-Chalk server transmits the audio, video and board streams and stores an archival copy. A remote viewer watches the class using an Internet browser.

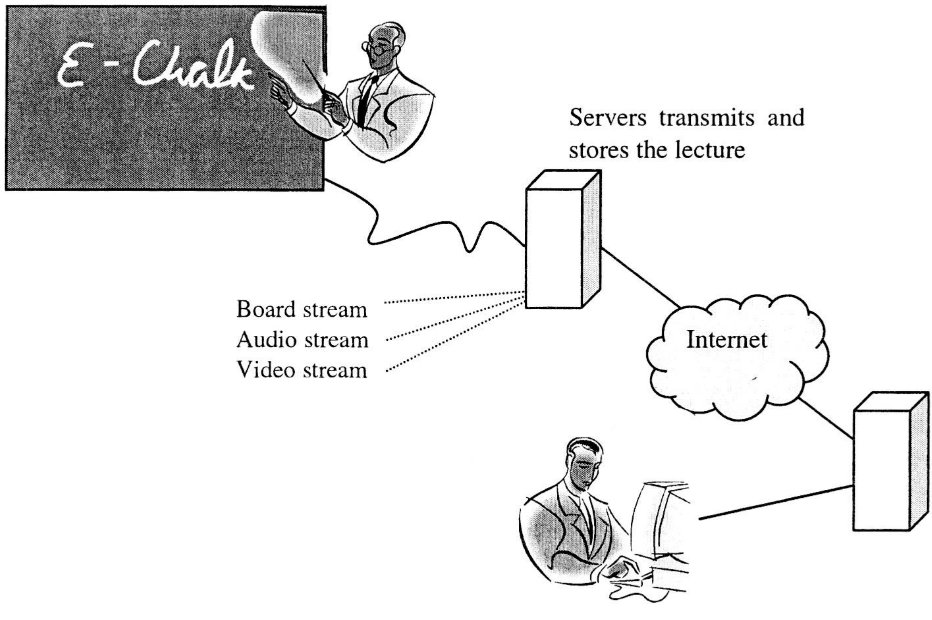

Fig. 1. The E-Chalk system. A lecturer writes on an electronic blackboard. Audio, video, and board contents are stored and are streamed through the Internet. Remote viewers watch a lecture using a Java enabled browser.

Fig. 2 is a screen dump of an actual lecture, as seen by a remote viewer in his browser. The look and feel of the screen is that of a lecture on a good blackboard. The use of color helps to emphasize some important aspects of the lecture.

The main feature of E-Chalk is to go beyond the original blackboard metaphor and provide “intelligence and information on demand”. This means, that a series of special programs is running in parallel with E-Chalk and is watching the user interacting with the screen. Certain programs can then become active when certain conditions are met. E.g., a program can observe the handwriting of the user and if a mathematical formula is entered, and if the user writes a special stop symbol, the program can interpret the formula. If it is an equation, it can solve it. This capability has been already implemented in E-Chalk, using Mathematica as the mathematical equation solver [8,9,10].

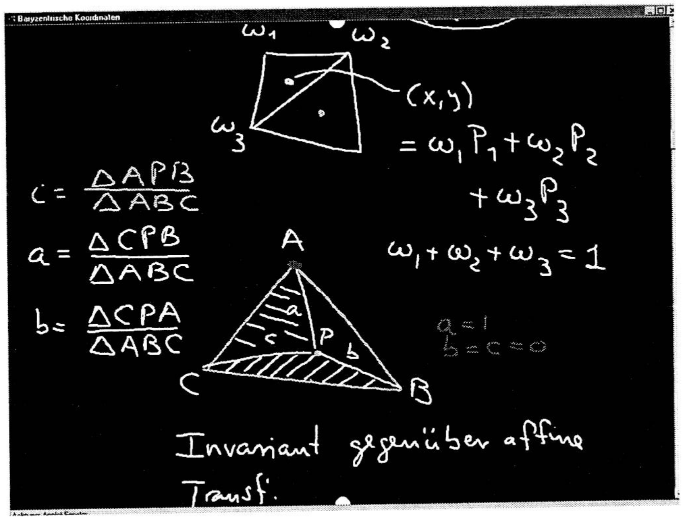

Fig. 2. A real E-Chalk lecture about geometric concepts.

Regarding algorithmic animations there are two things which come immediately to mind as possible extensions of the blackboard metaphor: a) the immediate execution of code written on the blackboard, b) the animation of algorithms started by the lecturer.

3 Executing a Programming Language in e-Chalk

In this section we describe a further educational innovation implemented for the E-Chalk system as. A simple interpreter for the programming language BASIC, was written, and the interpreter was coupled with the handwriting recognition machinery of E-Chalk. The lecturer can now write a BASIC program on the chalkboard and request its immediate execution. This implementation provides a glimpse of what will become possible in the future.

A handwriting recognizer is needed, if code written on the electronic blackboard is to be executed. The E-Chalk handwriting recognizer is a pattern recognition system developed by Ernesto Tapia at the FU Berlin [8].

Symbols are recognized by processing line strokes and extracting relevant features. Such features are, for example, the length of the line stroke, its centroid (in a unitary square), the distance between start and end of the stroke, divided by the total length. Also, the coordinates of a few points along the stroke can be added to the feature vector. The most relevant points for the shape are selected using a shape simplification algorithm. Once a symbol has been transformed into a feature vector, it is given to a neural network or support vector machine, which has been trained previously to recognize this symbol.

As a proof of concept for a programming system based on handwriting recognition, we defined a minimum subset of BASIC, which is nevertheless general purpose and universal. Our own version of BASIC is called *Tiniest BASIC* and consists of only two interpreter commands and six types of instructions.

The two instructions “RUN” and “LIST” can also be entered. RUN starts the program at the first line of code. LIST provides a listing of the current code lines. Non-trivial programs can be written with this language.

Fig. 3 shows JMATH, the program developed by Ernesto Tapia to train the character recognizer [10], coupled with our BASIC interpreter. A Tiniest Basic program has been entered. Each character is surrounded by a box and an identifier of the character which has been recognized. After writing the “LIST” command, the user closes the input by pressing on the B button (BASIC). The window to the lower left shows the output of the Tiniest BASIC interpreter.

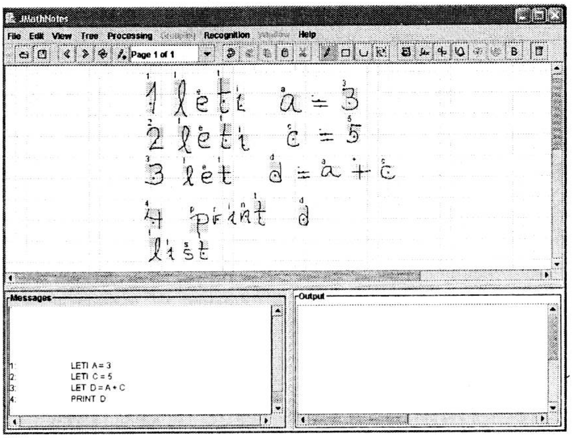

Fig. 3. Screenshot of JMATH, the editor and training program for formula recognition developed by E. Tapia, coupled to the BASIC interpreter.

Fig. 4 shows the same program, but now the command “RUN” has been entered. The output of the program is visible in the lower left window, it is the constant 8.

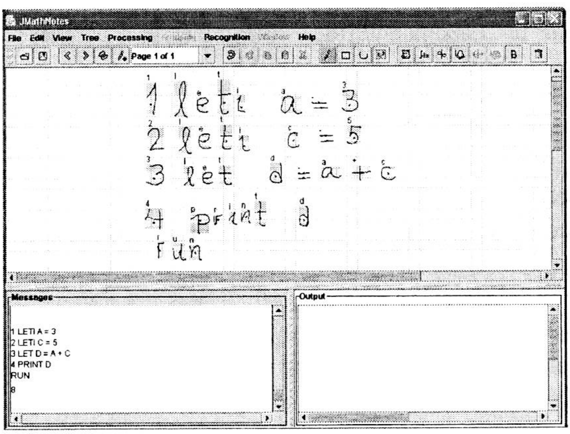

Fig. 4. Screenshot the Tiniest BASIC program after it has been executed.