Electronic Books for Programming Education: A Review and Future Prospects

R. Martínez-Unanue, M. Paredes-Velasco
Escuela Superior de CC. Experimentales y Tecnología
Universidad Rey Juan Carlos, Spain
r.martinez@escet.urjc.es
m.paredes@escet.urjc.es

C. Pareja-Flores
Escuela Univ. de Estadística
Depto. de Sistemas Informáticos y Programación
Univ. Complutense de Madrid, Spain
cpareja@sip.ucm.es

J. Urquiza-Fuentes and J. Á. Velázquez-Iturbide
Escuela Superior de CC. Experimentales y Tecnología
Universidad Rey Juan Carlos, Spain
j.urquiza@escet.urjc.es
a.velazquez@escet.urjc.es

ABSTRACT
Programming is a suitable field to design electronic books with a laboratory component, where the programming task is exercised in the theoretical context provided by the book. The goal of the paper is to make a review of current electronic books for programming education and identify future lines of research. First, we review a number of software tools and electronic books for programming education in order to give a broad vision of technological opportunities in programming education. Later, a comparative analysis of such electronic books is made. Finally, based on this overview and analysis, we identify aspects that either are currently poorly supported or are a subject of active research, thus constituting potential areas for future improvement.

Categories and Subject Descriptors
H.5.1 [**Information Interfaces and Presentation**]: Multimedia Information Systems. H.5.4 [**Information Interfaces and Presentation**]: Hypertext/Hypermedia. K.3.2 [**Computers and Education**]: Computer and Information Science Education – *computer science education*.

General Terms: Algorithms, Documentation, Design, Human Factors.

Keywords: Electronic books, programming education, programming tools, algorithm animation, exercises.

1. INTRODUCTION
Different good definitions of the concept of electronic book can be found in the literature. The following one [6] is a good representative: "An electronic book is an information system capable to make users available a set of pages, conceptually organized as a paper book, and amenable to interact with". The definition is very broad, and can be implemented as a multimedia application, CD-ROM, Web site, help system, etc. All of them can be based on the book metaphor and share a concern to be computer-based, use hypermedia, and include structured information where text plays an important role.

Last years, there have been dramatic achievements and improvements in the design and development of electronic books. A number of fields have contributed to such a development: digital typography, multimedia and Web technologies, mark-up languages, user interfaces, etc.

However, adoption of electronic books for education is far from being the rule. Although they exhibit many advantages with respect to paper books, also there are drawbacks:

- Lower quality (resolution) of text and graphics.
- Necessity of an electronic device (a computer or an e-book reader).
- The less comfortable, vertical position of the head required for reading in a PC.
- Less friendly user interface for handling and reading.

Much work is necessary to overcome the user interface provided by paper books, a result of centuries of innovation and improvement. As the designer Yves Zimmerman says, "it is as well conceived as a knife".

Electronic books have only succeeded in a few fields, such as encyclopedias, dictionaries, and textbooks. The later class can be enriched, among others, with facilities for problem solving and assessment. In particular, problem solving is especially important in scientific, technical and engineering disciplines, where electronic books can assist by means of "virtual laboratories".

We are concerned with the development of electronic books for the engineering discipline of computer programming. The "virtual laboratory" necessary for computer programming consists in the virtual activity par excellence: the development of programs. These "laboratories" should provide the tools common in programming environments (at least, editor, processor, debugger and support for mundane activities, such as file management).

ITiCSE'02, June 24-26, 2002, Aarhus, Denmark.

In this paper, we analyze past efforts to produce electronic books for programming, identify their merits and demerits, and outline future lines of work and research. In the following section, we briefly review a number of tools for programming education, in order to give a broad vision of technological opportunities for programming education. The third section contains an overview and comparative analysis of existing electronic books for programming education. Section 4 outlines prospects for future research. Finally, we summarize our conclusions.

2. TECHNOLOGIES TO SUPPORT PROGRAMMING EDUCATION

Current programming environments are integrated programming environments with comprehensive facilities for the professional programmer: editor, language processor, debugger, profiler, configuration manager, etc. However, from an educational point of view, they often are not that good. Thus, they neither provide a mental model of programming consistent with that taught at programming courses (they are based on the language implementation), nor assist students in building programming knowledge on previous knowledge (they are for programming experts). Explanation of errors, satisfactory for professional programmers, is typically too cryptic for novices: just a line number and a short message are given, but they are not further explained. Finally, they rarely give educational facilities, such as assessment of exercises, program visualization and animation, etc.

Last years, many efforts have been made to develop programming environments and tools [15] to assist in educational activities of different users (teachers, students or both). We enumerate several kinds of tools with high educational impact, although for the sake of space we cannot give an exhaustive list of references:

- Programming tools that support partial programming languages or incomplete programs. In some cases, they allow working with parts of the language or the language is divided into levels of difficulty for progressive teaching, whereas in other cases only a part of the program is required.
- Software visualization and animation. There are a large number of visualization or animation systems (e.g. read [18]), most of them designed for the imperative paradigm. In last years, efforts are being made to measure quantitatively and to improve their pedagogical effectiveness [2]. As a consequence, they are being enhanced by considering specific features with educational impact: integrating animations with explanations, into programming environments or with grading systems, allowing students to choose input data, or making "stop-and-think" questions.
- Aids for program analysis. Some tools assist in generating data input and output and in gathering performance data, while others analyze the programming style.
- Administration, generation and assessment of assignments. Many tools have been developed to assist in the many tasks related to assignments assessment, including generation, administration, automatic grading, and plagiarism detection.
- Collections of exercises. Large structured repositories of exercises are a valuable resource to reuse high quality materials elaborated by faculty. They can often be considered electronic books on exercises (e.g. eXercita [8]).

There are no definitive experimental results on whether the use of these tools improves programming education. However, there is a consensus on the importance of measuring their educational impact and about the fact that these tools, independently from their efficacy, increase students motivation.

3. A REVIEW OF ELECTRONIC BOOKS FOR PROGRAMMING EDUCATION

In this section, we review existing electronic books for programming education. We begin with a description of the most representative ones (up to our knowledge), and then we analyze them from two points of view: features of multimedia applications, and tools for programming education.

3.1 Electronic Books for Programming Education

Based on the broad definition of electronic books given in Section 1, we have chosen several representative electronic books.

Notice that we have filtered many related applications. In particular, we do not consider commercial multimedia applications with limited features, multimedia applications that do not fit the definition of electronic book, or direct dumps of textbooks to hypertext. We have also studied electronic books on related fields, such as theoretical computer science (e.g. hypertextbooks [3]). However, we do not include them here for the sake of space and because our conclusions are similar.

Algorithms in Action [19]. It is a multimedia tool for teaching algorithms. It is a Java application to be used with a browser, showing 3 separate windows with pseudocode, explanations, and animations. Based on the methodology of stepwise refinement, it allows the user expanding and contracting pseudocode of given algorithms; explanations and animations are correspondingly updated in the other two windows. It provides 4 learning modes, based on: explanations (the default mode), demonstrations (predefined animations), self-evaluation (with quizzes), and exploration (allowing the user to input test data). Some design guidelines were tested or derived after feedback from students.

CAT and JCAT [4]. JCAT (Java Collaborative Active Textbooks) is a Web-based algorithm animation system designed to be used in electronic classrooms. It consists of Web pages and algorithm animations, generated by a sophisticated algorithm animator that supports multiple views (updated simultaneously as the program runs), allows entering input data and includes a control panel and a speed control. The teacher controls the animation, and students can browse and customize their views.

HalVis [12]. HalVis (Hypermedia Algorithm Visualizations) is a hypermedia application to teach algorithms. Lessons correspond to different algorithms. For each lesson, it imposes a reading structure based on pedagogical considerations: motivation, explanation in detail, and global view. The three parts make an extensive use of animations. Detailed explanations are the most complex, including several synchronized windows with different aspects of an algorithm (e.g. an animation, pseudocode, values of variables, etc.). The user can control replay, speed, granularity of the animation steps, and enter input data. It includes quizzes,

some of them optional and others mandatory. Finally, it contains a fundamentals module, devoted to explain common elements in programming, which is accessible by links integrated in pseudocode. Its success has been validated with five experiments.

"Curso interactivo de programación en Pascal" [17]. It is a Web-based CD-ROM for teaching Pascal. Lessons are organized lineally. It contains navigation facilities by means of controls and indices. It includes a number of quizzes per lesson (123 in total) that can be used to generate "exams", that the book will grade. It allows interacting with an external editor, thus running simultaneously a programming environment such as Turbo Pascal.

ELM-ART [20]. It is an adaptive and interactive textbook to support learning programming in Lisp. It integrates an enhanced evaluator that allows modifying and evaluating examples. It assesses the reader's solutions by testing. When the user makes an error, explanations, counterexamples, and even hints to help fixing it are given.

Exploring Computer Science Concepts with Scheme [11]. It is a multimedia tool to accompany the author's book on computer science [10]. Lessons are organized lineally. It contains several tools for navigation: navigation controls, a sophisticated search facility on the main text, and user-defined bookmarks. It includes animations on the internal working of several functions in Scheme. A personal notebook is also included for self-study. It allows running simultaneously the Scheme interpreter.

KBS-Hyperbook *Introduction to Java Programming* [13]. The KBS-Hyperbook system permits to model and build adaptive, open hypermedia systems on the Web, such as the *Introduction to Java Programming* hyperbook. Student specify their learning goals by means of knowledge items, and they are given a course composed of several lectures consisting in a sequence of text units. Visual hints for navigation are provided, e.g. labeling links with semaphore lights. Each course is related to projects, and is assessed by means of portfolios.

ProgramLive [9]. It is a commercial multimedia application for teaching object-oriented programming and Java. It is a very comprehensive electronic book that includes a lessons book, animations, and quizzes. The book is divided into short lessons, complemented with expositions, i.e. short narrated explanations on specific concepts, sometimes relating animations to code. Navigation is supported by navigation controls over pages, several kinds of indices, a search facility, and user-defined bookmarks. Footnotes allow the reader obtaining additional information on lessons. Self-evaluation quizzes are filled in by drag-and-drop. Finally, it includes open problems and projects.

WWW-Based C++ Course [14]. It is a Web site devoted to the teaching of C++. It is structured in 3 layers: lessons, reference, and pragmatics, being the lessons layer the main one. There are a variety of resources to navigate (navigation controls, indices, formatting rules for links, and visual cues). The contents are structured in blocks that guarantee that scrolling is not necessary. Exercises are provided for each lesson, as well as the possibility of compiling and running programs in batch mode at the server.

3.2 Comparative Analysis

The main differences of the former electronic books are summarized in Table 1. Rows contain the given electronic books; columns are labeled with some relevant aspects of hypermedia applications and with educational and programming aspects. Each cell contains a brief description of the main features provided by an electronic book with respect to the corresponding aspect.

The common features we have found follow:

- Main media. Most of them are mainly based on text. Algorithm animations are also common. In fact, these electronic books fit two categories. The first three ones are hypermedia applications *on algorithms*, where the central element are animations; the remaining six resemble traditional books *on programming*, strongly based on textual explanations.
- Lessons structure. Programming books follow a traditional lineal structure. It is also common the existence of short lessons, complemented with additional information or algorithm animations. In some cases, structure is further refined, either by defining different layers of discourse or by substructuring explanations. On the other hand, books on algorithms consist of a set of independent lessons.
- Navigation and control facilities. Systems based on text exhibit a sophisticated navigation system, based on different techniques: navigation controls, search facilities, and user-defined bookmarks. On the other hand, systems based on algorithm animators include animation controls.
- Annotation tools. Two of them allow writing annotations, but only one of them includes a personal notebook.
- Evaluation tools. Most of them include a fixed set of solved quizzes or exercises. More advanced facilities can be found in a few cases: using a portfolio or assessing exercises by means of program testing.
- Programming tools. Some of them do not include any tools; others contain some tool, typically an independent interpreter or programming environment or an algorithm animator. Only in one case, a high-level evaluator is integrated in the book.

In summary, electronic books based on traditional books make an extensive use of text and exhibit a lineal structure. Electronic books based on algorithm animation contain independent lessons, and have sophisticated hypermedia features. All of them use with advantage computing facilities for either navigation or animation control. Other potential facilities, such as annotation and exercises are more modest. Finally, programming tools are seldom provided or they are given as independent applications.

4. FUTURE PROSPECTS

Based on the previous two sections, we identify here aspects of electronic books on programming that currently either are poorly supported or are a subject of active research, thus constituting potential areas for future improvement in these electronic books.

Our first conclusions are straightforward. In the first place, future books should provide, at least, currently established standards with respect to lessons structure and navigation. In addition, opportunities for improvement can be found in other hypermedia features and in their annotation features.

Table 1. Summary of aspects of electronic books on programming

	Main media	Lessons structure	Navigation/control facilities	Annotation tools	Evaluation tools	Programming tools
Algorithms in action	Text. Animations	Independent lessons. Hierarchical substructure	Expansion/contraction. Animation control	—	Quizzes	—
CAT and JCAT	Text. Animations	Independent lessons. Multiple views	Animation control. Different controls for teacher and students	—	—	—
HalVis	Text. Animations. Audio	Independent lessons. 3 sequential parts per lesson. 4 synchronized windows	Control of animation granularity. Animation control	—	Several classes of simple questions	—
"Curso interactivo de programación en Pascal"	Text	Lineal structure	Navigation controls. Several kinds of indices	—	Quizzes	Independent program editor
ELM-ART	Text	Independent lessons. Hierarchical substructure	Adaptive navigation by annotating links	Annotations with HTML-forms	Program testing	Syntax-directed editor. Evaluator and pretty-printing. Explanation of errors
Exploring Computer Science Concepts with Scheme	Text. Animations	Lineal structure	Navigation controls. Search facility on main text. User bookmarks	Notebook	—	Independent language interpreter
KBS- Hyperbook *Introduction to Java Programming*	Text. Images	Lessons composed of sequences of text units	Indices to information units. Hints for navigation. Navigation controls. User bookmarks	Annotations with HTML-forms	Portfolios	—
ProgramLive	Text. Animations. Audio	Lineal structure. Short lessons with expositions. Additional information	Navigation controls. Several kinds of indices. Search facility. User bookmarks	—	Drag-and-drop quizzes	—
WWW-Based C++ Course	Text	Lineal structure. 3 layers	Navigation controls. Several kinds of indices. Visual cues	—	Program testing	Server-side compiling and running

Although electronic books based on paper books seem to be adequate as a main reference for programming courses, an electronic version often requires different features [5]. Future books will make less use of text and more of other media, and will provide different paths for reading, often intended for different learning styles.

In order to obtain insight on future directions for electronic books on programming, we have also compared their educational and programming facilities with their counterparts in other educational programming applications (read Section 2). Thus, we find very different degrees of sophistication in several aspects:

- Algorithm animations. Enhancements to couple animations with other elements, such as explanations and exercises, have been very successful. We can even find taxonomies of animators with respect to their educational facilities [1, 7].

A limitation of algorithm animations is that they are usually constraint to a fixed set of (high-quality) predefined animations. Constructing other animations either is forbidden or is costly; in any case, they cannot be generated automatically for any algorithm, since they are high-level animations that exploit knowledge of the algorithm. However, generality can be provided if we lower the abstraction level [18]. In this case, we have program visualizations, which are not so impressive, but can display relevant elements of program execution (e.g. read [16] on visualizing the evaluation of functional expressions).

We still have not found electronic books on programming where automatic program visualizations are generated. This would augment their dynamism by allowing users to really experiment with open, virtual laboratories. The main obstacle

here is the technical complexity of incorporating language processing and program visualization.

- Programming tools. It is the most deficient aspect of current electronic books. Tools are typically provided in electronic books as independent applications, or they are simulations of actual tools. Compare this lack of support with facilities provided by educational programming environments for, at least, program execution and program analysis.

  Another way of understanding the consequences of this limitation is examining electronic books for other fields. For instance, electronic books on automata theory (e.g. [3]) allow the user to enter any grammar or finite state automata. Limiting the user to only experiment with given grammars or automata would be considered unacceptable.

  A consistent integration of tools in electronic books (as in programming environments) would be a qualitative shift (this problem has been dealt elsewhere [7] in relation to formal grammars). It is not a simple task: the complexity of the whole application increases substantially, since such tools are not trivial: language processors, sophisticated editors, etc. In addition, the user interface must be redesigned in order to integrate, in a consistent manner, facilities of both electronic books and programming tools.

  However, it is worth making such an effort if dynamism of electronic books for programming is to be improved. In other fields, there are electronic books where simulations (of music, physics, etc.) are a central part of their design and educational experience. Computer programming should not be constraint to the static experience provided today.

5. CONCLUSIONS

Different efforts have been made to produce electronic books for programming education. We have summarized the main features of the most relevant ones (up to our knowledge). We have made a comparative analysis of them based on features of hypermedia and educational applications, identifying common patterns and opportunities for improvement.

We have also compared their functionality with other tools for programming education. As a consequence, we claim that more efforts should be made to provide a dynamic experience to students. In particular, we have argued for better support to algorithm animation, and programming tools.

6. ACKNOWLEDGMENTS

This work is supported by the Spanish Research Agency CICYT under contract TIC2000-1413.

7. REFERENCES

[1] Anderson, J.M. and Naps, T.L. A context for the assessment of algorithm animation systems as pedagogical tools. In Proc. First Program Visualization Workshop (2001), University of Joensuu, 121-130.

[2] Ben-Ari, M. Program visualization in theory and practice. Informatik/Informatique, 2 (April 2001): 8-11.

[3] Boroni, C.M. et al. Engaging students with active learning resources: hypertextbooks for the Web. In Proc. 32nd SIGCSE Technical Symp. (2001), ACM Press, 65-69.

[4] Brown, M.H. and Raisamo, R. JCAT: Colaborative active textbooks using Java. Computer Networks and ISDN Systems, 29 (1997): 1577-1586.

[5] Caumanns, J., Apostopoulos, N. and Geukes, A. Computers are not books. In Proc. ED-MEDIA'99, AACE, 236-241.

[6] Díaz, P., Catenazzi, N. and Aedo, I. De la multimedia a la hipermedia. Ra-Ma, 1996 (in Spanish).

[7] Diehl, S. and Kerren, A. Levels of exploration. In Proc. 32nd SIGCSE Technical Symp. (2001), ACM Press, 60-64.

[8] Gregorio-Rodríguez, C. et al. EXercita: Automatic Web publishing of programming exercises. In Proc. ITiCSE 2001, ACM Press, 161-164.

[9] Gries, D. and Gries, P. ProgramLive. http://www.datadesk.com/ProgramLive

[10] Grillmeyer, O. Exploring Computer Science with Scheme. Springer-Verlag, 1998.

[11] Grillmeyer, O. An interactive multimedia textbook for introductory computer science. In Proc. 30th SIGCSE Technical Symp. (1999), ACM Press, 286-290.

[12] Hansen, S., Schrimpsher, D. and Narayanan, N.H. From algorithm animations to animation-embedded hypermedia visualizations. In Proc. ED-MEDIA'99, AACE, 1032-1037.

[13] Henze, N. and Nejdl, W. Extendible adaptive hypermedia courseware: Integrating different courses and Web material. In Brusilovsky, P., Stock, O., and Strapparava, C. (eds.), Adaptive Hypermedia and Adaptive Web-Based Systems, LNCS 1892, Springer-Verlag, 2000, 109-120.

[14] Hiltz, M. and Kögeler, S. Teaching C++ on the WWW. In Proc. ITiCSE'97, ACM Press, 11-13.

[15] Jiménez-Peris, R. et al. Towards truly educational programming environments. In T. Greening (ed.), Computer Science Education in the 21st Century, Springer-Verlag, 2000, 81-112.

[16] Naharro-Berrocal, F., Pareja-Flores, C. and Velázquez-Iturbide, J.Á. Foundations for the automatic construction of animations and their application to functional programs. In Proc. First Program Visualization Workshop (2001), University of Joensuu, 29-40.

[17] Sommaruga, L. et al. Curso interactivo de programación en Pascal. McGraw-Hill, 1997 (in Spanish).

[18] Stasko, J. et al. (eds.) Software Visualization. MIT Press, 1998.

[19] Stern, L., Sondergaard, H. and Naish, L. A strategy for managing content complexity in algorithm animation. In Proc. ITiCSE'99, ACM Press, 127-130.

[20] Schwarz, E., Brusilovsky, P. and Weber, G. World-wide intelligent textbooks. In Proc. ED-TELECOM'96 (1996), AACE, 302-307.