

Use of Hypermedia Tools for End-User Development

Sebastian S. Ortiz-Chamorro^{1,4}, Gustavo Rossi^{1,2}, and Daniel Schwabe³

¹ LIFIA, Universidad Nacional de La Plata, Argentina

² CONICET, Argentina

³ Departamento de Informática, PUC-Rio, Brazil

⁴ Departamento de Electrónica e Informática, Universidad Católica de Asunción, Paraguay
{sortiz,gustavo}@lifia.info.unlp.edu.ar, dschwabe@inf.puc-rio.br

Abstract. Software development tools aimed at end-users tend to employ various forms of visual programming because these users find textual programming very difficult to learn. However, visual programming has known scalability issues. As an alternative, we propose hypertextual programming; a technique that represents the program as hypertext and allows the user to both browse it and manipulate it mainly by using navigation. This technique leverages the users' ability to navigate in hyperspace, a widely available skill, to edit the program under development. In order to reap the benefits of this technique, adequate hypertextual editors must be built. Many of the lessons learned in the web engineering area can be used to deal with this problem. This paper discusses the state of the current research efforts behind this novel programming technique.

Keywords: hypertextual programming, hypermedial programming, end-user development, interfaces for end-user development, domain-specific languages, web engineering.

1 Introduction

Textual programming languages force the user to “learn the arcane syntax and vocabulary conventions of the language” [2]. Learning a text-based language constitutes a difficult and undesirable challenge for end-users doing software development.

To mitigate this problem, visual programming techniques [4] have successfully been used in end-user development [3]. However, visual programming is known to have scalability problems [5].

The potential length and complexity of the end-user's programs call for a representation of the program code as a set of manageable pieces that the user can both inspect and manipulate.

Hypermedia (a.k.a. hypertext¹) systems [6] provide interactive environments where users can navigate and manipulate a potentially vast network of interconnected pieces of information. This paper discusses the current research efforts on a programming technique based on the use of hypermedia development environments.

¹ We adopt the definitions in which these terms are synonyms.

1.1 Research Objectives

Research Question: How could hypermedia representations of programs be used to build better end-user oriented development tools?

General Objective: explore the potential use of hypermedia-based techniques in the creation of end-user oriented development environments.

Specific Objectives:

- Select, elaborate and integrate a set of concepts, models and tools to be used in the design and implementation of end-user oriented development environments that both use and provide adequate support to the representation of computer programs as hypermedia systems.
- Perform a detailed description of the fundamental features that a hypermedia-based programming environment must have and define adequate vocabulary to refer to these elements, consistent with the concepts used in specialized literature.
- Perform a critical comparison of the developed techniques against other editing and program visualization techniques.
- Select and adapt existing hypermedia development methodologies to be used in the construction of hypermedia-based development environments.
- Construct functional hypermedia-based development environments.
- Study the use of these development tools to assess their efficacy.

1.2 Research Methodology

A technique that aims to construct hypermedia-based software development tools has to be both adequately defined and validated.

In order to define this technique, we will provide: i) a formal definition of hypertextual programming, and ii) a theoretical analysis that compares textual, visual and hypertextual programming. This analysis aims at providing a clear distinction between these techniques.

In order to validate this technique, we will provide two hypertextual development environments that use hypertextual programming in different end-user oriented domains. One of the applications will be used for building general-purpose mashups. The second application, Benefit Catalog, uses hypertextual programming to build interactive rule-driven questionnaires to automatically generate health-care insurance policies.

These applications offer two different kinds of empirical evidence to assess the effectiveness of hypertextual programming. Benefit Catalog is a real-world application used by health insurance domain-experts. The mashup tool will be used in laboratory experiments with end-users. Using different domains will help to assess how general this technique is and will also provide guidelines to identify areas where it may be more effective.

2 Hypertextual Programming

Hypertext has been used in programming before. We reviewed three representative examples [10,11,12]. In general, all of these tools and techniques assume that there are one or more underlying programming languages and use hypertext to rearrange and/or link the potentially different program sources with other documents and products of the

software engineering process [1]. Although these development environments qualify as hypertext systems, they do not use navigation as a means to edit the source code.

2.1 Definition

We defined *hypertextual programming* (HTP, a.k.a. *hypermedial programming*) as a form of programming that uses navigation as the primary tool to inspect and edit the application code, and is supported by a computer system, called a *hypertextual editor* (a.k.a. *hypermedial editor*), that: i) represents the entire program source code as hypertext; and ii) allows all the possible finite language instances to be generated as navigation paths through it [1].

OOHDM [7], a general Web engineering technique, can be used to construct a hypertextual editor for a domain-specific language [1]. Other design guidelines like Web usability [8] and hypermedia design patterns [9,10] can be leveraged as well.

2.2 Comparison with Visual and Textual Programming

HTP does not force the end-user to learn a textual programming language. At the same time, HTP does not appear to have the vertical scalability problems that plague visual programming because large amounts of code can be organized in various interconnected views of the program code, if necessary, at different levels of abstraction [1].

3 Benefit Catalog

As a first example application, we participated in the development of Benefit Catalog, a hypertextual editor that is currently being used by end users to program dynamic health-care insurance policy configuration in a company in the United States [1].

In this development tool, the various program components such as product templates, questions, answers, cost sharing components, benefit options and benefit service-levels are represented as navigational objects. Programs are developed by creating, modifying and removing these objects.

Benefit Catalog is a hypertextual editor that has been used to effectively build complex programs by end-users in the health-care industry. This application is particularly important because the programs built by the end-users are very large and complex. This provides empirical evidence to substantiate the claim that this technique may be used to built scalable developing environments.

We are currently in the process of making a detailed study to quantify the size and complexity of these programs, but our preliminary findings show that the size of these programs—as measured by units of instructions comparable to lines of code—is above 10,000 units. We know of no other documented cases of programs developed by end-users that have this size and complexity.

4 Conclusions and Future Research

Thinking of a Web application as a development tool that both represents and enables the modification of an underlying programming language constitutes a novel application of Web technology.

Throughout this research we introduced the concept of hypertextual programming and provided a formal definition. We also provided guidelines to help construct development environments that employ this technique [1].

We have compared hypertextual programming against visual and textual programming. HTP does not force the user to learn the syntax of a text-based language and also does not suffer from the vertical scalability problems that plague visual programming. However, a more detailed discussion of the differences between these techniques and how they draw on different human abilities to support programming activities is needed.

The proposed research methodology has provisions to substantiate our claims around this technique both on theoretical and on empirical grounds. In the case of empirical evidence, we currently have a significant real-world example of this technique in the form of the Benefit Catalog application. A second hypertextual editor for building mashups and code-named Mashcraft is currently under development. We expect to perform laboratory experiments with this tool. Using these diverse forms of empirical validation will allow us to assess the effectiveness of this technique both under the view of domain-experts that need a practical solution and also under the controlled environment provided by laboratory experiments.

Millions of users navigate the World Wide Web. Hypertextual programming leverages this widely available end-user skill to facilitate the construction of computer programs.

Acknowledgements

Creating, studying and validating this technique constitute the core of a Ph. D.² thesis carried out by Sebastian S. Ortiz-Chamorro at the National University of La Plata in Argentina. The advisors for this thesis are Gustavo Rossi, Ph. D. from the same university and Daniel Schwabe, Ph. D. from PUC-Rio. Developing a hypermedia-based mashup development tool is a sub-part of this investigation that is being carried-out as a research project at the Catholic University of Asuncion, Paraguay. This project has received funding from Conacyt³.

References

1. Ortiz-Chamorro, S.S., Schwabe, D., Rossi, G.: Hypertextual Programming for Domain-Specific End-User Development. In: Pipek, V., Rosson, M.B., de Ruyter, B., Wulf, V. (eds.) IS-EUD 2009. LNCS, vol. 5435, pp. 225–241. Springer, Heidelberg (2009)
2. Cypher, A. (ed.): Watch What I Do: Programming by Demonstration. MIT Press, Cambridge (1993)
3. Lieberman, H., Paternò, F., Klann, M., Wulf, V.: End-User Development: An Emerging Paradigm. In: Lieberman, H., Paternò, F., Wulf, V. (eds.) End User Development, pp. 1–8. Springer, The Netherlands (2006)

² Carrera de Doctorado en Informática.

³ Conacyt, Consejo Nacional de Ciencia y Tecnología (National Council of Science and Technology) is a Paraguayan government agency.

4. Shu, N.: Visual Programming. Van Nostrand Reinhold, New York (1988)
5. Burnett, M.M., Baker, M.J., Bohus, C., Carlson, P., Yang, S., van Zee, P.: Scaling up Visual Programming Languages. *IEEE Computer* 28(3), 45–54 (1995)
6. Conklin, J.: Hypertext: an introduction and survey. *Computer* 20(9), 17–41 (1987)
7. Schwabe, D., Rossi, G.: An Object Oriented Approach to Web-Based Application Design. *Theory and Practice of Object Systems* 4(4) (1998)
8. Nielsen, J.: *Designing Web Usability: The Practice of Simplicity*. New Riders Publishing, Indianapolis (1999)
9. Rossi, G., Schwabe, D., Garrido, A.: Design reuse in hypermedia applications development. In: *Proceedings of Hypertext 1997*, pp. 57–66 (1997)
10. Hypermedia Design Patterns Repository,
<http://www.designpattern.lu.unisi.ch/index.htm>
11. Østerbye, K.: Literate Smalltalk Programming Using Hypertext. *IEEE Transactions on Software Engineering* 21(2), 138–145 (1995)
12. Anderson, K.M., Taylor, R.N., Whitehead, E.J.: Chimera: hypermedia for heterogeneous software development environments. *ACM Transactions on Information Systems* 18(3), 211–245 (2000)
13. Garg, P.K., Scacchi, W.: ISHYS: Designing an Intelligent Software Hypertext System. *IEEE Expert: Intelligent Systems and Their Applications* 4(3), 52–63 (1989)