Nonprogrammer Web Application Development

Jochen Rode
Virginia Polytechnic Institute and State University
3160 Torgersen Hall
Blacksburg, VA 24061, USA
jrode@vt.edu

ABSTRACT

We propose to investigate the feasibility of nonprogrammer web application development. The main target audience for this research is webmasters without programming experience – a group likely to be interested in building web applications. We choose a subset of web applications as the target for our analysis: basic web-based data collection, storage & retrieval applications. We propose to study the mental models of our target audience, collect requirements for a sufficiently powerful end-user programming tool, evaluate new programming paradigms, and implement a proof-of-concept prototype using participatory design techniques.

Author Keywords

End-User Programming, Web Application Development, Web Engineering.

ACM Classification Keywords

D.1.7 Visual Programming. D.2.2 Design Tools and Techniques. H.3.5 Online Information Services. H.5.2 User Interfaces. H.5.4 Hypertext/Hypermedia.

INTRODUCTION

The World-Wide-Web has become an important platform for interactive applications ranging from simple online-surveys, and web-based reference databases to complex transaction services such as online auctions. Even years after the rise of CGI-scripting, the creation of a web application is still difficult, requiring a broad range of skills and experience. The factors contributing to this situation are many and diverse, including the abundance of technologies and standards, inadequate technologies and integration between technologies, inconsistent implementation of standards and differences in end-user platforms. Despite the problems, *professional web programmers* are at least able to create interactive applications, while *non-programmers* are so far limited to creating static websites. Our preliminary studies [6] suggest that these limitations are not due to lack of interest but rather due to the difficulties inherent in interactive web development. Given the right tools and techniques even nonprogrammers may become web application developers.

By making web development possible for a wider audience, we may see a greater variety of useful applications, including applications not yet envisioned. For organizations that cannot afford a professional programmer, end-user programming (EUP) may help to streamline workflows, increase productivity and client satisfaction. Indeed, the WWW in general is an excellent example of what can happen when technology becomes accessible. Deshpande and Hansen [3], the co-founders of the web engineering discipline, argue that we need to "devise methods and processes to assist end users" which would help to increase the reliability of applications and "release the creative power of people." Our high expectations of EUP have been encouraged by positive user-reviews of VT Survey [5] - a basic web-based EUP tool that, alone at our organization, helps over 3000 users with the creation of online surveys.

We propose to make web development more accessible to nonprogrammers using a number of different but complementary approaches: by limiting scope, providing high-level abstractions, and employing a suitable programming paradigm.

SCOPE AND RESEARCH METHODS

We suggest that a good starting point is to analyze and develop tools for sophisticated end-users, individuals who have experience in web design but not in web programming.

Our preliminary research [6] indicates that a substantial fraction of end-users' needs for web applications are often quite simple and similar in nature. In a survey of webmasters at Virginia Tech we found that about one third (34%) of the web applications described by these users are *basic data collection, storage and retrieval applications*

CHI 2004, April 24-29, 2004, Vienna, Austria.
ACM 1-58113-703-6/04/0004.

such as surveys, registration systems, service request forms, or staff databases. This subset is particularly interesting to us, because functionality of such limited scope seems quite reasonable to provide via an end-user programming tool (25% were needs for advanced custom web applications; 41% were needs that could be addressed via generic web applications such as resource scheduling, shopping cart and payment, message board, content management, calendar). In order to validate our findings we intend to conduct a similar survey with a larger population.

For capturing the requirements for a programming tool that is targeted at the specific needs of its audience we plan to employ participatory design techniques as advocated by research in customization and tailorability [e.g. 9].

Clearly, the choice of the programming paradigm will have a big influence on the ease-of-use and efficiency of development. We plan to develop and evaluate a programming paradigm which is grounded in the knowledge of end-user programming. We call it "programming-at-runtime". This concept builds from the ideas of direct manipulation [8] and the "debugging into existence" behavior [7] studied in professional programmers. In its core it is similar to the automatic recalculation aspect in spreadsheet programs. A critical piece of the concept is that the user is able to both develop and use the application without switching back and forth between programming and runtime modes. That is, the application is always usable to the fullest extent that it has been programmed, and when its boundaries are tested, the environment provides useful feedback suggesting next steps for the "programmer" to take.

In order to better understand the end-users' mental models we adapt the methods of Pane, Ratanamahatana and Myers [4], who designed a "natural" programming language for children by first studying how children and adults use natural language to solve programming problems.

EXPECTED OUTCOMES

The research plan and expected outcomes are interdisciplinary, connecting and leveraging the two general areas of HCI and software engineering:

- An analysis of end-users' needs for web applications
- An analysis of the components, concepts and functionality of typical basic web applications
- An analysis of end-users' mental models, expectations and understandings of web technologies
- An analysis of critical obstacles in current web application development, obtained from experts
- An experimental study to test the efficacy of the "programming-at-runtime" paradigm
- Development and evaluation (formative; field studies) of a web development tool for nonprogrammers

We have finished the initial needs analysis [6], interviewed experienced web programmers, and began to study the mental models of our audience. The prototype development is in its early stages and will serve as an instrument to experimentally assess the viability of programming-at-runtime and shape the list of requirements and possible solutions for web development tools suitable for end-users.

RELATED WORK

Even before the rise of the web, end-user programming of basic data management applications has been a topic in research and industry, with Apple's HyperCard a well-known and successful EUP tool. With respect to web applications, research projects such as WebFormulate [1], or FAR [2], and commercial tools like Microsoft FrontPage, Macromedia Dreamweaver, www.codecharge.com have shown the viability of the idea. While being sufficiently powerful, most current commercial tools do not offer the level of ease-of-use we envision. Furthermore, the rationale behind their designs is not public and therefore difficult to reuse.

ACKNOWLEDGEMENTS

I thank my co-advisors Dr. Mary Beth Rosson and Dr. Pérez-Quiñones for their continuing support.

REFERENCES

1. Ambler, A. and J. Leopold (1998). Public Programming in a Web World. Visual Languages. Nova Scotia, Canada.
2. Burnett, M., S. K. Chekka, et al. (2001). FAR: An End-User Language to Support Cottage E-Services. IEEE HCC 2001; Stresa, Italy.
3. Deshpande, Y., S. Hansen (2001). "Web Engineering: Creating a Discipline among Disciplines." IEEE MultiMedia 8(2): 82-87.
4. Pane, J. F., C. A. Ratanamahatana, B.A. Myers (2001). Studying the language and structure in non-programmers' solutions to programming problems. International Journal of Human-Computer Studies. 54: 237-264.
5. Rode (2001). VT Survey. http://opensource.isc.vt.edu /products/survey/
6. Rode, J. and M. B. Rosson (2003). Programming at Runtime: Requirements & Paradigms for Nonprogrammer Web Application Development. IEEE HCC 2003. Auckland, New Zealand. Oct. 28-31
7. Rosson, M. B., J. M. Carroll (1996). The reuse of uses in Smalltalk programming. ACM TOCHI 3(3): 219-253
8. Shneiderman, B. (1983). Direct Manipulation: A Step Beyond Programming Languages. IEEE Computer. 16: 57-60.
9. Stiemerling, O., H. Kahler, V. Wulf (1997). How to Make Software Softer - Designing Tailorable Applications. Symposium on Designing Interactive Systems. 1997. 365-376.