API Usability:
CHI'2009 Special Interest Group Meeting

John M. Daughtry
Applied Research Laboratory
& College of Info. Sci. & Tech.
Pennsylvania State Univ.
University Park, PA 16802
daughtry@psu.edu
http://john.daughtryhome.com

Umer Farooq
User Experience Researcher
Microsoft Corporation
One Microsoft Way
Redmond, WA 98052
umfarooq@microsoft.com

Jeffrey Stylos
Computer Science Department
Carnegie Mellon University
5000 Forbes Ave
Pittsburgh, PA 15213
jsstylos@cs.cmu.edu

Brad A. Myers
Human Computer Interaction Inst.
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213-3891
bam@cs.cmu.edu
http://www.cs.cmu.edu/~bam

CHI 2009, April 4 – 9, 2009, Boston, Massachusetts, USA
ACM 978-1-60558-247-4/09/04.

Abstract
Programmers of all types from novice end-user developers to professional software engineers make use of application programming interfaces (API) within their various designs. And, while the use of these interfaces is ubiquitous, there is little research about their design. Recently, a number of researchers and practitioners have begun to treat API design as a first-order object of study and practice. The purpose of this special interest group meeting is to bring together the community of usability researchers and professionals interested in API usability. The time will be used to discuss attendees' ideas and opinions in order to stimulate this new and exciting emerging field that crosses the boundaries between human-computer interaction and software engineering.

Keywords
Psychology of Programming, Empirical Studies of Programmers (ESP), Natural Programming

ACM Classification Keywords
D.2.2: Software Engineering: Software Libraries

Introduction
The human-computer interaction community has long held an interest in the programming activity (e.g., [17,

A well-designed API can be a great asset to the organization that wrote it and to all who use it. Good APIs increase the pleasure and productivity of the developers who use them, the quality of the software they produce, and ultimately, the corporate bottom line. Conversely, poorly written APIs are a constant thorn in the developer's side, and have been known to harm the bottom line to the point of bankruptcy. Given the importance of good API design, surprisingly little has been written on the subject.

~Joshua Bloch [14]

18, 11]). Indeed, in the early years of the Conference on Human Factors in Computing Systems (CHI) many papers involved the study of programmers or had implications for programming. Further, there were more directed communities such as the Empirical Studies of Programmers workshop series (ESP). More recently, this work is reflected in the CHI community in the form of end-user programming (e.g., [13, 23]) and studies of programming in the broad (e.g., [14, 5]).

Despite the long and productive history of usability research in programming, the API has only recently been treated as a first-order artifact within the activity. Early work that foreshadowed this development includes studies of reuse (e.g., [16]).

APIs, frameworks, libraries, SDKs, toolkits all involve code in which an interface is exposed as a method of intended reuse by others. In this special interest group we use the term API broadly to indicate all of these.

Over time, it has become increasingly apparent that a theoretical base for API usability could have a great impact on the software development profession. From studies of programming (e.g., [16]) researchers began to recognize and condemn poor usability decisions in programming (e.g. – [15]) and recognize that the techniques and theory developed for usability should be applied directly to the API [12].

McLellan and colleagues [12], pioneering practitioners in this area, emphasize the importance of conducting API usability, comparing it to the cornerstone processes of running usability studies to evaluate Graphical User Interfaces (GUIs). They further highlighted the importance that documentation plays with respect to usability in API design.

Within a few years, it was reported by Clarke that this approach had been adopted by Microsoft by way of evaluating APIs as part of usability lab studies [6, 7]. His argument for studying API usability reiterated the points made by McLellan and colleagues. These reports distinguish themselves from the prior argument by briefly presenting their utilization of the cognitive dimensions framework. For example, one of the dimensions is "Abstraction Level", which describes the types of abstractions that the API exposes. These reports have had an impact by motivating interest in API usability, but the higher-level models (i.e. – personas) have not been made public.

Joshua Bloch at Google and Krzysztof Cwalina at Microsoft are two other notable leaders in the area of advocating the importance and application of API usability to improve the overall programming user experience (e.g., [3, 4, 8]).

Current Work

Recently, progress in the field has taken the form of academic research. Much of this work examines the usability of specific design patterns and programming paradigms as applied to the design of APIs for the purposes of enhancing software productivity. For example, Stylos and Myers map out the space of API design decisions [19]. Along with colleagues, they have examined the usage of the factory design pattern [10], redesigned APIs with respect to usability considerations [20], identified usability challenges in designing service-oriented APIs [2], studied the trade-offs for specific design options in object-oriented APIs [21], and

Potential Discussion

Building a Framework of API Usability

Challenges Evaluating API Usability

Usability and Abstraction

Metaphor

Organization of members

Intelligent APIs

Naming of Members

Rational & Creative Design

Types of APIs

evaluated design choices against usability evaluations [22]. Most importantly, the sum of this work has empirically confirmed the hypothesis that API usability is a significant problem for all programmers, from novices learning to program to experts programming professionally. In addition to the production of this research group, APIs have been studied with respect to the visualization of naming choices [1] perceived self-efficacy of use [9]. Exploratory work reflects other dimensions of API design such as, the challenges of intelligent APIs, the relationship between creativity and design rationale with respect to API design, and naming.

Special Interest Group

The goals of the special interest group are to bring together the community of usability researchers and professionals interested in API usability, and to solicit and discuss attendees' ideas and opinions in order to stimulate this new and exciting emerging field that crosses the boundaries between human-computer interaction and software engineering. In addition to discussing aspects of API usability, a cross-cutting discussion as to the positioning of API design and usability research with respect to the CHI community would be productive. Despite the recent production of API usability research, there has been no paper on API research at CHI. Further, it has been a long time since the pioneering API work has appeared in related journals such as ACM Transactions on Computer-Human Interaction.

Discussions could potentially delve into building a theory of API usability, unique challenges studying API usability, the relationship between usability and abstraction, metaphor, organization of members and modules, intelligent APIs (e.g. - those that do unexpected things by definition), naming of members and modules, the relationship between rational & creative design, and distinctions between varying types of APIs.

This special interest group will be of interest to those who have an interest in the psychology of programming, those who recognize the increased prominence and lowering usage barriers for external facing APIs (e.g. – APIs used in end-user programming such as Mashup development), those who expose high-power but challenging technologies (e.g. – AI algorithms), and those with experience designing APIs.

Example citations

[1] Anslow, C., Noble, J., Marshall, S., and Tempero, E., "Visualizing the Word Structure of Java Class Names", *Proc. OOPSLA*, 2008, 777-778.

[2] Beaton, J., Jeong, S.Y., Xie, Y., Stylos, J., and Myers, B.A. "Usability Challenges for Enterprise Service-Oriented Architecture APIs". *Proc. VL/HCC 2008*, IEEE Press, pp. 193-196.

[3] Bloch, J., *Effective Java: Programming Language Guide*, Addison-Wesley, 2001.

[4] Bloch, J., "How to Write a Good API and Why it Matters", Keynote Address for LCSD workshop at *OOPSLA*, 2005. Introduction and slides available at http://lcsd05.cs.tamu.edu/#keynote.

[5] Cherubini, M., Venolia, G., DeLine, R. and Ko. A. J. (2007). "Let's Go to the Whiteboard: How and Why Software Developers Draw Code", *Proc. CHI*, 557-566.

[6] Clarke, S., *API Usability and the Cognitive Dimensions Framework*. 2003. Available at http://blogs.msdn.com/stevencl/archive/2003/10/08/57040.aspx.

[7] Clarke, S., "Measuring API Usability". *Dr. Dobbs Journal*, May 2004, S6-S9.

[8] Cwalina, K., Abrams, B., *Framework Design Guidelines: Conventions, Idioms, and Patterns for Reusable .NET libraries*, Addison-Wesley, 2005.

[9] Daughtry, J.M. *Enabling Use: The Effects of Scenarios and Claims on the Self-Efficacy of Using Application Program Interfaces.* Unpublished master's thesis, The Pennsylvania State University, University Park, 2006.

[10] Ellis, B., Stylos, J., and Myers, B.A. "The Factory Pattern in API Design: A Usability Evaluation", *Proc. ICSE 2007*, ACM Press, pp. 302-312.

[11] Norcio, A.F., "Indentation, Documentation, and Programmer Comprehension", *Proc. CHI*, 1982, 118-120.

[12] McLellan, S.G., Roesler, A.W., Tempest, J.T., and Spinuzzi, C.I., "Building More Usable APIs", *IEEE Software*, 15(3), 1998, p. 78-86.

[13] Myers, B.A, Bernett, M.M., Rosson, M.B, Ko, A.J., and Blackwell, A, "End User Software Engineering: CHI'2008 Special Interest Group Meeting, *CHI 2008 Extended Abstracts*, 2008, 2371-2374.

[14] Patel, K, Fogarty, J., Landay, J.A., and Harrison, B., "Investigating Statistical Machine Learning as a Tool for Software Development". *Proc. CHI*, 2008, Florence, Italy, pp. 667-676.

[15] Pemberton, S., "Programmers are Humans Too, 2" SIGCHI Bulletin, 29(3), 1997, p. 64.

[16] Rosson, M.B., and Carroll, J.M., "The Reuse of Uses in Smalltalk Programming", *ACM Transactions on Computer-Human Interaction*, 3(3), pp. 219-253.

[17] Shneiderman, B., Software Psychology: Human Factors in Computer and Information Systems. 1980, Cambridge, MA: Winthrop Publishers.

[18] Soloway, E., Ehrlich, K., and Bonar, J., "Tapping into tacit programming knowledge", *Proc. CHI*, 1982, 52-57.

[19] Stylos, J. and Myers, B.A. "Mapping the Space of API Design Decisions", *Proc. VL/HCC 2007*, IEEE Press, 50-57.

[20] Stylos, J., Busse, D., Graf, B., Ziegler, C., Ehret, R., and Karstens, J., "Making Code Easier to Reuse: API Design for Improved Usability", *Proc. VL/HCC 2008*, IEEE Press, pp. 189-192.

[21] Stylos, J. and Clarke, S., "Usability Implications of Requiring Parameters in Objects' Constructors", *Proc. ICSE 2007*, ACM Press, pp. 529-539.

[22] Stylos, J., Clarke, S., and Myers, B.A. "Comparing API Design Choices with Usability Studies: A Case Study and Future Directions", *Proc. PPIG 2006*, pp. 131-139.

[23] Subrahmaniyan, N., Beckwith, L., Grigoreanu, V., Burnett, M., Wiedenbeck, S., Narayanan, V., Bucht K., Drummond, R., Fern, X., "Testing vs. code inspection vs. what else?: male and female end users' debugging strategies", *Proc. CHI*, 2008, 617-626.