Open Source as a Test Bed for Evaluating Creativity Support Tools

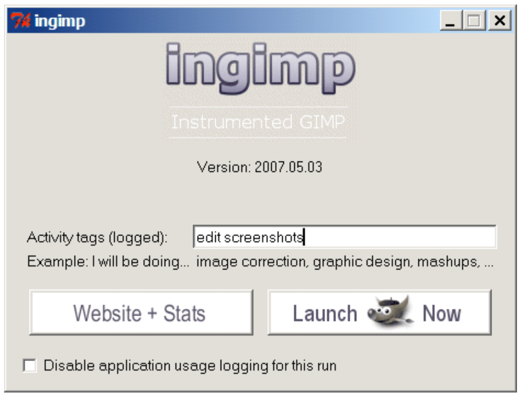
Michael Terry
David R. Cheriton School of Computer Science
University of Waterloo, Waterloo, ON, Canada
mterry@cs.uwaterloo.ca

ABSTRACT
A wide range of creativity support tools have been developed by the research community. However, few of these tools have been evaluated in real-world contexts for significant periods of time. This paper proposes mature open source applications as a means for conducting large-scale, longitudinal studies of creativity support tools in authentic situations. In particular, we argue that grafting a creativity support tool onto a mature open source application enables one to perform evaluations within highly authentic environments. Based on our work developing and deploying ingimp, an instrumented version of the open source GNU Image Manipulation Program (GIMP), we show how an instrumented open source application can support this kind of research in "the wild." We also provide evidence for the feasibility of this approach by considering the relative success of ingimp in attracting a sizable user base.

INTRODUCTION

Throughout its relatively young history, the computer has frequently been envisioned and positioned as an amplifier of our cognitive capabilities. Vannevar Bush [2], J.C.R. Licklider [9], Douglas Engelbart [5], and Alan Kay have all developed highly compelling visions of how the computer can tightly integrate with and extend our cognitive capabilities. Engelbart's goals to augment human intellect nicely reflect these individuals' aspirations [5]:

> *By "augmenting human intellect" we mean increasing the capability of a man to approach a complex problem situation, to gain comprehension to suit his particular needs, and to derive solutions to problems…*
>
> *[We seek] more-rapid comprehension, better comprehension, the possibility of gaining a useful degree of comprehension in a situation that previously was too complex, speedier solutions, better solutions, and the possibility of finding solutions to problems that before seemed insoluble… (p. 1)*



Version: 2007.05.03

Activity tags (logged): edit screenshots
Example: I will be doing... image correction, graphic design, mashups, ...

☐ Disable application usage logging for this run

Figure 1. ingimp is an instrumented version of the GNU Image Manipulation Program. ingimp automatically collects and transmits usage data to a public web server, where anyone can download and analyze the data.

Research in creativity support tools continues in this spirit, with a specific focus on designing tools that not only enable people to better solve problems[1], but to do so in a way that produces truly original, creative results.

Numerous computational tools have been developed to support the creative process, but the research community still wrestles with two fundamental questions [14]: What is important in the design of a creativity support tool, and how can these tools be properly evaluated? This paper focuses on the latter problem of evaluating creativity support tools.

[1] Here we use the notion of "problem" in a very broad and general sense, encompassing everything from design problems, to scientific problems, to the "problem" of creating an artistic artifact

Evaluating creativity support tools presents significant challenges due to the nature of the process itself: It is difficult to measure creativity, much less a tool's impact on the creative process. Creative endeavors also typically occur within well established, interconnected systems of people, tools, and practices. Thus, while short-term tests of proof-of-concept tools can show the promise of a tool, it is difficult to understand how a new tool will integrate into the larger creative problem solving *system*. Longitudinal studies are an obvious way to explore these issues, but it is costly to move a research system from a proof-of-concept to a production-quality system that can truly integrate with real-world problem solving environments.

This paper proposes mature open source software as a means for evaluating creativity support tools in meaningful, real-world contexts. Specifically, we propose grafting creativity support tools onto existing end-user applications to understand how well novel tools integrate with existing work practices. Using our work designing and deploying ingimp [17], an instrumented version of the open source application GNU Image Manipulation Program (GIMP) [8], we demonstrate the feasibility of this approach, discuss issues relevant to researchers considering this tack, and enumerate particular advantages to evaluating research tools within existing, mature open source applications.

The rest of this paper is structured as follows. First, we define the concept of a creativity support tool and review the various classes of these tools. Next, we consider the challenge of evaluating creativity support tools and suggest the need for long-term evaluations. We then propose open source software as a means by which one can conduct longitudinal studies of creativity support tools. We discuss the advantages to this approach, describe how we are exploring this method via ingimp, and reflect on the feasibility of this approach by conveying our experiences deploying ingimp to the public. We conclude with directions for future work.

BACKGROUND

Classes of Creativity Support Tool

In his article exploring ways to support innovation, Shneiderman enumerates eight ways that computers can assist creative processes [13]. For example, computers can assist with "what if?" explorations or by provide visualizations to aid comprehension of data. From these eight activities, as well as a consideration of prior work in creativity support tools, we can consider seven general categories of creativity support tools, each of which we describe in turn:

1. *Autonomous* creative tools
2. *Creative activity* tools
3. Domain-specific tools with creativity scaffolding
4. Domain-specific tools with limited or no scaffolding
5. Domain-independent process support tools
6. Computer-supported collaborative work (CSCW) tools
7. Research tools

Autonomous creative tools are intended to be *independently* creative, yielding creative results with little-to-no human intervention required. Cohen's Aaron is an example of an autonomous creative tool [11]: Aaron creates new, original paintings with few demands placed on a human. In the strictest sense, these applications are not truly creativity *support* tools since they are designed to work autonomously (rather than in conjunction with one or more individuals). We mention them here only to say they fall outside the scope of this paper.

Creative activity tools are tools designed to support activities intended to help people be more creative. Brainstorming and concept mapping are both examples of activities intended to help foster creative thinking about a problem. Creative activity tools explicitly support these practices and can help structure them.

Domain-specific tools with creativity scaffolding are tools designed to create and manipulate domain-specific data, with additional support to help individuals reach creative outcomes faster and more reliably. For example, Final Draft [6] provides a suite of tools to assist with screenwriting, including a set of templates to help with the initial structuring of a story. MasterWriter [10] provides a set of tools useful for songwriters, including a rhyming dictionary, alliteration dictionary, and pop-culture dictionary. These dictionaries cater to common creative practices of songwriters, thus aiding *creative practices*, rather than simply providing tools to create and edit content.

Domain-specific tools with limited or no scaffolding, on the other hand, do not provide tools to explicitly scaffold creative practices, but do provide tools to work with data at high levels of abstraction. At first, this capability may seem to have little to do with the creative process. However, working with data at greater levels of abstraction frees one to think about high-level details and goals. For example, Photoshop [1] provides a wide range of general-purpose tools to manipulate bitmap images, where one can think about operations on these images in high-level terms, such as adjusting an image's brightness and contrast, rather than in terms of the formulae necessary to manipulate each pixel in the desired fashion.

Domain-independent process support tools do not scaffold the creation or editing of data, but rather support common problem solving *practices* within the creative process. For example, one must often create and evaluate a set of alternatives when highly creative outcomes are desired. Tools such as undo and revision control systems (e.g., CVS [4]) both help scaffold these practices by enabling an individual to explore possibilities, without losing work.

CSCW tools build bridges between people, making collaborations easier. Revision control systems can provide this support, as can general-purpose communication tools, such as email, newsgroups, or wikis. Like the previous two types of creativity support tools, these tools do not necessarily provide functionality to explicitly scaffold creative practices, but they do help catalyze the social aspects of the creative process.

Finally, *research tools* assist with the process of researching what has previously been done in a field, or disseminating results. Search engines and citation tracking systems are both examples of tools that help individuals more quickly and thoroughly understand prior work in a field.

Across this range of creativity support tools, one of the significant challenges faced by toolmakers is assessing how their tool affects the creative process. We consider this problem next.

Evaluating Creativity Support Tools

Creativity support tools can be assessed via a range of traditional HCI techniques, including controlled studies, field studies, surveys, and deep ethnography [14]. To date, much of the research in creativity support tools has employed relatively short-term qualitative studies using proof-of-concept systems. These short-term studies constitute an important first step for testing the feasibility of a particular approach: They suggest what is promising about a design, while uncovering what needs to be improved.

Once one establishes the basic utility of a creativity support tool (e.g., users are satisfied with the tool, it appears to positively augment their creative process, etc.), the next step is to understand long-term use in authentic situations. However, these types of studies are rare, in part because of the costs associated with conducting longitudinal studies in real-world contexts. In particular, it is costly to implement production-quality tools. Furthermore, there are numerous logistic costs associated with running long-term studies.

The move from a proof-of-concept to a production-quality system represents significant costs: It takes time and effort to create a production quality version of the research tool. However, to be truly useful, one must also provide all other functionality necessary to do useful work on a day-to-day basis. This need further increases the costs of creating a real-world application for testing.

The logistics of conducting a long-term study also incur costs. One must recruit subjects and devise methods to collect, store, and analyze longitudinal data. With respect to recruitment, there is the burden of convincing subjects of the advantages of testing out a new tool, which will need to be learned and integrated into existing work practices.

Given the costs associated with conducting these long-term, real-world studies, it is no surprise that they are rare[2]. However, in spite of these costs, such studies are quite likely one of the most valuable ways to evaluate creativity support tools, as we explain next.

One of the most significant challenges in evaluating *any* creativity support tool is assessing its impact on the creative process. Since there is no way to directly measure creativity, assessment must consider multiple dimensions: efficiency, cognitive load, users' subjective assessments of the tools' value, and, most importantly, the creative output of the individual. However, measurement of the last dimension, creative output, is further confounded by the problem that truly creative work typically unfolds over significant stretches of time – days, weeks, months, or years [3].

[2] There are, of course, exceptions, such as the Lifelong Kindergarten group at MIT's Media Lab, which has deployed a number of hardware and software packages on the web and through computer clubhouses

Accordingly, understanding a creativity support tool's impact on one's creative output requires one to apply the tool across the potentially lengthy creative process. That is, until we can predict what types of tools and features reliably enhance the creative process, we *must* conduct long-term, authentic evaluations of creativity support tools to understand their ultimate effect on the creative process. In other words, we must follow the lead of Engelbart as he developed his revolutionary NLS system and continually analyze how users apply tool on a day-to-day basis in production environments. Open source software provides a catalyst to conduct precisely these types of studies by lowering the costs associated with these studies.

OPEN SOURCE SOFTWARE AND CREATIVITY SUPPORT TOOLS

Open source software, by definition, refers to software that includes access to the software's source code. Numerous philosophies and licenses intersect with this basic concept (e.g., free software [7] vs. open source software [12]), but all include the basic provision of providing access to the software's underlying source code. As we argue below, this access to the software's source code uniquely positions open source software as a vehicle for conducting long-term evaluations of creativity support tools.

The ability to freely alter an application is a powerful property of open source software. In the context of this paper, this capability means researchers can build on the work of others and graft their research tools into existing, mature end-user applications. This possibility opens new doors for evaluating research tools within production-quality, feature-rich applications as it lessens the burden of creating a truly useful, real-world application. While there are still costs associated with embedding one's tools in a production-quality application, they are arguably less than the cost of creating the entire application from scratch.

The availability of source code not only allows one to embed new tools within an application, it also allows one to instrument the application to collect longitudinal usage data. For example, deploying an instrumented application that includes novel creativity support tools allows one to determine whether the new tools are routinely used, or whether they are abandoned in favor of existing techniques. One can also determine how much a tool is used, its potential value in the problem solving process, what tools it is used in conjunction with, and the ways in which it might affect the problem solving process.

In the strictest sense, open source software refers to a particular type of software license. However, there also exists an open source software community. This community counts among its members many "early adopters," or people who routinely seek out and experiment with new technology. These individuals are noteworthy because they can provide some of the initial assessments of novel tools, with little need to convince them to "try out" the new systems. This "feature" of open source software further reduces the costs of longitudinal studies by making it easier to find subjects for testing new tools.

Given these favorable properties for conducting research with open source applications, we now describe our own work in this area. As we will show, we have evidence that suggests the feasibility of this particular approach.

ingimp: An Instrumented Open Source Application

As part of research into the usability open source software, we developed ingimp [17], an instrumented version of the GNU Image Manipulation Program (GIMP[3]) [8]. ingimp is a snap-in replacement for GIMP that users download, install, and use just like GIMP. In fact, with the exception of one additional start-up screen (Figure 1), end-users will perceive ingimp to be identical to the official version of GIMP. However, in the background, ingimp automatically logs basic interaction data – users' computing setup, the commands used, high-level features of their documents (e.g., number of layers, image size, but not pixel data), and so on. These data are automatically transmitted to a central server (http://www.ingimp.org) where they are summarized for the public. All collected logs are also available to enable public data analysis.

ingimp was designed with two primary purposes in mind. First, ingimp is an attempt to help the open source community understand how their software is used in practice. For general-purpose applications, such as image manipulation programs or office suites, there is a range of ways these applications could be applied in practice. At the same time, there will always be more bugs and feature requests than time to address these issues. Understanding real-world usage

[3] GIMP is an application similar in functionality to Adobe Photoshop

patterns can help prioritize development efforts so they serve the greatest number of users.

Second, ingimp was designed to support human-computer interaction research. In particular, we wanted to create an infrastructure specifically designed to allow HCI researchers to test new ideas in the context of a real-world application. We describe these capabilities next.

ingimp: Features for Researchers

ingimp's design provides features to assist researchers with logging data, labeling of log data, and subsequent data analysis.

Data Logging

The entire ingimp system (the instrumented application and associated website) provides a complete data collection infrastructure: ingimp automatically collects high-level data about its usage and makes all collected data publicly available on the ingimp website. By setting a flag within a configuration file, researchers can also elect to keep local copies of the generated logs. The availability of a central data collection service means researchers can embed new tools within ingimp (as described below), make the software publicly available, then analyze the data as it becomes available on the website; little-to-no effort is required on their part to collect log data.

Embedding new tools within GIMP (and by extension, ingimp) is possible via GIMP's well-defined plug-in architecture. This plug-in architecture is significant for researchers because it further lowers implementation costs. In particular, the plug-in architecture helps one avoid the need to alter (or understand) the application code. Instead, one need only understand the plug-in API. For many types of creativity support tools (e.g., domain specific tools with or without creativity scaffolding), this plug-in architecture may be sufficient. However, for other types of tools (e.g., process support tools), the developer may need to actually alter application code. In both cases, ingimp will automatically log any command invoked by the user, whether it is part of the original system or added by a third party. Thus, logging burdens on the researcher are also lessened.

Data Labeling

In addition to basic logging services, ingimp provides two means by which researchers can introduce arbitrary information to log data (such as experimental study data when conducting controlled experiments). The first means for adding data to a log is by using Activity Tags. When ingimp first starts, its start screen includes a field where users can describe how they intend to use the software (Figure 1). We call these descriptions *Activity Tags*. Any data entered into the Activity Tag field is copied directly into the log file, enabling one to add markers to data to help locate and differentiate logs.

The second means of injecting arbitrary information into a log file is via a text file. By editing ingimp's preference file, users can specify a text file whose contents will be copied directly into the log file when ingimp is started. Again, this mechanism is useful if researchers wish to include information to help them locate or differentiate log data. For example, one could conduct a study in which subjects use different tools for each test condition in the study. In this case, the researcher can specify different text files for each subject, where each text file contains data describing the specific experimental setup for the subject.

Notably, neither Activity Tags nor the text file require the researcher to edit the application's source code to include additional log data.

Data Analysis

The final feature of ingimp's design relates to data analysis. When initially released, we only provided a basic set of data summaries on the ingimp website and access to the raw log files. However, it soon became clear that there was a need to provide tools for the entire community to more easily analyze the collected data. Accordingly, we developed Stats Jam [16], an extension to MediaWiki that allows one to embed SQL queries within wiki text. Using Stats Jam, one can not only mine the data, but also specify how the returned data should be rendered within the web page (e.g., a table, graph, plot, etc.). This final component of the ingimp system allows one to take advantage of ingimp's automatic data collection facilities and more easily analyze the data collected. Furthermore, by embedding these capabilities within a wiki, collaborative analysis is facilitated, whether between researchers, or between the researcher and the community of users themselves.

Applying ingimp's Feature Set to Research Problems

Collectively, ingimp's feature set provides one example of how an open source application can be tuned to support research efforts. We have already started making use of this feature set to conduct experiments in the wild. Specifically, we have used these features to test new work in illustrated consent agreements.

Illustrated Consent Agreements

One of the issues we encountered in conducting this work is that of informed consent. ingimp includes an informed consent form to which users must agree before using the software. However, the consent form is written in English and the majority of our users currently are non-native speakers of English. Given our desire to truly inform our users about the implications of using this research software, we developed a series of wordless diagrams to accompany the text-based consent agreement. These diagrams describe what the software does (i.e., the fact that it collects data on usage) to increase the chance that users understand the implications of using the software.

After running tests of the new diagrams locally, we embedded the diagrams within ingimp and included code that randomly determines whether or not to show the diagrams to users. In essence, we are conducting an experiment of this new functionality, using ingimp as the test-bed for data collection and analysis. Based on the data returned, we were able to understand basic usage patterns of these new wordless diagrams and whether people seem to pay more attention to them than the text-based consent agreement. These data, in turn, are now feeding into new iterations of the illustrated consent agreements.

Assessing the Feasibility of the Approach

ingimp was initially released in May, 2007. In the year since its release, there have been over 800 installations and nearly 5,000 log files collected, collectively representing over 500,000 data points. Reception to the project has been highly positive: The open source community is extremely receptive to the idea of collecting information about how their software is used in practice and they seem to be extremely willing to explore new systems. As such, we believe this approach holds great promise in evaluating creativity support tools. To be sure, there is a significant amount of time required to produce a production-quality data collection environment, but once constructed, it becomes a useful test-bed for conducting research. However, one may find that they can tap into an existing open instrumentation effort, circumventing the need to implement an instrumentation infrastructure. For example, one can embed tools in ingimp and take advantage of its infrastructure, while the recent release of a Firefox instrumentation extension [15] opens up the possibility of conducting research in web browsers. Just as importantly, the freely available Stats Jam provides an application-independent data analysis system.

FUTURE WORK

This paper argues for the use of open source software applications as a substrate within which one performs long-term evaluations of creativity support tools in authentic conditions. ingimp demonstrates the feasibility of this approach and also shows the willingness of the open source community to take part in this research. Given this initial success, we encourage researchers in this area to seriously consider this approach: Image manipulation encompasses a range of creative activities all waiting to be augmented by creativity support tools. ingimp and its source code are publicly available, the logs are publicly available, and Stats Jam provides the tools for data analysis, all that is needed are your tools to enhance our creative powers.

REFERENCES

1. Adobe Photoshop. http://www.adobe.com
2. Bush, V. 1996. As we may think. *interactions* 3, 2 (Mar. 1996), 35-46.
3. Csikszentmihalyi, Mihaly. 1996 *Creativity: Flow and the Psychology of Discovery and Invention*. Harpercollins.
4. CVS. http://en.wikipedia.org/wiki/Concurrent_ Versions_System
5. Engelbart, D.C. "Augmenting Human Intellect: A Conceptual Framework", Summary Report AFOSR-3233, Stanford Research Institute, Menlo Park, CA, Oct 1962.
6. Final Draft. http://www.finaldraft.com
7. The Free Software Foundation. http://www.fsf.org
8. GNU Image Manipulation Program (GIMP). http://www.gimp.org
9. Licklider, J.C.R. "Man-Computer Symbiosis", *IRE Transactions on Human Factors in Electronics*, vol. HFE-1, 4-11, Mar 1960.
10. MasterWriter. http://www.masterwriter.com
11. McCorduck, P. 1991 *Aaron's Code*. W. H. Freeman & Co.
12. Open Source Initiative. http://www.opensource.org
13. Shneiderman, B. 2000. Creating creativity: user interfaces for supporting innovation. *ACM Trans. Comput.-Hum. Interact.* 7, 1 (Mar. 2000), 114-138.
14. Shneiderman *et al*. 2005. Report of Workshop on Creativity Support Tools. Available at: http://www.cs.umd.edu/hcil/CST/report.html

15. Spectator. http://wiki.mozilla.org/Spectator
16. Stats Jam. http://sourceforge.net/projects/statsjam
17. Terry, M., Kay, M., Van Vugt, B., Slack, B., and Park, T. 2008. Ingimp: introducing instrumentation to an end-user open source application. In Proceeding of the Twenty-Sixth Annual SIGCHI Conference on Human Factors in Computing Systems (Florence, Italy, April 05 - 10, 2008). CHI '08. pp. 607-616.