Techniques and heuristics for improving the visual design of software agreements

by

Matthew Kay

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Master of Mathematics
in
Computer Science

Waterloo, Ontario, Canada, 2010

Author’s declaration

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners. I understand that my thesis may be made electronically available to the public.

Abstract

End users must regularly agree to lengthy software agreements prior to installing software or using software services. However, despite the fact that these agreements contain terms of direct concern to users—e.g., data collection policies—software agreements are typically read by less than 2% of the population [30]. This thesis presents techniques and heuristics for improving the presentation and visual design of software agreements, to better capture reader attention and improve comprehension. In contrast to other techniques, these techniques are applied to the full agreement content, rather than a summary, as summaries have been found to distract readers from the full content of the agreement [44,56].

This thesis introduces two techniques for improving software agreements: *narrative pictograms* and *textured agreements*. Narrative pictograms are a pictorial technique designed to improve the communication of agreement terms to non-native readers of the language of an agreement. An experimental study shows that they can successfully communicate the basic concepts of a data collection policy without words. Textured agreements are visually redesigned software agreements that highlight information relevant to users. A pair of experimental studies shows that they increase both reading time—by 30 seconds, from 7 in the first experiment and 20 in the second—and comprehension of agreement content—by 4/16 points, from 0.

Finally, a solid understanding of users' attitudes towards specific agreement content is needed to inform the design of improved software agreements. To that end, this thesis presents an analysis of EULAscan, an online community of anonymous reviewers of software agreements. An open coding is used to categorize 191 EULAscan reviews. From this analysis, functionality emerges as the most prevalent concern. The wide variety of other concerns across reviews suggests that static designs of software agreements would inadequately serve a large population of users. Instead, this thesis proposes a focus on end-user tools that identify and highlight clauses of possible interest to a given user—for example, terms that the user has not seen before.

Acknowledgements

This thesis would not have been completed without the help and encouragement of numerous people. Foremost among those, I would like to thank Michael Terry, my supervisor, for introducing me to Human-Computer Interaction and thereby renewing my flagging interest in Computer Science. Without his continual prodding for higher level perspectives on any and every problem, even when such perspectives seem out of reach, this thesis would not be what it is. I would also like to thank Ed Lank for his feedback on my work, both as a reader of this thesis and as a valuable and approachable source of feedback in the HCI Lab. Thanks to Dan Berry, too, for his valuable feedback as a reader, and for his congenial attitude in general.

Thanks also go to my friends and colleagues in the HCI Lab at the University of Waterloo: Ben Lafreniere, Adam Fourney, Richard Fung, Jaime Ruiz, Janna Weber, and Matei Negulescu, among others. They have all provided feedback on this work in lab meetings, given extensive and valuable feedback on presentations of this work, and generally have just been a pleasure to know and work with.

I must also thank my friends and family. My parents, Margaret and Stephen, have always given me unwavering support in whatever I do, and the value of that support cannot be overestimated. My sister, Rachael, has always been a source of wise advice in the ways of university life, and life in general, for which I am truly grateful. My brother, Nathaniel, has always been a source of much-needed distractions from work, often in the form of a night or two of good (bad) movies.

Portions of this thesis are based on work published at alt.chi [43] and SOUPS [44]. I would like to thank my paper reviewers, who have given me valuable feedback that has been incorporated both into paper revisions and this thesis. Finally, I would like to thank my participants for their time and patience, and the Natural Sciences and Engineering Research Council for funding support.

Contents

List of figures

List of tables

Chapter 1

Introduction

1.1 Motivation

End users must often agree to lengthy terms of use prior to installing software or using software services. These terms of use may come in many forms, such as website privacy policies [24,41,45,56], EULAs shown during software installation [30,31,33], and consent agreements accompanying research software [83]. While these various forms of agreement each serve slightly different purposes, they are referred to collectively in this thesis as *software agreements*.

These agreements often contain terms of direct concern to users. For example, agreements may describe data collection policies, or inform the user that additional software will be installed, such as digital rights management (DRM) software or software that delivers targeted advertising. Although these agreement terms can subsequently have an impact on the user's privacy, security, or overall user experience, the terms are typically read by less than 2% of the population (in the case of EULAs [30]), often because the agreements do not contribute to users' more immediate goals, such as using the software to accomplish a specific task [27,28].

Currently, software agreements are communicated solely via text. Text provides great precision for communicating the abstract concepts of software agreements. However, these agreements are often of considerable length and utilize sophisticated language [33] that can require the ability to read at a university or postgraduate level [41]. These aspects of software agreements make it challenging for users to read and comprehend the agreement terms, especially for a user whose native language differs from that of the software agreement; these challenges often deter people from reading the agreements. The same aspects of software agreements that discourage reading—length and sophistication of language—also make them difficult and costly to translate into alternative locales.

A growing number of proprietary software companies have recognized some of these issues, and have begun to present agreements in ways that are more easily understood by their users.

For example, the makers of Aviary (a Web-based suite of multimedia applications) provide a license agreement that includes a summary of the agreement's terms in a side column [9]. This practice represents a concerted effort to build a trusted relationship between the company and its users through an improved software agreement process.

Besides producers of proprietary software, there is a range of other software producers who would benefit from and welcome an improved software agreement process. For example, producers of free or open-source software, such as the Free Software Foundation[1], often wish others to learn about their movement and its philosophical underpinnings. Methods for better communicating the terms and perceived benefits of their licenses—such as the GNU General Public License [26]—would help further their cause.

As a second example, researchers who conduct studies using remotely distributed software—such as the ingimp project [83]—have an ethical obligation to obtain informed consent from their participants. However, if participants ignore the consent agreement included with the software, they may not fully understand the risks associated with participating in the research; these risks can include the potential for sensitive information to be inadvertently made public [83]. There is a clear need for an improved software agreement process in these situations.

Governments are increasingly considering legislation to both standardize and improve software agreements to help protect consumers [68]. This presents an additional motivation for research in this space: if legislative measures are desired, they should be crafted in a way that results in demonstrably better software agreement processes.

Recognizing the inadequacy of current software agreement methods, alternative strategies have been proposed by the research community to improve this process. Many of these techniques introduce additional material, such as a summary of an agreement's content [30,31]. However, this additional material distracts users from the agreement; as a result, readers may miss any information contained only in the agreement itself [56]. The problem of consistency between the summary and its summarized content also arises, particularly when the latter undergoes change.

There exists a range of motivations for improving software agreement processes, including economic, ideological, and ethical motivations. Together, these motivations suggest a strong need for greater consumer advocacy in this space—whatever the impetus—and therefore a need for a better understanding of how best to communicate software agreement content to users.

---

1 http://www.fsf.org/

1.2 Thesis statement

In agreements with a large number of terms of interest to users, it is difficult to identify a specific set of terms of interest to all users. One way address this problem is to attract attention to the full range of agreement content by using techniques that modify the primary object of interest—the software agreement itself. This thesis tackles several of the problems in designing such techniques:

- capturing reader attention and directing it to the primary object of interest—the agreement itself—rather than a summary,
- improving comprehension of the full agreement's terms,
- facilitating localization and the communication of agreements terms to linguistically diverse audiences, and
- determining what aspects of an agreement are of particular importance to readers.

1.2.1 Contributions

This thesis makes the following contributions:

Narrative pictograms. A narrative pictogram is a diagram that makes limited use of words to describe a software agreement's content. Importantly, as presented here, a set of narrative pictograms are not meant to replace a text-based agreement, but rather, to supplement an agreement and facilitate an individual's comprehension of the agreement's content. In particular, a narrative pictogram helps improve the comprehension of an agreement for individuals who are not native readers of the language of the agreement text. Example pictograms for the data collection policies of research software are presented along with the composition rules used to create them. The pictograms are also evaluated using a modification of the ISO 9186-1 test for graphical symbols.

Textured agreements. A textured agreement is a software agreement that employs visual design techniques, such as typography and layout, to create a well-defined information hierarchy without modifying the underlying text of a software agreement. These techniques also help capture and retain a reader's attention. For example, a textured agreement might use warning symbols to highlight terms of an agreement that may affect a user's privacy and employs visual variety to create interest throughout the document.

To validate the design of textured agreements, two between-subjects laboratory experiments were conducted. Both experiments employed deception via a distractor task that included the need to install software and agree to its terms. The studies demonstrated that reading time for textured agreements was 30 seconds longer than for a plain-text agreement, up from 7 seconds in the first experiment and from 20 seconds in the second. On an open-ended comprehension quiz median comprehension of a textured agreement's terms was 4 out of 16 points higher than that of a plain-text agreement, which in fact had median comprehension of 0 out of 16 points.

Recommendations for selecting salient content. It is necessary within the attention-starved environment of a software agreement process to maximize the useful potential information a reader may gain from an agreement. This can be done by making content that users find important more salient. In order to better understand users' attitudes towards specific agreement content, I conducted an analysis of EULAscan, an online community of anonymous reviewers of software agreements. This thesis discusses the range of concerns raised by EULAscan users in their reviews, classified according to their prevalence as primary and secondary concerns. The analysis of EULAscan reviews forms the basis of recommendations for designing improved presentations of software agreements. In particular, the results of this study suggest that, besides those having an impact on functionality, the terms users consider important may be a moving target. As an alternative, I suggest developing tools to help each user identify terms in software agreements that they find interesting or unusual—for example, terms they have not seen before.

1.3 Organization

Chapter 2 of this thesis reviews related work in three areas: software agreements, visual design, and user attitudes towards agreement content.

Chapter 3 describes narrative pictograms, and is based on work originally published at alt.chi [43]. It presents an example of narrative pictograms as applied to the data collection agreement for a publicly distributed research application, as well as the composition rules used to create the diagrams. It also describes results from formative and summative evaluations of the diagrams.

Chapter 4 describes textured agreements, and is based on work originally published at SOUPS [44]. It first presents the design method that led to the creation of textured agreements. It also presents the specific techniques that compose textured agreements, and then describes the two experimental studies conducted to evaluate the effectiveness of the agreements. Results and implications from both studies are presented, along with perspectives from my local internal review board on the feasibility of deploying textured agreements in real-world situations.

Chapter 5 describes a study of EULAscan, an online community of anonymous reviewers of software agreements. It presents a characterization of the EULAscan community based on various factors, including websites linking to EULAscan. Next, It discusses the range of concerns raised by EULAscan users in their reviews, classified according to their prevalence as primary and secondary concerns. It also describe concerns specific to reviewers of open-source licenses. Finally, It presents recommendations for designing improved presentations of software agreements based on the analysis of EULAscan reviews.

Chapter 6 concludes with a summary of findings and implications for future work.

Chapter 2

Related work

2.1 Presenting software agreements

Software agreements are presented to users in a variety of forms, such as shrink-wrap agreements packaged with software, Web privacy policies shown or linked to during the sign-up process for an online product or service, and EULAs shown during software installation. An example of the latter is shown in Figure 2.1.

Recent research suggests that less than 2% of the population read EULAs when installing software [30]. Similarly, few individuals appear to read website privacy policies [41]. These results are unsurprising, as users often see the agreement process as something of low utility that does not contribute to the goal of using the software [27,28]. Recognizing the inadequacy of current software agreement methods, a number of strategies have been developed in both industry and the research community to improve this process. Many of these strategies aim to reduce the length or complexity of the agreements, as described next.

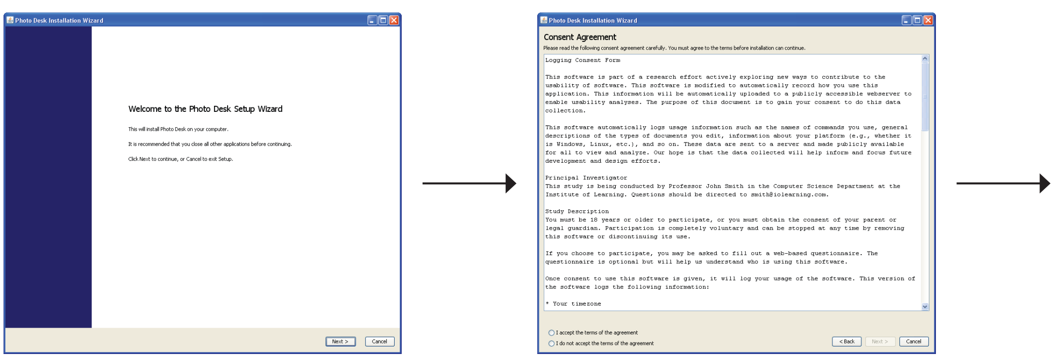


Figure 2.1 *The initial screens of a typical Windows software installation process involving a EULA. After the installer's welcome screen, the user is shown a EULA. In order to proceed with the installation of the software, the user must indicate acceptance of the agreement terms by clicking "I agree" then "Next".*

To assist users in comprehending the terms of lengthy agreements, Good *et al.* drew inspiration from techniques used in fields such as medical informed consent [18] and financial privacy notices [3] to create single-screen *summaries*[1] of EULAs [30,31]. These summaries have been found to significantly reduce the number of installations of spyware[2] when presented as part of the software installation process. However, as will be shown later, this approach runs the risk of reducing the likelihood that users read the full agreement should they proceed past the summary.

Kelley *et al.* [45] took inspiration from nutrition labels in designing a summary of privacy information for P3P (Platform for Privacy Preferences)-compliant websites. These *privacy labels* use a table of privacy policy information to indicate how users' data are used by a website and what data collection practices users can opt in or out of. While these labels were found to improve users' ability to find relevant information in Web privacy policies, it is unclear how well the technique would generalize to other forms of software agreements.

Patrick and Kenny approached the problem of lengthy agreements by proposing Just-In-time Click-Through Agreements (JITCTAs) [68]. Instead of requiring the user to agree to a large agreement prior to using the software, smaller JITCTAs are displayed when performing an action that causes personal information to be collected by the software. In essence, this approach segments the agreement into smaller components that are shown at situationally appropriate times, increasing the chance that users will notice, comprehend, and act on the agreement terms. While promising, this technique is only appropriate when an agreement can be broken into smaller context-sensitive pieces that do not overload the user with requests as they work.

To address the problem of capturing user attention in warning dialogs, Brustoloni *et al.* [15] created *polymorphic dialogs*, which rearrange the order of buttons on the interface or temporarily make buttons inactive. These strategies are intended to prevent users from learning a fixed path through the interface. This approach has been shown to increase the likelihood that users take the time to understand dialogs that outline the security risks corresponding to the user's intended action—for example, opening an email attachment that may contain viruses. However, it is unclear whether similar techniques would be effective in the context of software agreements, where the content is longer and denser.

In contexts outside of software agreements, research has explored a number of other approaches aimed at helping individuals understand lengthy agreements. To understand how to

---

1 The summary approach is often called *layered notices* [18,19,30,31,56], with the implication that it could be extended beyond simply presenting a summary with the full text, to presenting multiple, progressively more detailed layers—an approach that has not, to my knowledge, been tested.

2 Good *et al.* define spyware as software that "may track user activities online and offline, provide targeted advertising and/or engage in other types of activities that users describe as invasive or undesirable." [31]

improve comprehension of medical agreements, Campbell *et al.* [17] compared the efficacy of video, text, and enhanced text agreements (agreements that use improved headers, pictures, and other visual enhancements). They found that subjects with lower literacy rates were able to better comprehend the enhanced text agreements than the other treatments. However, it is unknown if similarly enhanced software agreements would garner similar improvements in comprehension, as software agreements and medical agreements differ in both content and purpose: medical consent agreements involve choices relating to immediate impacts on health, whereas software agreements involve more abstract concepts, like security and privacy. In addition, participants in Campbell *et al.*'s study were explicitly asked to read the agreements; consequently, this study does not answer the important question—particularly for software agreements—of whether enhanced text agreements can compel users to read.

While this past work has made some important strides to improving the software agreement process, there is still much room for improvement. For example, while summaries have been shown to be effective in some circumstances, summaries do not, by definition, include all terms of an agreement. Thus, in cases where a user wishes to proceed with software installation, that user is still faced with a lengthy agreement. The visually enhanced medical agreements of Campbell *et al.* show promise in addressing this problem, but these have not been tested in software contexts.

Among the many approaches that have been explored, visually enhanced agreements represent a particularly attractive avenue for further research: current and proposed solutions to the software agreement process are largely conducted using the medium of *plain text*, with minimal use of typography or other visual design techniques applied directly to agreement content. Given the rich history of visual design, there is an opportunity to draw on this field to enhance the software agreements themselves, rather than introducing additional material. To that end, the thesis next reviews related work in visual design.

2.2 Visual design

A rich repertoire of visual design practices have been developed to suggest the value in taking notice of information and to communicate information to wide audiences more effectively. This section reviews two such areas of visual design: the design of informationally dense documents and wordless communication.

2.2.1 Designing informationally dense documents

A number of techniques exist to improve the readability and accessibility of dense documents that convey large quantities of information. To construct information visualizations, Tufte advocates the practice of *layering* and *separation* of information in order to visualize dense data sets graphically [85]. This practice refers to constructing visualizations with visually distinct layers that are both coherent as a whole and can be read separately. For example, a graph of data from a heart monitor may include two layers: the background grid and the line plotted on that grid. Tufte observes that if both are given the same visual weight—i.e., the same stroke width and colour—the two are difficult to distinguish [85]. However, if the grid is set in a lighter colour and thinner stroke, both layers—the grid and the plotted data—can be read clearly, and complement, rather than interfere with, each other [85]. Proper use of layering and separation ensures that the usefulness of data on lower layers is not lost in higher layers of the visualization—it creates a *hierarchy of information*, making it possible to read a composition at a multiple levels [85].

In a similar vein, Kress and van Leeuwen discuss *framing* and *salience* [49]. Framing refers to the use of dividing lines—explicit or implied—to create relationships between elements [49]; for example, the use of a dividing line to separate a page into body text and associated figures. Salience refers to lending importance or visual weight to an element through the use of placement, size, contrast, or sharpness [49]; for example, a large photograph that dominates a magazine spread is naturally the most salient element, and catches the reader's attention. These principles impart a structure on the composition. Kress and van Leeuwen stress the importance of composition in facilitating non-linear reading of dense documents, which allows readers to more quickly find personally relevant information [49].

In a survey of research into the effectiveness of various methods of presenting technical information, Wright similarly found that design elements such as headings can make skimming a document easier, and even assist in the integration of content for readers who do not skim [94]. Wright also notes that design can be used effectively to influence the pace at which a dense document is read [94].

These techniques formed the basis of my work in developing textured agreements, described in Chapter 4. Although these techniques can capture readers' attention and improve the communication of dense documents to readers familiar with the language of the text, they do not specifically address problems faced by non-native language readers. Because not all software is localized for every language, users may encounter software agreements in languages other than their native tongues. While the obvious solution to this problem is to localize content for all likely locales, software localization requires time and resources that may not be available to all software developers. Wordless communication is one solution to the problem of localization that has been proposed in other domains, as described next.

2.2.2 Wordless communication

The success of wordless communication in other contexts suggests it may be a promising approach to reducing localization requirements for software agreements. Some of that work is reviewed here.

Pictograms, compact graphical symbols representing a single concept or object, have been highly successful in informational signs and road signs. For example, Otl Aicher's 1972 Munich Olympics signage [60] reduces the various Olympic sports to simplified, iconic representations that can be understood across the linguistically diverse audience of the Olympics. In a similar vein, the US Department of Transportation (DOT) in 1974 commissioned the design of a set of transportation symbols to standardize the means by which services were communicated to travelers [60]. In both of these examples, the designs strive to remove as much detail as possible in line, form, and colour, leaving only the essence of the object or action to aid quick comprehension.

More elaborate wordless diagrams have been employed successfully in other contexts, for example, the welcome mats created by Patrick Hofmann for HP (Hewlett-Packard) to illustrate how one assembles and uses a printer [37,38]. These wordless welcome mats improved usability and eliminated the need for the production of 16 separate, localized manuals, saving HP hundreds of thousands of dollars in printing costs [38].

In the HCI community, recent work by Medhi *et al.* has shown that text-free user interfaces can communicate effectively to illiterate populations [59]. They developed design recommendations for interfaces for semiliterate populations, which include using semiabstract—as opposed to primarily abstract—representations and providing voice feedback [59].

Similarly, the Aphasia Project has explored the use of interfaces relying more heavily on visual and auditory cues to assist those with aphasia [57,82]. They developed a system of visual instructions that assist individuals with language impairments in following cooking recipes; these instructions use a syntactic template of *[start-state]–action–end-state* such that each individual instruction, such as “mix the ingredients with a spoon”, follows a consistent visual layout [82].

These previous successes form the basis of my work in designing narrative pictograms, described in Chapter 3. The name of the technique—*narrative pictograms*—directly reflects the basic principles of using simplified depictions of objects and actions, within a consistent overarching scenario, to communicate concepts.

2.3 User attitudes towards software agreements

Closely tied to questions of how to better present software agreements are questions of which content within the agreements themselves is actually of particular concern to users. Ideally, the content of most concern to users would be made most salient in an improved software agreement. Past research into the question of users' concerns with software agreement content falls into two categories: studies of users' privacy attitudes specifically and studies of software agreement terms more generally.

2.3.1 Privacy concerns

Researchers have found that users often trade off long-term privacy concerns against short-term benefits when deciding whether or not to engage with some software or service [4,31,79]. For example, Good *et al.* found that users are often willing to install spyware as a trade-off for desired functionality, which is considered to be the most important part of the application [31]. However, when the functionality of software is not a factor—e.g., when selecting between software of equal functionality—users will choose software based on other factors, such as degree of privacy-invasiveness [31].

In an online survey of 2468 adults in the United States, Culnan and Milne found that more than 70% of those surveyed had concerns about providing too much personal information to companies [21]. However, other research has found that users' stated privacy concerns and their steps to protect privacy do not always align [4,79]. For example, users often trade off privacy for other short-term benefits—e.g., a product discount [4]—or they may be drawn into an interactive experience and reveal information contrary to their stated privacy attitudes [79]. At the same time, Acquisti and Grossklags found that most people will act to protect their privacy at times, but definitions of privacy are personal, and do not always imply the same concerns or strategies for different people [4]. Good *et al.* similarly found that most people object to the installation of spyware on their systems, but their definitions of spyware varied [31].

Research has consistently found that users fall into three broad groups on the basis of their privacy concerns: privacy fundamentalists, who object to any data collection on principle; privacy pragmatists, who have specific concerns and strategies that are not shared uniformly; and the marginally concerned [4,20,79]. Cranor *et al.* found that sharing of information and collecting identifiable information are generally-held concerns, but that marginally concerned users objected more to certain practical effects of disclosing information, such as an increase in unsolicited communication [20].

Taken together, these results suggest certain software agreement terms—such as those related to information sharing or the collection of personally identifiable information—may be important to call attention to when presenting an agreement. However, since this research has generally focussed on user's concerns with privacy policies, it is unclear how privacy concerns rate in terms of severity compared to other terms found in software agreements.

2.3.2 Other concerns with software agreements

In a survey of 92 contract law students' attitudes towards online contracts in e-commerce (i.e., online purchases and online subscriptions), Hillman found that, similar to results from privacy literature, users generally do not read the agreements and do not shop around for better deals [35]. Being "in a hurry" was the most common reason for respondents not to read the agreement, given by 65% [35]; this is also consistent with findings from privacy research that users consider software agreements to be distractions from their primary task. Among those that did read, the price of the good or service and unfamiliarity with the vendor were the most likely motivations to do so [35].

Hillman also asked respondents who stated they may read agreements what type of terms would induce them to do so. He found two broad categories of content considered particularly important: warranties, guarantees, or return policies; and product information, disclosures, and warnings [35]. Both of these categories are fairly broad, and little fine-grained detail is provided. Echoing some of the concerns found by Hillman, Wu [95] summarizes various objections to terms of software agreements made by legal and consumer advocacy groups, including provisions allowing for backdoors to be installed on users' systems, post-transaction disclosure of terms[1], lack of warranties for known defects, restrictions on users' rights to publicly review the software, and prohibitions against reverse engineering. However, there is little information on the relative importance ascribed these terms by the broader population of users.

Hillman also discussed terms that were not of interest to users, finding that details of arbitration and choice of forum[2] were not considered important by those surveyed [35]. He further suggests that forum selection and dispute resolution clauses are likely not considered important by users due to the fact that users often treat low probability risks as zero probability risks, and

1 Post-transaction disclosure occurs when users are not provided with the agreement terms until after a sale is completed. Braucher provides an in-depth discussion of this practice on the Web, wherein she estimates that 87.5% of online sales of software employ post-transaction disclosure [14].

2 As an example, a choice of forum clause may require disputes to be settled in a specific legal jurisdiction.

thus readily trade off any potential future difficulties with settling legal disputes against the utility of the software [36].

Both this work and work in the privacy space suggest a number of categories of agreement terms that are of interest to readers, for example, information sharing, the collection of personal information, warranties, guarantees, disclosures, and warnings. However, these concerns have not been synthesized into a comprehensive picture of their relative importance; without this understanding, a designer of a software agreement would have difficulty selecting the most important terms in an agreement. For example, are users more concerned about information sharing, or warranties? Existing research does not address their relative importance. Further, several of these categories are quite broad—e.g., product information, disclosures, and warnings—giving no indication of the relative importance of terms within those categories. Finally, some of the rankings of these categories are based on data from closed surveys (e.g., Hillman [35]); as a result, these categories may not represent the full spectrum of user concerns. There is therefore a need for a more ecologically valid understanding of user's concerns with software agreements, with a focus on developing concrete guidelines for designers of those agreements.

In addition to advancing understanding of what terms readers consider important, there is a need to identify terms of little interest to users. Agreements often restate rules that apply due to copyright law and are not of interest to users [76], potentially making salient information difficult to find. This fact, along with previously discussed issues of complex language use, linguistic barriers, and uninviting presentation, strongly suggests the need for better approaches to communicating the content of software agreements to users. The next chapter discusses one such approach, called narrative pictograms, which aims to improve the communication of agreement terms across linguistic barriers.

Chapter 3

Narrative pictograms of data collection

3.1 Introduction

This chapter introduces *narrative pictograms*, a pictorial technique specifically designed to improve the process of communicating software agreement terms to a diverse population of users (e.g., Figure 3.1). This chapter first describes the specific communication goals that guided the creation of narrative pictograms. It then presents the full technique and its corresponding composition rules with illustrative examples. Finally, it describes results from formative and summative evaluations of the diagrams.

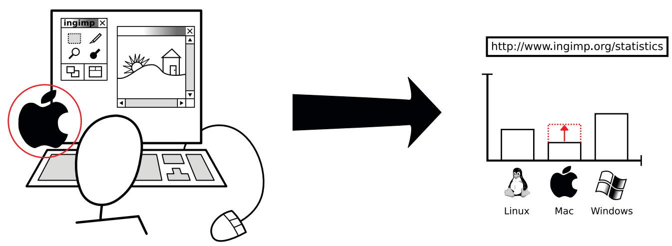


Figure 3.1 *A narrative pictogram. This diagram illustrates the fact that the software will collect data about which operating system the user uses.*

3.2 Goals and challenges

This work began with the need to create a consent agreement to accompany a research application, ingimp [83], which is a publicly available version of an open-source bitmap graphics editor, the GIMP[1], that collects software usage data. Over time, this project amassed a user base that includes users from nearly a dozen different locales. Because this research involves human

1 http://www.gimp.org/

subjects, and because the software collects usage data, ingimp includes a consent agreement that outlines the data collected and the ways the participant could be placed at risk by using the software. However, this consent agreement is written in English and can thus be difficult to read for a user whose native language is not English. Ideally, all ingimp's users would take the time to read and understand the consent agreement, but the project has limited resources for localizing the content. Thus, there is a real challenge for gaining truly informed consent from users.

In attempting to improve the communication of ingimp's consent agreement to its linguistically diverse audience, my specific goals were to:

- augment, not replace, the existing text-based consent agreement,
- increase the chance that the user understands the purpose and functionality of the software, specifically, the types of data collected and the types of data not collected, and
- reduce the need to localize the diagrams themselves; more specifically, the diagrams should contain minimal text, and one should be able to read the diagrams, or learn how to read the diagrams, without the need for auxiliary aids.

Achieving these goals presents challenges particular to this space. While wordless diagrams have been applied successfully to communicate concrete objects and actions—e.g., cooking instructions [82] or printer setup [37,38]—the concepts involved in a data collection policy are more abstract. As a result, it is not possible to rely entirely a pre-existing visual vocabulary of objects and actions; instead, readers must be instructed both in how to read the diagrams and in the meaning of certain graphical elements—using the diagrams themselves.

To explore possible solutions to these goals and challenges, I engaged in an iterative design process and formative evaluation, discussed later in this section. The next section describes the resultant diagrams while introducing some of the composition rules used in their creation, following that is a section describing the full set of composition rules in more depth.

3.3 Narrative pictograms of data collection

The narrative pictograms I developed are segmented into four collections of diagrams, each serving a different purpose in describing the data collection policies of the instrumented software. Users see these in the order outlined below to create a narrative that teaches them how to read the diagrams; this narrative gradually instructs users on what data are, and are not, collected. The four sets of diagrams are:

1. ***The functional overview.*** A diagram that provides a functional overview of how the software works. This overview establishes a context for the narrative, and most importantly teaches readers the vocabulary and syntax of the remaining diagrams.

2. ***Environmental data collection.*** These diagrams show what data are logged without explicit user action, e.g., the user's operating system, their locale, etc.

3. ***Interaction data collection.*** These diagrams show what data are logged as a result of explicit user action, e.g., interacting with windows.

4. ***Privacy sensitive data collection.*** These diagrams depict data collected that may be considered private or sensitive in nature.

With the exception of the functional overview, each set of diagrams is comprised of multiple individual diagrams, each of which corresponds to a single concept or scenario of use. In contrast, the functional overview is a single, larger diagram that conveys an establishing narrative. Each set of diagrams is described in detail below.

3.3.1 Functional overview

The first diagram displayed in the narrative pictograms is the functional overview (Figure 3.2). The functional overview depicts a basic interaction sequence from the time the application is started to when it is closed. The basic use of the application is complemented with graphs suggesting data collection: as the user in the diagram interacts with the mouse or inputs a filename, bar graphs corresponding to these inputs are incremented proportionally. When the application is closed, the user's personal bar graphs are shown being transmitted to the research website.

The functional overview addresses the significant challenge of teaching the user how to read the diagrams; it does this by presenting a concrete, simplified series of familiar actions. This sequence establishes a context in which more novel elements can be introduced, and is an example of the first composition rule: *employ literal contexts*. The conventions and visual elements established in this overview are repeatedly used in subsequent diagrams and, when necessary, purposefully violated in order to draw attention to important information. This basic strategy

The overview begins by establishing a concrete context: the launch and use of an application.

The overview then introduces an easily-identified anomaly: graphs, initially blank. The purpose of these graphs may not be immediately obvious to many readers; this is expected.

Next, the meaning of the graphs is clarified through two concrete examples: a mouse click and, in this example, typing. Each action and its consequences—the data collected—are highlighted to help the reader understand what the graph represents. Pictures of the graph's source data are used as labels to improve the correspondence between the character's actions and the graphs.

Finally, the overview depicts the transmission of the collected data—the graphs—to the ingimp servers, via the Internet.

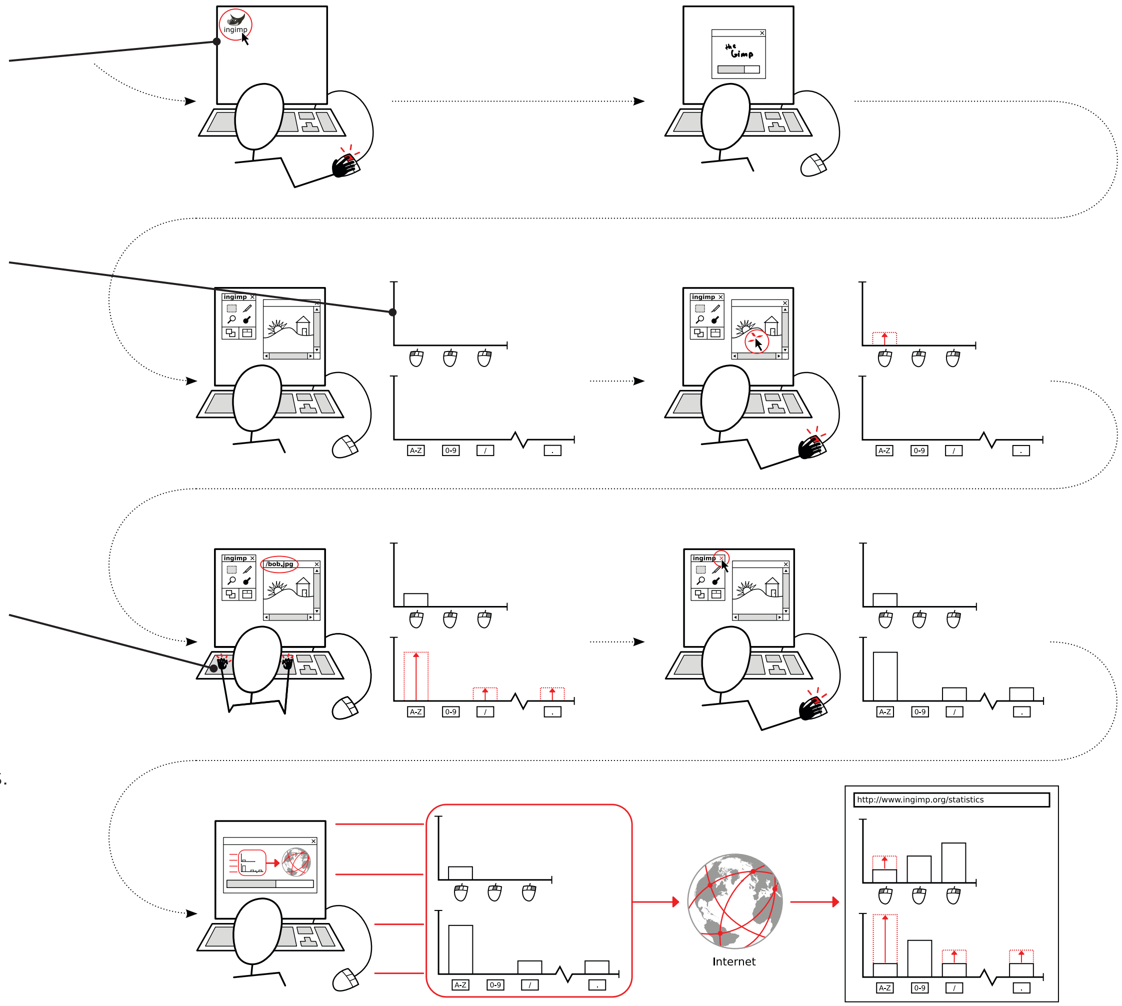


Figure 3.2 *The functional overview provided to describe the fact that the ingimp software collects usage data that is sent to a website.*

permeates the diagrams' design: a familiar scenario is presented to the viewer, with a select set of novel elements introduced into that scenario. Over time, repeated use of novel elements turns them into known, familiar elements, allowing further augmentation of scenes with other, novel elements. In this way, the viewer is incrementally introduced to increasingly sophisticated concepts. This is an example of the second composition rule: *use repetition to establish patterns and conventions.*

The next three sets of diagrams build on the conventions established in this overview to represent what data are, and are not, collected. They accomplish this goal by repeating the same motif developed in the overview: a user shown at the computer with important actions and concepts highlighted in red, and a summarization of the data collected shown in a data collection graph.

3.3.2 Environmental data collection

The second set of narrative pictograms depicts *environmental data* collected, or data that describe a user's task environment. For example, the diagrams illustrate that the user's operating system and the size of their monitor are logged (Figures 3.3 and 3.4).

In contrast to the functional overview, which utilized a narrative composed of a string of related scenes, this series of diagrams reduces the depiction of logging to single, independent scenes. As mentioned, the character is shown in front of the computer with the data being collected highlighted in red. An arrow placed between the user and the data collection graph suggests the collection and transmission of the data to a remote website (Figures 3.3 and 3.4). In this series, the data collected are not data that result from specific user action, such as the user's time zone. Accordingly, the user is shown sitting in front of the computer with no hands visible.

Like the graphs shown in the functional overview, the data collection graphs have very literal graph labels; this again follows the rule of employing literal context as opposed to abstract symbols. For example, icons for popular operating systems are used to indicate that the type of operating system is logged (Figure 3.3). Similarly,

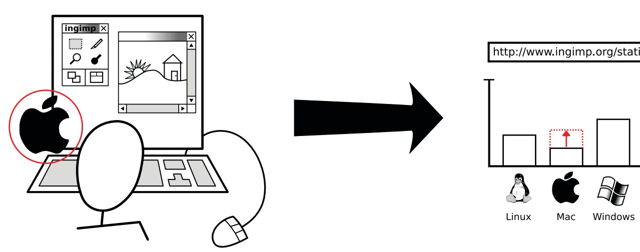


Figure 3.3 *A diagram in the environmental data collection set. This diagram illustrates the fact that the software will collect data about which operating system the user uses.*

common screen resolutions—e.g., 1024×768—are used as labels to indicate the capture of screen resolution (Figure 3.4).

The order of diagrams within this series was also derived from two composition rules: employing literal context and using repetition to establish patterns and conventions. These rules suggest the strategy of first showing objects and concepts that are most likely to be familiar and known to users. Thus, the first scene depicts the logging of the user's operating system using familiar icons, followed by time zone data using a shaded world map (Figure 3.5). The collection of locale information, a slightly more abstract concept, is shown after these more concrete concepts, at a point when users' repeated exposures to the common elements of the first few diagrams have made them more likely to understand the purpose of the individual diagrams—an example of using repetition to establish conventions. I represent locale data using countries' flags associated with the keyboard (Figure 3.6).

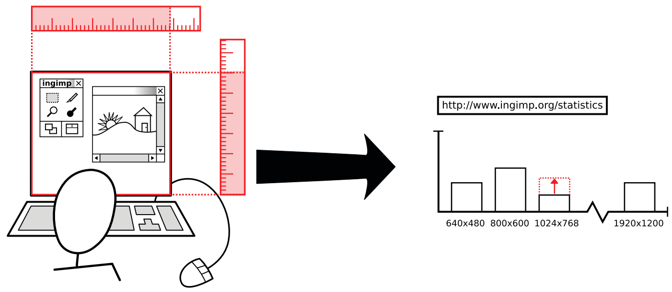


Figure 3.4 *A diagram in the environmental data collection set, depicting the capture of screen resolution. Graph labels with common screen resolutions are used to indicate that monitor resolutions are recorded.*

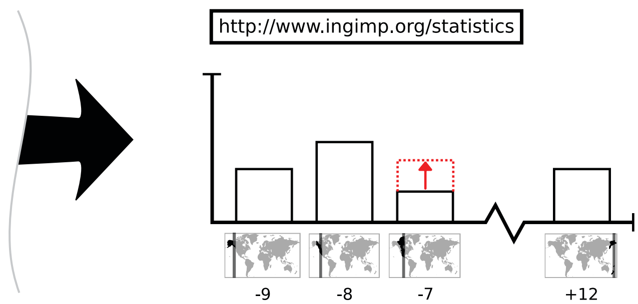


Figure 3.5 *Cropped view of a diagram showing the collection of time zone data. As time zones allow a relatively familiar, concrete depiction—shaded world maps—this diagram is placed early in the Environmental Data Collection set.*

Figure 3.6 *Cropped view of a diagram showing the collection of locale information. This more abstract concept is introduced later in the Environmental Data Collection set.*

3.3.3 Interaction data collection

The next series of diagrams, interaction data collection, depicts scenes in which an explicit user action results in some data being collected. For example, the system records that the mouse or keyboard was used, so use of the mouse and keyboard is shown, alongside appropriate data collection graphs. As in the previous set, the pictograms start with concepts more likely to be familiar: mouse clicks, keyboard use, and tool use (Figure 3.7).

For this particular application, the actual mouse location and specific keystrokes are not logged, just the fact that the mouse or keyboard was used. To make these points clear, I alter a number of elements in the previously established visual motif. As an example, the keyboard data collection diagram (Figure 3.8) shows the user typing on the keyboard, but its data collection graph displays only a single bar labeled with an icon of an entire keyboard. This depiction suggests that all of the keyboard activity is summarized in one dimension only, since individual bars for each key are not shown. Formative evaluations of the diagrams (discussed in more detail later in this chapter)

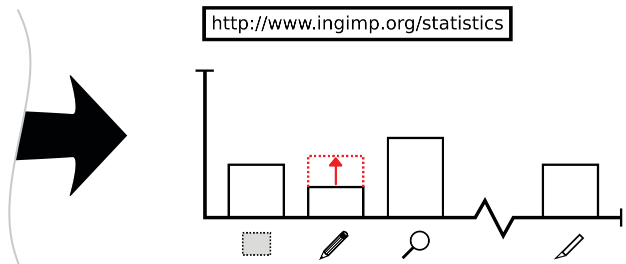


Figure 3.7 *Cropped view of a diagram showing the collection of tool use. This diagram makes use of concrete labels drawn from ingimp: tool icons.*

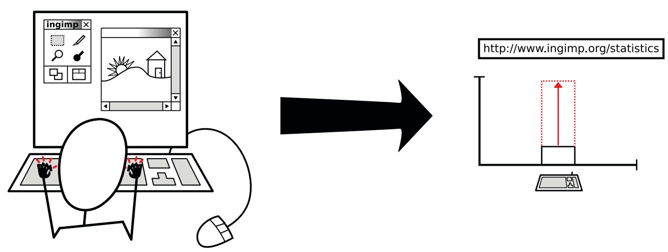


Figure 3.8 *A diagram in the interaction data collection set, depicting the capture of keyboard activity. From the single bar in the graph most viewers correctly infer that individual keystrokes are not recorded.*

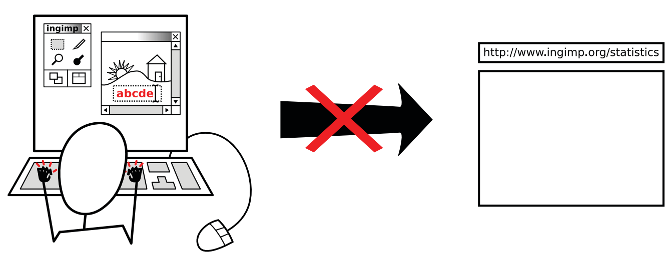


Figure 3.9 *This diagram follows Figure 3.8 to emphasize that individual keystrokes are not collected. This is done by breaking previously established conventions for data collection: the arrow and graph.*

revealed that the absence of information within a diagram suggests the absence of that data being collected by the application, making this a useful strategy to help convey this concept.

The ability to infer the absence of data collection is useful, but I also wanted to make it explicit that certain types of data are not collected. To make this fact clear, the diagram in Figure 3.9 is shown immediately after the diagram in Figure 3.8 to clarify that the actual, typed text is not recorded. For this class of diagram, I modify the typical data collection convention by placing a red "X" through the data transmission arrow and showing a blank website. Having already shown several examples of data being collected and represented on the website, these two modifications to an established convention help communicate that data are *not* transmitted, and are examples of the third composition rule: *break established patterns and conventions to draw attention to details*.

3.3.4 Privacy sensitive data collection

The previous two diagram sets employ the convention of showing data being collected and aggregated with other data, suggesting a level of anonymity to the data collection process. However, some data collected by this software is recorded without any summarization or anonymization applied. For example, ingimp allows users to describe their planned uses of the software by entering keywords in a special box shown at start-up. These keywords—referred to as *activity tags*—are logged without modification, and are publicly accessible, as is all other data collected by this software.

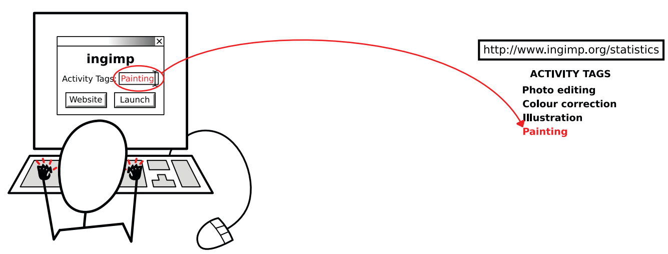


Figure 3.10 *This diagram also breaks conventions established in previous diagrams—the large arrow—to highlight data that are recorded without any summarization or anonymization, in this case, activity tags.*

To highlight the sensitive nature of this type of data collection process, a previously established convention is again broken—the large arrow indicating data collection—and data are instead shown being taken directly from the user's screen and placed into the public website's data set (Figure 3.10). This direct correspondence between information on the character's screen and the public website underscores the associated privacy risks.

3.4 Composition rules

These composition rules, with the exception of the last rule, were developed in tandem with the diagrams during my design phase. The composition rules are as follows:

1. Employ literal contexts and simplified representations of objects, rather than abstract symbols

2. Use repetition to establish patterns and conventions

3. Break established patterns and conventions to draw attention to details

4. Avoid conventions of manuals to avoid creating the impression that the diagrams are instructional or imperative, rather than informative

5. Avoid reliance on domain knowledge, or provide additional context for complex domain-specific concepts

These composition rules, in and of themselves, are not necessarily novel: graphic designers and technical illustrators regularly apply them in various contexts. However, it is the careful selection and application of these techniques to this particular problem domain that results in a novel contribution. The following subsections expand on each rule in turn.

3.4.1 Employ literal contexts

Narrative pictograms are presented to users in the highly unique context of a software agreement process, e.g., during software installation. They are also intended to function across cultures, without the need for localized text to assist in comprehension. These factors require care in the design process so that the intent of these unfamiliar and unexpected diagrams is not misinterpreted. Accordingly, the diagrams employ very literal narratives, with familiar objects and actions, to establish a grounding context. The narratives also help establish the fact that the diagrams are informative and not tutorials or advertisements. With this context established, more abstract concepts can be introduced to communicate more sophisticated material.

3.4.2 Use repetition to establish patterns and conventions

Hofmann, in writing about methods to create cross-cultural informational illustrations [38], suggests starting instructions with a base illustration that is gradually changed. The repetitive use of the same base illustration, with only minor modifications, enables the reader to more easily establish what is changing. This same convention is used within narrative pictograms.

Repetition in my diagrams works in several complementary ways. First, by repeating elements without change, the pictograms suggest that the repeated element is not very significant; simultaneously, elements that *do* change are more easily identified and granted significance. Second, repetition helps reinforce the concepts being conveyed. For example, the convention of data being transmitted from the person's machine to the public website is used repeatedly. If a user does not initially understand this convention, as they move through the diagrams they will be repeatedly exposed to it, giving them a number of opportunities to interpret it in different contexts. Once they finally understand the convention, they can easily recognize that all of the diagrams are attempting to convey the logging and transmission of data. In my formative evaluation, participants often exhibited this behaviour, and were able to go back and understand previous diagrams which were initially unclear to them.

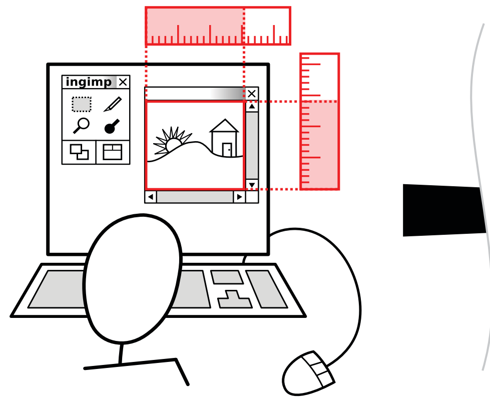


Figure 3.11 *Cropped view of a diagram showing the collection of image size. This diagram reuses the visual convention for "measurement" previously established in the diagram depicting the collection of screen resolution (Figure 3.4).*

3.4.3 Break established patterns and conventions to draw attention to details

Once a pattern has been established, it can be broken or modified to draw attention to particular details, or to create new conventions. For example, the diagrams break conventions to highlight notable exceptions of what data are, and are not, collected. As previously discussed, the convention of an arrow used to indicate data collection is broken by some diagrams to indicate data not collected (Figure 3.9) and data collected without summarization (Figure 3.10). Both of these instances establish new conventions that are used in subsequent diagrams.

3.4.4 Avoid conventions of manuals

Testing of the diagrams revealed that the use of many conventions from instructional illustrations—e.g., assembly instructions—would immediately lead readers to consider the diagrams to be a user manual.

As an example, the earliest diagrams used numbers in a variety of contexts to connote sequences. I originally numbered the sequences of the functional overview to emphasize the order in which they should be read, since reading direction is not universal. However, this numbering suggested that the diagrams were trying to instruct the viewer on how to accomplish some task. Consequently, I adopted other strategies to suggest sequences, e.g., the use of arrows in the functional overview (Figure 3.2).

When originally designing diagrams to describe what data are not collected, I placed a prohibition sign (a circle with a line through it) over a diagram showing data collection. Like the numbering, participants felt they were being instructed, but in this case, instructed *not* to do something. To avoid this connotation, I instead place a red "X" over the arrow to indicate that transmission does not occur (Figure 3.9). By using an "X" instead of the prohibition sign, and by placing it over the arrow rather than the activity, I avoid the suggestion that certain user actions are prohibited.

3.4.5 Avoid reliance on domain knowledge

My summative evaluation identified an additional issue with a subset of the diagrams: overreliance on domain knowledge. For example, one of the pictograms (Figure 3.12) is intended to depict the collection of image histogram data. Such histograms are commonly employed by professional users of photo manipulation applications. However, the histograms may not be understood by novices to the domain. In my summative evaluation, some participants were provided with text from the ingimp agreement, which describes that image histograms are generated from frequency counts of image pixel values. These

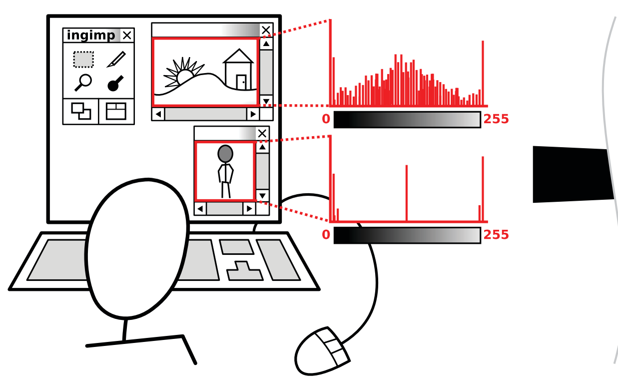


Figure 3.12 *Cropped view of a diagram showing the collection of image histogram data. Many participants had difficulty with this diagram due to a lack of domain knowledge.*

participants were able to acquire the necessary knowledge to interpret the diagram. In contrast, participants without that text expressed confusion about the meaning of the histograms. This finding suggests that more complex, domain-specific concepts require additional written or pictorial context to be understood by relative newcomers to the domain.

As mentioned, the majority of these composition rules arose out of my initial formative evaluation, which is described next.

3.5 Design process and formative evaluation

My review of existing wordless diagrams yielded an initial set of techniques—simplified representations of objects embedded in a surrounding, pictorial narrative—to use to describe the data collection policies of ingimp. These techniques formed the basis of a series of paper-based diagrams conveying the primary concepts of the ingimp software agreement, such as the software's functionality and purpose, and the data it collects. The diagrams were further developed using an iterative design process combined with formative evaluations, described in this section.

3.5.1 Method

The diagrams were iteratively developed using a hybrid evaluation method incorporating the ISO 9186-1 test method for graphical symbols [39], a think-aloud protocol, and Wizard-of-Oz prototyping. The ISO procedure prescribes two different types of tests when evaluating graphical symbols: a comprehension test and a judgment test. In the comprehension test, for each symbol to be tested, each participant is first presented with a context in which they might see that symbol. The participant is then shown the symbol and asked to interpret its meaning in that context. In the judgment test, the participant is informed what the intent of each symbol is, then asked to estimate how many people are likely to understand the symbol in the given target population. This latter test is meant to assist in selecting the most effective designs among a set of alternative designs.

In the iterative testing procedure, I presented each participant with paper-based prototypes and indicated that the pictograms would be seen when starting ingimp for the first time. I then asked the participant to interpret the pictograms, thinking aloud. I took notes, recorded the participant's comments, and observed the participant's behavior while studying the pictograms. I did not provide any assistance as the participant interpreted the pictograms. Instead, I instructed each of them to say "I don't know" if they could not understand a pictogram. When significant communication failures occurred with the pictograms, I sketched new, alternative pictograms and asked the participant to interpret the new designs.

After each participant was done interpreting the pictograms, I described the pictograms' meanings and asked the participant to estimate the percentage of users who would understand them. This step allowed us to gauge the effectiveness of the designs and design alternatives, and guided the selection and refinement of the designs for the next participant.

3.5.2 Participants

For this study, I recruited 14 participants, 7 native English speakers and 7 non-native English speakers; all were fluent in English and studying in English. There were 4 female and 10 male participants. All participants were university students, with 11 undergraduates and 3 graduate students. Participants were not remunerated for their participation.

3.5.3 Results

The diagrams and composition rules outlined previously in this chapter, with the exception of the fifth rule, are a result of the design process and formative evaluation. Initial results suggest the effectiveness of the final set of diagrams produced

For the first four participants, the initial designs failed to effectively convey the intended concepts of the software agreements. As noted earlier, a significant issue with this space is teaching users the vocabulary and syntax of the diagrams, particularly when the concepts being conveyed are of such an abstract nature as data collection. Many of the initial designs attempted to portray the full life cycle of the ingimp software, including its use, the data collected, and the use of that collected data by ingimp developers to improve subsequent releases of the software. However, this life cycle was not understood by participants, and these designs were unsuccessful.

These designs underwent significant modifications in the presence of the participants and in between trials. It was during this time that the concept of a functional overview was developed. The functional overview addressed the most significant barrier to users' understanding by teaching them how to read the diagrams. At this point, I began to have success in communicating, at a minimum, that the software collects data.

After these first four participants, a set of designs and a uniform set of composition rules was converged upon. These designs were iteratively refined with the last 10 participants of this study. Each of the last 10 participants was able to correctly interpret all but 1 or 2 of the resulting 20 diagrams, with no one diagram being consistently misinterpreted by all participants.

3.6 Summative evaluation

To understand the effectiveness of the narrative pictograms produced, I designed an experimental evaluation to compare their comprehensibility with and without the software agreement.

3.6.1 Experimental design

As one of the most significant challenges in developing narrative pictograms is to teach users how to read them—accomplished by the functional overview—I wanted to understand how well participants could comprehend the diagrams without any additional information. This poses a difficult problem for experimental design, as an experiment in which participants are asked specific questions about the agreement content in itself provides participants additional information about that content through the questions asked. As a result, I decided to explore a different route, based on the aforementioned ISO 9186-1 method for testing comprehension of graphical symbols [39], slightly altered for my needs.

I employed two conditions: a diagrams-with-text condition and a diagrams-only condition. The diagrams-with-text condition places the diagrams in-line with a full text agreement, positioned near the text corresponding to the diagram content. This setup gives participants the maximum possible information to complete the comprehension test, representing a best-case scenario. In the second condition, participants are shown only the diagrams without any supplementary information. This condition provides an indication of how well these diagrams can be interpreted for those who may not have the benefit of understanding the language of the software agreement.

The ISO procedure is designed for testing graphical symbols. The procedure requires that the participant be shown each symbol on a sheet of paper or computer screen, and told the context in which that symbol would be used, e.g. "At an airport". They are then asked the question, "What do you think this symbol means?" For this study, I have made the following modifications to the ISO protocol to better suit the needs of this research:

1. I use the word "diagram" in place of "symbol" throughout the test, since the word "symbol" can be confusing when participants are presented with the comparatively richer narrative pictograms of this work, which often comprise many symbols.

2. I describe the context of the diagrams at the beginning of the test, rather than for each diagram individually. Specifically, in written instructions, participants are informed that they would encounter the diagrams during the installation of a piece of software.

3. The ISO test does not allow participants to go back and change answers, as it was designed for testing independent symbols. Since narrative pictograms are meant to be read as a whole, it is expected, rather than discouraged, that comprehension of one diagram might influence comprehension of another. Thus, I explicitly allow participants to go back and change answers if they wish. However, I ask them to cross out old answers with a single line so that I can better document their process of understanding the diagrams.

4. The ISO test recommends testing only 15 symbols per participant. Since the diagrams are meant to be read as a whole, I preferred not to remove any from the test. The complete set consists of 20 diagrams, which I did not consider excessive.

3.6.2 Method

After obtaining written consent, each participant was given written instructions and allowed to ask questions about the study. Each participant in the diagrams-with-text condition was provided with two booklets: a booklet containing a text agreement with in-line diagrams labelled "Diagram A", "Diagram B", etc., and an answer booklet. The answer booklet contained the question, "What do you think this diagram means?" for each diagram in the agreement. Two booklets were used in this condition as there was insufficient space to include both answer boxes and diagrams in-line with the full agreement text (see Appendix A). Each Participant in the diagrams-only condition was given one booklet containing only the diagrams, with the question, "What do you think this diagram means?" under each diagram. Diagrams and the answer space provided in both conditions were printed at the same size.

After completing the comprehension test, each participant filled out a short post-task questionnaire. This questionnaire asked the participant to rate the diagrams' visual appeal, informativeness, and comprehensibility on a five-point scale. The questionnaire also collected basic demographic information.

Finally, a short interview was conducted to understand what diagrams participants found particularly difficult to comprehend, and why.

3.6.3 Performance measures

Each response on the comprehension test was assigned a score as specified by ISO 9186-1 as one of (see Table 3.1): correct (score 1), incorrect (2), response is "don't know" (3), or no response (4). Wrong answers are further sub-divided into two subcategories: wrong answers (2a) and wrong answers with the opposite meaning of that intended by the diagram (2b). This distinction is crucial in the context of data logging: if users misinterpret a diagram as indicating no data are sent when data actually are, users' privacy may be inadvertently affected.

3.6.3.1 *Scoring*

Participants' written answers were scored by two raters not involved with the research project. Raters were given verbal and written instructions on the ISO 9186-1 scoring procedure, as well as answer keys with sample answers representing responses that would be scored 1, 2a, or 2b. Disagreements in scores were resolved by assigning the score that suggests lower comprehension, where 2b is lowest.

Category	Meaning
1	Correct
2a	Wrong
2b	Wrong and the response given is the opposite of the intended meaning
3	The response given is "Don't know"
4	No response is given

Table 3.1 *Comprehension test response categories, taken from ISO 9186-1 [39].*

I calculated the *total correct score* as the number of diagrams correctly interpreted by each participant (score 1). A perfect score would thus be 20.

Raters were instructed to give the 2b score (wrong and opposite of intended meaning) for each answer in which the participant thought some data was not being collected when it actually was, or vice versa.

3.6.4 Participants

20 participants were recruited in a university setting. Participants were compensated with a $10 gift certificate for a coffee chain. 15 females and 5 males participated, aged 17-42 years old (mean=24.95, SD=5.74). All participants were native English speakers to prevent possible confounds caused by second-language participants being unable to adequately express their interpretation of a diagram in English (ISO 9186-1 also requires the test be administered in each participant's native language [45]). While one motivation for this work was to address localization concerns associated with software agreements, I wished to first establish how well these diagrams could communicate the abstract concepts of software agreements with a relatively homogenous set of participants, with respect to culture and language. However, the condition without the

text-based agreement provides us with an approximation of how well these agreements might fare with individuals whose native language is different than the that of the software agreement.

3.6.5 Results

Figure 3.13 shows histograms of total correct score by condition. Inter-rater reliability (percentage agreement) for the scoring was 82.8%. Across all responses, only two were scored as "wrong and opposite". In both cases, these misinterpretations did *not* arise from users believing some data was not being collected when it actually was. This result is noteworthy: While the diagrams are not always understood, participants did not misinterpret them in ways that could adversely affect their privacy.

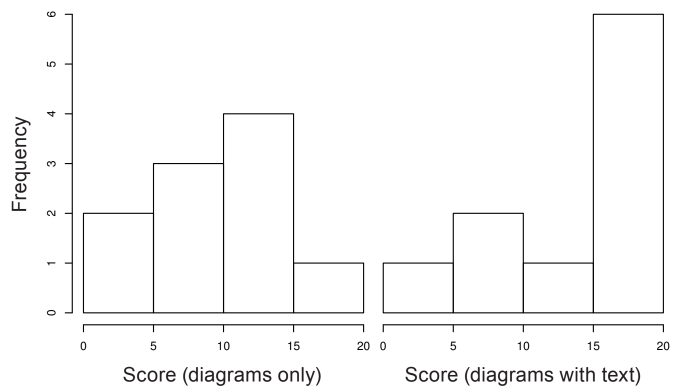


Figure 3.13 *Histograms of total correct score, by condition*

Mean total correct scores were 10.0 out of 20 (SD=5.77) and 13.6 out of 20 (SD=5.97) for the diagrams-only and diagrams-with-text conditions, respectively. Median scores were 11.5 (min=0, max=16) and 16 (min=0, max=19). The distributions did not differ significantly, but there was a trend (Wilcoxon rank sum W = 27, $p < 0.1$ two-tailed). Visual inspection of the histograms supports this observation of a trend; there appears to be a clear tendency to perform better when agreement text is available (See Figure 3.13).

In terms of the questionnaire data, mean and median informativeness ratings were 3.4 and 4 out of 5, respectively (SD = 0.94), where 5 is "most informative". Visual appeal was rated a mean of 3.26 and a median of 3 out of 5 (SD = 0.73). Comprehensibility was rated a mean of 3.25 and median of 3.5 out of 5 (SD = 0.85). Significant differences between conditions were not seen for informativeness (Wilcoxon rank sum W = 46, $p > 0.7$ two-tailed) or visual appeal (Wilcoxon rank sum W = 41, $p > 0.7$ two-tailed). A significant difference was not seen between conditions for comprehensibility; however, there was a trend (Wilcoxon rank sum W = 32.5, $p < 0.2$ two-tailed), with a median comprehensibility rating of 3 in the diagrams-only condition and 4 in the diagrams-with-text condition.

3.7 Discussion

The primary goal of the experiment was to understand whether participants could comprehend the narrative pictograms, with or without text. The diagrams-only condition provided the most rigorous test of these diagrams, because participants needed to correctly infer what the diagrams depict without the benefit of explanatory text. Since these diagrams are the first to convey the abstract concepts of data collection policies, effective communication in the absence of supporting text is made even more difficult. Thus, to help understand the best one could hope these diagrams to communicate in conditions without text, the condition with agreement text serves to establish an upper bound on performance.

In each condition, most participants successfully interpreted the majority of the diagrams. As one would expect, participants in the diagrams-with-text condition performed better on average than those in the diagrams-only condition, with a trend towards significance. However, participants in the diagrams-only condition performed nearly as well. The fact that this latter group did nearly as well as those with more information available to them bodes well for the use of these diagrams in situations where users may not be able to speak the same language as that of the software agreement.

Given the communication challenges in teaching users to read the diagrams, the results of the study are very promising for two reasons: many of the problems with comprehension seen in the diagrams-only condition can be attributed to problems with particular diagrams; and not only did a majority of participants in both conditions correctly interpret the majority of diagrams, but the data suggest that correct interpretation of the functional overview may be key to understanding subsequent diagrams, a result that validates our composition strategies. These two results are discussed in greater detail below.

3.7.1 Difficulties with particular diagrams

As noted, on average, participants in the diagrams-only condition did not perform as well as those in the diagrams-with-text condition. However, this difference in performance can be attributed to a few problematic diagrams, as shown in Figure 3.14. This figure compares the number of correct responses for each diagram, with the two conditions paired together. What is clear from this figure is that much of the difference in performance scores is due to a few problematic diagrams located on the right side of the graph.

The two most problematic diagrams both suffered from a similar problem that was uncovered during post-test interviews. Consider Figure 3.15: in this diagram, it was my intention to convey that use of menu commands was recorded. In my interviews, however, participants noted that it was not clear that they should generalize the content of the diagram. Consequently, they tended to think that the diagram was informing them that use of the contrast tool is recorded. Three of the worst-performing five diagrams suffered from similar problems with generalization. It is not clear what prevented participants from generalizing in these cases, since many other diagrams require one to generalize, such as those depicting time zone, locale, and operating system (e.g., Figure 3.1). It may be that singling out a very specific command unique to this domain, rather than a generic command (such as "Menu Item 1" or "Menu Item 2") contributed to this problem, but more research is needed to understand and effectively address this issue.

Another problem among the worst-performing diagrams was that of domain knowledge. As a result of this issue, I amended my original set of composition rules to suggest considering the use of additional context for diagrams that depict complex domain-dependent information (see the composition rules, discussed earlier).

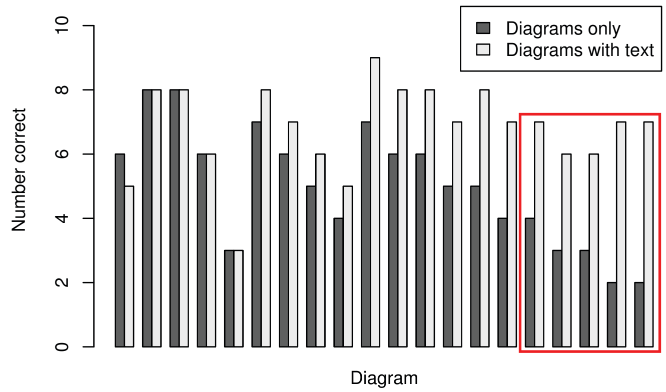


Figure 3.14 *Number of participants who correctly interpreted each diagram, ordered by the difference between conditions. The worst-performing five diagrams are circled in red.*

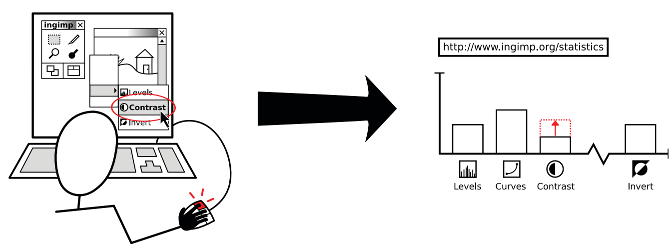


Figure 3.15 *Diagram depicting collection of command names; many users thought this referred only to use of contrast.*

3.7.2 Effect of functional overview

For a reader who is unfamiliar with narrative pictograms, learning how to interpret the visual syntax and vocabulary of narrative pictograms is key to understanding subsequent diagrams. As previously described, the functional overview—the first diagram shown to each reader—was designed to teach readers this syntax and vocabulary; therefore, the functional overview should be crucial to a reader's understanding of the overall set of diagrams. In fact, the data suggest that understanding the functional overview is key to understanding subsequent diagrams. The median score for all participants who did not correctly interpret the functional overview was 4.5 (n = 6, min = 0, max = 10); among those who correctly interpreted the functional overview, the median was 15.5 (n = 14, min = 8, max = 19). These results suggest that, when successful, the functional overview taught participants the diagrams' syntax, allowing them to understand the other diagrams.

3.7.3 Limitations

My evaluation provides evidence that the abstract concepts of data collection can be effectively conveyed using wordless diagrams. However, two specific issues have been identified that cause certain diagrams to underperform without accompanying text—problems with generalization, and lack of domain knowledge—and more work is needed to determine how best to address these issues. In addition, research is needed to understand how well the composition rules used for narrative pictograms generalize to other types of software agreements, such as EULAs. Finally, my summative evaluation, in contrast to my iterative design process and formative evaluation, considered only native English speakers. While the diagrams-only condition shows results comparable to the diagrams-with-text condition, and may be indicative of the experience of non-native speakers, future work should examine the effectiveness of this set of narrative pictograms with a more culturally diverse population.

3.8 Summary

This chapter presented narrative pictograms, an approach for communicating software agreement content using near-wordless diagrams. This technique helps to improve the comprehension of data collection policies for individuals who are not native readers of the language of the agreement text. This chapter presented composition rules for the creation of narrative pictograms, as well as example pictograms for the data collection policy of a piece of research software, and an evaluation of those pictograms. This work is one of the first attempts to comprehensively investigate the problem of localizing consent agreements, a significant problem for commercial software developers, open-source software developers, and those conducting research on the Internet.

While this work offers first steps towards tackling localization issues, the problem of capturing and retaining reader attention remains. The next chapter introduces a technique called textured agreements, which attempts to address this issue.

Chapter 4
Textured agreements

4.1 Introduction

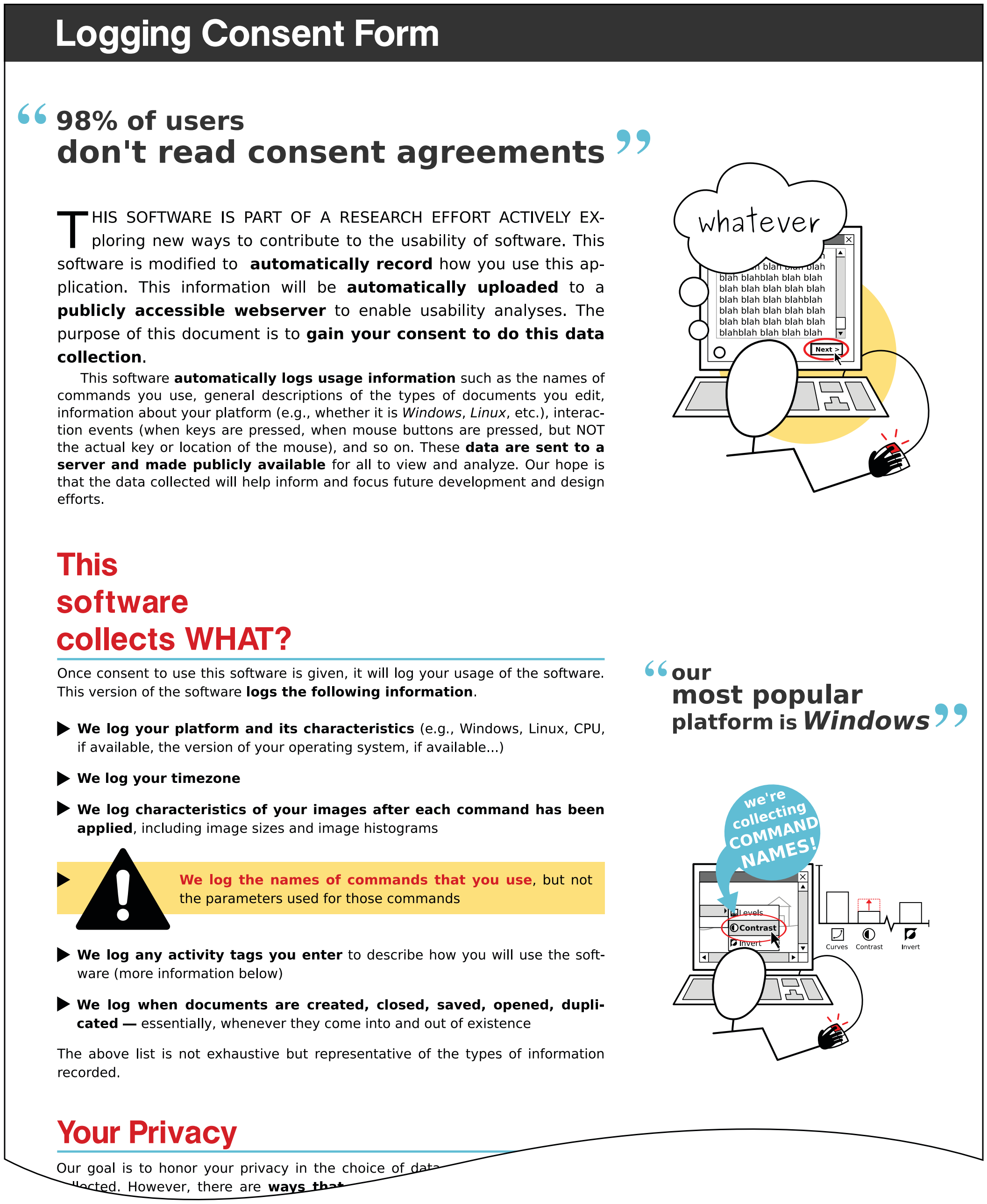
This chapter presents visually redesigned software agreements called *textured agreements* (e.g., Figure 4.1), which aim to capture readers' attention and direct it to the full content of the agreement. This chapter first describes the design method that led to the creation of textured agreements. It then presents the specific techniques that compose textured agreements, and describes the two experimental studies conducted to evaluate the effectiveness of the agreements. Finally, it discusses results and implications from both studies.

4.2 Design process

I wanted to develop an improved consent process that would capture reader attention and direct it to the primary object of interest—the agreement itself—thereby improving comprehension of the full agreement's terms. To explore the space of possibilities for improving the software agreement process, I engaged in rapid, low-fidelity prototyping and formative evaluations of a range of alternative methods for presenting agreements. An affinity diagram was used to group prototypes conceptually and to aid in identifying areas of the design space that had not yet been explored (see Figure B.1).

This phase of the research had 21 participants evaluating a range of prototypes. I used paper-based prototypes and computer-based mock-ups to explore the potential effectiveness of different approaches. Each prototype was presented to the participant as part of a mock software installation process, mimicking the standard context in which users encounter software EULAs; computer-based prototypes actually included simulated installers.

Each participant was asked to imagine they were downloading and installing the software for the purpose of evaluating the software's usability. That is, the participant was not initially



Logging Consent Form

“98% of users don't read consent agreements”

THIS SOFTWARE IS PART OF A RESEARCH EFFORT ACTIVELY EXploring new ways to contribute to the usability of software. This software is modified to **automatically record** how you use this application. This information will be **automatically uploaded** to a **publicly accessible webserver** to enable usability analyses. The purpose of this document is to **gain your consent to do this data collection**.

This software **automatically logs usage information** such as the names of commands you use, general descriptions of the types of documents you edit, information about your platform (e.g., whether it is *Windows*, *Linux*, etc.), interaction events (when keys are pressed, when mouse buttons are pressed, but NOT the actual key or location of the mouse), and so on. These **data are sent to a server and made publicly available** for all to view and analyze. Our hope is that the data collected will help inform and focus future development and design efforts.

This software collects WHAT?

Once consent to use this software is given, it will log your usage of the software. This version of the software **logs the following information**.

- **We log your platform and its characteristics** (e.g., Windows, Linux, CPU, if available, the version of your operating system, if available...)
- **We log your timezone**
- **We log characteristics of your images after each command has been applied**, including image sizes and image histograms
- **We log the names of commands that you use**, but not the parameters used for those commands
- **We log any activity tags you enter** to describe how you will use the software (more information below)
- **We log when documents are created, closed, saved, opened, duplicated —** essentially, whenever they come into and out of existence

The above list is not exhaustive but representative of the types of information recorded.

“our most popular platform is ***Windows***”

Your Privacy

Our goal is to honor your privacy in the choice of data ... llected. However, there are **ways that** ...

Figure 4.1 *Excerpt from an example textured agreement (full agreement in Figure C.4). Basic typographic manipulations are applied: headings, bullets, bold text, and a more distinct lead paragraph. Important content is also made more salient using factoids, vignettes, and iconic symbols.*

informed that the focus of the study was on understanding the effectiveness of the altered agreements in capturing and retaining users' attention. Each participant typically viewed multiple prototypes, so this minor deception was effective for only the first prototype. Therefore, to randomize learning effects, each participant was shown the prototypes in a different order. This same basic approach of supplying a distractor task was used in later experiments to attempt to replicate realistic scenarios in which users encounter software agreements.

4.2.1 Design insights

Dozens of prototypes were evaluated, employing techniques ranging from supplementary videos and illustrations to enhancements of the text-based agreement itself. I explored this design space by drawing inspiration from related fields—e.g., advertising, technical communications, and comics—and the various approaches attempted in previous work—e.g., summaries and video.

Video and animation was explored using prototypes derived in large part from narrative pictograms. A range of videos were developed, as well as prototypes that used *semantic skimming*, which—by analogy to *semantic zooming* [10]—shows high-level summaries of video content on the timeline and in the video frame when the user scrubs the video (e.g., Figure 4.2).

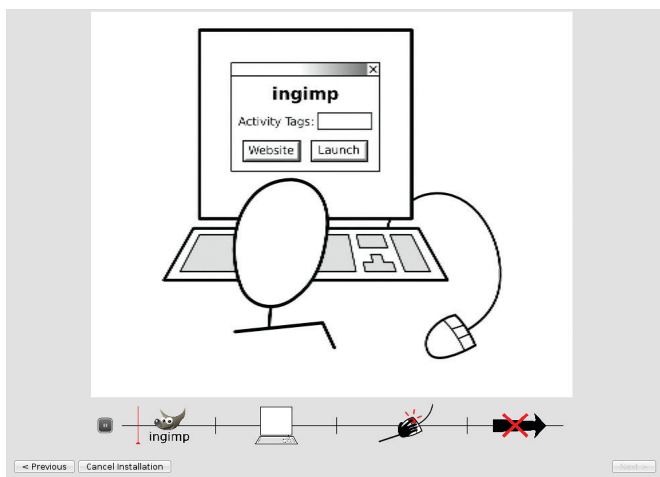


Figure 4.2 *Snapshot of a prototype video agreement that uses semantic skimming—note the high-level icons representing categories of video content on the timeline.*

When participants were shown additional material, such as videos or illustrations, they either expressed confusion or disinterest in the additional content. Most participants mistook the videos for advertisements, as advertising videos or slide shows are sometimes shown during the installation of other software. The poor performance of video echoes the results of Campbell *et al.* [17], who found that participants tuned out video presentations of medical consent agreements.

Among the many prototypes tested, visually redesigned text-based agreements appeared to be the most promising. In contrast to video or animation, when shown text-based agreements that made use of typography and graphic design, participants often stated they felt the software distributors were *making an explicit effort to communicate with them*. Participants saw these enhancements as an attempt to make the agreement process accessible and meaningful to them.

As a result of these responses, I decided to focus my prototyping efforts exclusively on cultivating a set of visual design strategies tuned to improving the software agreement process. In order to do so, I drew inspiration from visual design techniques used in popular media, such as magazines, newspapers, technical manuals, and how-to guides. These media face similar challenges in enticing their audiences to take notice and retaining that attention once captured.

To explore this design space, I obtained samples of a variety of popular media, ranging from *National Geographic*, to *Time Magazine*, to *Us Weekly*. Representative examples of possible techniques were posted on a wall, where they were annotated and then clustered into a large affinity diagram (Figure B.2). These design techniques were then abstracted and re-categorized using a second affinity diagram (Figure B.3). This approach yielded a set of candidate design techniques, which were used to create prototypes of redesigned software agreements. These designs were iteratively developed, tested, and refined with participants during the latter half of the design process. The result of this prototyping process was a repertoire of techniques that collectively form textured agreements.

4.3 Textured agreements

This section introduces *textured agreements*, visually enhanced software agreements. It is divided into two subsections: 1) A description of the overall goals I wish to achieve with the visual redesign of software agreements, and 2) a description of the specific techniques employed to reach these goals.

4.3.1 High-level goals

In redesigning software agreements, I wish to improve readers' comprehension of the agreement content. In order to do so, I established two high-level goals. First, I wish to introduce an *information hierarchy* to impart a clear visual organization to the material. This information hierarchy should serve to highlight important information in the agreement and suggest its personal relevance to the reader. Second, I wish to *capture and retain reader interest*. The chief means of accomplishing this latter goal is through the use of *visual variety* in the designs.

4.3.1.1 *Provide a clear information hierarchy*

The typical software agreement has a clear structure, with sections, subsections, clauses, and so on; however, the typical plain-text presentation of these agreements does not reflect that structure. Given the typical information density of a software agreement, there is a need to provide a clear visual hierarchy to its contents. For my purposes, I will broadly consider the concept of a "visual information hierarchy" to encompass the principles behind Tufte's layering and separation techniques in constructing visualizations [85], and Kress and van Leeuwen's use of salience and framing [49] in visual design.

A good information hierarchy will lend agreement clauses of greater importance greater visual weight, where—ideally—importance is defined from the user's perspective. For example, a clause that describes features that affect the user's privacy, such as the automatic collection of personal information, should be emphasized over a more common clause that the user will expect to be in an agreement, such as a standard limitation of liability. Identifying clauses of particular importance to users is discussed in greater detail in Chapter 5.

Imparting a clear and distinct visual hierarchy helps counter common problems with existing text-based agreements: the impression that documents are irrelevant, the inability to skim their content with ease, and the impression that they are long, difficult, and cumbersome to read. When done well, the resultant visual hierarchy yields a clear path for navigating the content, making it easier to locate pertinent information and giving an impression that the document contains information that is interesting and worthwhile. The full range of techniques employed to create this visual hierarchy—including typographic manipulation, white space, factoids, vignettes, and iconic symbols (Figures 4.3-4.5)—are described in detail later; a brief illustrative example of one technique is provided here.

Figure 4.3 *Factoids present interesting information related to the nearby content in the consent agreement.*

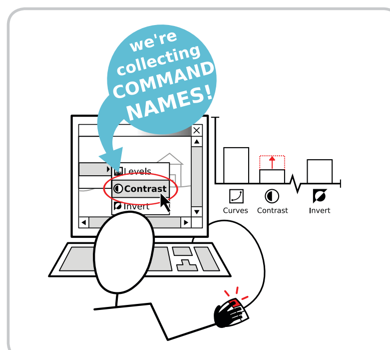


Figure 4.4 *Vignettes draw the reader in and communicate agreement content through mini-narratives.*

Figure 4.5 *Iconic symbols (e.g. warnings) highlight important information.*

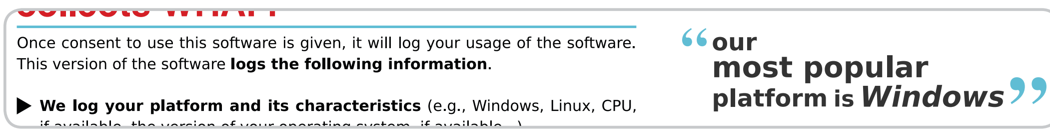


Figure 4.6 *This factoid hooks the reader with an interesting fact related to a nearby clause describing the collection of operating system data by the software.*

It is difficult for users to skim a plain-text agreement to find information they consider personally relevant as everything is contained in long paragraphs. To break up these swaths of text and highlight information likely to be of interest to users, one can use *factoids*, or tidbits of interesting information that stand apart from the text. For example, consider an agreement that describes the data collection policies of some software, where one piece of data collected is the user's operating system. One can highlight this fact with a factoid that states the most popular platform amongst the software's users (Figure 4.6). This introduces an interesting and memorable bit of information into the agreement and simultaneously clues the user into a specific consequence of accepting the agreement—that data are being collected—lending a sense of transparency.

Note that this factoid is also positioned directly next to the agreement text that describes the data logged by the application, illustrating an important feature of a good information hierarchy: readers can readily transition between levels of the hierarchy. More specifically, readers can enter the document at a high level—say, by skimming headers and factoids—then easily find related information nearby in lower levels. This skimmability lowers the barrier of entry for the entire document: when readers feel they can skim the agreement without negotiating paragraphs of text—and actually find interesting information while skimming—there is a greater chance they will subsequently read further into the content.

4.3.1.2 *Capture and retain attention with visual variety*

While the need for an information hierarchy dictates that one should raise information to higher levels of the visual presentation, as one begins to raise more elements higher in the hierarchy, one risks over-using specific techniques. For example, during formative evaluations, I found that users would not read more than 3 or 4 warning boxes (Figure 4.5) per document; overuse of those elements rendered them unable to retain interest. It is therefore important to employ *visual variety* when selecting amongst techniques in order to continually *capture and retain interest* in an otherwise lengthy document. For example, one could use headlines, warning symbols, or vignettes to highlight privacy information in an agreement. The choice of which technique to use, where, and when, is partially informed by the need to create visual variety in the overall design.

At a large scale, textured agreements use *progressive exposure to different visual design techniques* to create visual variety. New visual elements are continually introduced throughout the agreements to entice readers to take notice. Devices like factoids and vignettes (Figure 4.4) are particularly effective means for introducing variety, since they are independent of the primary body of text and can thus be placed nearly anywhere, within reasonable proximity to related information. Balancing the use of these various techniques is important: concentrating all instances of a technique in one area of the document will render it ineffective. The need to maintain this balance constrains how much information can reasonably be placed at higher levels of the document.

When a document uses an information hierarchy and visual variety together effectively, it suggests an accessible *pacing*—or reading rate—for that document. Pacing influences a reader's assessment of the effort required to read a document. As an example, the frequency of headings can communicate to the reader how quickly a document can be read, and how easily it can be skimmed [94]. Plain-text agreements—with no headings, or typographic manipulation in general—create an impression that the document is lengthy and arduous to read [94]. Through application of the aforementioned techniques, textured agreements create a *textual pattern* that suggests one can easily move through the document, at various levels of detail, to glean the most relevant information from it.

Having established these high level goals, the next section describes how to select and apply the various techniques to satisfy these goals.

4.3.2 Specific techniques

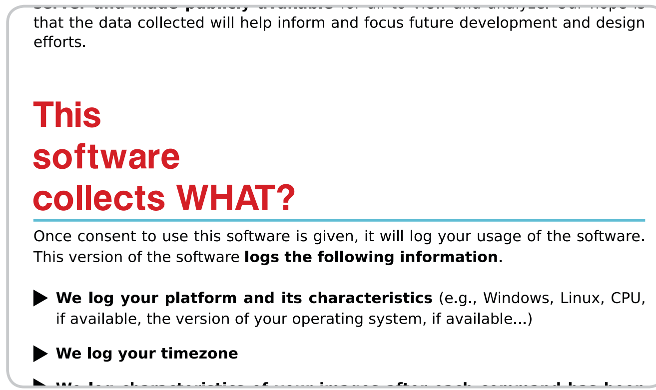
The primary techniques used to achieve the goals described above are typographic manipulation, pull-quotes, vignettes, and iconic symbols (Figures 4.3-4.5). While these methods are commonly used in other media (see the aforementioned [49,85,94]; practical examples of various techniques are also discussed elsewhere [25,88]), my contributions lie in the selection and adaptation of these strategies to the design of software agreements, and in the evaluation of the effectiveness of these techniques. This section describes how these techniques are used to create textured agreements[1].

---

1 Further examples of these specific techniques, along with template agreements to facilitate adoption, are available here: http://hci.uwaterloo.ca/research/textured.

4.3.2.1 *Typographic manipulation*

Given that software agreements are most commonly presented as plain text, it is not surprising that the overwhelming impression among users is that agreements are both irrelevant and boring. As a first step towards improving the presentation of an agreement, one can break up unformatted and uninviting walls of text using simple typographic manipulation: one can bold key terms and phrases, and add headers and bullet points. For example, the body text of the agreement in Figure 4.7 has been made more inviting and relevant by adding headers and bolding key terms of interest to readers. These additions create the first level of the information hierarchy: readers can now skim key phrases and headers in the document.


that the data collected will help inform and focus future development and design efforts.

This software collects WHAT?

Once consent to use this software is given, it will log your usage of the software. This version of the software **logs the following information**.

- **We log your platform and its characteristics** (e.g., Windows, Linux, CPU, if available, the version of your operating system, if available...)
- **We log your timezone**

Figure 4.7 *An example of the use of typographic manipulation: headers, bold text, and bullet points make text more inviting.*

4.3.2.2 *Pull-quotes and factoids*

While typographic manipulation has made this agreement more inviting, one would like certain information to be even more accessible to users when skimming the agreement. A *Pull-quote* is a catchy or interesting quote taken from the primary text, shown separately from the main text. A pull-quotes uses a different font, often larger than the body font, to make it visually distinct and noticeable. They are used in a variety of media, such as websites, newspapers, and magazines, to improve the skimmability of documents [88] and to convey key concepts through quickly consumed, bursty nuggets of information. In the case of software agreements, the text is typically not pithy—and in fact is generally rather dry—making actual pull-quotes impractical. Instead, I take the approach of writing factoids, which are laid out similarly to pull-quotes, but do not use text taken directly from the agreement body.

In my agreements, I have taken two approaches to writing factoids: highlighting interesting facts that arise as a consequence of agreement terms—such as the earlier example where I hint at data collection by stating the most popular platform among the software's users (Figure 4.6)—and using humorous or even guilt-inducing facts to hook users at the top of the agreement, such as a factoid that exclaims, “98% of users don’t read consent agreements” (Figure 4.8). The evaluation shows that this particular factoid contributed to many users reading the rest of the document: it simultaneously acknowledges the tedium of reading a software agreement while suggesting that this particular agreement is different, accessible, and worth reading.

It is important to keep factoids germane to the software agreement or the agreement process. At the same time, a factoid’s content should be chosen and worded to suggest that the document is *accessible* and includes interesting information that is relevant to the user, rather than just boilerplate legal text. When done effectively, a factoid will lend a human, personal element to the agreement, elevate important agreement content to a higher level of the hierarchy, and pique reader interest in the agreement content.

> **“98% of users**
> **don't read consent agreements”**

Figure 4.8 *An example of a factoid used at the top of the textured agreement in Figure 4.1 to draw readers in.*

4.3.2.3 *Vignettes*

A vignettes is a mini-narrative related to the content of the agreement. Vignettes grew out of the work on narrative pictograms, but a vignette has a less restricted visual syntax, and is meant to be self-contained—it does not require a functional overview. As with a factoid or pull-quote, a vignette helps both to capture user interest and to create a clear information hierarchy by raising important content to higher levels of the hierarchy. One potential strength of a vignette is to make a more direct, literal connection with the reader, since a vignette depicts users interacting with the software—or even with the software agreement (Figure 4.9)—through

Figure 4.9 *An example of a vignette used to achieve an effect similar to that of the factoid in Figure 4.8.*

a comic-like medium. For example, the agreement in Figure 4.1 employs a vignette that depicts the user clicking on the contrast tool. The vignette highlights that this information is collected with the exclamation, "we're collecting COMMAND NAMES!" (close-up in Figure 4.10). The informal nature of the illustration, its suggestion of an underlying narrative, and its sensational text add interest, while the involvement of the reader by implication—as a potential user of the software—adds personal relevance to the content.

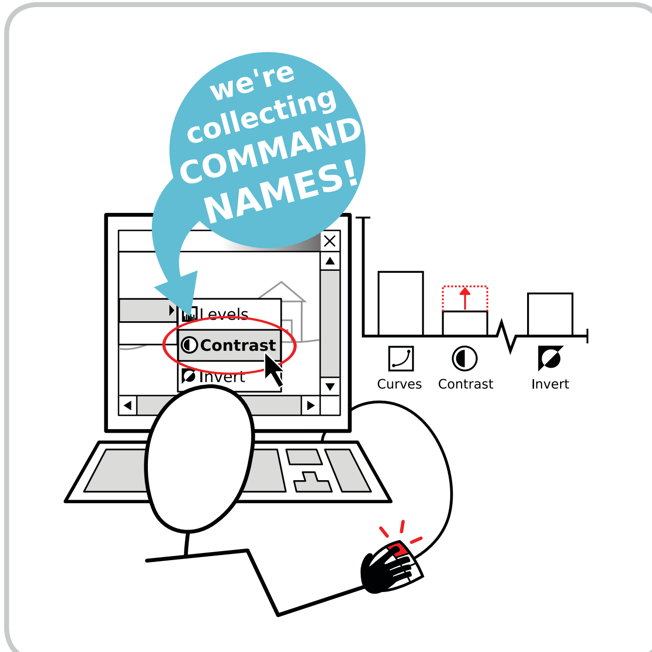


Figure 4.10 *A vignette that highlights the collection of command names by the software.*

4.3.2.4 *Iconic symbols*

Operational manuals and technical manuals often use warnings to alert readers to information vital to their personal safety. Warnings are similarly used in the workplace (see Wogalter and Laughery's summarization of research on workplace signs and labels for more on the topic [93]). These warnings are frequently accompanied with an icon to demarcate and classify the warnings (as in Figure 4.11).

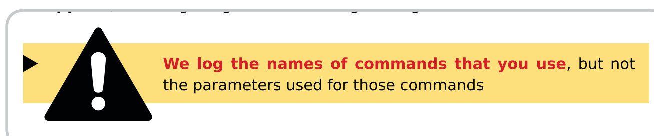


Figure 4.11 *A warning that highlights the collection of command names by the software.*

To help capture reader interest and create an information hierarchy, textured agreements use warning symbols and colored boxes to highlight particularly sensitive information in the software agreement, with the hope that even those who quickly skim the document will stop and read the content associated with the warning.

My formative evaluations suggest that warning symbols must be used sparingly to ensure they are perceived as highlighting truly exceptional or hazardous features of the agreement. Otherwise, users become desensitized to them and read only the first few warnings they encounter. This is consistent with previous work showing that warning symbols must be used only with truly hazardous information in order to retain their *arousal strength* (e.g., [8,34,93]).

4.3.3 Summary

The techniques discussed above must be applied with the overarching goals of building an information hierarchy and balancing visual variety. Evaluations of textured agreements will show that, using these techniques, it is possible to break down the monotonous presentation typically associated with software agreements, creating a more interesting and accessible document. Note that there is no single template to achieve these goals. Instead, textured agreements represent a set of *strategies* one applies to the visual design of a software agreement to achieve these ends.

The sections that follow present results from two experiments which suggest the effectiveness of these techniques.

4.4 Experiment 1

This first experiment was primarily concerned with understanding textured agreements' overall effectiveness in capturing and retaining readers' attention compared to existing approaches. Experiment 2, presented later in this chapter, will examine whether increased attention paid to the agreement translates into greater comprehension of the agreement content.

4.4.1 Experimental design

A between-subjects deception experiment was devised to evaluate the ability of textured agreements to capture reader attention. The study employed five conditions corresponding to five different agreement styles: three conditions with textured agreements' visual techniques applied to varying degrees (minimal, moderate, heavy), a pre-installation summary condition similar to that of Good *et al.*'s study [30], and a control condition with plain-text software agreements. The summary condition was included to partially replicate Good *et al.*'s previous study and provide another point of comparison for the textured agreements.

Participants were asked to download, install, and use three image manipulation applications from a mock webpage (Figure 4.12) for the purpose of choosing the best application for rotating images in a digital photo collection. Each application's installer was a custom-written installation program that could be manipulated to show a different agreement. The installer was instrumented to collect interaction events—in particular, the time spent on each screen of the installer and scroll events in the agreement itself. After choosing an application, participants completed a questionnaire.

To increase ecological validity, I carefully designed the study to minimize the chance that participants would artificially focus on the software agreements. I took two measures to avoid this potential bias. First, deception was employed. Participants were told that the purpose of the experiment was to learn how they choose software when multiple choices exist. Second, a verbal consent script was used to obtain initial consent to participate in the study. This approach was motivated by the observation during my formative evaluations that written consent agreements primed participants to look at the software agreements: reading and agreeing to a paper consent agreement made them more cognizant of text-based consent processes in general. Participants were debriefed upon completion about the true nature of the experiment and given a second, paper-based consent form to provide consent to keep their data. No participants refused to accept the second consent form.

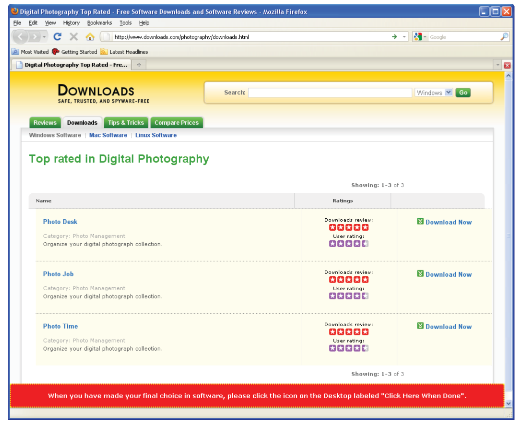


Figure 4.12 *Mock webpage shown to participants in experiment 1. The order of the application names on this webpage was counterbalanced between participants.*

4.4.2 Procedure

Participants were given a written scenario and instructions after obtaining verbal consent. The scenario indicated that they had recently received a digital camera, but lacked software to perform basic manipulations of the images. Accordingly, they were told to imagine they had just found a website with three different photo applications. The instructions indicated that they could download, install, and use any of the applications. Their specific task was to decide which software they would choose to use for the purpose of rotating images. A folder of improperly-oriented photographs was provided to assist with their evaluations. Once they had made a choice, the instructions indicated that they should start a questionnaire to record their final selection.

After receiving and reading the instructions, participants had the opportunity to ask questions. They were then seated at a desktop computer with a Web browser already opened to the download webpage to perform the task. Participants could then install and use any of the three applications provided. The applications were image library management applications

A. Minimal B. Moderate C. Heavy

Figure 4.13 *Examples of each textured agreement condition from experiment 1 (see Appendix C for full agreements from each condition). Approximately the same portion of each agreement is shown.*

that support basic image manipulation, such as scaling and rotation. After completing the task, participants launched a post-task questionnaire on the computer. After completing the questionnaire, participants were debriefed about the actual intent of the study and given a written consent agreement.

4.4.3 Experimental conditions

I developed three separate plain-text software agreements for the applications, drawing from existing software agreements. The content was designed to be consistent in form and presentation across all three agreements. Each agreement indicated that the application was instrumented to collect data, though the specifics of what was collected, why, and by whom, varied per agreement.

Three textured agreement *templates* were developed representing minimal, moderate, and heavy application of the techniques. These templates were applied to each of the plain-text agreements, yielding three separate instantiations of a template per condition (Figure 4.13 gives examples of each experimental condition).

In more detail, the conditions in the experiment were as follows:

1. ***Control condition.*** A plain-text software agreement.

2. ***Minimal condition.*** Based on lessons learned in the formative evaluation, I hypothesized that merely adding visual decoration to a software agreement would not be enough to increase reading times. Instead, my experiences suggested that one needed to conscientiously apply the textured agreements' techniques. To test this hypothesis, the minimal condition represents an aesthetically pleasing software agreement with visual adornments. However, while

aesthetically pleasing, the visual design does not otherwise strive to reinforce the *content* of the agreement. (Figure 4.13A).

3. ***Moderate condition.*** This condition represents what I feel is a balanced application of the strategies of textured agreements. This agreement uses a more scholarly heading font, warning boxes for three important agreement clauses, and factoids relevant to agreement content. (Figure 4.13B).

4. ***Heavy condition.*** This condition was designed to incorporate as many of the techniques as possible to make a visually dense, deeply layered agreement (Figure 4.13C).

5. ***Summary condition.*** One-page summaries for the three software agreements were developed using the heuristics used by Good *et al.* [30,31] (Figure 4.14).

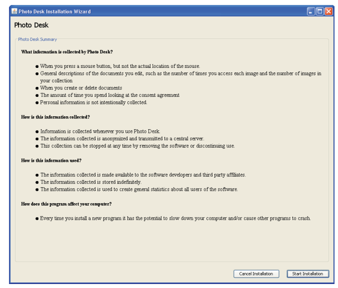


Figure 4.14 *Example summary condition from experiment 1.*

To avoid order effects, the order of agreements paired to installers was counterbalanced. The order of the software applications' names on the webpage was also counterbalanced. To avoid potential effects due to pairing the same application names to the same agreements, there were six different application name-agreement pairings. Applications were given the fake names "Photo Time", "Photo Job", and "Photo Desk" to avoid the effect of brand recognition; consequently, I had to vary application name-agreement pairings to avoid effects caused by one name bring more intrinsically attractive than another. This yielded a 5×6 between-subjects design with 5 conditions and 6 application name-agreement orderings.

4.4.4 Performance measures

I measured the effect of each treatment by recording the amount of time participants spent on individual installer screens and the maximum amount the software agreement was scrolled.

Because I strove to create a more realistic experience, participants could run each installer multiple times. However, this capability complicates measures of time spent in the installer screens. Accordingly, my timing measures are derived from the *first* time an installer is run, which I refer to as *first-run timings*. This measure is likely to more closely correspond to actual practices, since people typically only run an installer once. For each participant, an average timing is calculated from first runs of the installers. If a participant does not run a particular

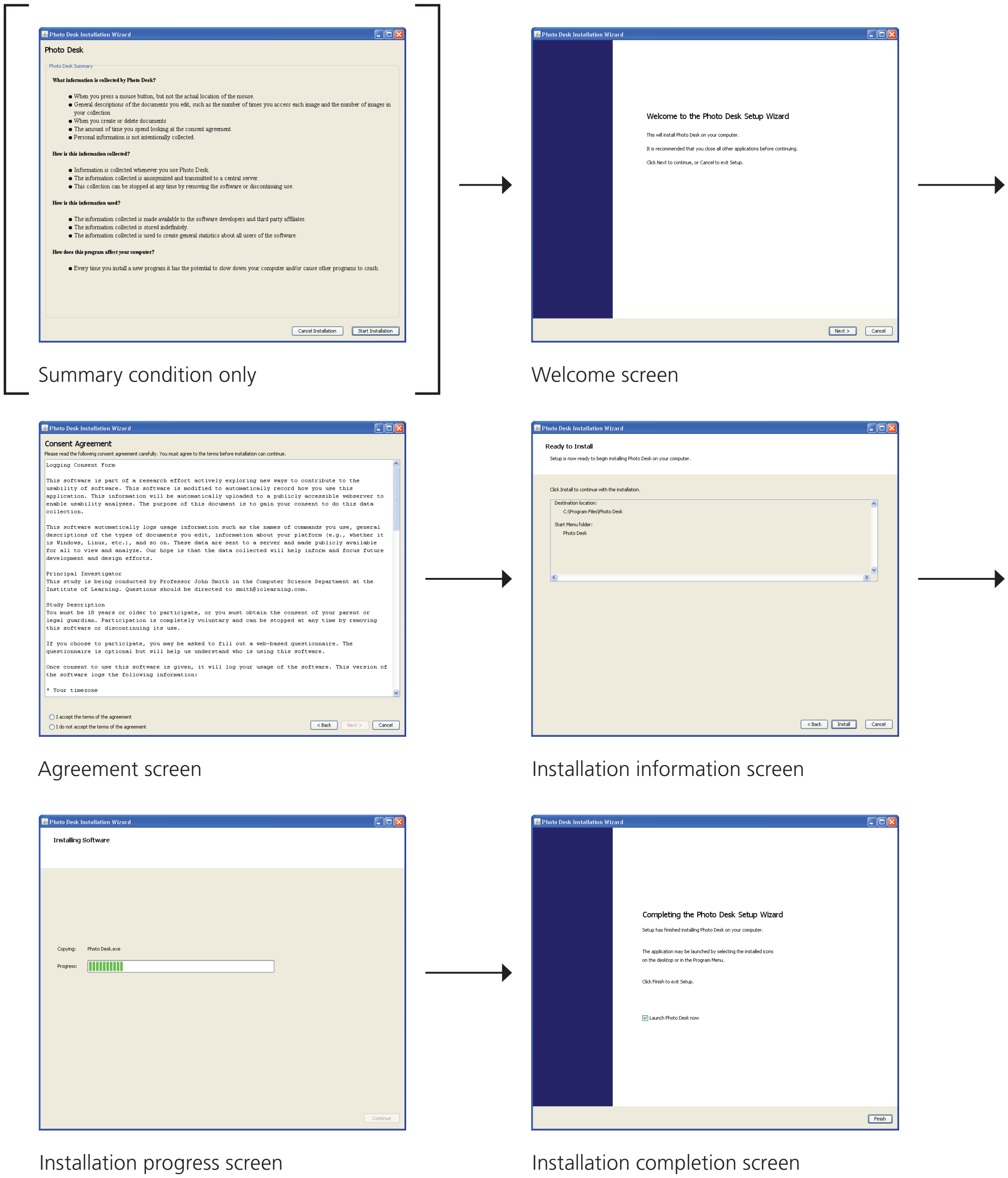


Figure 4.15 *Installation process from experiment 1. The summary screen (in brackets) was shown only in the summary condition. The remaining screens were shown in all conditions.*

installer, it is not included in the calculation of the participant's average time. The maximum amount an agreement is scrolled is calculated the same way.

4.4.5 Experimental apparatus

The study was conducted using a basic Windows XP installation on a VMWare virtual machine that had the Internet connection disabled. Using the snapshot feature of VMWare, I was able to have identical start conditions for every participant. A setup script was run before each session to set the experimental conditions for each participant based on their participant number.

The applications' installer was a custom-developed installer written in Java. Its design used the Windows look-and-feel, and mimicked the appearance and feature set of a typical Windows installer. The installer was instrumented to record when it was opened and closed, the time spent on each screen, interactions with controls, and whether participants canceled or completed the installation.

With the exception of the summary condition, each installation process was identical (see Figure 4.15). The first screen was a welcome screen; the second, the software agreement screen; the third, a screen informing the user where the software will be installed; the fourth, a screen showing the installation progress; and the fifth, a screen indicating the installation was finished. The summary condition added an additional screen, a summary of the software agreement, which was shown before the welcome screen. This ordering of screens mirrors that of Good *et al.*'s study, though I did not include a blank screen before the welcome screen in non-summary conditions. While Good *et al.* included this screen to act as a control for the presence of the summary screen, I wanted to increase the ecological validity of my study, and thus did not include it.

The installation program did not actually install the application—the applications were already installed—but did copy all of the files to a temporary directory to simulate the installation process. It also added shortcuts to the Start Menu.

4.4.6 Participants

90 participants were recruited in a university setting. Six dropped out, providing 84 participants, or 17 participants per condition, with the exception of the heavy condition, which had only 16. Participants were compensated with a $10 gift certificate for a coffee chain. 43 females and 41 males participated, aged 17-47 years old (mean=24, SD=6). Participants' self-reported computer expertise on a five point scale was an average of 3.4 (SD=1) with 5 being "most expert".

4.4.7 Results

4.4.7.1 *Timings, scrolling behavior, and reading self-reports*

An analysis of agreement screen timings indicates three outliers, one each in the minimal, summary, and control condition. These three participants spent an average time of 250, 579, and 433 seconds, respectively, on the agreement screen, each of which is more than 3 standard deviations from the mean of their respective conditions. These outliers were dropped from the timing analyses and are not represented in any graphs presented here. Apart from these outliers, there was considerable variation in reading habits. In the post-task questionnaire, participants were asked to self-report their tendency to read software agreements in general on a five-point scale: "never noticed before", "never reads", "rarely reads", "often reads", or "always reads". This self measure of reading habits was a contributing factor to the scores, and thus I include it as a factor in my analyses.

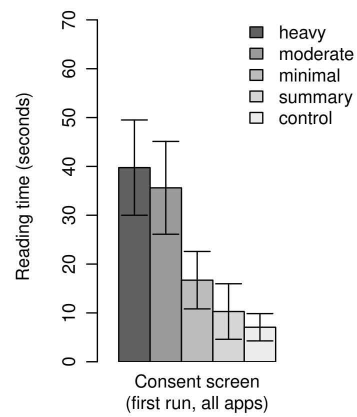


Figure 4.16 *Mean time spent on the first run of the consent screen (seconds with standard error).*

Figure 4.16 presents a plot of the first-run agreement screen timings for each condition. The heavy condition features the longest agreement screen time (mean = 39.8 seconds, SD = 39), followed by the moderate condition (mean = 35.6, SD = 39.2). Mean times in the summary and control conditions were 10.3 and 7.1 seconds, respectively. An ANOVA indicates significant differences between conditions ($F_{4,76} = 5.65$, $p < 0.01$). Post-hoc Tukey tests indicate significant differences between: heavy and control ($p < 0.01$), heavy and summary ($p = 0.026$), moderate and control ($p = 0.026$), and a trend for significance between moderate and summary ($p = 0.063$). No other significant differences were found between conditions.

An ANOVA also found significant differences between conditions for scrolling ($F_{4,76} = 3.96$, $p = 0.014$). Post-hoc Tukey tests demonstrate significant differences between heavy and control ($p = 0.04$), and heavy and summary ($p = 0.02$).

The questionnaire asked participants to self-report how much they read each software agreement on a five-point scale. Self-reported reading amounts were found to be significant with respect to condition ($F_{4,76} = 3.16$, $p < 0.05$), with post-hoc analysis indicating the differences are due to participants reporting they read the agreements more in the moderate condition than the control condition ($p < 0.05$).

4.4.7.2 *Aggregate timings across all installations*

Participants had a tendency to cancel installations at the *summary* screen. This observation echoes Good *et al.*'s findings and suggests the effectiveness of the summaries in communicating information. However, this tendency also slightly complicates comparisons of first-run time spent on the agreement screen, since participants may not reach that screen the first time the installer is run. Accordingly, I also examined *total pre-install time*: the sum of the time spent on all screens up to, and including, the agreement screen. In the summary condition, this includes the summary, welcome, and agreement screens. For all other conditions, it includes only the latter two screens. For this measure, I also sum the time spent in these screens across *all* installation runs. While this measure is not perfect—there is variability in how often people ran an installer more than once, and each run adds to this measure—it helps assess the potential impact of summaries by summing time across screens. I define the *total maximum scroll amount* in the same way. Figures 4.17 and 4.18 include summaries of these measures.

An ANOVA found no significant differences between the treatment conditions for total pre-install time, though there is an apparent trend ($p = 0.089$). This trend appears to be due to

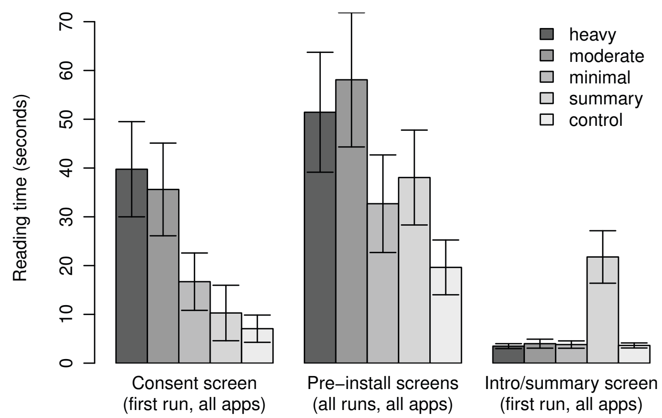


Figure 4.17 *Mean time spent on installer screens (seconds with standard error): time on consent screen only (from Figure 4.16); time on summary, welcome, and consent screens; and time on summary and welcome screens only.*

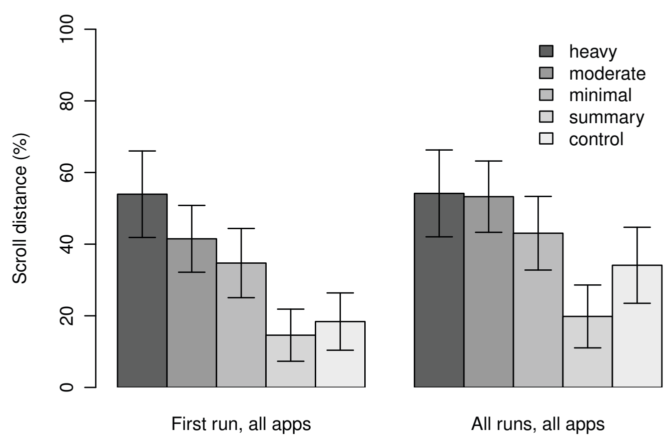


Figure 4.18 *Maximum scroll distance into the consent agreement (% of document with standard error): average over first runs only and average over all runs.*

both the moderate and heavy conditions. This hypothesis is supported by an ANOVA of scrolling behavior across all installation runs—rather than just the first run of each installer—which shows significance ($F_{4,76} = 3.36$, $p = 0.026$). Post-hoc analysis indicates differences between the heavy and summary condition ($p < 0.05$) and the moderate and summary condition ($p < 0.05$).

4.4.7.3 *Assessing the summary condition*

The summary condition was not found to significantly affect the reading time of the actual software agreements. However, this finding does not indicate that summaries are ineffective. A comparison of time spent on the summary screen to time spent on the welcome screen of the other conditions indicates that participants *do* read the contents of the summary screen, spending 18 seconds longer on this screen than the welcome screen of the other conditions ($p < 0.0001$) (see Figure 4.17). However, the data indicate that there are side effects associated with reading the summary. In particular, the data argue that summaries reduce the likelihood that people spend time reading the *actual* software agreement, or are less likely to read as far into the summaries. This echoes findings from a previous study by McDonald *et al.* [56] that examined participants' accuracy in a multiple-choice quiz on the content of privacy policies with different presentations. That study found that participants were significantly less accurate for summary agreements than plain text agreements when the answer to a question was not present at the summary layer. As hinted at in the study, participants were less likely to read into the full text to find the answer. Thus, while summaries have the potential to effectively communicate a *condensed* version of the agreement, they do so at the cost of discouraging the reading of the *full* agreement.

4.4.7.4 *Preferences*

Participants were asked to rate the visual appeal of the software agreements. An ANOVA indicates a significant difference in visual appeal between conditions ($F_{4,76} = 7.61$, $p < 0.001$), with post-hoc analysis revealing significant differences between heavy and control ($p < 0.01$); heavy and summary ($p < 0.01$); moderate and control ($p < 0.05$); and moderate and summary ($p < 0.05$). The minimal condition was also found more appealing than the summary condition ($p < 0.05$) and trended towards being more appealing than the control condition ($p = 0.10$).

4.4.7.5 *Qualitative feedback*

In the questionnaire, participants were asked to provide qualitative feedback regarding the visual appearance of the software agreements and their overall informativeness. A number of comments suggest the techniques worked as intended. For example, a participant in the heavy condition commented:

It got me to read them, when I install other programs, I NEVER read them. Big letters, organized points, and cartoons help. I think the organization was the most important.

A participant in the moderate condition suggests the effectiveness of the documents' pacing:

Compared to other agreements on other programs, this one is appealing because of the headings; which are placed similar to a newspaper to get one's attention. On other programs, it is just a bunch of words bunched together, similar to a contract but on the monitor.

Comments from participants in the minimal condition indicate they noticed the changes in visual appearance. However, they did not comment on the organization of information or the notion of being compelled to read, as in the moderate and heavy conditions.

While the techniques were generally successful, a few comments indicated that not everyone found the redesigns visually appealing. For example, a participant in the heavy condition wrote:

I think I would have read it more closely if it had been a little less over-the-top. I did really like that it was different and caught your attention.

Another participant noted that the heavy style was "somewhat obnoxious in coloration and layout." These comments suggest that while the heavy application of techniques attracted attention for some, it may be too much; the moderate application may strike a better balance.

4.4.8 Summary

This experiment saw increased reading times in the moderate and heavy conditions. The lack of an observed increase in the minimal condition hints that the techniques have an effect beyond the initial impact of the design or its novelty; however, I wanted to determine the extent of that effect on participants' retention of agreement content. I designed a second experiment to focus on this question.

4.5 Experiment 2

4.5.1 Experimental design

As in the first study, I conducted a between-subjects deception experiment, but with only two conditions: a textured agreement and a plain-text control condition. Participants were asked to download, install, and use a single image manipulation application, as opposed to three in the first experiment. The same instrumented installation environment was used as before. However, the distractor task of using the application was not actually performed by participants, though the instructions asked them to use the software after installing it to rate its usability. Instead, participants were interrupted after reaching the point in the software installation process where the software would actually be installed. Instead of installing the software, the participant was stopped and given a content quiz to test how much information they absorbed from the agreement process. This approach minimized the time between exposure to the agreements and taking the quiz.

4.5.2 Procedure

As before, participants were given a written scenario and instructions after obtaining verbal consent. The scenario indicated that they had recently received a digital camera, but lacked software to perform basic manipulations of the images. Accordingly, they were told to imagine they had just found the website of an image manipulation application. The instructions asked them to download, install, and evaluate the application, and to decide whether they would continue to use this program on their home computer. Once they had reached a decision, the instructions indicated that they would be given a questionnaire.

After receiving and reading the instructions, participants had the opportunity to ask questions. They were then seated at a desktop computer with a Web browser already opened to the download page of the application, called “Program A”. Participants were then able to download and run the installer. After clicking the “Next” button on the software agreement screen in the installer, a full-page screen informed participants that the task portion of the study was complete, and that the researcher will set up the questionnaire. This screen had no visible controls, to prevent participants from accidentally or instinctively skipping past it.

The same VMWare setup and installation apparatus from the first study was also used in this experiment.

4.5.3 Experimental conditions

This experiment had two conditions: control and textured. An agreement from experiment 1 was used as the plain-text control.

In the first experiment, the lack of an increase in reading time for the minimal condition emphasized the need for a careful application of my techniques. Using feedback from the first experiment, I further iterated on the design of the textured agreements. I developed a single design that makes more use of vignettes and sensationalism than the moderate condition, but which is more judicious than the heavy condition in their use (see Figure 4.19). The improved design, applied to the same text as in the control condition, forms the textured condition for this experiment.

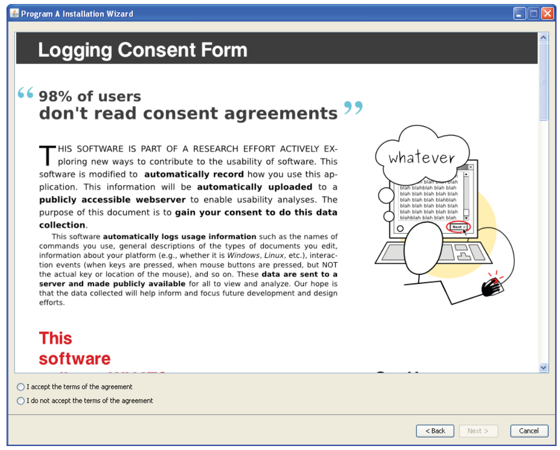


Figure 4.19 *Consent screen showing a textured agreement from the experimental condition in experiment 2. This is the same agreement shown in Figure C.4.*

4.5.4 Performance measures

As before, I measure the amount of time spent on individual installer screens and record the agreement scroll position over time. Since this experiment has only one application, and participants are interrupted immediately after completing the agreement screen, I can measure the time spent on the agreement screen without needing to consider multiple runs of the installer.

As part of the computer-based questionnaire, an open-ended content quiz was given to participants in order to gauge how much information they retained from the agreement. Participants were given 8 questions of varying difficulty that required them to recall content from the agreement. Each question began with the phrase, “According to Program A’s license agreement...” Some example questions include:

- According to Program A’s license agreement, what characteristics of your images does Program A record?
- According to Program A’s license agreement, if you have questions about the software or the study, who can you contact?

Questions were chosen from content in all areas of the agreement. Potential questions were piloted and refined to ensure that the questions could be answered fully if the entire agreement had been read.

Questions were scored out of 2 points each, making a maximum possible score of 16. A score of 2 was given to complete answers, and a score of 1 given to incomplete answers that nevertheless indicated some awareness of the content in question. For example, to the second question listed above (asking who participants can contact for more information) one participant answered:

*The primary investigator, John Smith, via email at smith@***learning.***.*

This answer was given a score of 2. Another participant answered:

There was an email address provided but I don't remember what it was.

This answer was given a score of 1.

4.5.5 Participants

28 participants were recruited in a university setting. Participants were compensated with a $10 gift certificate for a coffee chain. Participants were screened to ensure all were native English speakers to minimize potential effects arising due to language ability. 17 females and 11 males participated, aged 19-31 years old (mean=24, SD=3). Participants' self-reported computer expertise on a five point scale was an average of 3.5 (SD=1) with 5 being "most expert".

Measure		Control	Textured
Time on consent screen	*mean*=	19.7	53.6
(seconds)	*standard deviation*=	22.2	48.9
Maximum scroll distance	*mean*=	47.4	71.4
(% of document length)	*standard deviation*=	50.0	46.9
Open-ended quiz score	*median*=	0	4
(out of 16)	*interquartile range*=	4	5.75

Table 4.1 *Summary of experiment 2 results across conditions*

4.5.6 Results

4.5.6.1 *Timings and scrolling behavior*

As in experiment 1, participants were asked to self-report their tendency to read software agreements on a five-point scale. This measure of reading habits was again a contributing factor to the scores, and I include it as a factor in the analyses.

Table 4.1 summarizes the time spent on the agreement screen (in seconds) and the maximum scroll distance into the document (as a percentage of document length). As in experiment 1, the textured condition shows longer reading times than does the control (mean of 53.6 seconds versus 19.7 seconds), and an ANOVA shows that these differences are significant ($F_{1,26} = 7.54$, $p < 0.05$). Maximum scrolling distance into the document was also greater in the textured condition (mean of 71.4% versus 47.4%), and these differences were also significant ($F_{1,26} = 7.16$, $p < 0.05$).

In addition to measuring the reading time and the maximum scrolling distance, I examined the scrolling patterns of participants in finer detail. Figure 4.20 comprises a series of graphs representing the scrolling behaviour for 18 of the 28 participants. Shown are the

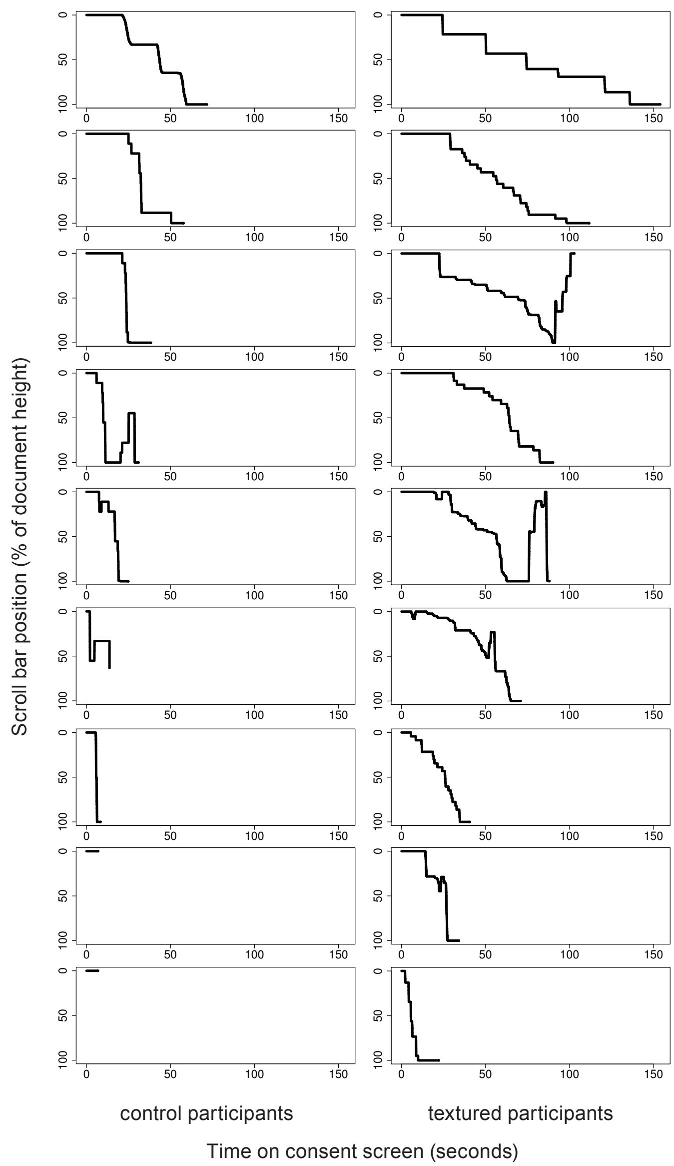


Figure 4.20 *From experiment 2, the scroll bar position as a function of the time since the consent screen was visible. The 9 longest-reading participants in each condition are shown.*

9 participants from each of the 2 conditions who spent the most time reading the agreement within their condition; the remaining 5 participants from each condition had similar scrolling behaviour to the lowest ranked participants shown. Participants are arranged in two columns: control condition on the left and textured on the right. Each column is sorted by reading time. From these graphs, it appears that participants in the textured condition are more inclined to read or skim through the entire agreement at a slower pace, while those in the control condition tend to skip directly to the end of the document, though they were not required by the software to do so. Only one or two participants in the control condition exhibit behaviour comparable to the top 6 in the textured condition.

4.5.6.2 *Content quiz*

Participant scores on the content quiz are also summarized in Table 4.1. As can be seen in Figure 4.21, both distributions are heavily skewed; as a result, median, not mean, is a more appropriate measure of the central tendency of these distributions. The median score in the textured condition was 4 (IQR = 5.74), compared to 0 (IQR = 4) in the control condition. An ANOVA on ranks suggests this difference is significant ($F_{1,26} = 4.18$, $p = 0.052$). Quiz scores were also found to be highly correlated with the amount of time participants spent reading the agreement ($r = 0.83$, $p < 0.0001$) (see Figure 4.22).

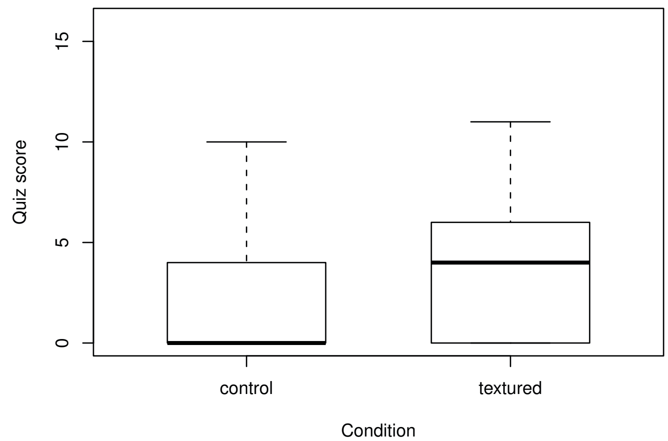


Figure 4.21 *Box plot of quiz scores in experiment 2.*

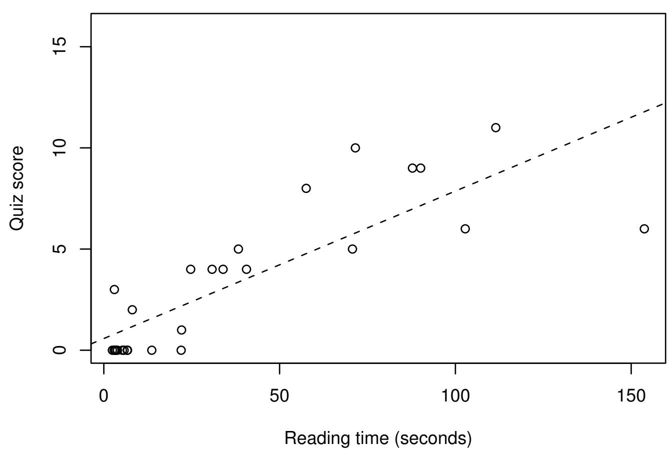


Figure 4.22 *Quiz score versus reading time in experiment 2 with linear regression (r = 0.83).*

4.5.6.3 *Qualitative feedback*

As in experiment 1, qualitative feedback was collected in a questionnaire and a post-task interview. In particular, participants were asked why they read less, more, or the same amount of content in comparison to their normal behaviour when installing software. Participants in the control condition all responded that they read the same amount as usual, and often made comments such as, "All of the EULAs are basically the same", or, "The same amount was read, since they generally all say the same things."

By contrast, a number of participants in the textured condition stated they read more content than usual, and cited elements of the agreements that pulled them in; for example:

> *I read more due to the diagram stating that 98% of the users don't read the License Agreement.*

Both in written responses and during interviews, participants in the textured condition noted being pulled in by prominent elements at the top of the document—such as the pull-quote mentioned in the above quotation (Figure 4.8), or a vignette serving a similar purpose (Figure 4.9)—and then were compelled to continue reading further into the document. These findings provide evidence that such attention-grabbing embellishments can pull participants into the content of the rest of the document.

4.6 Discussion

The results of both experiments support the notion that textured agreements compel people to engage with the agreements more than with plain-text agreements. In experiment 1, textured agreements increased the time spent on agreement screens from an average of 7 seconds in the control condition to an average of 36-40 seconds in the moderately and heavily designed treatments. Notably, these effects were not observed in the minimal condition, suggesting this increase in time cannot be attributed to novelty alone.

Experiment 2 saw a similar increase in reading time between conditions of 34 seconds, from 20 to 54 seconds. Longer reading times were also found to correlate strongly with higher quiz scores. This, combined with the higher scores in the textured condition, suggests that textured agreements are successfully capturing and retaining attention long enough for participants to absorb more information from the agreements. That participants are engaging more with the textured agreements is also evident in the change in reading behaviour seen in Figure 4.20:

more participants in the textured condition appear to be spending time looking through the *entire* document than in the control condition; this is also reflected by the increase in maximum scrolling distance between conditions.

The success of these techniques in increasing reader engagement is promising for two reasons. First, the techniques employed represent only a subset of the wide repertoire of visual design techniques available; there is significant room for further improvement. Second, the technique achieved its effectiveness by operating on the *primary object of interest*, namely, the software agreement itself. As discussed next, there are good reasons to consider improving this document rather than introducing auxiliary documents.

4.6.1 Human ethics perspective

Having observed the positive effects of the textured agreements, we met with three members of our internal review board who regularly review human ethics applications. We presented the textured agreements to gain their perspectives and understand potential issues in using them in practice.

The reaction to the agreements was extremely positive. Compelling study participants to read consent agreements is a problem they struggle with in study designs, so they welcomed the visual redesigns. However, they did have some suggestions for improving the designs and for future research. In particular, they observed that the heavily textured agreements could be problematic for seniors. This population might find the dense clustering of information distracting or difficult to comprehend.

4.6.2 Limitations and future work

That seniors may find heavily textured agreements difficult to read raises an important issue for future work: examining how these techniques affect readability and reading behaviors among different demographic groups, particularly seniors. In addition, while the results of the minimal condition in the first experiment suggest that more than novelty is required to engage readers, long-term evaluations of textured agreements are needed to fully evaluate their robustness with respect to desensitization and habituation. There is also a need for a more fine-grained understanding of the relative effectiveness of techniques at capturing attention—for example, factoids versus vignettes, or guilt-inducing factoids (such as "98% of users don't read consent agreements") versus informative factoids (such as "our most popular platform in Windows").

The success of textured agreements in capturing attention opens up other promising avenues for future work: there is a rich, ever-expanding repertoire of visual communication methods that can be drawn upon to further improve the consent process, of which only a portion has been explored in this work. Thus, this strategy has significant room to grow with respect to improving the consent process in software systems.

4.7 Summary

This chapter presented textured agreements, visually redesigned software agreements that draw upon strategies used in other visual media to gain users' attention, retain that attention, and highlight information of personal relevance. Studies demonstrate that these techniques show promise in improving the software agreement process—both reading time and comprehension were improved in textured agreements over plain-text agreements. My results also suggest caution in using techniques that partition the agreement process into multiple phases, as is done with summaries of agreements. While summaries are effective at conveying a synopsis of the agreement, they can lead users to ignore the full agreement.

While these techniques have been successful in capturing reader attention, the poor performance of the minimal condition suggests that these techniques must be applied with particular attention to users' concerns, lest users ignore the agreements. To that end, the next chapter explores users' attitudes towards a range of terms found in software agreements.

Chapter 5

User attitudes towards agreement content

5.1 Introduction

Since readers devote little attention to software agreements, it is necessary to maximize the useful potential information each reader may gain from an agreement within the short time they look at it. As discussed in Chapter 4, if an agreement calls attention to information a user does not find salient, that user will quickly lose interest in the agreement process. Consequently, whatever type of interface is used to present a software agreement—be it a summary, a textured agreement, narrative pictograms, or some other form—it is important to highlight[1] only information that users are likely to be interested in.

This chapter centres around an open coding of the reviews on the EULAscan website[2], a repository of "community-written EULA reviews". It first describes the EULAscan website and the method used to analyze the EULAscan data set. It then presents the results of that analysis: a characterization of the EULAscan participants, a breakdown of the types of products and services reviewed, and an analysis of the concerns raised by EULAscan reviewers. These concerns are broken into three categories: primary concerns (for example, data collection and data sharing), secondary concerns (for example, choice of forum), and open-source–specific concerns (for example, license compatibility). Finally, based on my findings, I make recommendations about what information to highlight in software agreements, and how.

1 In this section, the term "highlight" is meant in an interface-agnostic way. For example, in a summary agreement, "highlighting" refers to placing a piece of information in the summary layer; in a textured agreement, it might involve using a warning symbol, factoid, or vignette.

2 http://www.eulascan.com/

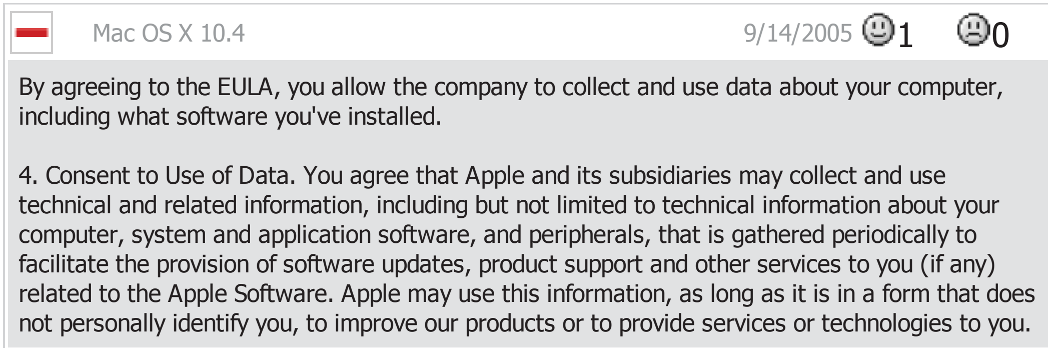
5.2 EULAscan

EULAscan is a website on which anyone can read or write reviews of license agreements anonymously. The About page for EULAscan describes the site as follows:

> *Simply put, EULAscan is a place where you may comment on End User License Agreements (known as EULAs), or read comments left by others.*
>
> *Have you found something suspicious in a EULA, and want to let others know about it? Add your comments to the site.*
>
> *Are you about to install a new program, and want to see if that 100 page EULA has something in it that you should be aware of? Search on the product's name, and read what other people have said.*
>
> *Why do people agree to EULAs without reading them? The two main concerns are comprehension and time. EULAscan hopes to combat that by letting the community work as a whole. Now, one comment can both explain what all that legalese means, and, save others the effort of reading through pages of text.*
>
> *Please contribute comments when you can. The more you share, the smarter the site becomes.*

Users can search for and comment on existing products or create entries for missing products they wish to review. While its stated focus is EULAs, EULAscan's participants have co-opted the website in order to review a range of types of licenses, including website privacy policies, open-source licenses, and even the agreement for the software automatically installed by a music CD—Van Zant's *Get Right with the Man*, one of the CDs involved in the 2005 Sony/BMG copy-protection scandal [58]. As a result, the set of reviews on EULAscan reflects user opinions on a variety of types of software agreements.

Each product has its own page on the site, containing the reviews for that product in chronological order. An example review is shown in Figure 5.1, and the corresponding review input form is shown in Figure 5.2. Review content consists of three primary pieces of information: a binary rating of "good" or "bad"—shown as a plus or minus symbol—the product version, and a free-form plain-text comment. The date of submission is also attached to each review;

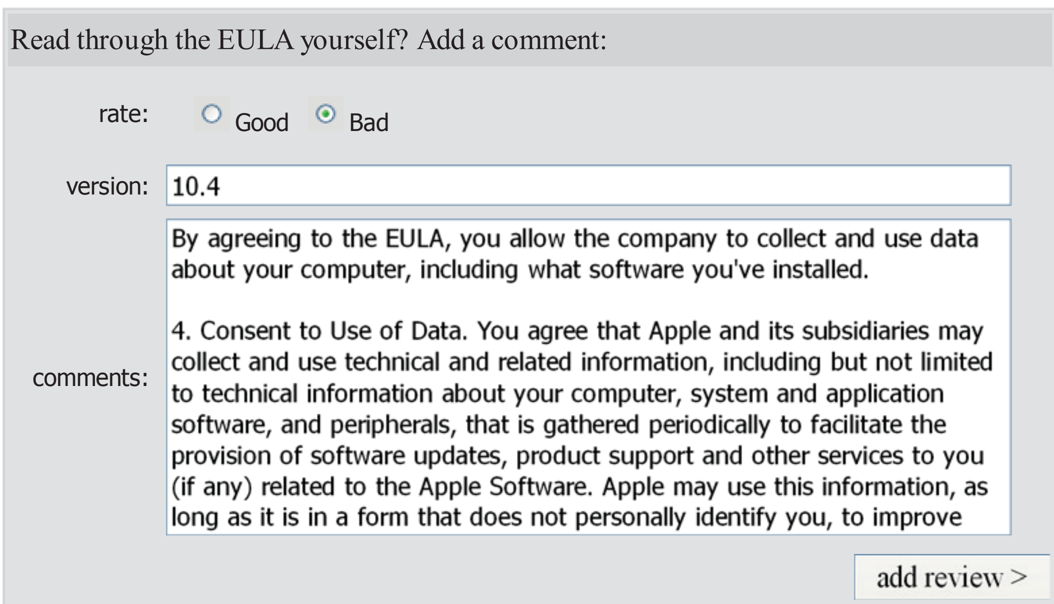


Mac OS X 10.4 9/14/2005 1 0

By agreeing to the EULA, you allow the company to collect and use data about your computer, including what software you've installed.

4. Consent to Use of Data. You agree that Apple and its subsidiaries may collect and use technical and related information, including but not limited to technical information about your computer, system and application software, and peripherals, that is gathered periodically to facilitate the provision of software updates, product support and other services to you (if any) related to the Apple Software. Apple may use this information, as long as it is in a form that does not personally identify you, to improve our products or to provide services or technologies to you.

Figure 5.1 *Example EULAscan review of the agreement for Mac OS X. The red minus sign in the upper left signifies a negative review (this is a binary indicator that may be plus or minus). The face icons in the upper right indicate the number of users that have "liked" or "disliked" this review.*



Read through the EULA yourself? Add a comment:

rate: Good Bad

version: 10.4

comments: By agreeing to the EULA, you allow the company to collect and use data about your computer, including what software you've installed.

4. Consent to Use of Data. You agree that Apple and its subsidiaries may collect and use technical and related information, including but not limited to technical information about your computer, system and application software, and peripherals, that is gathered periodically to facilitate the provision of software updates, product support and other services to you (if any) related to the Apple Software. Apple may use this information, as long as it is in a form that does not personally identify you, to improve

add review >

Figure 5.2 *Example EULAscan review input form showing the input that would produce the review seen in Figure 5.1.*

as of March 15, 2010, the earliest review is dated September 14, 2005, the most recent Jan 17, 2010. In addition, once submitted, other users may click on the "smile" or "frown" icons for a review to indicate that they like or dislike it; these data are recorded as a simple tally next to the corresponding icons. In addition to these simple feedback mechanisms, the chronological order of the reviews means that reviewers occasionally use product pages as discussion boards, posting responses to previous reviewers' comments.

5.3 Method

A snapshot of the EULAscan website was taken on March 15, 2010, including 143 products and services and 223 reviews[1] of those products' respective licenses. Of these, 7 products and 27 reviews were determined to be spam and were omitted from the analysis. Spam entries were either empty (one product), consisted of random strings of characters (several products and reviews), or consisted of URLs clearly intended for spamming purposes (a number of reviews). 5 additional reviews were exact duplicates of the preceding review, likely due to a user accidentally pressing submit twice, and were omitted. As well, 38 products had no reviews provided, or all provided reviews were spam, leaving 98 products and 191 reviews for further analysis.

An open coding of the reviews was conducted to examine both the concerns raised by EULAscan participants and the manner in which those concerns were raised. The corpus of EULAscan comments was printed out and systematically grouped using an affinity diagram (Figure D.1), generating an initial set of codes. These codes were then used to label the EULA-scan corpus, during which the codes were further refined. The codes were then grouped into higher-level categories using a second affinity diagram (Figure D.2), and the reviews were re-labelled with these categories. These categories form the basis of the analysis of user concerns later in this chapter.

Given that legal text is often difficult to interpret accurately, particularly for laypeople, it is important to note that whether or not users' interpretation of the legal text in question was accurate is not always relevant to this analysis. It is more important to understand how a user interpreted the text than whether that interpretation was accurate, since I am primarily concerned with determining what aspects of an agreement impact users' positive or negative assessments of that agreement.

1 Due to a limit on comment length, at least one user split reviews with comments longer than the limit into a series of reviews posted consecutively; this was observed in 2 cases). These chains were concatenated into single reviews prior to analysis.

5.4 Results

5.4.1 Participants

EULAscan participation was highest during its launch year: 76, or 39.8%, of the reviews were submitted in 2005. The project has had somewhat diminishing use since that time (see Figure 5.3).

Unfortunately, since all EULAscan reviews are anonymous, it is not possible to determine how many unique users have participated; however, it is possible to get a rough characterization of the population by examining websites that link to EULAscan, as these are likely sources of participants. A Google Web search for "eulascan" returned 124 possible webpages. From these results, 41 unique and relevant pages were found—i.e., non-spam pages linking to or discussing EULAscan. The creation dates of these pages are summarized in Figure 5.4. There is a close correspondence with Figure 5.3, indicative of a possible relationship between webpages *about* EULAscan and posts *to* EULAscan. This is not surprising, as it seems likely that people inclined to write about EULAscan on the Web are also likely to take the time to participate in the project. Given this insight, I counted the number of uniquely identifiable authors of webpages about EULAscan—uniquely identifiable by name; pseudonyms were not considered unique. This number—23—represents a probable lower bound on the number of EULAscan users.

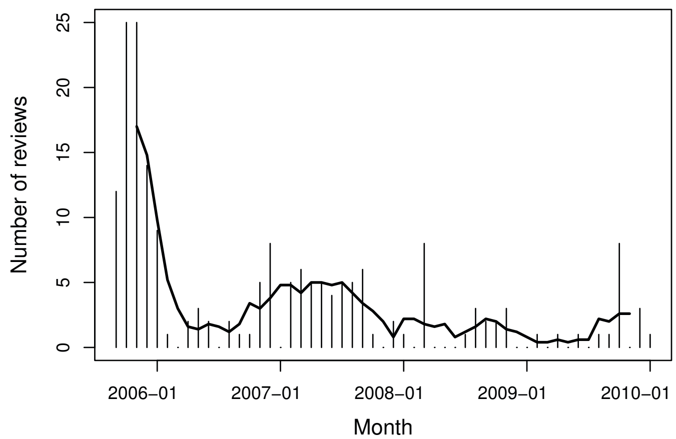


Figure 5.3 *Number of non-spam reviews posted on EULAscan each month since its inception, with 5-month moving average.*

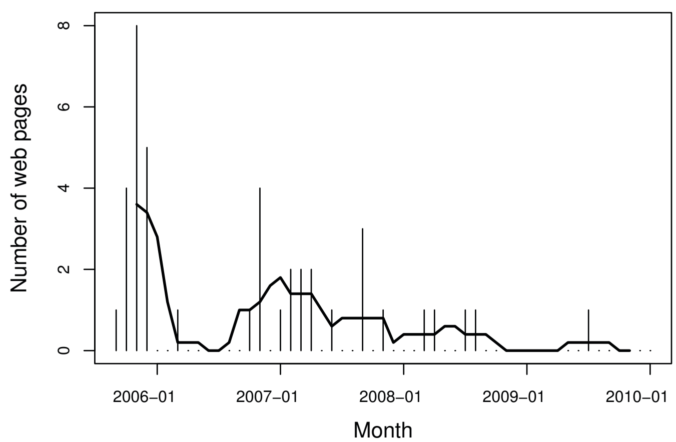


Figure 5.4 *Number of webpages linking to EULAscan made each month since its inception, with 5-month moving average.*

The webpages and posts written about EULAscan appeared on a variety of websites that fall into four broad categories:

Technology websites. This category consists of blogs and websites with broad interests in technology, such as the fairly high-traffic[1] Lifehacker, which describes itself as a blog that "covers tips and tricks for streamlining your life with computers (and sometimes without)" [52]. A post about EULAscan appeared on Lifehacker in October of 2005 [51], just over a month after the earliest reviews in the EULAscan database. In addition to Lifehacker, EULAscan has been mentioned on various smaller technology-oriented blogs [23,42,47,48,62,70,78,81].

Malware prevention and security websites. Some of the websites linking to EULAscan have a particular focus on malware prevention and/or security. References to EULAscan have appeared in a post on ZDNet's Spyware Confidential blog [86], a post on 2-Spyware.com [2], and comments on posts to Bruce Schneier's well-known blog, Schneier on Security [11,53], among other places [67].

Technology law websites. EULAscan has been mentioned on a number of blogs with a focus on intellectual property or legal issues surrounding technology, such as Techdirt [55]—a blog concerned with law and governmental policy as they relate to technology—and Between Lawyers [46], among others [13,29,32,40,61,63,66,80].

Application-specific discussion boards. Mention of EULAscan has also appeared on a number of discussion boards, forums, and mailing lists for particular software applications. This includes the mailing list for GNU Privacy Guard [71,72,74], a piece of encryption software; various Apple discussion boards [1,73,87]; and the Gmail help board [84], among others [5,6,12,22,69,77]. A number of these posts consisted of questions about the license agreements of the software, or responses to such questions that directed the question asker to EULAscan.

Finally, mention of EULAscan was made in the external links section of the entry for "Software license agreement" (or equivalent) in the English (September 2005 [89]), Slovak (March 2006 [92]), and Finnish (October 2006 [91]) versions of Wikipedia. It was subsequently removed from the English version of Wikipedia in February 2007 [90] and remains on the other two versions as of this writing.

Assuming most EULAscan participants arrived through one of these websites, the bulk of users likely consists of people with interests in technology, and particularly interests in security, malware prevention, and the legal issues surrounding license agreements; other users may also

1 As of June 4, 2010, Lifehacker is ranked the 817th most popular website worldwide (361st in the United States) according to Alexa [7]. Lifehacker was also named one of *Time*'s "50 Coolest Websites of 2005", the same year EULAscan was launched [16].

Product category	Products
Multimedia playback and multimedia file–management software	18
Open-source software and licenses	13
Development software: IDEs, SDKs, revision control systems, and databases	12
File-management software: archivers, backup/sync, and desktop search	11
Online services: music stores, flight booking, and ad management	8
Office and productivity software	7
Web browsers and browser add-ons	5
Security software: anti-malware, anti-virus, firewalls	5
Video games	5
Multimedia authoring software: video editors and graphics editors	4
Encryption software	4
Instant messengers	4
Hardware drivers and utilities	3
Financial management software	3
Operating systems	2
Mapping and GPS software	2
Desktop notification software	2
Download managers and file sharing software	2
Uncategorized	*5*
n=	**98**

Table 5.1 *Product categories. Some products are counted in multiple categories. Uncategorized products are: an Internet Service Provider contract, a language-learning tool, a screen capture tool, a speech recognition tool, and the EULA for the software automatically installed by a music CD.*

have come across EULAscan due to posts on application-specific forums or Wikipedia. Given the mandate of EULAscan, it is not surprising that it attracts these types of participants. However, despite this seemingly narrow constituency, the results presented later in this section indicate that a wide range of products and agreements were reviewed, a variety of different concerns were raised in those reviews, and the reviewers often attributed quite different severities to the same agreement clauses. These findings suggest a diversity of user needs and opinions within the EULAscan community.

5.4.2 Categories of products, services, and licenses

The set of products was coded using an open coding scheme. The resulting codes were then grouped into 18 categories of products, with 5 additional categories containing only one product (the categories are summarized in Table 5.1). Of the top 5 categories, 3 particularly reflect the possible technological bias of the sample population suggested earlier: open-source software and licenses, development software, and file-management software. However, a range of other product types are also represented, including multimedia playback software and online services—the other top two categories—as well as Web browsers, video games, and multimedia management software. Both the variety of types of products and the number of products in each category—4 categories contain more than 10 products, and the largest category, multimedia management and playback software, has 18—suggest the that EULAscan community consists of users having a variety of needs.

5.4.3 Effort invested in good and bad reviews

Of the 209 reviews examined, only slightly more reviews rated an agreement bad (100, or 52.4%) than good (91, or 47.6%). However, users writing negative reviews appear to have invested

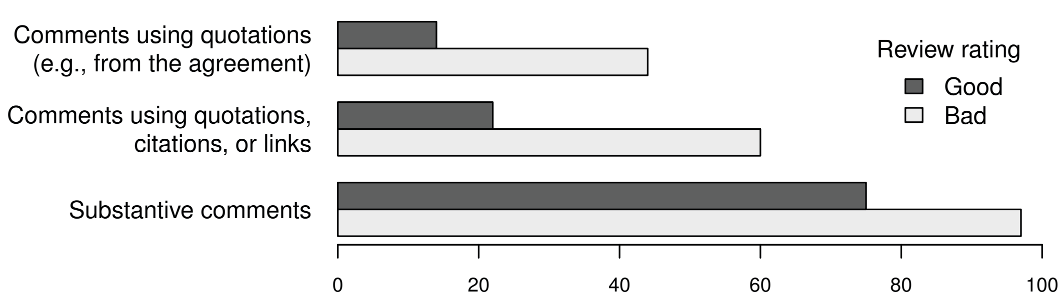


Figure 5.5 *Number of reviews exhibiting each measure of engagement, by review rating.*

more effort in the process than those writing positive reviews, as indicated by a number of measures (summarized in Figures 5.5 and 5.6).

First, in nearly half (44.0%) of bad reviews the author took the time to include direct quotations from the agreement content; direct quotations were found in only 15.4% of good reviews. If this category is expanded to include any use of additional references as justification for a review—including quotations, citations of paragraph numbers from the agreement, and links to related webpages—the number of bad reviews exhibiting such behaviour rises to 60.0%, considerably more than the 24.2% of good reviews.

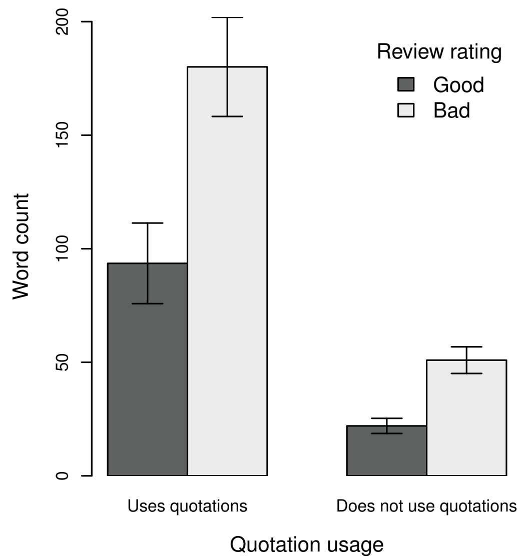


Figure 5.6 *Word count by review rating and quotation usage.*

Second, it is possible to distinguish between comments that elaborate upon the review rating and those that provide no additional substantive information. For example, a comment of "It is alright" attached to a review with a good rating is not a substantive comment, because the comment adds no information not already reflected in the rating. Fully 97.0% of bad reviews had some additional substantive content, compared to 82.4% of good reviews. Put another way, in only 3 cases did a participant submit a bad review without some additional content attached, while in 16 cases a good review was submitted without substantive additional content.

Finally, bad reviews had a higher average word count than good reviews: 107.8 compared to 33.0. Since negative reviews tended to contain quotations, word count was also examined when broken down by quotation usage. Bad reviews had a higher word count within both groups: within reviews that use quotations, bad reviews had a mean word count of 180.1 compared to 93.6 for good reviews; within reviews that did not use quotations, the means were 50.9 and 22.0, respectively (see Figure 5.6).

Most comments in good reviews were short and to the point, with observations that "nothing [was] out of the ordinary", or indications that the reviewer found "nothing fishy" in the agreement. In contrast, comments in bad reviews tended to be longer and to address a wider range of concerns with agreement content. The next two sections outline those concerns: the

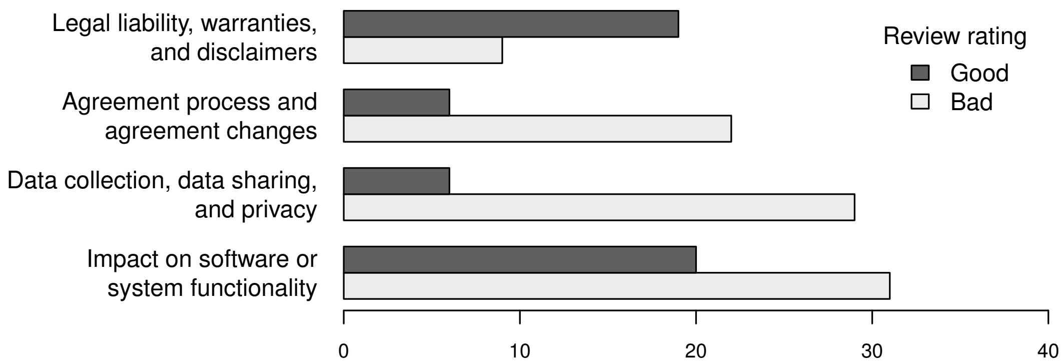


Figure 5.7 *Number of reviews in each of the primary discussion categories, by review rating.*

primary concerns—those raised most often or generating large amounts of discussion—and the secondary concerns—those raised less often, or generating little discussion.

5.4.4 Primary concerns

Recall that reviews were grouped into high-level categories based on an open coding of the EULAscan corpus. The majority of users' specific concerns with agreement content fell into four categories, which are the largest four high-level categories of user concerns identified during the coding process (Figure 5.7 shows the number of reviews in these four categories, broken down by rating):

1. Legal liabilities, warranties, and disclaimers.

2. Agreement process and notification of updated agreement terms.

3. Data collection, data sharing, and privacy.

4. Software or system functionality.

5.4.4.1 *Legal liabilities, warranties, and disclaimers*

A number of the agreements reviewed contained limitations of legal liability, warranty disclaimers, or statements that the software is provided *as-is* (this is equivalent to a disclaimer of warranty [54]). Such clauses were considered standard by most reviewers, garnering comments like, "Fairly standard, mostly referring to limitations of liability in use with the product", or, “This is about as harmless as it gets. Nothing beyond the standard disclaimers in here.” Phrases like “standard”, “short and sweet”, “fairly benign”, “seems fair”, “nothing fishy”, or “not bad” were often used by reviewers giving these agreements a good rating.

Some reviewers saw warranty disclaimers as a trade-off, particularly in the context of software that is available at no cost (freeware or open-source software):

Basically, all it states is that, as a free product, there is no warranty. Seems fair.

At least one review went further, expressing some understanding of the risks associated with using software having such clauses, still giving the agreement a good rating:

Short and sweet, nothing bad in here.

In fact, I love this part:

“Good data processing procedure dictates that any program be thoroughly tested with non-critical data before relying on it. The user must assume the entire risk of using the program.”

Seriously.

This reviewer agreed with the implication contained in the agreement that the responsibility for testing the software in mission-critical settings rests with the user of the software, not the software provider.

As reflected in Figure 5.7, a minority of reviewers who raised the issue of warranty disclaimers gave the agreement a bad rating; for example:

While Microsoft's EULA may not be entirely evil on any one point, they are indemnifying themselves against any and all lost profits due to using their software. For a game, OK, I can see that being a non-issue: don't install video games on your work PC.

But for an OS company that charges businesses THE MOST for its product, targets businesses THE MOST to sell its product, and CHARGES you to receive the most basic product support, AND does not provide anyone with a full instruction manual of their *software (the source code), they're pretty bad. No product liability whatsoever - that's not typical of consumer products. Microsoft gets away with it only because they have more money than our own government to fight any such lawsuits that complain about this lack of consumer protection.*

This reviewer contrasts the use of limited liability clauses in computer software with their expectations of consumer protection in other consumer products, finding the use of such clauses in software agreements to be offensive, particularly because the software in question—the Windows operating system—costs money and may be used in a business context. In this sense, the underlying stance of this reviewer may not differ significantly from those reviewers who gave freeware with a warranty disclaimer a good rating: no warranty is appropriate for some software, but not others.

Other negative comments objected that if a flaw in the software caused a system to be "hacked", or for a virus to be installed on the computer, the software provider would not be responsible: "If there's a flaw that compromises your system's security, and you get hacked, it's not their fault and the only remedy is for you to go away". In contrast to some good reviews, these reviewers felt that responsibility for problems with the software rest primarily with the software provider, and not the user.

Clearly, there is some disagreement amongst the reviewers as to the severity of warranty disclaimers and limited liability clauses in software agreements. Some reviewers regard such clauses as a trade-off against cost: appropriate for freeware, but not for paid software or software used in a business context. Others disagree on whether the responsibility for judging the suitability of a particular piece of software for some task lies with the user or with the software provider. In any case, the majority of reviewers—including several giving bad ratings to such clauses—appear to regard these clauses as a standard fixture. Most reviews addressing these clauses (19 compared to 9) rated the agreements containing them as good rather than bad; this is probably in part a reflection of users' expectations that warranty disclaimers and limitations of liability are the rule, rather than the exception, in software agreements.

5.4.4.2 *Agreement process and notification of updated agreement terms*

Several comments raised concerns about the agreement process itself, and in particular, the policies and procedures surrounding updated versions of the agreement. In this category, the bad reviews far outweighed the good (22 to 6). Many of the agreements cited had terms stating that new versions of the agreement can be released at any time, and outlined the mechanism by which users become bound to the new agreement—for example, through notification by email. Users often took issue with such mechanisms, and what constitutes acceptance of a new agreement, as in this negative review of Winamp:

> *In section 9, it states that if you provide Nullsoft with your email address, this gives them the power to send "ANY PRIVACY OR OTHER NOTICES, AGREEMENTS, DISCLOSURES, REPORTS, DOCUMENTS, COMMUNICATIONS, OR OTHER RECORDS". However, "The delivery of any Notice from Nullsoft is effective when sent by Nullsoft, regardless of whether you read the Notice when you receive it or whether you actually receive the delivery."*
>
> *This means they can send you a change in their privacy policy or other terms, and have it take effect even though you never receive their email.*

This comment notes that the email mechanism places a burden on the users, rather than the service provider, to ensure that users are aware of updated terms, drawing on the fact that email without acknowledgement is an unreliable communication mechanism. This comment also hints at a fundamental mistrust that companies will act in good faith when updating agreement terms. This mistrust was echoed in the language used by several reviewers. For example, one reviewer referred to a similar policy from the iTunes Music Store as "rotten to the core". Others noted that the lack of an opt-out for policies like data collection or software updates was untrustworthy.

Another mechanism employed by several agreements is to refer to an additional external policy, often a privacy policy, which can be updated by the service provider at any time. For example, in a negative review of Firefox's EULA, one reviewer wrote:

> *A shifting privacy policy? It's my responsibility to periodically check for changes? This stinks.*

Again, this reviewer takes explicit issue with the responsibility of checking for changes being shifted to the user, a concern echoed in many other comments. Indeed, the bulk of comments addressing the issue of external policies were negative, often taking issue with this shift in responsibility. However, at least one reviewer gave a positive review to such an agreement (for McAfee VirusScan):

Although McAfee had previously disclaimed their privacy policy, it seems they have turned around and licensed the TRUSTe privacy program.

Make sure to reference http://us.mcafee.com/root/aboutUs.asp?id=privacy *for their privacy policy.*

This reviewer rated the agreement positively, citing McAfee's use of TRUSTe[1]. TRUSTe is a service that reviews and vets online privacy policies. For this reviewer, the addition of a third party authority has given them greater trust in the use of an external policy, though they still recommend others read that policy.

On the whole, reviews in this category expressed displeasure at how agreements were made and how users are notified of updated terms. Some users wished for opt-out procedures up front, but more often users expressed concern that agreements could change without adequate notice, or that an external—possibly shifting—policy was included by reference; this latter concern may be mitigated if the external policy has been vetted by a third party. In general, users were displeased that it was the their responsibility, and not the software provider's, to ensure that notifications of updated agreement terms are received.

5.4.4.3 *Data collection, data sharing, and privacy*

A number of the agreements reviewed contained clauses involving the collection, use, and possible sharing of a variety of user data. The types of information collected ranged from system information—such as IP address, computer name, and lists of the software installed on users' computers—to "license-specific" information, to real-time collection of usage data—such as internet connection use. The majority of agreements containing data collection clauses garnered universal condemnation: 29 reviews gave such agreements a bad rating, while only 6 gave them a good rating. However, some aspects of data collection—for example, keeping track of installed software—did provoke mixed reactions. Consider the following exchange, discussing Mac OS X:

[Bad rating] *By agreeing to the EULA, you allow the company to collect and use data about your computer, including what software you've installed.* […]

1 http://www.truste.com/

> [Good rating] *So what? All the 'Consent to Use of Data' says is that data that will get collected anyway (such as the list of installed software the MUST get collected for the software update feature to work) may be used for other purposes in the aggregate. They are not keeping track of what you have installed, although they may be tracking that x% of users have application foo installed.* […]
>
> [Bad rating] *Still it is a privacy violation*
>
> [Good rating] *It's only a privacy violation IF the information is personally identifiable or traceable to the originator. Using your debit card to buy groceries allows the supermarket to personally identify you and your bank account. I don't see this much differently.*

In this exchange, one reviewer expresses concern that the software collected data about the software installed on users' computer, and therefore rates its agreement negatively. A second reviewer responds to that concern with a positive rating and a rationale: the list of software is needed to support functionality—software updates. The next comment, possibly from the first reviewer, defends the original position by insisting that the collection of these data constitutes a privacy violation, seemingly regardless of their use. This is countered by a comment stating that a privacy violation exists only if certain types of data are collected, specifically personally identifiable information).

This exchange highlights a crucial difference between two categories of comments: those that express unspecified concerns about the collection and sharing of data, and those comments indicating fear of specific, practical consequences of data collection. An example of the latter:

> *Whatever someone might claim was the intent of that consent clause, it does sound like the excellent new iMac, thin enough to hang on the wall, is also being allowed to be a spy panel like in Orwell's novel "1984". […] With OS 10.4 Apple computers are going to be beautiful but poisonous Apples, like in Snow White and the Seven Dwarfs.*

This comment, while hyperbolic, expresses the writer's genuine fear of being spied on as a result of using Mac OS X. A less extreme example of a practical concern was expressed about the FriendGreetings EULA: "Agreeing to this EULA means you allow this software to go through your email contacts list and mail your contacts with an ad." Rather than expressing a broad rejection of data collection, this user balks specifically at the use of personal data by FriendGreetings to send emails to members of their contact list. This distinction—between users who a broad reject all data collection on principle and those with specific concerns—is consistent with Cranor *et al.*'s distinction between privacy fundamentalists and privacy pragmatists [20].

Other comments expressed concerns about the ambiguous nature of data collection policies, calling some agreements "too broad". Consider this comment on XMLSpy:

> *Quoting the privacy policy as of this review:*
>
> *"In order to prevent unlicensed or illegal use of Altova software, the software contains a software activation system that allows for an exchange of license related data such as operating system, IP address, date/time, software version, and computer name, along with other information between your computer and an Altova license server at the time of installation, registration or update."*
>
> *The line "along with other information" is the most distressing.*

This comment expresses concern that the range of data collected is left deliberately ambiguous, leaving the software open to collect any data not specified in the agreement. While this does not represent a specific, concrete threat to privacy, it leaves the potential for any number of privacy issues, and so the reviewer rejects the agreement.

Some comments took issue with policies for data sharing and use, rather than data collection. For example:

> *Once Lavasoft has your IP address, it "will be transmitted to and stored by Google on servers in the United States." Google will compile reports on your website activity for "website operators". Who exactly these website operators are is not mentioned (Lavasoft? Google? Driveby marketers? Greasy stalkers?). Conveniently, this information will be linked to any other personally-identifiable information "you choose to furnish" on the Lavasoft website.*

This review of Ad-aware notes that data collected by the software provider (Lavasoft), possibly including personally-identifiable information, may be shared with "website operators". The reviewer is particularly concerned that the term "website operators" is ambiguous, and may result in personal information being leaked to unwanted third parties. As in previous examples, the clause in question has the potential for specific consequences—e.g., if data are released to marketers, the user may be targeted by advertisements—but in this context the reviewer is concerned with consequences that result from data sharing policies rather than data collection policies.

Finally, a small number of agreements included terms allowing the software provider to audit the users' machines in order to determine if users are in compliance with the agreement:

> *However, agreeing to this EULA allows Red Gate to physically come to your office and do an audit of your computers!* [...] *So, if you like companies you buy things from rummaging through your computer, say yes to this EULA.*

When discussed, such auditing was condemned by all but one reviewer, who reasoned that it applied primarily to volume licensees, which they were not.

Reviewers took issue with data collection clauses for a range of reasons. Some users rejected data collection outright, without making exception for how the data are used. More often, reviewers had specific, practical consequences in mind—sometimes trading data collection for functionality, as in the case of software updates—or objected to the broad nature of many data collection policies—in other words, objected when they were unable to determine what data would be collected, or the consequences of allowing that collection. Generally speaking, broad or ambiguous data collection and sharing was universally condemned, with some indication that users will trade off data collection against other considerations, such as functionality, so long as the policies are defined clearly enough that users can determine the impact accepting such an agreement might have on their privacy.

5.4.4.4 *Software or system functionality*

The largest discussion category was that of software or system functionality, accounting for 31 bad reviews and 20 good reviews. This category includes discussions of the relative functionality or usability of different software, as well as impacts that clauses in the agreement have on the functionality of the software or the user's system as a whole.

It should not be surprising that—in the context of deciding whether or not to accept a software agreement—consideration of software functionality is an important factor. Previous research has demonstrated that users, installing software for a particular task, often skip over agreements because the agreements are not perceived to be relevant to their goal of accomplishing that task [27,28]. In other words, when users consider whether or not to accept—or even read—software agreements, the value of software functionality—the utility derived from actually using the software—is often more important than other concerns, such as legal ramifications or effects on privacy [31,79]. Some of these trade-offs were discussed above in the context of data collection policies. One reviewer in the discussion of the data collection policy for Max OS X made this trade-off explicit:

> *I think people are paranoid drama queens. WTF do you care if Apple collects what hardware and software you use, as long as it makes for a better/seemless computing experiance, i care not.*

In contrast to many of the other reviewers, this user considers the greater usability of Mac OS X an acceptable trade-off for allowing Apple to collect data on hardware and software use. It is indicative of the importance of software functionality to users that, on a website focused primarily on legal and privacy issues, the utility derived from a piece of software is still a central part of the discussion and can trump other concerns. While this kind of decision—imparting a *positive* review on an agreement because of the software's functionality despite the risk of some other negative impact, e.g., a privacy risk from data collection—was reflected in several comments, another pattern was also common: imparting a *negative* review on an agreement due to some perceived negative impact on functionality caused by clauses in the agreement.

A variety of such negative impacts on functionality were mentioned, from limitations imposed on the use of the software (e.g., restrictions on professional use) to negative impacts on the functionality of other software (e.g., by the installation of adware) to negative impacts on the functioning of the entire system (e.g., by the use of computer resources like processor time or bandwidth). For example, one reviewer took issue with the Google Services Terms of Service, which they found when looking for the Google AdWords agreement:

> *"The Google Services are made available for your personal, non-commercial use only. You may not use the Google Services to sell a product or service, or to increase traffic to your Web site for commercial reasons, such as advertising sales."*
>
> *That doesn't really coincide with what AdWords are for, and also goes against many things that many folks use Google for.*

This user is concerned that this clause limits use of AdWords to non-commercial use, which is unusual since AdWords is an advertising service often used by commercial entities. A comment posted in reply to this review pointed out that the Google Terms of Service do not apply to AdWords; nevertheless, this review is indicative of users' concerns for how software agreements might limit their use of software in professional contexts. As another example of this, one review, though positive, warned that the Quicktime agreement prevents certain MPEG codecs from being used in some contexts, and may also place limitations on professional use.

Reviewers also objected to the installation of unwanted software on their computer that would have a negative impact on the functionality of existing programs. Commonly cited were examples of adware and examples of Digital Rights Management (DRM) software. Consider the following comment:

> *(1) We all hate Google add-ons. If you quickly click through the install buttons, you'll get Google Chrome as a "bonus". The middle button will let you "install Ad-Aware only".*
>
> *(2) As per the eula, you hand over your "consent" that the Ad-Aware software is configured to automatically download "news and offers". The agreement says you need to change the settings in the program to "Manual Download" to undo this.*

Ironically, Ad-Aware (an adware removal program) is cited here as an example of software that installs adware. DRM also raised concerns about functionality, as in this review of Windows XP's agreement:

> *You also agree to let microsoft automatically install updates to make their DRM more restrictive...*
>
> *"download onto your computer such security updates that a secure content owner has requested that MS, Microsoft Corporation, or their subsidiaries distribute."*

Several comments on the Windows XP agreement raised concerns with Microsoft automatically installing updates, some of which, like the above, were concerned in particular with the installation of DRM software. Interestingly, the resulting discussion also produced further examples of positive comments citing the functionality provided as a reasonable trade-off; as one reviewer put it: "Nothing wrong with that. They are providing a service."

Some agreements grant the software (usually limited) use of computer resources not for the user's direct benefit. One reviewer took issue with a clause in Skype granting the software use of processor and bandwidth:

> *Article4: You give permission to Skype "to utilize the processor and bandwidth of Your computer for the limited purpose of facilitating the communication between Skype Software users" (note it doesn't say that you will be one of the users).*

This comment also provides another example of the mistrust of software providers' intentions mentioned earlier: the language of this clause is not entirely clear whether the specified computer resources will be used for the user in question or for others, but the reviewer assumes the worst. In another, more clear-cut case—for Yapta, which has an agreement explicitly granting use of resources to perform queries for other users—a reviewer simply quotes the relevant portion of the agreement, without additional comment.

The concerns discussed in this section reflect a range of ways in which considerations of software or system functionality affect users' opinions of an agreement. These concerns range from the functionality of the software itself, to limitations on use imposed by the software agreement, to ways in which the software affects the functionality of the user's system. At times, the same clause evoked opposite reactions from two reviewers, both having functionality as an underlying concern. This is perhaps best demonstrated by the reactions elicited by the Windows XP agreement: some reviewers found the negative impact on the functionality of their system caused by auto-updates installing DRM to be unacceptable, while others found this to be a reasonable trade-off for the functionality and ease of use of the auto-update service. This range of opinion is reflected in the large number of both good and bad reviews in this category.

5.4.5 Secondary concerns

Some other less common—or less contentious, and therefore less discussed—concerns were also raised. These include:

Quality of presentation—length, clarity of language, and conciseness. A common complaint across discussion categories was that the agreement presentation was poor, or that language was ambiguous. Agreements were described as "very broad", "ambiguous", "sloppy", "poorly written", or containing "ugly legalese". While some of this response is likely due to the complex language found in these documents [41], these complaints were also directed at language that readers felt was deliberately broad, such as the overly-inclusive data collection policies discussed above.

Choice of forum. Some of the agreements reviewed had clauses stipulating that the agreement was governed by the laws of a particular jurisdiction; e.g., Opera's EULA states, "This Agreement shall be governed by Norwegian law". Choice of forum was raised in some comments as a passing concern. In only one comment did a reviewer raise this issue and give the agreement a bad rating, and that comment devoted significantly more space to objections the reviewer had to other content in the agreement.

Damages to provider. Some reviews made mention of clauses that allow the software provider to seek damages against the user under certain circumstances, for example, if the user violates the agreement. Two agreements (for Pro Softnet IBackup and for LiveVault Backup) stipulated that damages could be sought against the user for negative reviews posted about the products themselves. Both of these reviews gave the agreement a bad rating; presumably the reviewers did not accept the agreement before writing their review.

Restrictions on resale. Finally, several reviews complained about clauses restricting resale or transfer of the software to other users. Writing of one such agreement, a reviewer noted, "You can't resell this software. They told me it was a service, not a software, but I distinctly bought a software package." Those concerned about such clauses generally expressed similar sentiments that such clauses were dishonest.

5.4.6 Open-source concerns

Open-source software and licenses accounted for 13.3% of the products and 15.2% of the reviews on EULAscan. Response to open-source licenses was mixed, with 17 good reviews and 12 bad reviews. Compared to other products reviewed, the discussion topics surrounding open-source licenses were notably different. These topics include:

Copyleft. A contentious topic on the open-source licenses was that of *copyleft*, or licenses like the GNU General Public License that allow the source code of licensed software to be modified and redistributed, so long as it retains the same license; some reviewers referred to these clauses as "viral". Readers disagreed whether or not such an agreement is truly "free", making comparisons to other open-source licenses without copyleft clauses.

License compatibility and difficulty integrating code into other projects. Another common topic of discussion within open-source licenses was the ease with which a particular license allowed use of the software's source code in projects with other licenses. This is a common problem within the open-source software community, as certain types of agreements, such as copyleft agreements like the GPL, are difficult to integrate into projects without similar clauses in their license agreements [75].

Policing licenses. Finally, some reviewers policed software producers and software licenses for what they saw as violations of open-source licenses and open-source principles. As an example of the former, a comment on LockDisk noted, "LockDisk is based on FreeOTFE, and violates the FreeOTFE licence." As an example of the latter, a bad review of TrueCrypt stated, "The TrueCrypt Collective License does not meet the Open Source Definition", in reference to the Open Source Initiative's list of approved open-source licenses[1]. In comparison, reviewers of proprietary software agreements were not as concerned with how well an agreement held up to an external organization's standard. One exception was the review (discussed earlier) that cited TRUSTe as a trustworthy third-party authority for vetting privacy policies, but TRUSTe was cited in only that instance.

5.5 Recommendations for design

As mentioned in the introduction to this chapter, it is necessary to highlight information that users will find relevant and interesting, lest the limited attention they have for the agreement process is exhausted. This section draws from the results of this chapter in order to outline recommendations for presenting agreement content in such a way as to increase the likelihood that it will be of interest to readers. Note that these recommendations focus on proprietary software and services, since open-source concerns were markedly different, and the number of comments addressing those concerns was too small to confidently draw design implications from.

These recommendations are aimed at any designer wishing to capture readers' attention during consent processes, or who aims to improve the communicative value of software agreements by aligning agreement presentation more closely with users' concerns. Such a designer might be a designer of third party tools for presenting software agreements, such as EULAlyzer[2]—a tool that allows users to browse agreement content organized by key terms and phrases—a regulator, a promoter of industry standards, or a software producer who wishes to build trust with users by engaging them more directly during the consent process.

In developing recommendations for designing software agreements, or third-party tools for presenting agreement content, it is necessary to ask how to choose what to highlight. This section first discusses why a blanket prescription of the type of terms to highlight is inadequate in the long term, and then presents an alternative approach that may be more successful in capturing and retaining reader interest.

1 http://www.opensource.org/docs/osd

2 http://www.javacoolsoftware.com/eulalyzer.html

5.5.1 Static recommendations are not robust

It is tempting to suggest highlighting terms based on the content the majority of people found objectionable. From the study results, this strategy would lead to a simple set of recommendations: highlight impacts on functionality, agreement update practices, data collection, data sharing, and privacy clauses. Other content is either not considered important by a large number of users—e.g., choice of forum clauses—or received a mixed response but is anyway considered by users to be standard—e.g., warranty disclaimers—and so highlighting it adds no value.

Unfortunately, this approach represents a moving target. Warranty disclaimers, for example, may now be considered standard: the majority of reviewers who discussed warranty disclaimers or limitations of liability gave the agreements containing those clauses a good rating. Even among reviewers who gave such agreements a bad rating, there is an expectation that most agreements contain such clauses, and this expectation is not unfounded: in a survey of 647 software license agreements, Marotta-Wurgler found that 90% disclaimed implied warranties and 89% have limitations on liability for consequential, incidental, or special damages [54].

In some sense, users' awareness of the ubiquity of warranty disclaimers represents a success of regulation, as warranty disclaimers are often required to be "conspicuous" by law [65], where "conspicuous" means setting the clauses in all caps or bold text [64]. This awareness may also be indicative of a shift in user attitudes that has arisen from regulation; Hillman, for example, found a greater concern for warranty disclaimers in previous work [35]. Users now expect that all software has a warranty disclaimer, and therefore consider it non-negotiable. Regulations must be made carefully, as they have the potential to shift user opinions of what agreement content is appropriate. As users become habituated to highly visible clauses, general attitudes towards those clauses may shift over time, making it difficult to give a robust, static set of recommendations. This shift echoes Lehman's observations on *E-type programs*—programs embedded in the real world: that such a program becomes a part of the world it is embedded in, and thereby changes its own requirements [50].

Besides shifting over time, concern for specific clauses also differs across users: while a minority, some users are concerned about choice of forum, damages, or resale restrictions; some users consider warranty disclaimers to be egregious, others standard fare. These findings echo previous research suggesting that privacy concerns are not uniform across users [4,31]. A blanket prescription of terms to highlight, therefore, may not be the best approach.

5.5.2 Recommendations

Having discussed the difficulties in developing a canonical set of terms that users consider important, this section will now provide a more nuanced set of recommendations: to highlight impacts on functionality, to highlight unusual terms, and to select techniques that allow rephrasing for clauses of particular complexity.

Highlight impacts on functionality—both positive and negative. While most terms represent a moving target, functionality has consistently been found to be an overriding concern for users when selecting software [31,79]. Indeed, functionality was the most frequently used rationale for both positive and negative ratings of agreements by EULAscan reviewers. While other concerns may shift over time, it is unlikely that functionality will cease to be a primary concern for users; therefore, clauses that impact functionality should be highlighted. This will make positive and negative aspects of the agreement more readily accessible to a reader who might wish to make trade-offs against functionality when accepting an agreement. In addition, as software functionality is clearly highly relevant to most users, highlighting it may act as a hook into agreement content: readers, upon discovering that an agreement does have information relevant to them, may be more inclined to read further into that agreement.

Highlight unusual terms. Outside of functionality, selecting important terms to highlight could be done by considering how much a clause deviates from other agreements—either standard agreements, or agreements the user has seen before. The latter option would require the use of end-user tools, likely provided by a third party. The aforementioned EULAlyzer is an example of such a tool, which already retains a database of license agreements the user has seen before for users to review. Given such a database, there is great potential for assisting users in finding terms that are truly personally relevant, rather than terms that have been selected in advance. It would be possible to indicate how much a new agreement deviates from previous agreements the user has accepted or rejected, and in what way. Users might also indicate what particular clauses they find egregious within agreements they have seen, and these data could be used to help predict the relative importance of clauses in subsequent agreements. In general, end-user tools offer the ability to identify terms that are unusual or important according to the user's perspective, rather than according to a predefined, static set of recommendations.

Where complex language is used, apply techniques that allow that language to be rephrased. On top of improved design, there is a need for clearer, less ambiguous, and more accessible language in software agreements. Highlighting techniques can employ various approaches to clarify underlying content. In particular, factoids, vignettes, and summaries offer the opportunity to clarify difficult to understand text from the agreement; by contrast, warnings and bold text highlight important information without clarifying it. Therefore, when choosing between possible techniques for highlighting a clause in an agreement, consider the complexity of the clause and apply techniques like factoids and vignettes to the most complex clauses—data collection policies in particular were often the subject of complaints about language clarity. Techniques like warnings and bold text can then be used for easier to understand text that need not be rephrased. While work is being done to improve the language used in legal contracts[1], this may take time; in the short term, alternative presentations of agreement content can selectively improve the most difficult to understand clauses.

5.6 Summary

In this chapter, I analyzed the concerns raised by reviewers of license agreements on EULAscan, an online community of anonymous reviewers of license agreements. I identified three categories of concerns: primary concerns, such as warranties/disclaimers, notification of updated agreement terms, privacy, and functionality; secondary concerns, such as quality of presentation, choice of forum, damages to the provider, and restrictions on resale; and open-source concerns, such as copyleft, license compatibility, and policing of licenses. Noting the barriers to developing effective static designs—particularly, the varying of concerns between users and over time—this chapter also gave more nuanced recommendations for designers who wish to present license agreements in a manner that reflects users' concerns: to prioritize highlighting impacts on functionality, to develop ways of identifying and highlighting terms of interest to each user—e.g., through end user tools—and to use techniques that allow language to be rephrased on clauses that are particularly complex.

1 See the Plain Language Association InterNational (http://www.plainlanguagenetwork.org/) and Clarity, an "international association promoting plain legal language" (http://www.clarity-international.net/).

Chapter 6

Conclusion

6.1 Contributions

This thesis makes contributions in three areas of improving the communication of software agreement content to end-users:

Narrative pictograms: near-wordless techniques for communicating to linguistically diverse audiences. Narrative pictograms are one of the first attempts to comprehensively investigate the problem of localizing software agreements, a significant problem for commercial software developers, open-source software developers, and those conducting research on the Internet. The lessons learned and the composition rules derived from this work provide a foundation for future exploration of this problem space.

Textured agreements: visually enhanced agreements that capture reader attention and direct it towards the full agreement. Textured agreements achieve improved reading times and comprehension by operating on the primary object of interest—the agreement—rather than creating secondary objects, such as summaries. This approach has particular advantages as it does not partition the consent process into multiple stages.

Recommendations for the design of future tools for presenting software agreements. I developed a set of recommendations for designers who wish to present license agreements in a manner that reflects users' concerns: to prioritize highlighting impacts on functionality, to develop ways of identifying and highlighting unusual terms, and to apply techniques that allow rephrasing to complex clauses. I also caution against adopting static sets of recommendations that will quickly become obsolete as users' attitude shift or as technology changes.

6.2 Future work

The success of textured agreements suggests that we can convince users to read software agreements without changing the underlying text. However, adoption of these techniques by proprietary software companies remains a barrier—currently, there is little incentive for software providers to change agreement formats, and little push-back from consumers. At the same time, the results of Chapter 5 suggest that static re-designs of software agreements may not adequately account for variation in users' concerns.

Both of these factors suggest that further developing end-user tools for identifying and highlighting content that is potentially interesting to the user is a more promising way forward. Such a tool has the potential to make agreements a more competitive factor in software choice by allowing consumers to be better informed about the agreements for the software they might wish to install.

An end-user tool could attempt to determine salient content using techniques from natural language processing, such as latent semantic analysis. Clauses of interest might be identified based on deviation from a pre-existing corpus of standard agreements, or deviation from agreements the user has already read. In order to assist such a tool with identifying clauses of interest, users could also classify clauses they find particularly interesting—or egregious—in agreements they have already read. Once salient information has been identified, such a tool could build upon existing work by applying techniques from summaries, narrative pictograms, or textured agreements, as appropriate, to improve the communication of that content to the end-user.

Finally, in light of the effect that standardization has had on user awareness—and acceptance—of warranty disclaimers, there is the potential for static designs to have a profound impact on how users perceive software agreement content through habituation to that content. Future work should strongly consider studying dynamic approaches to presenting software agreements as possible solutions to the problems of habituation and desensitization.

Appendix A

Software agreement with narrative pictograms

This appendix contains a copy of the data collection policy with embedded narrative pictograms used for the diagrams-with-text condition of the summative evaluation in Chapter 3.

ingimp Logging Consent Form

Introduction

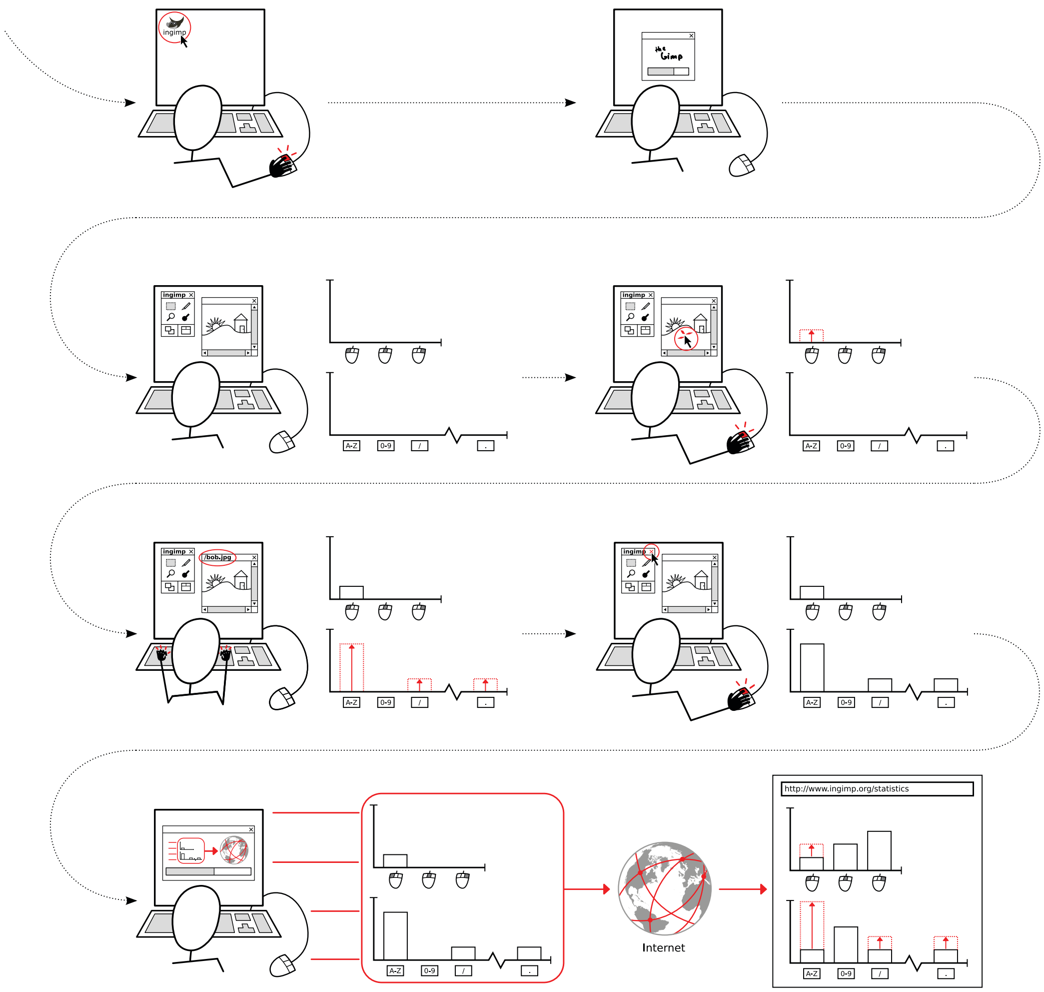


Diagram A

This software, ingimp (Instrumented GIMP) is part of a research effort actively exploring new ways to contribute to the usability of open source software. This software is modified to automatically record how you use this application. This information will be automatically uploaded to a publicly accessible webserver to enable usability analyses. The purpose of this document is to gain your consent to do this data collection.

ingimp automatically logs usage information such as the names of commands you use, general descriptions of the types of documents you edit, information about your platform (e.g., whether it is Windows, Linux, etc.), interaction events (when keys are pressed, when mouse buttons are pressed, but NOT the actual key or location of the mouse), and so on. These data are sent to a server and made publicly available for all to view and analyze. Our hope is that the data collected will help inform and focus future development and design efforts.

Principal Investigator

This study is being conducted by Professor Michael Terry in the School of Computer Science at the University of Waterloo. Questions should be directed to ingimp@cs.uwaterloo.ca.

Study Description

You must be 18 years or older to participate, or you must obtain the consent of your parent or legal guardian. Participation is completely voluntary and can be stopped at any time by removing this software or discontinuing its use.

If you choose to participate, you may be asked to fill out a web-based questionnaire. The questionnaire is optional but will help us understand who is using this software.

Once consent to use this software is given, ingimp will log your usage of the software. This version of the software logs the following information:

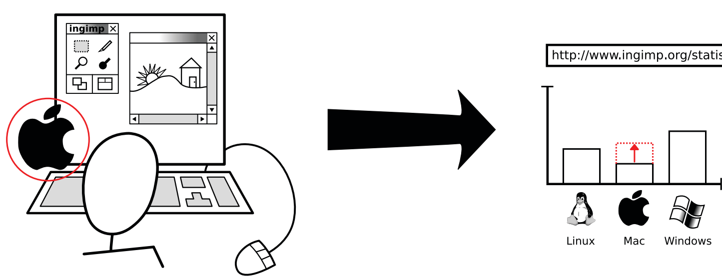


Diagram B

- ▶ Your platform and its characteristics (e.g., Windows, Linux, CPU if available, the version of your operating system, if available...)

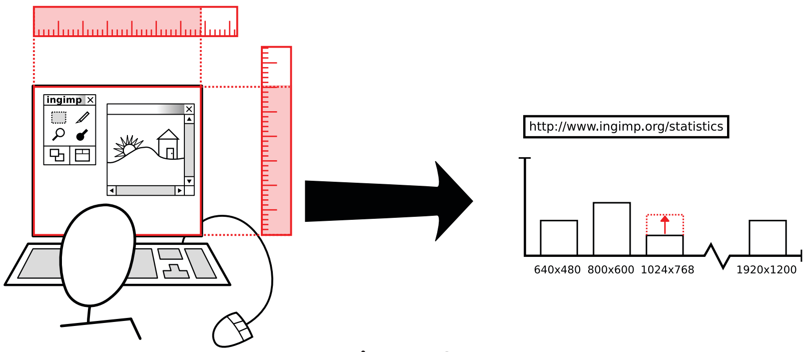


Diagram C

- The resolution of your computer screen

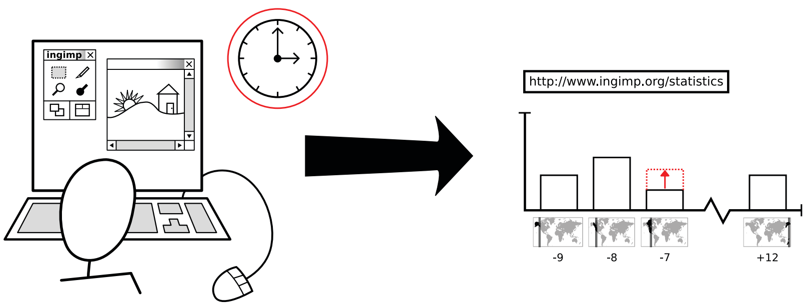


Diagram D

- Your timezone

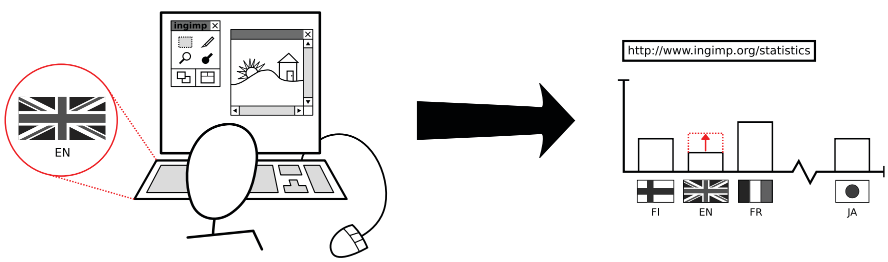


Diagram E

- Your locale

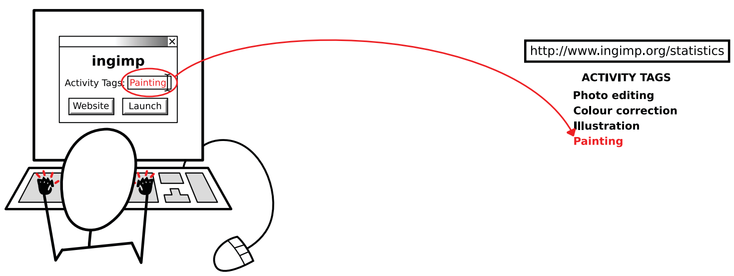


Diagram F

- ▶ Any activity tags you enter at start-up to describe how you will use the software (more information below)

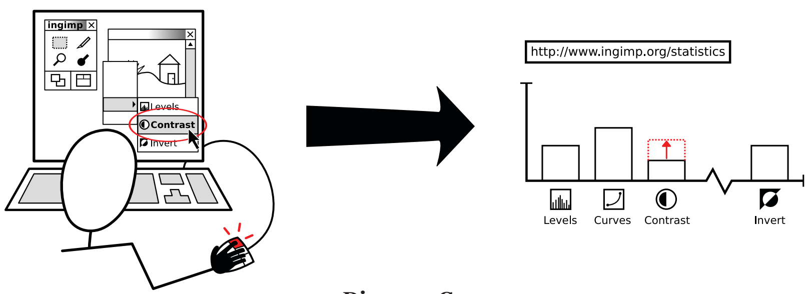


Diagram G

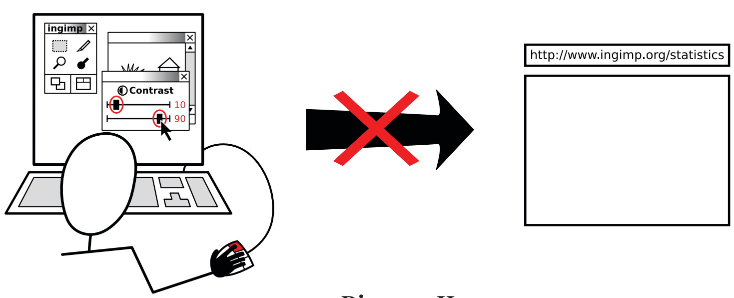


Diagram H

- ▶ The names of commands that you use, but not the parameters used for those commands

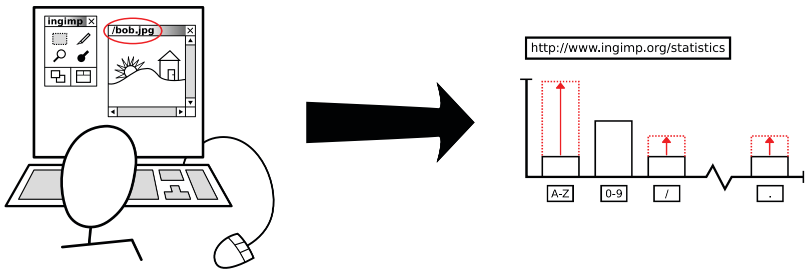


Diagram I

- The number of letters, digits, spaces, punctuation marks, and forward/backward slashes in your file names, but NOT the file names themselves

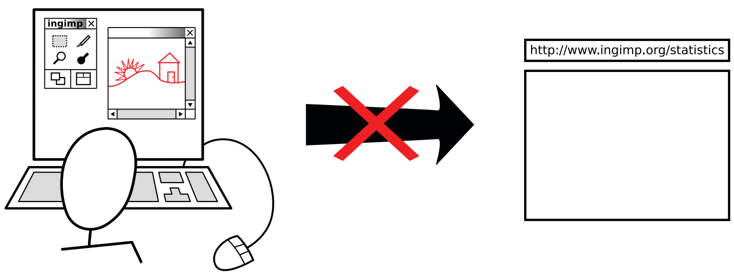


Diagram J

- Characteristics of your document after each command has been applied, including:
  - the number of layers in the image
  - layer visibility, opacity, and combination mode
  - the number of letters, digits, spaces, and punctuation marks in your layer names, but NOT the layer names themselves
  - whether a layer is linked, floating, or removed

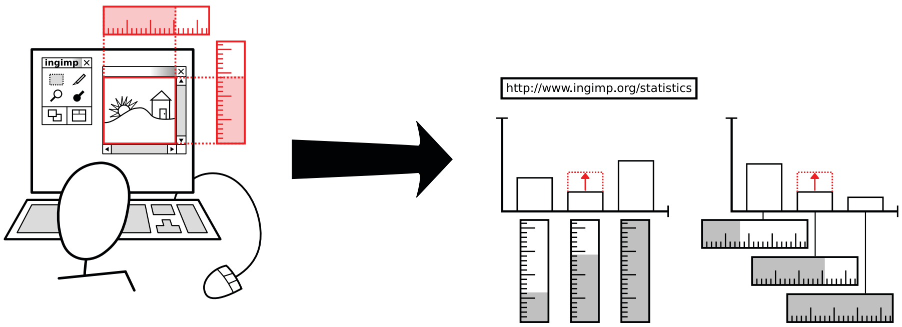


Diagram K

- ◇ the sizes of the respective layers
- ◇ the resolution of your image

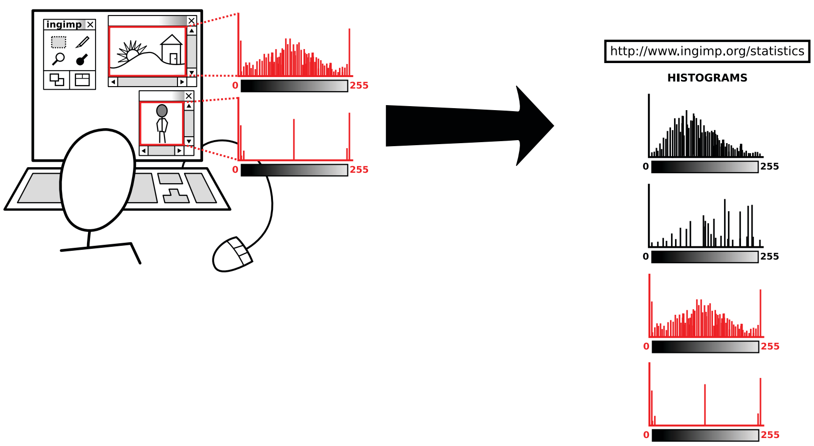


Diagram L

- ◇ pixel counts that indicate the relative frequency of dark and light pixels in the image (an image histogram), along with statistics for these pixel counts (count, max, mean, median, standard deviation)

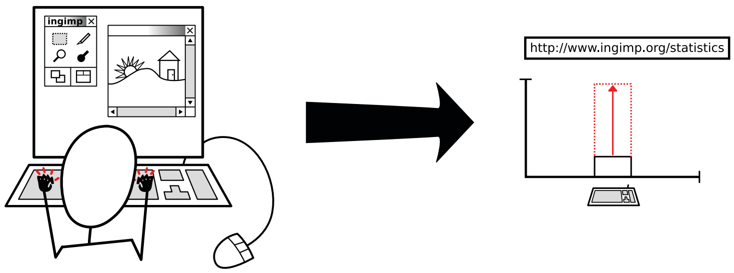


Diagram M

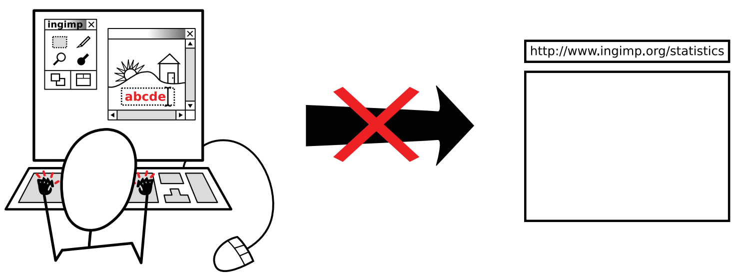


Diagram N

- ▶ Keyboard and mouse activity, but NOT the actual keys pressed or the location of the mouse button. The state of the modifier keys (Shift, Control, Alt, etc.) ARE recorded, but the actual key pressed is NOT recorded.

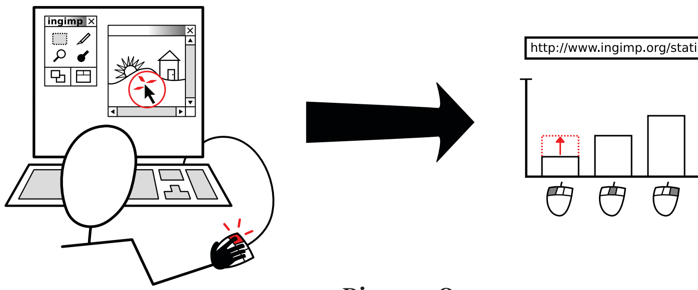


Diagram O

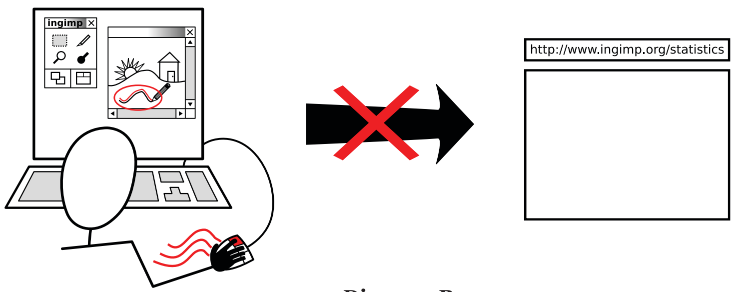


Diagram P

Which mouse button is pressed/released is recorded (button 1, 2, 3), but NOT the location of the cursor

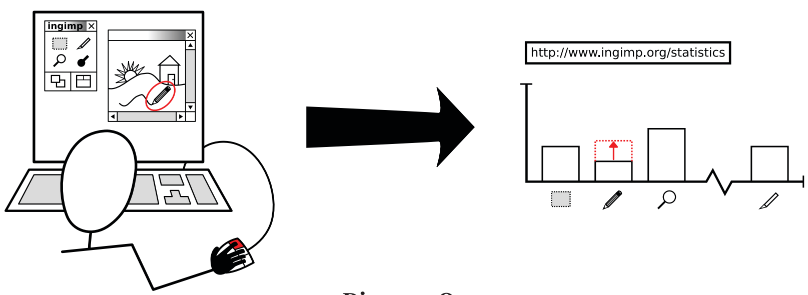


Diagram Q

- When and what tools (paintbrush, eraser, etc.) are selected
- When foreground/background color, brush opacity, paint mode, pattern, gradient, palette, and fonts change, but NOT the details about what the new selection actually is (e.g., the new foreground color is NOT recorded when it is changed)
- When a document window loses and gains focus (that is, when it is in front or pushed back)

- When a document window is resized, maximized, minimized, iconified, deiconified, shown/hidden, and what its new size and location are
- When documents are created, closed, saved, opened, duplicated -- essentially, whenever they come into and out of existence

The above list is not exhaustive but representative of the types of information recorded.

Your Privacy

Our goal is to honor your privacy in the choice of data collected and how it is collected. However, there are ways that personal information could inadvertently be logged by ingimp:

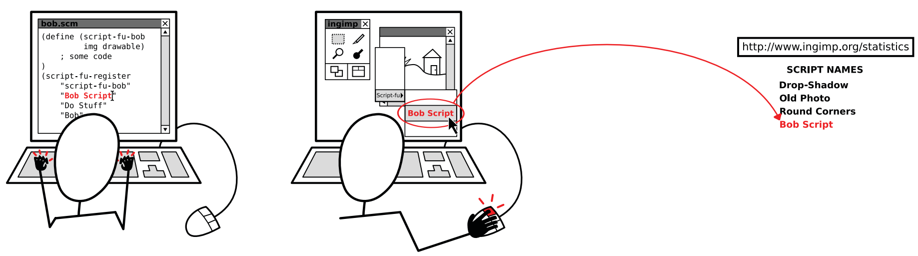


Diagram R

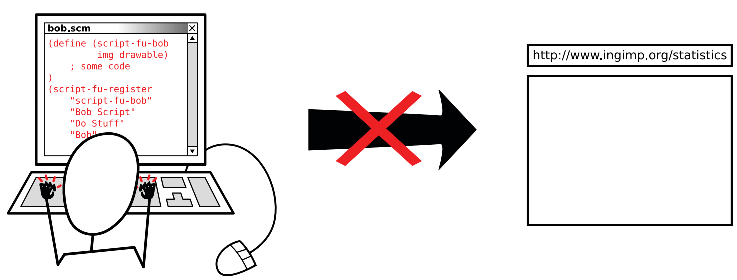


Diagram S

- If you create your own custom scripts, your script names will be recorded because ALL command names are recorded. Thus, if you named your script “john_smiths_acme_script”, that script name would be recorded whenever you used it.
- When you first start ingimp, you will have the opportunity to enter “activity tags,” or free-form descriptions of how you plan on using the software. These are purely optional. You are free to enter any text in the activity tag field, but should remember that these contents will be available for anyone to see in your log file. Thus, if you enter, “Removing Aunt Edna from

John Smith's family photo" in the Activity Tags field, this text will be available for anyone to read in the uploaded log files.

Your privacy is important to us. Consequently, we want to emphasize that we do not intentionally collect any personal information and that our data collection is restricted to activity within this piece of software only. However, as a final assurance, you are free to download, compile, and inspect the source code for this software. The distribution is available at http://www.ingimp.org. You may also inspect a typical log file at: http://www.ingimp.org.

Periodically, the software will attempt to transmit the logging information to a central server where it will be stored indefinitely. The general public will have access to all usage data (including yours) via http://www.ingimp.org.

Compensation

You will not be compensated for participation in the study. However, the information you provide may lead to more usable software as well as an increased understanding of how users of open source software can contribute to usability efforts.

Confidentiality and Data Retention

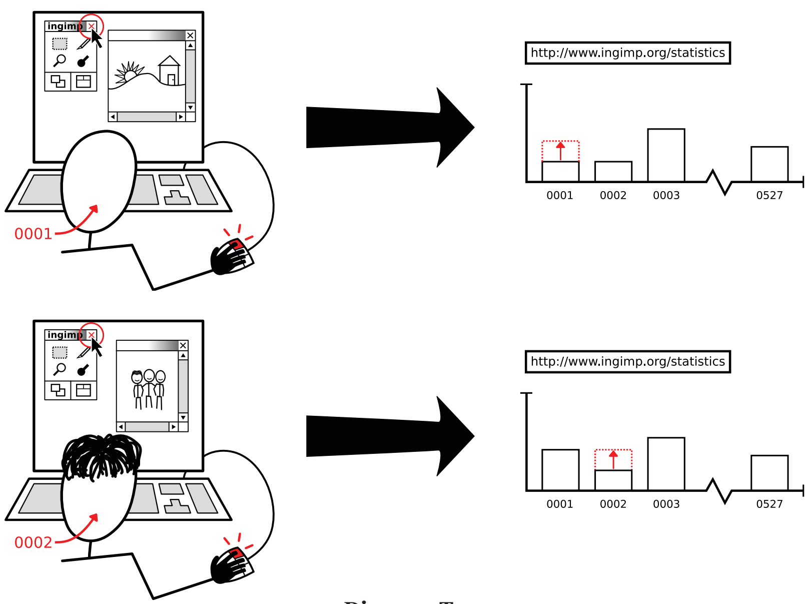


Diagram T

All questionnaire data and logged data will be kept indefinitely by the researcher. All collected data will be made publicly available. You should be aware that the public availability of your log data means that we make no guarantees regarding confidentiality, even though we do not intentionally collect personal information.

Questions

We offer no support for this software. However, if you have any questions about participation in this study, during or after the study, please contact Dr. Michael Terry at ingimp@cs.uwaterloo.ca.

Consent

By selecting "I agree to participate in this study," you are indicating your consent to participate in this study and are acknowledging that you are at least 18 years old, or that you are a parent or legal guardian granting permission for your minor to participate in this study.

Your consent indicates that you have read and understood this consent letter and have had the opportunity to receive any additional details about the study. Further, it indicates you understand that you may withdraw consent at any time by terminating your use of this software.

This project has been reviewed by, and received ethics clearance through, the Office of Research Ethics at the University of Waterloo. By granting consent, you are indicating that you have been informed that if you have any comments or concerns resulting from participation in this study, you may contact the Director, Office of Research Ethics at +1 (519) 888-4567 ext. 36005, or via email at ssykes@uwaterloo.ca.

Appendix B

Affinity diagrams from the textured agreement design process

Figure B.1 Affinity diagram of candidate approaches to redesigning software agreements. Initial prototypes were developed from these approaches, including video, illustrations, and enhanced text agreements.

Figure B.2 *Affinity diagram used to categorize examples of possible design techniques taken from popular media for use in textured agreements.*

Figure B.3 *Affinity diagram of design techniques from Figure B.2, abstracted from particular examples and re-categorized.*

Appendix C

Example textured agreements

Logging Consent Form

This software is part of a research effort actively exploring new ways to contribute to the usability of software. This software is modified to **automatically record** how you use this application. This information will be **automatically uploaded** to a **publicly accessible webserver** to enable usability analyses. The purpose of this document is to **gain your consent to do this data collection.**

This software **automatically logs usage information** such as the names of commands you use, general descriptions of the types of documents you edit, information about your platform (e.g., whether it is *Windows*, *Linux*, etc.), interaction events (when keys are pressed, when mouse buttons are pressed, but NOT the actual key or location of the mouse), and so on. These **data are sent to a server and made publicly available** for all to view and analyze. Our hope is that the data collected will help inform and focus future development and design efforts.

Principal Investigator

This study is being conducted by *Professor John Smith* in the *Computer Science Department* at the *Institute of Learning*. Questions should be directed to **smith@iolearning.com.**

Study Description

You must be **18 years or older** to participate, or you must obtain the consent of your parent or legal guardian. Participation is completely voluntary and can be stopped at any time by removing this software or discontinuing its use.

If you choose to participate, you may be asked to fill out a web-based questionnaire. The questionnaire is optional but will help us understand who is using this software.

Once consent to use this software is given, it will log your usage of the software. This version of the software **logs the following information**:

- **Your timezone**
- **Your platform and its characteristics** (e.g., *Windows*, *Linux*, CPU, if available, the version of your operating system, if available...)
- **Any activity tags** you enter at start-up to describe how you will use the software (more information below).
- **The names of commands** that you use, but not the parameters used for those commands
- **The number of letters, digits, spaces, punctuation marks, and forward/ backward slashes in your file names**, but NOT the file names themselves
- **Characteristics of your images after each command has been applied**, including:
  - *image sizes*
  - *image histograms*
- **When and what tools** are selected

The above list is not exhaustive but representative of the types of information recorded.

Your Privacy

“privacy *n.* ~
The state of being
private”
Wiktionary
The Open Content Dictionary

Our goal is to honor your privacyin the choice of data collected and how it is collected. However, there are **ways that personal information could inadvertently be logged**:

- **If you create your own custom scripts** (available only in the professional version), **your script names will be recorded because ALL command names are recorded.** Thus, if you named your script "john_smiths_acme_script", that script name would be recorded whenever you used it
- When you start the software, you may occasionally be given the opportunity to enter free-form descriptions of how you plan on using the software called "**activity tags**". These are purely optional. You are free to enter any text in the activity tag field, but should remember that these contents will be available for anyone to see in your log file. Thus, if you enter, "*Removing Aunt Edna from John Smith's family photo*" in the Activity Tags field, this text will be **available for anyone to read** in the uploaded log files

Your privacy is important to us. Consequently, we want to emphasize that we do not intentionally collect any personal information and that our data collection is restricted to activity within this piece of software only. You may inspect a typical log file at **http://www.iolearning.com.**

Periodically, the software will attempt to transmit the logging information to a central server where it will be stored indefinitely. The general public will have access to all usage data (including yours) via http://www.iolearning.com.

Confidentiality and Data Retention

All questionnaire data and **logged data will be kept indefinitely** by the researcher. All collected data will be made publicly available. You should be aware that the public availability of your log data means that **we make no guarantees regarding confidentiality**, even though we do not intentionally collect personal information.

Questions

We offer no support for this software. However, if you have any questions about participation in this study, during or after the study, please contact:

Dr. John Smith, **smith@iolearning.com**

Remuneration

You will not be remunerated for participation in the study. However, the information you provide may lead to more usable software and an increased understanding of how end-users can contribute to usability efforts.

Consent

“to agree
v. (intransitive) ~
To yield assent;
to accede”
Wiktionary
The Open Content Dictionary

By selecting "I AGREE TO PARTICIPATE IN THIS STUDY," you are **indicating your consent** to participate in this study and are acknowledging that you are **at least 18 years old**, or that you are a parent or legal guardian granting permission for your minor to participate in this study.

Your **consent** indicates that **you have read** this consent letter and have had the opportunity to receive any additional details about the study. Further, it indicates you are aware that you may withdraw consent at any time by terminating your use of this software.

This project has been reviewed by, and received ethics clearance through, the *Office of Research Ethics* at the *Institute of Learning*. By granting consent, you are indicating that you have been informed that if you have any comments or concerns resulting from participation in this study, you may contact the *Director, Office of Research Ethics* at **+1 (519) 888-4567 ext. 36005**, or via email at **syks@iolearning.com.**

Figure C.1 *Example of a minimal application of textured agreement techniques. This is one of three agreements used for the minimal condition in Experiment 1 in Chapter 4.*

Logging Consent Form

“ **98% of users**
don't read consent agreements ”

This software is part of a research effort actively exploring new ways to contribute to the usability of software. This software is modified to **automatically record** how you use this application. This information will be **automatically uploaded** to a **publicly accessible webserver** to enable usability analyses. The purpose of this document is to **gain your consent to do this data collection.**

This software **automatically logs usage information** such as the names of commands you use, general descriptions of the types of documents you edit, information about your platform (e.g., whether it is *Windows*, *Linux*, etc.), interaction events (when keys are pressed, when mouse buttons are pressed, but NOT the actual key or location of the mouse), and so on. These **data are sent to a server and made publicly available** for all to view and analyze. Our hope is that the data collected will help inform and focus future development and design efforts.

Principal Investigator

This study is being conducted by *Professor John Smith* in the *Computer Science Department* at the *Institute of Learning*. Questions should be directed to **smith@iolearning.com**.

Study Description

You must be **18 years or older** to participate, or you must obtain the consent of your parent or legal guardian. Participation is completely voluntary and can be stopped at any time by removing this software or discontinuing its use.

If you choose to participate, you may be asked to fill out a web-based questionnaire. The questionnaire is optional but will help us understand who is using this software.

Once consent to use this software is given, it will log your usage of the software. This version of the software **logs the following information**:

- **Your timezone**
- **Your platform and its characteristics** (e.g., *Windows*, *Linux*, CPU, if available, the version of your operating system, if available...)
- **Any activity tags** you enter at start-up to describe how you will use the software (more information below).

“ **our most popular platform is *Windows*** ”

- **The number of letters, digits, spaces, punctuation marks, and forward/backward slashes in your file names**, but NOT the file names themselves
- **We log the names of commands that you use**, but not the parameters used for those commands
- **Characteristics of your images after each command has been applied**, including:
  - *image sizes*
  - *image histograms*
- **When and what tools** are selected

The above list is not exhaustive but representative of the types of information recorded.

Your Privacy

Our goal is to honor your privacyin the choice of data collected and how it is collected. However, there are **ways that personal information could inadvertently be logged**:

- **If you create your own custom scripts** (available only in the professional version), **your script names will be recorded** because ALL command names are recorded. Thus, if you named your script "john_smiths_acme_script", that script name would be recorded whenever you used it
- When you start the software, you may occasionally be given the opportunity to enter free-form descriptions of how you plan on using the software called "**activity tags**". These are purely optional. You are free to enter any text in the activity tag field, but should remember that these contents will be available for anyone to see in your log file. Thus, if you enter, "*Removing Aunt Edna from John Smith's family photo*" in the Activity Tags field, **this text will be available for anyone to read** in the uploaded log files

Your privacy is important to us. Consequently, we want to emphasize that we do not intentionally collect any personal information and that our data collection is restricted to activity within this piece of software only. You may inspect a typical log file at **http://www.iolearning.com**.

“ **We've collected nearly 4000 publicly available logs so far** ”

Periodically, the software will attempt to transmit the logging information to a central server where it will be stored indefinitely. The general public will have access to all usage data (including yours) via http://iolearning.com.

Confidentiality and Data Retention

All questionnaire data and **logged data will be kept indefinitely** by the researcher. All collected data will be made publicly available. You should be aware that the public availability of your log data means that **we make no guarantees regarding confidentiality**, even though we do not intentionally collect personal information.

Questions

We offer no support for this software. However, if you have any questions about participation in this study, during or after the study, please contact:

Dr. John Smith, **smith@iolearning.com**

Remuneration

You will not be remunerated for participation in the study. However, the information you provide may lead to more usable software and an increased understanding of how end-users can contribute to usability efforts.

Consent

By selecting "I AGREE TO PARTICIPATE IN THIS STUDY," you are **indicating your consent** to participate in this study and are acknowledging that you are **at least 18 years old**, or that you are a parent or legal guardian granting permission for your minor to participate in this study.

Your **consent** indicates that **you have read** this consent letter and have had the opportunity to receive any additional details about the study. Further, it indicates you are aware that you may withdraw consent at any time by terminating your use of this software.

This project has been reviewed by, and received ethics clearance through, the *Office of Research Ethics* at the *Institute of Learning*. By granting consent, you are indicating that you have been informed that if you have any comments or concerns resulting from participation in this study, you may contact the *Director, Office of Research Ethics* at **+1 (519) 888-4567 ext. 36005**, or via email at **syks@iolearning.com**.

Figure C.2 *Example of a moderate application of textured agreement techniques. This is one of three agreements used for the moderate condition in Experiment 1 in Chapter 4.*

Logging Consent Form

"98% of users don't read consent agreements"

whatever

THIS SOFTWARE IS PART OF A RESEARCH EFFORT ACTIVEly exploring new ways to contribute to the usability of software. This software is modified to **automatically record** how you use this application. This information will be **automatically uploaded** to a **publicly accessible webserver** to enable usability analyses. The purpose of this document is to **gain your consent to do this data collection**.

This software **automatically logs usage information** such as the names of commands you use, general descriptions of the types of documents you edit, information about your platform (e.g., whether it is *Windows*, *Linux*, etc.), and so on. These **data are sent to a server and made publicly available** for all to view and analyze. Our hope is that the data collected will help inform and focus future development and design efforts.

OUR USERS a poll

90 percent of people using this software with a digital camera

average size of our users' photo collections **121**

28 average age of the people using this software

This software collects WHAT?

Principle Investigator: This study is being conducted by *Professor John Smith* in the *Computer Science Department* at the *Institute of Learning*. Questions should be directed to **smith@iolearning.com.**

You must be **18 years or older** to participate, or you must obtain the consent of your parent or legal guardian. Participation is completely voluntary and can be stopped at any time by removing this software or discontinuing its use.

If you choose to participate, you may be asked to fill out a web-based questionnaire. The questionnaire is optional but will help us understand who is using this software.

Once consent to use this software is given, it will log your usage of the software. This version of the software logs the following information.

We log:

- **Your timezone**
- **Your platform and its characteristics** (e.g., *Windows*, *Linux*, CPU, if available, the version of your operating system, if available...)
- **Characteristics of your images after each command has been applied**, including:

image sizes

Pwwease don't cowwect my data

image histograms

stuff we log

Photo copyright Pirate Scott. Used under Creative Commons Noncommercial Attribution License **http://www.flickr.com/photos/piratescott/467458969/**

"our most popular platform is *Windows**"

*based on data automatically collected by this software

- **When documents are created, closed, saved, opened, duplicated** -- essentially, whenever they come into and out of existence
- **We log the names of commands that you use**, but not the parameters used for those commands
- **Any activity tags you enter** to describe how you will use the software (more information below)

The above list is not exhaustive but representative of the types of information recorded.

we're collecting COMMAND NAMES!

Your Privacy

Our goal is to honor your privacy in the choice of data collected and how it is collected. However, there are **ways that personal information could inadvertently be logged:**

- **If you create your own custom scripts** (available only in the professional version), **your script names will be recorded** because ALL command names are recorded. Thus, if you named your script "john_smiths_acme_script", that script name would be recorded whenever you used it
- When you start the software, you may occasionally be given the opportunity to enter free-form descriptions of how you plan on using the software called "**activity tags**". These are purely optional. You are free to enter any text in the activity tag field, but should remember that these contents will be available for anyone to see in your log file. Thus, if you enter, "*Removing Aunt Edna from John Smith's family photo*" in the Activity Tags field, **this text will be available for anyone to read** in the uploaded log files.

WHAT?!

www.ingimp.org
User 38047598357
Tags
Removing Aunt Edna from John Smith's Family Photo

Inspect a typical log file at **http://iolearning.com/**

Your privacy is important to us. Consequently, we want to emphasize that we do not intentionally collect any personal information and that our data collection is restricted to activity within this piece of software only. You may inspect a typical log file at **http://www.iolearning.com.**

"We've collected nearly **4000** publicly available logs so far"

Periodically, the software will attempt to transmit the logging information to a central server where it will be stored indefinitely. The general public will have access to all usage data (including yours) via http://www.iolearning.com.

Your logs are publicly accessible

Confidentiality and Data Retention

All questionnaire data and **logged data will be kept indefinitely** by the researcher. All collected data will be made publicly available. You should be aware that the public availability of your log data means that **we make no guarantees regarding confidentiality**, even though we do not intentionally collect personal information.

Questions

We offer no support for this software. However, if you have any questions about participation in this study, during or after the study, please contact:

Dr. John Smith, **smith@iolearning.com**

Remuneration

You will not be remunerated for participation in the study. However, the information you provide may lead to more usable software and an increased understanding of how end-users can contribute to usability efforts.

Do you know what we're doing with YOUR DATA?

C O N S E N T

BY SELECTING "I AGREE TO PARTICIPATE IN THIS STUDY," YOU are **indicating your consent** to participate in this study and are acknowledging that you are **at least 18 years old**, or that you are a parent or legal guardian granting permission for your minor to participate in this study.

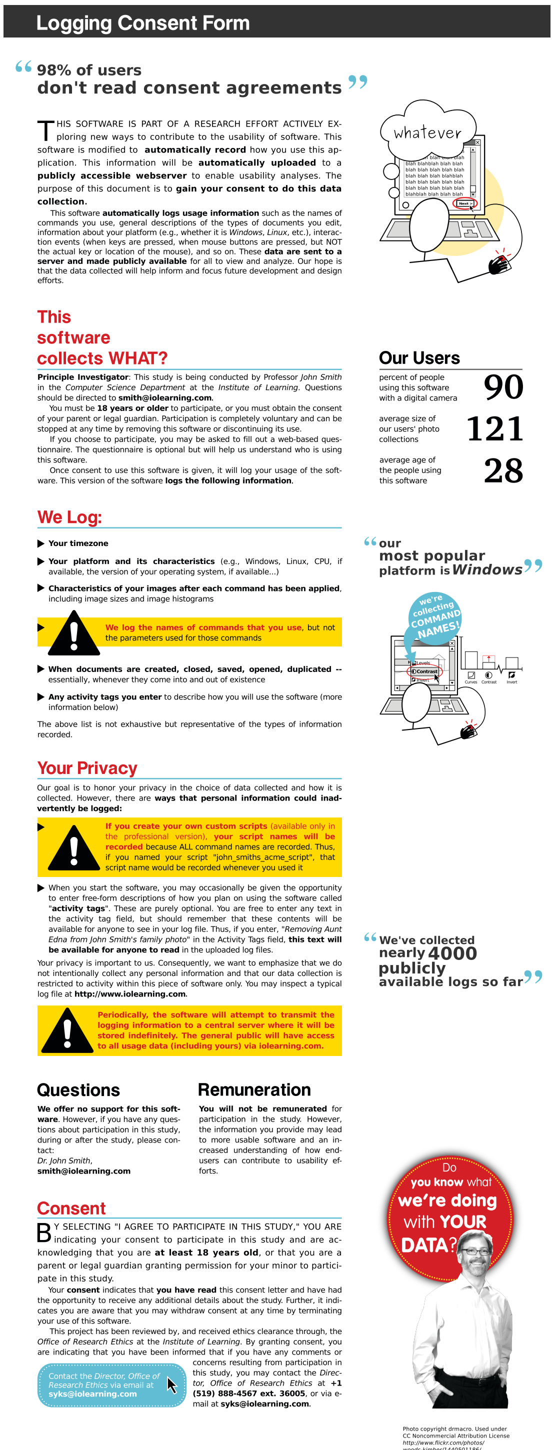
Your **consent** indicates that **you have read** this consent letter and have had the opportunity to receive any additional details about the study. Further, it indicates you are aware that you may withdraw consent at any time by terminating your use of this software.

This project has been reviewed by, and received ethics clearance through, the *Office of Research Ethics* at the *Institute of Learning*. By granting consent, you are indicating that you have been informed that if you have any comments or concerns resulting from participation in this study, you may contact the *Director, Office of Research Ethics* at **+1 (519) 888-4567 ext. 36005**, or via email at **syks@iolearning.com.**

Contact the *Director, Office of Research Ethics* via email at **syks@iolearning.com**

Photo copyright drmacro. Used under CC Noncommercial Attribution License *http://www.flickr.com/photos/woods-kimber/1440501186/*

Figure C.3 *Example of a heavy application of textured agreement techniques. This is one of three agreements used for the heavy condition in Experiment 1 in Chapter 4.*



Logging Consent Form

“98% of users
don't read consent agreements”

THIS SOFTWARE IS PART OF A RESEARCH EFFORT ACTIVELY EXploring new ways to contribute to the usability of software. This software is modified to **automatically record** how you use this application. This information will be **automatically uploaded** to a **publicly accessible webserver** to enable usability analyses. The purpose of this document is to **gain your consent to do this data collection**.

This software **automatically logs usage information** such as the names of commands you use, general descriptions of the types of documents you edit, information about your platform (e.g., whether it is *Windows*, *Linux*, etc.), interaction events (when keys are pressed, when mouse buttons are pressed, but NOT the actual key or location of the mouse), and so on. These **data are sent to a server and made publicly available** for all to view and analyze. Our hope is that the data collected will help inform and focus future development and design efforts.

This software collects WHAT?

Principle Investigator: This study is being conducted by Professor *John Smith* in the *Computer Science Department* at the *Institute of Learning*. Questions should be directed to **smith@iolearning.com**.

You must be **18 years or older** to participate, or you must obtain the consent of your parent or legal guardian. Participation is completely voluntary and can be stopped at any time by removing this software or discontinuing its use.

If you choose to participate, you may be asked to fill out a web-based questionnaire. The questionnaire is optional but will help us understand who is using this software.

Once consent to use this software is given, it will log your usage of the software. This version of the software **logs the following information**.

Our Users

percent of people using this software with a digital camera	90
average size of our users' photo collections	121
average age of the people using this software	28

We Log:

- **Your timezone**
- **Your platform and its characteristics** (e.g., Windows, Linux, CPU, if available, the version of your operating system, if available...)
- **Characteristics of your images after each command has been applied**, including image sizes and image histograms
- **We log the names of commands that you use**, but not the parameters used for those commands
- **When documents are created, closed, saved, opened, duplicated --** essentially, whenever they come into and out of existence
- **Any activity tags you enter** to describe how you will use the software (more information below)

The above list is not exhaustive but representative of the types of information recorded.

“our
most popular
platform is *Windows*”

Your Privacy

Our goal is to honor your privacy in the choice of data collected and how it is collected. However, there are **ways that personal information could inadvertently be logged:**

- **If you create your own custom scripts** (available only in the professional version), **your script names will be recorded** because ALL command names are recorded. Thus, if you named your script "john_smiths_acme_script", that script name would be recorded whenever you used it
- When you start the software, you may occasionally be given the opportunity to enter free-form descriptions of how you plan on using the software called "**activity tags**". These are purely optional. You are free to enter any text in the activity tag field, but should remember that these contents will be available for anyone to see in your log file. Thus, if you enter, "*Removing Aunt Edna from John Smith's family photo*" in the Activity Tags field, **this text will be available for anyone to read** in the uploaded log files.

Your privacy is important to us. Consequently, we want to emphasize that we do not intentionally collect any personal information and that our data collection is restricted to activity within this piece of software only. You may inspect a typical log file at **http://www.iolearning.com**.

Periodically, the software will attempt to transmit the logging information to a central server where it will be stored indefinitely. The general public will have access to all usage data (including yours) via iolearning.com.

“We've collected
nearly **4000**
publicly
available logs so far”

Questions

We offer no support for this software. However, if you have any questions about participation in this study, during or after the study, please contact:
Dr. John Smith,
smith@iolearning.com

Remuneration

You will not be remunerated for participation in the study. However, the information you provide may lead to more usable software and an increased understanding of how end-users can contribute to usability efforts.

Consent

BY SELECTING "I AGREE TO PARTICIPATE IN THIS STUDY," YOU ARE indicating your consent to participate in this study and are acknowledging that you are **at least 18 years old**, or that you are a parent or legal guardian granting permission for your minor to participate in this study.

Your **consent** indicates that **you have read** this consent letter and have had the opportunity to receive any additional details about the study. Further, it indicates you are aware that you may withdraw consent at any time by terminating your use of this software.

This project has been reviewed by, and received ethics clearance through, the *Office of Research Ethics* at the *Institute of Learning*. By granting consent, you are indicating that you have been informed that if you have any comments or concerns resulting from participation in this study, you may contact the *Director, Office of Research Ethics* at **+1 (519) 888-4567 ext. 36005**, or via e-mail at **syks@iolearning.com**.

Contact the *Director, Office of Research Ethics* via email at **syks@iolearning.com**

Figure C.4 *Final textured agreement design based on successful elements from the moderate and heavy treatments. This example shows the full textured agreement used in Experiment 2 in Chapter 4.*

Appendix D

Affinity diagrams from the analysis of EULAscan reviews

Figure D.1 *Affinity diagram used for the initial coding of the EULAscan reviews. This diagram includes every review from the EULAscan corpus analyzed in Chapter 5. A large number of short, one- or two-sentence reviews are grouped in stacks in this diagram, including those with non-substantive comments; see the analysis of reviewer engagement in Chapter 5 for a discussion of substantive versus non-substantive comments. The set of codes developed from this diagram was used as a starting point for coding the entire set of EULAscan comments, during which the codes were further refined.*

Figure D.2 *Affinity diagram used to categorize the codes applied to EULAscan reviews. After the reviews were coded (see Figure D.1), these codes were further grouped into high-level categories using this affinity diagram. Reviews were subsequently re-coded according to the high-level categories. These categories include the various high-level concerns users had with agreement content discussed in Chapter 5.*

Bibliography

1. **030108** [pseud.]. Re: is there a free way to record skype calls. *MacRumors: Forums*. April 2008. http://forums.macrumors.com/showthread.php?p=5341232#post5341232. Last checked June 16, 2010.

2. **2-Spyware.com.** EULAscan—for those who hate reading deadly boring license agreements. *2-Spyware.com*. November 2005. http://www.2-spyware.com/news/post18.html. Last checked June 4, 2010.

3. **Abrams,** Martin E., Margaret P. Eisenhauer, and Lisa J. Sotto. Response to the FTC request for public comments in the Advance Notice of Proposed Rulemaking on Alternative Forms of Privacy Notices under the Gramm-Leach-Bliley Act. *Center for Information Policy Leadership, Hunton and Williams*. March 2004. http://www.hunton.com/files/tbl_s47Details/FileUpload265/685/CIPL_Notices_ANPR_Comments_3.29.04.pdf.

4. **Acquisti,** Alessandro, and Jens Grossklags. Privacy and rationality in individual decision making. *Security & Privacy, IEEE*. 2005, vol. 3, no. 1, pp. 26-33.

5. **Ahlstrom,** Chris. Re: Company Dumps .NET Software, Finds Happiness in Open Source. *Google Groups: comp.os.linux.advocacy*. April 2007. http://groups.google.com/group/comp.os.linux.advocacy/msg/3b912c35fbe6b2f4. Last checked June 17, 2010.

6. **Ahlstrom,** Chris. Re: iTunes surpasses 2-Billion song downloads. *Google Groups: comp.os.linux.advocacy*. January 2007. http://groups.google.com/group/comp.os.linux.advocacy/msg/fb9ebc2b7d86464a. Last checked June 17, 2010.

7. **Alexa.** Lifehacker. *Alexa*. June 2010. http://www.alexa.com/siteinfo/lifehacker.com. Last checked June 4, 2010.

8. **Amer,** T. S., and Jo-Mae B. Maris. Signal Words and Signal Icons in Information Technology Exception and Error Messages. Northern Arizona University. Working Paper Series. SSRN, 2004.

9. **Aviary.** Terms of Use. *Aviary*. March 2008. http://aviary.com/terms. Last checked April 5, 2009.

10. **Bederson,** Benjamin B., and James D. Hollan. Pad++: a zooming graphical interface for exploring alternate interface physics. *UIST '94: Proceedings of the 7th annual ACM symposium on User interface software and technology*. New York, NY, USA: ACM, 1994, pp. 17–26.

11. **Beverly** [pseud.]. Re: Understanding Apple's DRM. *Schneier on Security*. March 2007. http://www.schneier.com/blog/archives/2007/03/understanding_a.html#c153491. Last checked June 4, 2010.

12. **Biggs,** William. Re: KnowBrainer Review of NaturallySpeaking Ver. 10. *emrupdate.com*. July 2008. http://www.emrupdate.com/forums/p/16066/87950.aspx. Last checked June 14, 2010.

13. **Bishop,** David. Weblinks. *ipWatch*. March 2008. http://mellowfellow.co.uk/ipwatch/index.php?option=com_weblinks&view=category&id=34%3Arights-groups&Itemid=48. Last checked June 14, 2010.

14. **Braucher,** Jean. Delayed Disclosure in Consumer E-Commerce as an Unfair and Deceptive Practice. *Wayne Law Review*. 2000, vol. 46, pp. 1805–1868.

15. **Brustoloni,** José Carlos, and Ricardo Villamarín-Salomón. Improving security decisions with polymorphic and audited dialogs. *SOUPS '07*. New York, NY, USA: ACM, 2007, pp. 76–85.

16. **Buechner,** Maryanne Murray. 50 Coolest Websites 2005: Blogs. *Time*. June 2005. http://www.time.com/time/business/article/0,8599,1072872,00.html. Last checked June 5, 2010.

17. **Campbell,** Frances A., Barbara D. Goldman, Maria L. Boccia, and Martie Skinner. The effect of format modifications and reading comprehension on recall of informed consent information by low-income parents: a comparison of print, video, and computer-based presentations. *Patient Education and Counseling*. Elsevier, 2004, vol. 53, no. 2, pp. 205–216.

18. **Center for Information Policy Leadership, Hunton and Williams.** HIPAA Highlights Privacy Notice. *Center for Information Policy Leadership, Hunton and Williams*. April 2003. http://www.hunton.com/news/news.aspx?nws_pg=7&gen_H4ID=10102. Last checked Sep 19, 2008.

19. **Center for Information Policy Leadership, Hunton and Williams.** Ten steps to develop a multilayered privacy notice. *Center for Information Policy Leadership, Hunton and Williams*. 2007. http://www.hunton.com/files/tbl_s47Details%5CFileUpload265%5C1405%5CTen_Steps_whitepaper.pdf. Last checked May 13, 2010.

20. **Cranor,** Lorrie Faith, Joseph Reagle, and Mark S. Ackerman. Beyond concern: Understanding net users' attitudes about online privacy. *The Internet Upheaval: Raising Questions, Seeking Answers in Communications Policy*. MIT Press, 2000, pp. 47–70.

21. **Culnan,** Mary J., and George R. Milne. The Culnan-Milne Survey on Consumers & Online Privacy Notices. Federal Trade Commission. 2001.

22. **d-tracks** [pseud.]. Re: Need a EULA (end user licence agreement). *Streamerp2p Forum*. February 2007. http://streamerp2p.com/forum/viewtopic.php?f=10&t=1878&start=0. Last checked June 17, 2010.

23. **de Icaza,** Miguel. Re: Reading Comprehension and the English Language. *tirania.org blog comments*. September 2007. http://groups.google.com/group/tiraniaorg-blog-comments/msg/990ffd17c5ffa186. Last checked June 17, 2010.

24. **Egelman,** Serge, Janice Tsai, Lorrie Faith Cranor, and Alessandro Acquisti. Timing Is Everything? The Effects of Timing and Placement of Online Privacy Indicators. *CHI '09*. New York, NY, USA: ACM, 2009.

25. **Frascara,** Jorge. *Communication design: principles, methods, and practice*. Allworth Communications, Inc., 2004.

26. **Free Software Foundation.** GNU General Public License, version 3. *Free Software Foundation*. June 2007. http://www.gnu.org/copyleft/gpl.html. Last checked May 13, 2010.

27. **Friedman,** Batya, Ian Smith, Peter H. Kahn Jr., Sunny Consolvo, and Jaina Selawski. Development of a Privacy Addendum for Open Source Licenses: Value Sensitive Design in Industry. *Lecture notes in computer science*. Springer, 2006, vol. 4206, pp. 194.

28. **Friedman,** Batya, Peyina Lin, and Jessica K. Miller. Informed consent by design. *Security and usability: designing secure systems that people can use*. Ed. Lorrie Faith Cranor and Simson Garfinkel. O'Reilly Media, Inc., 2005.

29. **Gahtan,** Alan. Bad and Unfair End User Agreements. *Gahtan's Technology and Internet Law Blog*. November 2005. http://www.gahtan.com/techlawblog/2005/11/29/bad-and-unfair-end-user-agreements/. Last checked June 4, 2010.

30. **Good,** Nathaniel S., Jens Grossklags, Deirdre K. Mulligan, and Joseph A. Konstan. Noticing notice: a large-scale experiment on the timing of software license agreements. *CHI '07*. New York, NY, USA: ACM, 2007, pp. 607–616.

31. **Good,** Nathaniel S., Rachna Dhamija, Jens Grossklags, David Thaw, Steven Aronowitz, Deirdre K. Mulligan, and Joseph A. Konstan. Stopping spyware at the gate: a user study of privacy, notice and spyware. *SOUPS '05*. New York, NY, USA: ACM, 2005, pp. 43–52.

32. **Greenwood,** Patrick. EULAScan - Another Precursor of Open Source Lawyering?. *Blawgcoop*. December 2005. http://www.blawgcoop.com/pgreenwood/?p=19. Last checked June 15, 2010.

33. **Grossklags,** Jens, and Nathaniel S. Good. Empirical Studies on Software Notices to Inform Policy Makers and Usability Designers. *Financial Cryptography and Data Security*. 2008, pp. 341–355.

34. **Hellier,** Elizabeth, Daniel B. Wright, Judy Edworthy, and Stephen Newstead. On the stability of the arousal strength of warning signal words. *Applied Cognitive Psychology*. 2000, vol. 14, no. 6, pp. 577–592.

35. **Hillman,** Robert A. On-line Consumer Standard-Form Contracting Practices: A Survey and Discussion of Legal Implications. Cornell Law School. Legal Studies Research Paper Series. SSRN, 2005.

36. **Hillman,** Robert A. Online Boilerplate: Would Mandatory Website Disclosure of E-Standard Terms Backfire?. *Michigan Law Review*. JSTOR, 2006, vol. 104, no. 5, pp. 837–856.

37. **Hofmann,** Patrick. Away with words! How to create wordless documentation. *IPCC '98*. IEEE International, 1998, vol. 2, pp. 437–438.

38. **Hofmann,** Patrick. Localising and Internationalising Graphics and Visual Information Commentary. *IEEE Transactions on Professional Communication*. 2007, vol. 50, no. 91–92.

39. **International Organization for Standardization.** *ISO 9186-1: Graphical symbols—Test methods—Part 1: Methods for testing comprehensibility*. 2007.

40. **Jacobson,** David. End user licence agreements (EULA). *Langes*. December 2005. http://www.langes.com.au/technologyip/2005/12/11/end-user-licence-agreements-eula/. Last checked June 15, 2010.

41. **Jensen,** Carlos, and Colin Potts. Privacy policies as decision-making tools: an evaluation of online privacy notices. *CHI '04*. New York, NY, USA: ACM, 2004, pp. 471–478.

42. **Kantor,** Andrew. iTunes and Pinnacle users: Guess what you agreed to. *Andrew Kantor's Place*. October 2005. http://www.andrewkantor.com/2005-10/itunes-and-pinnacle-users-guess-what-you-agreed-to/. Last checked June 15, 2010.

43. **Kay,** Matthew, and Michael Terry. Communicating software agreement content using narrative pictograms. *CHI EA '10: Proceedings of the 28th of the international conference extended abstracts on Human factors in computing systems*. New York, NY, USA: ACM, 2010, vol. 5, pp. 2715–2724.

44. **Kay,** Matthew, and Michael Terry. Textured agreements: re-envisioning electronic consent. *SOUPS '10: Proceedings of the 6th symposium on usable privacy and security*. New York, NY, USA: ACM, 2010.

45. **Kelley,** Patrick Gage, Joanna Bresee, Lorrie Faith Cranor, and Robert W. Reeder. A "nutrition label" for privacy. *SOUPS '09*. New York, NY, USA: ACM, 2009, pp. 1–12.

46. **Kennedy,** Dennis M. EULAScan - Another Precursor of Open Source Lawyering?. *Between Lawyers*. December 2005. http://betweenlawyers.corante.com/archives/2005/12/09/eulascan_another_precursor_of_open_source_lawyering.php. Last checked June 4, 2010.

47. **Kerouanton,** Bruno. Re: Conditions Générales d'Utilisation. *Zythom - Blog d'un informaticien expert judiciaire*. September 2007. http://zythom.blogspot.com/2007/08/conditions-gnrales-dutilisation.html?showComment=1188978720000#c562831795511190 7603. Last checked June 17, 2010.

48. **Kinner,** Tony. Time to get EULAgising. *ITProPortal*. November 2005. http://www.itproportal.com/portal/news/article/2005/11/30/time-to-get-eulagising/. Last checked June 17, 2010.

49. **Kress,** Gunther, and Theo van Leeuwen. *Reading images: the grammar of visual design*. Routledge, 2006.

50. **Lehman,** Meir M. Programs, life cycles, and laws of software evolution. *Proceedings of the IEEE*. 1980, vol. 68, no. 9, pp. 1060–1076.

51. **Lifehacker.** Community-driven EULA feedback. *Lifehacker*. October 2005. http://lifehacker.com/133402/community+driven-eula-feedback. Last checked June 4, 2010.

52. **Lifehacker.** Lifehacker Frequently Asked Questions. *Lifehacker*. http://lifehacker.com/software/announcements/lifehacker-faq-28869.php. Last checked June 4, 2010.

53. **macFriend** [pseud.]. Re: Macintosh Security. *Schneier on Security*. August 2008. http://www.schneier.com/blog/archives/2007/10/macintosh_secur.html#c299901. Last checked June 8, 2010.

54. **Marotta-Wurgler,** Florencia. Competition and the Quality of Standard Form Contracts: An Empirical Analysis of Software License Agreements. New York University School of Law. Law and Economics Research Paper Series. 05-11, 2005.

55. **Masnick,** Mike. Fight Back Against Bad EULAs. *Techdirt*. November 2005. http://gamma.serverbox.net/articles/20051129/0318233.shtml. Last checked June 4, 2010.

56. **McDonald,** Aleecia, Robert Reeder, Patrick Kelley, and Lorrie Cranor. A Comparative Study of Online Privacy Policies and Formats. *PETS '09: Proceedings of the 9th International Symposium on Privacy Enhancing Technologies*. 2009, pp. 37–55.

57. **McGrenere,** Joanna, Rhian Davies, Leah Findlater, Peter Graf, Maria Klawe, Karyn Moffatt, Barbara Purves, and Sarah Yang. Insights from the aphasia project: designing technology for and with people who have aphasia. *CUU '03*. New York, NY, USA: ACM, 2003, pp. 112–118.

58. **McMillan,** Robert. Sony Ships Sneaky DRM Software. *PCWorld*. November 2005. http://www.pcworld.com/article/123362/sony_ships_sneaky_drm_software.html. Last checked June 3, 2010.

59. **Medhi,** Indrani, Aman Sagar, and Kentaro Toyama. Text-free user interfaces for illiterate and semiliterate users. *Information Technologies and International Development*. 2007, vol. 4, no. 1, pp. 37–50.

60. **Meggs,** Philip B., and Alston W. Purvis. *Meggs' History of Graphic Design*. Wiley, 2005.

61. **Milles,** Jim. EULAScan - Another Precursor of Open Source Lawyering?. *Out of the Jungle*. December 2005. http://outofthejungle.blogspot.com/2005/12/eulascan-another-precursor-of-open.html. Last checked June 4, 2010.

62. **miramar** [pseud.]. Re: Hackintosh + Apple OS X Tiger retail pack. Legal?. *Lowyat.NET Forum*. June 2007. http://forum.lowyat.net/topic/473095/+0#entry12085051. Last checked June 16, 2010.

63. **Morrell,** Harvey. Out of the Jungle: EULAScan - Another Precursor of Open Source Lawyering?. *The University of Baltimore Law Library Weblog*. December 2005. http://ubaltlawlibrary.blogspot.com/2005/12/out-of-jungle-eulascan-another.html. Last checked June 16, 2010.

64. **National Conference of Commissioners on Uniform State Laws and the American Law Institute.** *Uniform Commercial Code § 2-103*.

65. **National Conference of Commissioners on Uniform State Laws and the American Law Institute.** *Uniform Commercial Code § 2-316*.

66. **Noveck,** Beth Simone. EULAlyzer. *Cairns*. October 2005. http://cairns.typepad.com/blog/2005/10/eulalyzer.html. Last checked June 17, 2010.

67. **Parker,** Avery J. Community site to help users understand EULA's better. *Computer Tips -Tech Info*. November 2005. http://www.averyjparker.com/2005/11/29/community-site-to-help-users-understand-eulas-better/. Last checked June 15, 2010.

68. **Patrick,** Andrew S., and Steve Kenny. From Privacy Legislation to Interface Design: Implementing Information Privacy in Human-Computer Interactions. *Privacy Enhancing Technologies*. 2003, pp. 107–124.

69. **Placebo** [pseud.]. Addon/Mod Ownership - Q&A. *Bohemia Interactive Forums*. November 2005. http://forums.bistudio.com/showthread.php?t=44973. Last checked June 17, 2010.

70. **Plutchok,** Jonathan. EULAppreciate This: Help with Required Reading, and Flame Retardant. *Jonathan's Tool Bar & Grill*. November 2006. http://jonathanstoolbar.blogspot.com/2006/11/6-eulappreciate-this-help-with-required.html. Last Checked June 4, 2010.

71. **Reynolds,** Tom. Re: [Announce] GnuPG 2.0.1 released. *GnuPG-users mailing list*. November 2006. http://www.mail-archive.com/gnupg-users%40gnupg.org/msg04752.html. Last checked June 17, 2010.

72. **Reynolds,** Tom. Re: 1.4.7 packages for OS X. *GnuPG-users mailing list*. March 2007. http://www.mail-archive.com/gnupg-users%40gnupg.org/msg05367.html. Last checked June 16, 2010.

73. **Reynolds,** Tom. Re: Leopard & earlier iBooks. *iBook mailing list*. September 2007. http://listserver.themacintoshguy.com/pipermail/ibook/2007-September/014529.html. Last checked June 17, 2010.

74. **Reynolds,** Tom. Re: OpenPGP Smartcard and MacOS. *GnuPG-users mailing list*. July 2009. http://www.mail-archive.com/gnupg-users%40gnupg.org/msg11764.html. Last checked June 4, 2010.

75. **Rosen,** Lawrence. *Open Source Licensing: Software Freedom and Intellectual Property Law*. Prentice Hall, 2004.

76. **Schechter,** Roger E. The Unfairness of Click-On Software Licenses. *Wayne Law Review*. 2000, vol. 46, pp. 1735–1804.

77. **severin** [pseud.]. iNuke. *sidux.com Forum*. April 2007. http://sidux.com/index.php?name=PNphpBB2&file=viewtopic&t=3033&start=0. Last checked June 17, 2010.

78. **Shoemate,** Ben. End User Agreements – Stealing customer rights with a sneak attack in the fine print. *Ben Shoemate*. November 2007. http://www.benshoemate.com/2007/11/29/end-user-agreements-stealing-customer-rights-with-a-sneak-attack-in-the-fine-print/. Last checked June 4, 2010.

79. **Spiekermann,** Sarah, Jens Grossklags, and Bettina Berendt. E-privacy in 2nd generation E-commerce: privacy preferences versus actual behavior. *Proceedings of the 3rd ACM conference on Electronic Commerce*. New York, NY, USA: ACM, 2001, pp. 38–47.

80. **Sternberg,** Andy. EULA tools, guides, forums, and decoders. *The Small Print Project*. November 2006. http://smallprint.netzoo.net/eula-tools-guides-forums-and-decoders/. Last checked June 15, 2010.

81. **Stirling,** Scott. EULAscan: community-written EULA reviews. *Exploring and Evolving*. November 2005. http://scottstirling.blogspot.com/2005/11/eulascan-community-written-eula.html. Last checked June 4, 2010.

82. **Tee,** K., K. Moffatt, L. Findlater, E. MacGregor, J. McGrenere, B. Purves, and S.S. Fels. A visual recipe book for persons with language impairments. *Proceedings of the SIGCHI conference on Human factors in computing systems*. 2005, pp. 510.

83. **Terry,** Michael, Matthew Kay, Brad Van Vugt, Brandon Slack, and Terry Park. Ingimp: introducing instrumentation to an end-user open source application. *CHI '08*. New York, NY, USA: ACM, 2008, pp. 607–616.

84. **Tones411** [pseud.]. Question about Terms and Conditions. *Gmail Help Discussion*. February 2007. http://groups.google.com/group/Gmail-Help-Organizing-Messages-en/browse_thread/thread/251d614fa99b45a7/fb9cf0dfdf2f1553#fb9cf0dfdf2f1553. Last checked June 17, 2010.

85. **Tufte,** Edward R. *Envisioning Information*. Cheshire, Conn: Graphics Press, 1995.

86. **Turner,** Suzi. Cool New Resource for EULAs. *ZDNet*. November 2005. http://www.zdnet.com/blog/spyware/cool-new-resource-for-eulas/710. Last checked June 4, 2010.

87. **two** [pseud.]. I need gcc 4 for OS 10.3.9. *Apple Support - Discussions*. November 2006. http://discussions.apple.com/message.jspa?messageID=3508471. Last checked June 17, 2010.

88. **White,** Jan V. *Editing by design*. Bowker New York, 1982.

89. **Wikipedia (English).** Software license agreement (comparison of revision as of 05:35, 12 September 2005 and revision as of 06:41, 13 September 2005). *Wikipedia (English)*. September 2005. http://en.wikipedia.org/w/index.php?title=Software_license_agreement&action=historysubmit&diff=23149331&oldid=23084593. Last checked June 17, 2010.

90. **Wikipedia (English).** Software license agreement (comparison of revision as of 23:04, 20 February 2007 and revision as of 23:21, 22 February 2007). *Wikipedia (English)*. February 2007. http://en.wikipedia.org/w/index.php?title=Software_license_agreement&action=historysubmit&diff=110185916&oldid=109672030. Last checked June 17, 2010.

91. **Wikipedia (Finnish).** EULA (comparison of revision as of 19.47, 16 October 2006 and revision as of 19:49, 26 October 2006). *Wikipedia (Finnish)*. October 2006. http://fi.wikipedia.org/w/index.php?title=EULA&action=historysubmit&diff=1806395&oldid=1806389. Last checked June 15, 2010.

92. **Wikipedia (Slovak).** Softvérová licencia (revision as of 11:59, 22 March 2006). *Wikipedia (Slovak)*. March 2006. http://sk.wikipedia.org/w/index.php?title=Softv%C3%A9rov%C3%A1_licencia&oldid=281599. Last checked June 17, 2010.

93. **Wogalter,** Michael S., and Kenneth R. Laughery. WARNING! Sign and Label Effectiveness. *Current Directions in Psychological Science*. 1996, vol. 5, no. 2, pp. 33–37.

94. **Wright,** Patricia. Presenting technical information: a survey of research findings. *Instructional Science*. Springer, 1977, vol. 6, no. 2, pp. 93–134.

95. **Wu,** Sandy T. Uniform Computer Information Transactions Act: Failed to Appease Its Opponents in Light of the Newly Adopted Amendments. *Southwestern University Law Review*. 2004, vol. 33, pp. 307–326.