

Communicating Software Agreement Content Using Narrative Pictograms

Matthew Kay, Michael Terry

HCI Lab, David R. Cheriton School of Computer Science

University of Waterloo

{mjskay, mterry}@cs.uwaterloo.ca

ABSTRACT

This paper presents *narrative pictograms*, diagrams designed to convey the abstract concepts of a software agreement. Narrative pictograms arose out of a need to increase the chance that subjects of any culture or language could understand the purposes and intent of a consent agreement accompanying publicly available experimental software. Accordingly, the diagrams presented in this paper are designed to be used *without* the aid of explanatory text. We first present our iterative design process and initial formative evaluation of the diagrams. We then present example diagrams designed to describe the data collection policies of research software, and the more general composition rules used to create them. Finally, we demonstrate the diagrams' ability to effectively communicate concepts by presenting results from an experimental study based on the ISO 9186-1 comprehension test for graphical symbols.

Author Keywords

Informed consent, pictograms, wordless diagrams, open source

ACM Classification Keywords

H5.m. Information interfaces and presentation (e.g., HCI): Miscellaneous.

INTRODUCTION

End users must often agree to lengthy terms of use prior to installing software or using software services. These terms of use may take the form of End User License Agreements (EULAs), privacy policies, or, in the case of computer-based human subjects research, consent agreements. While these various agreements each serve slightly different purposes, we collectively refer to them as *software agreements*.

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 that can require the ability to read at a university or postgraduate level [10]. These aspects of software agreements have two important consequences. First,

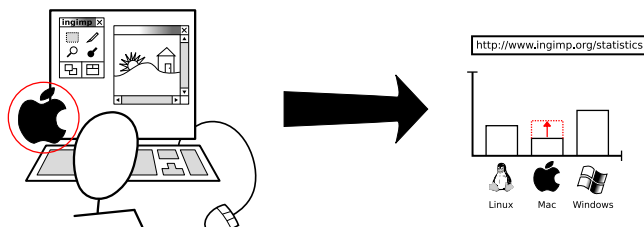


Figure 1. A narrative pictogram. The diagram illustrates the fact that the software will collect data about which operating system the user uses.

they make it challenging for users to read and comprehend the agreement terms, especially for users whose native language differs from that of the software agreement. Second, they make software agreements difficult and costly to translate into alternative locales. As a result, both users and software producers would benefit from techniques that improve the ability to effectively communicate agreement terms to a diverse audience.

This paper introduces *narrative pictograms*, a pictorial technique specifically designed to improve the process of communicating software agreement terms to a diverse population of users (Figure 1). Narrative pictograms are diagrams that make limited use of words to describe a software agreement's content. Importantly, as presented here, these illustrations are not meant to replace text-based agreements, but rather, to supplement agreements and facilitate the ease with which individuals can comprehend the agreement's content.

The contributions of this paper are twofold. First, we present a set of visual composition rules to guide the production of narrative pictograms. These composition rules were developed through an iterative design process driven by the goal of communicating the data collection policies of a publicly distributed research application. The diagrams produced by these rules make little use of text, making them of potential interest to emerging areas of HCI, such as interfaces for illiterate users [13].

Second, this paper describes results of a study that demonstrate the efficacy of these diagrams, even in the absence of the actual text-based agreement. Specifically, using a modified version of the ISO 9186-1 test method (a standard test used to measure comprehension of graphical symbols [9]), we demonstrate that it is possible to convey privacy-sensitive data collection policies using these diagrams.

The rest of this paper is structured as follows. First, we review related work in two areas, software agreements and the

design of wordless diagrams. We then describe our iterative design process for designing narrative pictograms to accompany publicly distributed research software. Next, we present specific examples of the narrative pictograms developed and the composition rules used for their creation. Finally, convey results of an evaluation of their comprehensibility, and conclude with avenues for future research.

BACKGROUND

As mentioned in the Introduction, software agreements tend to be lengthy documents containing sophisticated language. In this section, we first review past work aimed at mitigating these issues. We then show how global audiences introduce localization needs that can be costly to address. Finally, we show how wordless diagrams are a promising strategy for addressing these various issues.

Compelling Users to Read Agreements

Recent research suggests that less than 2% of the population reads EULAs when installing software [6]. Similarly, few individuals appear to read website’s privacy policies [10]. Users often see the agreement process as something of low utility that does not contribute to the goal of using the software [3,4]. Recognizing these issues, a number of methods have been developed to increase the chance that individuals actually take the time to read information presented, or to improve comprehension of that content.

Work by Good *et al.* [6] experimentally demonstrated that summarizations of EULAs are an effective way to increase the chance that some terms of use are read and acted upon. In a similar vein, Kelly *et al* [11] took inspiration from nutrition labels in designing a summary of privacy information for P3P (Platform for Privacy Preferences)-compliant websites. These “privacy labels” were found to improve users’ ability to find information in the agreements.

Other approaches purposefully disrupt the typical flow of interaction to focus users’ attention on pertinent information. For example, when showing a license agreement, software installers often disable a “Next” button until the user selects a checkbox or radio button to signify agreement. This practice is sometimes augmented by requiring the user to fully scroll through the information presented. However, we are unaware of work that empirically examines the effectiveness of these particular techniques. In contrast, research has demonstrated that rearranging user interface elements is an effective means to increase the chance that users take time to understand informational messages [2].

Localization and Software Agreements

The techniques described above assume that the individual has the ability to comprehend the information if they take the time to read it. However, not all software is localized for every language, meaning that users may encounter software agreements in languages other than their native tongue. 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. For example, volunteer-driven open source software projects and researchers conducting studies via pub-

licly distributed software typically have limited resources to address these issues. We faced precisely these issues, as we describe.

In a separate research effort, ingimp [14], we publicly released a modified version of an open source bitmap graphics editor (the GIMP [5]) that collects software usage data. Over time, we have amassed a user base that includes users representing nearly a dozen different locales. Because this research involves human subjects, and because the software collects usage data, we include 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 users whose native language is not English. Ideally, we would like all of our users to take the time to read and understand the consent agreement, but are limited in our ability to localize the content for our prospective user base. Thus, there is a real challenge for gaining (truly) informed consent from users in these situations.

The localization issues we encountered are not unique to our project. As mentioned, open source projects face similar challenges, as do many companies and researchers. Recognizing these needs, we note that one promising approach is to develop a more “universal” representation of the software agreement, with fewer localization requirements. The success of wordless diagrams and pictograms in contexts such as public transportation and instructional manuals suggests the promise of this particular approach.

Wordless Diagrams

The emergence of a highly mobile, global community over the past century has led to interest in visual communications that do not rely on written language. Numerous efforts have been made to create “universal” symbols or images across a range of domains, including informational signs, road signs, and manuals [7]. While these wordless diagrams are not immune to localization needs (many examples exist of symbols meaning different things in different cultures [8]), a single, well designed diagram can effectively communicate across cultures without words. These types of wordless illustrations are best represented by two classes of pictorial communication: pictograms and instructional diagrams.

Pictograms

Some of the initial efforts in creating signs designed for an international audience can be seen in the 1972 Munich Olympics signage created by Otl Aicher [12]. The symbols are a highly simplified form of visual communication that reduces the various Olympic sports to iconic representations. While these graphics are quite minimalist, they nonetheless bear a clear resemblance to the activity which they represent. That is, they are not abstract symbols lacking any resemblance to the objects and activities depicted.

In 1974, the US Department of Transportation (DOT) hired the American Institute of Graphic Arts to design a set of transportation symbols to standardize the means by which services were communicated to travelers [12]. These now appear in airports, train stations, and other locations. Like the Olympic icons of 1972, the symbols are highly simpli-

fied representations of the objects they depict, though there is some use of abstract symbols, such as a question mark to represent information.

The symbols for the Olympics and the US DOT are best classified as pictograms [1]: compact graphical symbols representing a single concept or object. Their 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. These pictograms provided one inspiration for our narrative pictograms work, described later.

Instructional Manuals

Commercial products often have a global market and benefit from standardized visual communications in the design of their interfaces and documentation. In recent years, it has become commonplace to use wordless diagrams as components of instructional manuals. One of the more successful efforts in this vein is the work of Patrick Hofmann in designing HP's wordless "welcome mats" for product setup. These welcome mats are one of the first items unpacked when opening a product package and are intended to illustrate how one assembles and uses the product. The wordless welcome mats ultimately eliminated the need for the production of 16 separate, localized manuals, saving HP hundreds of thousands of dollars in printing costs [8]. Furthermore, testing revealed that these wordless diagrams increased the usability of the documentation.

The objects depicted in the welcome mats necessarily have more detail than pictograms, but in contrast to technical illustrations, they often omit some details when representing objects. The slightly simplified forms of the objects help focus attention and clarify intent. Thus, the diagrams are neither photorealistic nor oversimplified representations of the objects.

The illustrated sequence of steps in the welcome mats explicitly establishes a concrete context with which to understand the individual elements in the diagrams. Pictograms lack this context within their graphics. Instead, their placement in space provides the necessary context. For example, a question mark (?) gains clarity of meaning in an airport as a place where information can be found. Placing that same symbol on the ground or on a wall, with no help desk in sight, would render the symbol ambiguous and uninformative.

As we will show, these basic rules—simplified, yet literal depictions of objects and actions, and the use of a concrete scenario to contextualize the drawings and their elements—formed the basis for our work in designing narrative pictograms. (The name of the technique—*narrative pictograms*—directly reflects these basic principles of using simplified depictions of objects and actions, within an overarching scenario, to communicate concepts.) Through an iterative process, we refined these basic rules and discovered the need to augment them with a set of additional diagram conventions. In the section that follows, we outline how we developed narrative pictograms, then present the technique in detail.

NARRATIVE PICTOGRAMS OF DATA COLLECTION

In this section, we first describe the specific communication goals that motivated the creation of narrative pictograms, then describe the lessons learned from an iterative design process employed to develop and hone these diagrams. We then present the full technique and its corresponding composition rules with illustrative examples.

Goals of the Illustrations

This work began with the need to create a consent agreement to accompany a research application that collects software usage data (e.g., the commands people use). Our 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.
- Reduce the need to localize the diagrams themselves. More specifically, the diagrams should contain minimal text and, to the extent possible, one should be able to read the diagrams, or learn how to read the diagrams, without the need for auxiliary aids.

To achieve these goals, we reviewed prior work in wordless diagrams (e.g., pictograms and instruction manuals, as previously discussed), then engaged in an iterative design process, discussed next.

Design Process and Formative Evaluation

Our 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*. We used these techniques to create a series of paper-based illustrations conveying the primary concepts of our software agreement, such as the software's functionality and purpose, and the data it collects.

The diagrams were iteratively developed using a hybrid evaluation methodology incorporating the ISO 9186-1 test method for graphical symbols [9], a thinkaloud 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 the subject is presented with a context in which they might see the sign or symbol. They are then shown a symbol and asked to interpret its meaning in that context. The judgment test informs the subject what the intent of the symbol is, then asks them to estimate how many people are likely to understand it 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 our iterative testing procedure, we presented users with paper-based prototypes and indicated that the illustrations would be seen when starting our software for the first time. We then asked subjects to interpret the illustrations for us, thinking aloud as they did so. We took notes, recorded their

comments, and observed their behavior as they studied the illustrations. We did not provide any assistance as they interpreted the illustrations. Instead, we instructed them to say “I don’t know” if they could not understand an illustration. When significant communication failures occurred with the illustrations, we sketched new, alternative illustrations and asked the participant to interpret the new designs.

After each subject was done interpreting the illustrations, we described the illustrations’ meanings and asked the subject 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.

For this study, we recruited 14 subjects, 7 native English speakers and 7 non-native English speakers; 4 were female and 10 were male. All subjects were university students, with 11 undergraduates and 3 graduate students. We describe initial results that suggest the effectiveness of the final set of diagrams produced, saving discussion of the lessons learned until we present the full set of composition rules.

Initial Results

For the first four subjects, our initial designs failed to effectively convey the intended concepts of the software agreements. These designs underwent significant modifications in the presence of the subjects and in between trials. After these first four subjects, we converged on a set of designs and a uniform set of composition rules. These designs were iteratively refined with the last 10 subjects of this study. These last 10 subjects were each able to correctly interpret all but 1 or 2 of the resulting 20 diagrams, with no one diagram consistently misinterpreted. We now present this final set of diagrams and the composition rules used to produce them.

Narrative Pictograms of Data Collection

The narrative pictograms we developed are segmented into four collections of diagrams, each serving a different purpose in describing the data collection policies of the instrumented software:

1. *The Functional Overview*. This illustration provides a functional overview of how the software works.
2. *Environmental Data Collection*. These illustrations 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 illustrations 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 illustration that conveys a narrative. Each set of diagrams is described next.

Functional Overview

The first illustration displayed in our narrative pictograms is the functional overview (Figure 2). The functional overview depicts a basic interaction sequence from the time the application is started to when it is closed. This basic use of the application is complemented with graphs suggesting data collection: as the user depicted in the illustration 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 teaches the user how to read the diagrams by presenting a concrete, simplified series of familiar actions. This sequence establishes a context in which more novel elements can be introduced. In this case, the depiction of basic software use (e.g., start application, edit a document, quit) is augmented with an easily noticed anomaly: a graph. The graph changes based on the character’s actions, with actions and consequences highlighted in red to help the reader understand what the graph represents. Pictures of the graph’s source data are used as labels to improve one to notice the correspondence between the character’s actions and the graphs. Finally, when the application is closed, the graph is shown being transmitted to a website to indicate that the data are sent to a remote site.

The conventions and visual elements established in this overview are repeatedly used in subsequent diagrams and, when necessary, purposefully violated to draw attention to important information. This basic strategy permeates the diagrams’ design: A “familiar” scenario is presented to the viewer with a select set of novel elements introduced into the 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.

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.

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.

In contrast to the functional overview, which utilized a narrative composed of a string of related scenes, this series of illustrations 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 1 and 3). In this series, the data collected are not data that result from specific user

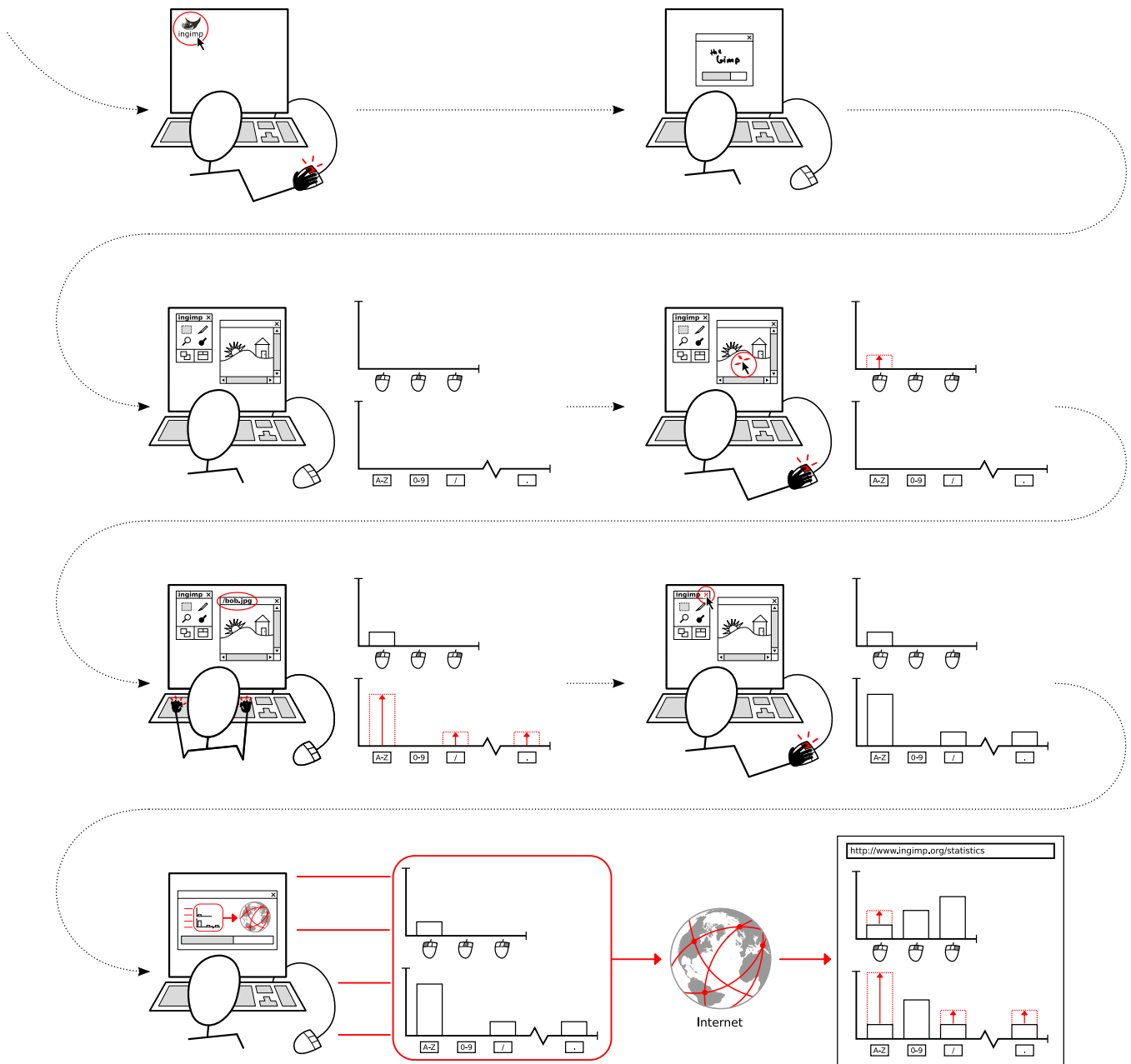


Figure 2. The functional overview provided to describe the fact that the gimp software collects usage data that is sent to a website

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. For example, icons for popular operating systems are used to indicate the type of operating system is logged (Figure 1). Similarly, common screen resolutions (e.g., 1024x768) are used as labels to indicate the capture of screen resolution (Figure 3).

The order of illustrations within this series follows our strategy of showing objects and concepts that are most likely to be familiar and known to users. Thus, we first show a scene depicting the logging of the operating system (which uses familiar icons), followed by time zone data using a shaded

world map. The collection of locale information, a slightly more abstract concept, is shown after these more concrete concepts, at a point when the user is more likely to understand the purpose of these individual diagrams. We represent locale data using countries' flags associated with the keyboard.

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 are shown, along appropriate data collection graphs.

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, we alter a number of elements in the previously established visual motif. As an example, the keyboard data collection diagram (Figure 4) 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 graph suggests that all of the keyboard activity is summarized in one dimension only, since individual bars for each key are not shown. Our formative study 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 a process is useful, but we also wanted to make it explicit that certain types of data were not collected. As a result, some of the diagrams in this set show what data are *not* collected. As an example, the diagram in Figure 5 is shown immediately after the diagram in Figure 4 to clarify that the actual, typed text is not recorded. For this class of diagram, we 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.

Privacy Sensitive Data Collection

The previous two diagram sets employ the convention of showing data being collected and aggregated with other data. This aggregation suggests a level of anonymity to the data collection process. However, there is some data collected by this software that is recorded without any summarization or anonymization applied. In particular, users can describe their planned use of the software by entering keywords in a special box shown at start-up. These keywords are recorded directly in the log file without modification, and are publicly accessible (as is all other data collected by this software). The names of custom scripts are also recorded and made available with the collected dataset, which may constitute a privacy concern for some individuals.

To highlight the sensitive nature of these data collection processes, we break a previously established convention—the large arrow indicating data collection—and instead show data being taken directly from the user’s screen and placed directly in the public website’s dataset (Figure 6). This direct correspondence between information on the character’s screen and the public website underscores the privacy risks associated with these particular actions.

Having described the four sets of narrative pictograms developed for this specific application, we now describe the composition rules employed to create these diagrams.

Composition Rules

Composition rules were developed in tandem with the diagrams during our design phase. The composition rules are as follows:

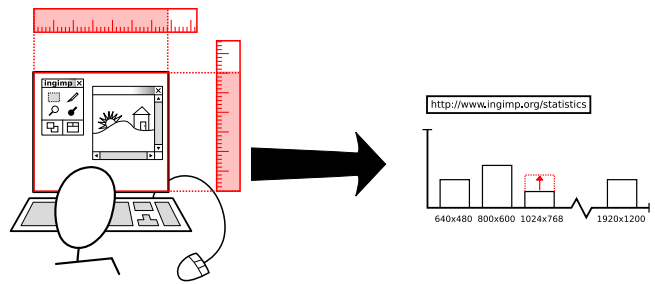


Figure 3. A diagram in the “environmental data collection” set, depicting the capture of screen resolution. Labels with common screen resolutions reinforce that monitor resolutions are recorded.

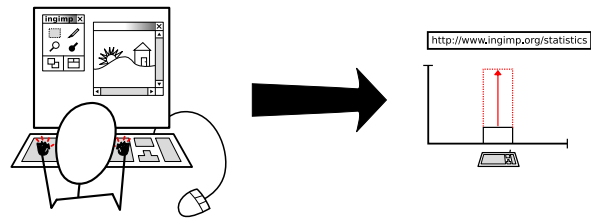


Figure 4. A diagram in the “interaction data collection” set, depicting the fact that keyboard activity is recorded. By using a single bar in the graph, most viewers correctly assume that individual keystrokes are not recorded.

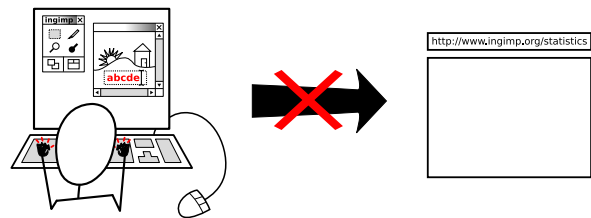


Figure 5. Previously established conventions for data collection (the arrow and graph) are broken to display an example of data not collected (specific keys/text typed).

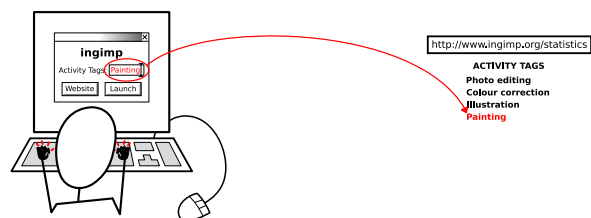


Figure 6. This diagram breaks conventions established in previous diagrams (the large arrow) to highlight data that are recorded without any summarization or anonymization.

- Employ literal contexts and simplified representations of objects, rather than abstract symbols
- Use repetition to establish patterns and conventions
- Break established patterns and conventions to draw attention to details
- Avoid conventions of manuals to avoid creating the impression that the illustrations are instructional, rather than informational

These composition rules, in and of themselves, are not necessarily novel – graphic designers, technical illustrators, and others in the visual arts 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. We expand on each of these rules next.

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 so the intent of these unfamiliar and unexpected illustrations are not misinterpreted. Accordingly, the diagrams employ narratives with familiar objects and actions to establish a context. These narratives, with their simplified representations of concrete objects, also hint at the communicative purposes of the diagrams. Just as importantly, they hint at what the diagrams are *not* (such as tutorials or advertisements). As mentioned, the functional overview described above (Figure 2) establishes this context. With this context established, more abstract concepts can be introduced to communicate more sophisticated material.

Use Repetition to Establish Patterns and Conventions

Hofmann, in writing about methods to create cross-cultural informational illustrations [8], suggests starting instructions with a base illustration that is gradually changed. The repetitive use of the same base illustration, with only minor changes between steps, enables the reader to more easily establish what is changing. This same convention is used throughout the narrative pictograms.

Repetition in our illustrations works in several complementary ways. First, by repeating elements without change, one indicates that the repeated element is not very significant. At the same time, the elements that *do* change are more easily identified and thereby granted special meaning. 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 our formative evaluation, subjects often exhibited this behavior of understanding a particular, repeated convention after viewing several instances of it in different contexts. They could then go back and understand previous diagrams.

Breaking 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.

A pervasive example of breaking a convention to draw attention is the use of colour. The diagrams are largely monochromatic. However, this property is routinely broken by the introduction of red strokes to draw attention to actions or data being collected; this predictable use of red then confers specific meaning to this use of colour.

The diagrams also break conventions to highlight notable exceptions of what data are, and are not, collected. For example, most of the diagrams depict data being collected using an arrow pointing to the data collection graph. The diagram in Figure 5 breaks that convention with a large “X” over the arrow and a blank space in place of the graph indicating the absence of data collection.

Avoid Conventions of Manuals

Our testing of the diagrams revealed that the conventions of instructional illustrations (e.g., assembly instructions for products) could not be blindly applied. The presence of symbols common to manuals (e.g., numbers and prohibition signs) immediately led viewers to consider the illustrations to be a user manual.

As an example, our earliest illustrations used numbers in a variety of contexts to connote sequences. We originally numbered the sequences of the functional overview to emphasize the order in which it should be read (since reading direction is not universal). However, these numberings suggested that the diagrams were trying to instruct the viewer on how to accomplish some task. Consequently, we eliminated these numbers and relied on other strategies to suggest sequences (for example, the use of arrows, see Figure 2).

When originally designing illustrations to describe what data is not collected, we used the strategy of placing a prohibition sign (a circle with a line through it) over an illustration showing data collection. Like the numbering, subjects felt they were being instructed, but in this case, instructed *not* to do something. To avoid this connotation, we instead place a red “X” over the arrow to indicate that transmission does not occur (Figure 5). By using an “X” instead of the prohibition sign, and by placing it over the arrow rather than the activity, we avoid the suggestion that we are prohibiting certain user actions.

As mentioned, these composition rules arose out of our initial formative evaluation. We now describe an experimental study that more carefully examines how well these diagrams are understood.

EXPERIMENTAL EVALUATION

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

Experimental Design

The experimental design of this experiment is based on the aforementioned ISO 9186-1 method for testing graphical symbols [9], slightly altered for our needs (described below).

We employed two conditions: a “diagrams with text” condition and a “diagrams only” condition. The “diagrams with text condition” places the illustrations inline 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, subjects 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, we have made the following modifications to the ISO protocol to better suit the needs of this research:

1. We 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. We 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, we explicitly allow participants to go back and change answers if they wish. However, we ask them to cross out old answers with a single line so that we can better document their process of understanding the diagrams.
4. The ISO test recommends only testing 15 symbols per participant. Since the diagrams are meant to be read as a whole, we preferred not to remove any from the test. The complete set consists of 20 diagrams, which we did not consider excessive.

Method

After obtaining written consent, participants were given written instructions and allowed to ask questions about the study. If they were in the “diagrams with text” condition, they were then provided with two booklets: a booklet containing a text agreement with inline illustrations labelled “Diagram A”, “Diagram B”, etc., and the answer booklet. The answer booklet contained the question “What do you think this diagram means?” for each diagram in the agreement. Participants in the “diagrams only” condition were 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, participants filled out a short post-task questionnaire. This questionnaire asked participants 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 subjects found particularly difficult to comprehend, and why.

Performance Measures

Each response on the comprehension test was assigned a score as specified by ISO 9186-1: correct (score 1), incorrect (2), “don’t know” (3), and 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 is sent when it actually is, users’ privacy may be inadvertently affected.

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).

We calculate *total 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 answers in which the subject thought some data was not being collected when it actually was (or vice versa).

Participants

20 subjects were recruited in a university setting. Subjects 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 [11].) While one motivation for this work was to address localization concerns associated with software agreements, we wished to first establish how well these diagrams could communicate the abstract concepts of software agreements with a relatively homogenous set of subjects, 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.

RESULTS

Figure 7 shows histograms of total score by condition. Interrater reliability (percentage agreement) for the scoring was 82.8%. Recall that a perfect total score is 20.

Mean scores were 10.0 (SD=5.77) and 13.6 (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.

In terms of the questionnaire data, mean and median “informativeness” ratings were 3.4 and 4, respectively (SD=0.94), out of 5 (where 5 is the “most informative”). Visual appeal was rated a mean of 3.26 and a median of 3 (SD=0.73), out of 5. Finally, comprehensibility was rated a mean of 3.25 and median of 3.5 (SD=0.85), out of 5.

DISCUSSION

The primary goal of this experiment was to understand whether subjects could comprehend the narrative pictograms, with or without text. The condition without the software agreement text available provided the most rigorous test of these diagrams because subjects needed to correctly infer that the diagrams depict the data collection policies of a software application. Since these diagrams are the first to convey such concepts, 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.

Given these communication challenges, the results of the study are very promising. In both conditions, most subjects 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. In short, these techniques appear to be an effective means at conveying rather abstract concepts without significant use of words.

Difficulties with Particular Diagrams

As we noted, on average, participants in the “diagrams only” condition did not perform as well as those in the condition with the agreement text. However, this difference in performance can be attributed to a few problematic diagrams, as can be seen in Figure 8. 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.

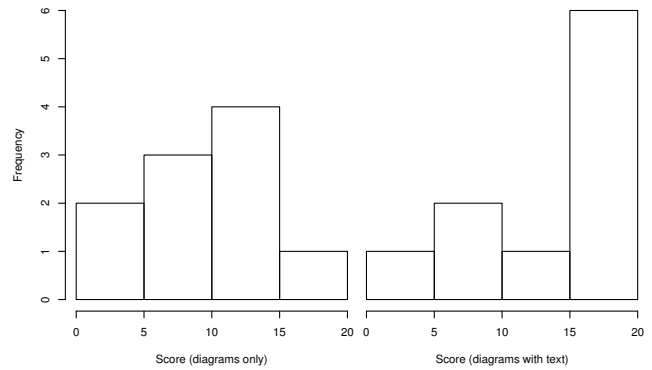


Figure 7. Histograms of total score, by condition

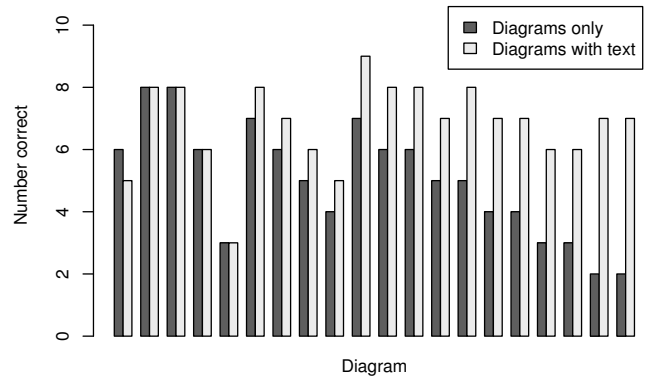


Figure 8. Number of subjects for whom correctly interpreted each diagram (ordered by the difference between conditions).

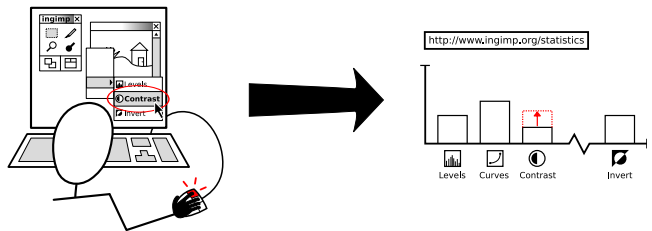


Figure 9. Diagram depicting collection of command names; many users thought this referred only to use of contrast.

The two most problematic diagrams both suffered from a similar problem that was uncovered during our post-test interviews. Consider Figure 9: in this diagram, it was our intention to convey that use of menu commands was recorded. In our interviews, however, users 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 1). It may be that singling out a very specific command unique to this domain, rather than a generic command (such as “Undo”) 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. Consider Figure 10, which is intended to depict the collection of image histogram data. These histograms are commonly employed by professional users of photo manipulation applications. However, the histograms may not be understood by novices to the domain. In the condition with the agreement text, there is a passage that describes that image histograms are generated from frequency counts of pixel values. Consequently, participants in that condition were able to acquire the necessary knowledge to interpret the diagram. In contrast, participants in the other condition expressed their confusion in both answers and the post-task interview about the meaning of the histograms. This finding suggests that the more complex, domain-specific concepts may require additional pictorial context to be understood by relative newcomers to the domain.

Effect of Functional Overview

As previously described, the functional overview was designed to teach users how to read the diagrams. Thus, one question we wished to answer was whether these overviews were accomplishing this goal. Our data suggest that understanding the functional overview is key to understanding subsequent diagrams. The median score for 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 the functional overview served an important function in helping participants understand subsequent diagrams.

Limitations

This study provides evidence that the abstract concepts of data collection can be effectively conveyed using wordless diagrams. However, further work is required in two areas. First, research is needed to understand how well these concepts can be applied to other types of software agreements, such as EULAs. Second, our evaluation only considered native English speakers. While the condition without the software agreement shows comparable results, 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.

CONCLUSION AND FUTURE WORK

This paper has presented narrative pictograms, an approach for depicting software agreements using illustrations only. A study demonstrates that these diagrams can be effectively comprehended with or without accompanying text. These findings have implications for the design of materials intended to convey abstract concepts to audiences who may not be able to fully read a document, such as user interfaces for illiterate users or those in a foreign locale. The lessons learned and the composition rules derived from this work provide a foundation for future exploration of this problem space.

REFERENCES

1. Abdulla, R. and Hubner, R. *Pictograms, Icons and Signs*. Thames & Hudson Ltd. 2006.
2. Brustoloni, J. C. and Villamarín-Salomón, R. 2007. Improving

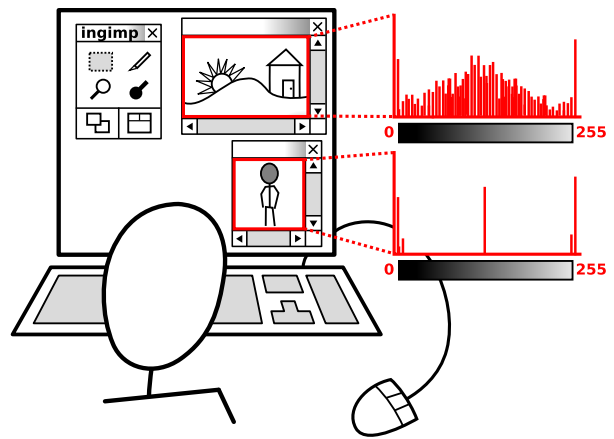


Figure 10. Cropped view of a diagram showing the collection of image histogram data. Many participants had difficulty with this illustration due to a lack of domain knowledge.

- security decisions with polymorphic and audited dialogs. *Proc. of SOUPS '07*, vol. 229. ACM Press, New York, NY, 76-85.
3. Friedman, B., Lin, P., & Miller, J. K. (2005). Informed consent by design. In L. Cranor and S. Garfinkel (Eds.), *Designing secure systems that people can use*. Cambridge, MA: O'Reilly and Associates. 495-521
4. Friedman, B., Smith, I. E., Kahn, P. H., Jr., Consolvo, S., & Selawski, J. (2006). Development of a privacy addendum for open source licenses: Value Sensitive Design in industry. *Proc. of Ubicomp '06*. New York: Springer-Verlag. 194-211.
5. GIMP. <http://www.gimp.org>
6. Good, N. S., Grossklags, J., Mulligan, D. K., and Konstan, J. A. 2007. Noticing notice: a large-scale experiment on the timing of software license agreements. *Proc. of CHI '07*. ACM Press, New York, NY. 607-616.
7. Hofmann, P. Away with words! How to create wordless diagrams. *Proc. of IPCC '98*. IEEE International. Vol 2, 437-438.
8. Hofmann, P. Localising and Internationalising Graphics and Visual Information Commentary. In *IEEE Transactions on Professional Communication*. June 2007, vol 50(2), 91-92.
9. ISO 9186-1. http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=41090
10. Jensen, C. and Potts, C. 2004. Privacy policies as decision-making tools: an evaluation of online privacy notices. In *Proc. of CHI '04*. ACM Press, New York, NY, 471-478.
11. Kelley, P. G., Bresee, J., Cranor, L. F., and Reeder, R. W. A "nutrition label" for privacy. *Proc. of SOUPS '09*. ACM Press, New York, NY. 607-616.
12. Meggs, P.B. and Purvis, A.W. *Meggs' History of Graphic Design*. Wiley. 2005.
13. Medhi, I., Sagar, A., and Toyama, K. Text-free user interfaces for illiterate and semilliterate users. *Information Technologies and International Development*. (2007). 4, 37-50.
14. Terry, M., Kay, M., Van Vugt, B., Slack, B., and Park, T. In-gimp: introducing instrumentation to an end-user open source application. *Proc. CHI '08*. ACM, 607-616.