

Contents

Abstract	1
Introduction	2
Background and Prior Work	3
Intelligent User Interfaces	3
Scenario Based Design	4
Bayesian Inference	4
Method	5
Develop narratives	5
Assign probability statements	5
Visualization	6
Integration with real usage data	7
Data assimilation	7
Case Study	8
Introduction	8
Assigning Probabilities	8
Generating PGCs and Visualizing Them	9
Assessing real usage vs. expected usage	12
Assimilating real usage data into pSBD scenarios	15
Discussion	19
Advantages of pSBD over traditional SBD	19
limitations	19
Future research goals	19
Conclusion	19

Abstract

Constructing hypothetical scenarios and user narratives is a common technique for communicating the envisioned user experience (UX) of a tool. Using this approach, application developers, designers, and stakeholders rapidly build stories of expected use of a tool that concretize the intended UX. This design strategy is effective, cheap, flexible, and simple to implement. However, most narrative scenarios informal textual or visual sketches, lacking a mathematical basis suitable for statistical testing and visualization of usage patterns. In this paper, I introduce a new method, probabilistic scenario-based design (SBD) for underpinning narrative scenarios with probability statements. Using this technique, tool developers modify traditional scenarios to include formal statistical statements that describe the interactivity and design of each interface component. This approach offers at least three advantages that complement traditional scenario-based design approaches. First, it offers the potential for novel visualizations of usage patterns, both real and hypothetical. These visualizations provide alternate and more concrete view of the tool's intended UX. Once an early release of the interface is released, pSBD can enable formal statistical testing of whether intended usage patterns conform with patterns expected by the UX designers. Finally, pSBD provides a mechanism of improving interface interactivity and design that statistically incorporates both real usage data and designer intuition by specifying probability distributions that can be used in data assimilation routines. This process would allow the interface to adapt to user needs throughout its lifetime while using much less data than traditional AI-based personalization algorithms. The process of pSBD and the derived visualizations and analyses are illustrated with a synthetic example based on a web-based interactive mapping

tool. pSBD is designed to be operationalized as part of the an existing user centered design or scenario based design workflow.

Introduction

Powerful, easy-to-use programming frameworks and widespread consumer access to low-cost, high-speed, internet-enabled computing devices have resulted in a host of highly interactive, richly-featured applications on the Web. These apps rely on a two-way communication model that encourages the production of user-generated content and social interaction among users. As cloud services gain popularity, these web-based app distribution is becoming increasingly common. Many common tasks, including email, word processing, and data analysis are increasingly done ‘in the cloud’, facilitated through software-as-a-service (SaaS) that enables synchronous information retrieval and preference persistence across devices to create a seamless user experience [1,2]. Often, these applications serve multiple user groups, with different interests, motivation, or skills [3]. While it is possible to simultaneously support multiple user groups through carefully engineered design decision, artificial intelligence (AI) and machine learning (ML) algorithms are often applied to recognize user actions, identify likely sequences of interactions, recommend suggested products, and adapt the user interface (UI) to likely preferences to maintain a positive user experience for all users [4–7]. Typically, these approaches improve the tool’s UX by exposing the user to less information and visual stimuli, allowing cognitive function to become more focused on specific tasks. While AI and ML personalization and customization approaches are widely-used and efficient, particularly for large applications with many users, these techniques have two important drawbacks. First, because they typically model the user based on the sequence of interactions taken by a user during an interaction session (i.e., the clickstream) these algorithms often require large amounts of data from many user sessions [8]. Such large amounts of real usage data may not be available for early stage applications still under active development. Second, AI approaches do not typically account for expected usage as envisioned by the application developers and domain experts; rather, user models are generally constructed from the application’s usage data [9]. However, in many cases, application designers, developers, domain experts, and other stakeholder groups go to great lengths to characterize the user and their expected user experience (UX) with the interface during a user centered design (UCD) process.

Specifically, application developers often work with a team of designers and domain experts to develop narrative scenarios about the expected UX of a tool in a process known as scenario based design (SBD) [10,11]. Scenarios provide a common vocabulary for communication between stakeholders by describing how a user will interact with a tool [12]. SBD is used in a wide variety of fields and is not limited to development of software systems [13,14]. Within software development, it is often used to inform the requirements of a proposed tool during their negotiation and to envision the intended use of the software by multiple user groups. Scenarios provide a flexible and cheap method of communicating concrete use cases and have been shown to improve utility and usefulness of the resulting tool [12]. By design, the scenarios produced during SBD are informal [10,15]. If a scenario is formalized, it typically changes in both form and purpose from an illustrative device for communication to a rigorous document describing the functional requirements of the proposed tool [13]. Requirements documents are often more concerned with the feasibility of the tool than the utility and usability by the envisioned user. However, a framework for formalizing scenarios while maintaining the flexibility and central focus on the user would improve communication among tool builders by ground discussion with statistical and visual evidence of differences in UX among scenarios. Moreover, formal scenarios could be used to inform personalization algorithms, resulting in more efficient customization.

In the present study, I develop a method, probabilistic scenario based design (pSBD), for creating formal user scenarios that maintain the connection to the envisioned users by enhancing traditional SBD scenarios with probability distributions. In this method, each scenario consists of probability distributions that describe the interactivity and design of each proposed interface component, function, or logical element in the envisioned interface. In a simple case, these distributions can simply represent the probability that the actor in the scenario will use a specific component. In more complex cases, distributions can be specified to describe the dimensions of a user-configurable layout or the geographic center of a map component, for example.

By introducing a probability model, pSBD facilitates improved inference of user interaction patterns, even

before the interface has undergone extensive use. Rather than informally constructed narratives that rely on relaxed language, pSBD scenarios become suitable for formal statistical testing of user interaction patterns, distinction between real usage and envisioned usage, and prediction of success of future interfaces at a component level. Because the probability model is developed during the planning and development stages along with the UI, AI customization systems have data to work with immediately that formally account for the developers’ intuition for how user groups will interact with a tool.

In the present paper, I describe and illustrated three potential advantages of a pSBD design approach. First, this method would allow for novel visualizations techniques capable of showing, in a concise manner, how UX would differ among target user groups. Second, pSBD enables the formal statistical testing of interaction with an interface. Specifically, pSBD allows the testing of if real usage patterns conform with the expectations of the development team. This could be particularly useful during design interactions to improve the interface to meet expectations. Finally, pSBD is amenable to Bayesian data assimilation, where observations of actual usage can be integrated as the application is deployed. The interface can then adapt, in an intelligent way, to reflect both the envisioned usage of the designer and the real world usage patterns of the target audience. While not as sophisticated as many of the AI personalization systems currently on the market pSBD is flexible, inexpensive, and requires relatively little data.

The balance of this paper proceeds as follows. First, I outline contemporary techniques in user interface personalization and recommendation, review pertinent literature on SBD, and discuss existing methods Bayesian inference and prediction in the design and function of user interfaces. Second, I describe the method for enhancing narrative scenarios from SBD with probability distributions. Third, I illustrate the derived statistics, visualizations, and prediction possible with the new method using a synthetic case study.

Background and Prior Work

Intelligent User Interfaces

Modern web apps help us choose what music to listen to, which roads to drive on, which friends to talk to, and what products to buy. Well crafted UIs and efficient UXs instill positive feelings of success and competency, receding into the background as users focus on their work, exploration, or pleasure [???]. In many sophisticated apps, AI and user modeling can play important roles in helping the interface to disappear by intelligently limiting the content to which the user is exposed. By processing large amounts of historical usage data and building statistical models of user preference, AI systems can limit consumer exposure to only items that the user is likely to be interested in or partially execute tasks the user is likely to wish to accomplish. These systems are highly profitable, particularly in ecommerce – by limiting a customer’s exposure to a large catalogue of goods, the system encourages consumer focus on a profitable task (buying an item) rather than an enabling task (choosing which item to purchase).

At the heart of many AI personalization algorithms are user models that describe and quantify the traits of the application users. The construction of user models is a focus of active research in contemporary human-computer interaction study, and is important in recommendation systems, social computing, intelligent search algorithms, and adaptive interfaces[16]. Specifically, user modeling involves inferring unobservable information about the user, such as his or her thought processes, intentions, etc, from observable information, such as his/her actions [17,18]. While user modeling need not be quantitative, statistical user models allow an application to anticipate behavior, including goals, actions, and preferences [17]. Models can be constructed as top-down, in which a theoretical understanding of user preferences are prescribed by the model developer, or bottom-up in which associations between sequences of user actions are learned organically. However, most contemporary user modeling approaches can be seen as hybrids that combine aspects of these two categories [19].

In addition to generating personalized recommendations, user models can be used to underpin adaptive user interfaces (AUIs) which adapt in look, feel, and interactivity to that most likely to be preferable to a new user, based on their series of actions. AUIs can provide just-in-time assistance by predicting the user’s most

likely actions and then performing one or more of those actions on the user's behalf [20]. AUIs can adapt to the needs of different users for a variety of tasks [6]. AUIs have commonly been implemented in the context of intelligent tutoring and online education systems. In these cases, the system must adapt to a user's learning style and pace, and a user model is used in tracking how the user is progressing towards an educational goal [21,22]. Typically, AUIs work by identifying membership in a user group based on a series of interaction events, which requires tracking all user interaction sequences [20]. Moreover, the more adaptive an interface, the larger the amount and quality of feedback it needs from users to determine if it is adapting in a favorable way [6].

Multi-level interfaces are an important target for AI assistance. Multi-level interfaces are specifically designed to support multiple tasks of increasing difficulty for users of different skill levels, motivations, or expertise. For example, a novice user could receive a more detailed and longer sequence of dialog steps than an expert familiar with the system [23]. Such systems would identify the appropriate user model, then assemble the user interface components most suitable for the identified context of use [23].

Scenario Based Design

SBD scenarios are flexible, low-cost, and evocative narratives of a designer's envisioned use for a tool. SBD has been applied to a variety of contexts; however, while the details may differ between implementations of SBD, all are aimed at concretely describing the use of a tool early in its development [10]. These narratives generally contain four elements: (1) a setting, (2) an actor with personal motivation and skills, (3) assumptions and context about the actors and their objectives, and (4) sequences of actions and events in which the actors manipulate the tools and objects surrounding them [11,13,15]. Typically, actors execute a sequence of actions and events that lead to some outcome [10]. Scenarios can be expressed in a variety of ways, such as through text, videos, mockups, or storyboards.

It is important to note that scenarios, by design, are informal and do not attempt to outline the functional requirements of the interface under development. Scenarios serve as a sketch of the envisioned UX of the tool, capturing the essence of future uses of the tool [15], evoking reflection in the context of design [11]. Rather than enumerating requirements, they can be used as a communication tool to ground conversations about the design and interactivity of the application. Scenarios can facilitate brainstorming between development team members, inform UI design choices, and act as a guide when developing formal requirements [12,15]. Like other user-centered design approaches, scenarios maintain a central focus on the target user of the tool, and are thereby able to effectively communicate tradeoffs between design decisions for those specific user groups [11]. Moreover, the products created during SBD can be used as design rationale in later phases of the design cycle.

While scenarios provide a clear communication mechanism and concrete products on which to guide future development and design activities, they typically lack a mathematical or statistical basis. A previous attempt to quantify a scenario with a preference matrix was described in [16], but focuses primarily on the affinities of the user, rather than the attributes of the proposed interface.

Bayesian Inference

Bayesian approaches to knowledge representation are common in many fields, including in user modeling and adaptive user interface design. Bayesian statistical inference involves drawing concrete conclusions about unobservable qualities of a system, in the presence of uncertainty. These claims are represented in terms of probability statements, conditional on the analyst's belief of the true nature of the system and the observed dataset [???]. In classical statistics, it is difficult to take into account prior knowledge when testing hypotheses and statistical experiments demand large sample sizes. Moreover, it is difficult to use the results from one experiment to predict the outcome of a future experiment [???]. The Bayesian paradigm provides a coherent approach for combining information from new and existing information in a probabilistic framework [24]. Bayesian inference allows the a probability model to be fit to a new dataset, and the results summarized by probability distributions on both the parameters of the model and unobserved, or unobservable, latent

qualities of a system [???]. Bayesian inference is often used in the context of probabilistic forecasting or data assimilation, where an existing numerical or physical model is used in conjunction with a set of observations to update knowledge about the true state of a system [24].

Bayesian belief networks (BBNs) are common technique for representing user models in AI-based personalization systems. Bayesian belief networks are directed acyclic graphs (DAGs) in which each node represents a conditional probability of a particular event’s occurrence. The probability that an event will occur can be estimate by traversing the graph and calculating the conditional probability at a node, given that all prior events have occurred. BBNs are a powerful structure for representing knowledge and reasoning about future events under conditions of uncertainty [???]. BBN-based user models can take into account a user’s background, actions, and previous search queries when reasoning about what the user’s intention is most likely to be [22,25]. BBNs have been used in a variety of contexts related to user modeling, including in Microsoft Office Assistant [22], educational tools and interactive tutoring applications [21,22], and health-related smartphone apps [26].

BBNs require that the entire state-space is described, a process that can be difficult to accomplish manually. Many early AI interfaces relied on manual specification of all conditional probabilities in the graph, leading to a bottleneck in which humans were the limiting factor in the derivation of new knowledge[17]. In many modern contexts, alternate techniques are used to build state-space graph and calculate conditional probabilities, from formal descriptions of expert knowledge [21] or by creating them from the underlying user interaction data [4,27].

Data assimilation involves fusing observations and prior knowledge together is a statistical framework to obtain an estimate of the distribution of the true state of the underlying process [24]. In a Bayesian context, this can be accomplished by using Bayes theorem to obtain a posterior distribution through the use of a likelihood function and prior function. Data assimilation is often used in a spatiotemporal context for numerical weather and climate models [24,28,29], as well as in ecological and phylogenetic modeling [30–32], among other fields.

Method

Underpinning a user-case scenario with probabilities is an iterative process that may involve designers, developers, and stakeholders. While it is not essential, it may be helpful to have low-fidelity wireframes [33] of the intended interface. These wireframes, rough visual outlines of the intended tool, can be used to identify, name, and visualize the components during pSBD negotiations. If statistical testing and data assimilation is desired, an alpha- or beta- release prototype of the application capable of capturing use feedback or user generated configurations, so that real user content can be compared to the developer’s expected use cases.

Develop narratives

The first step in pSBD is to develop one or more clearly defined narrative scenarios for the tool’s UX. These scenarios should contain the four essential elements of traditional SBD, namely actions, context, goals and objects, and actions and events that lead to an outcome [13], to clearly describe the intended usage of the application. These narratives may be articulated in visual or textual form, according to the preference of the development team. For reference on building traditional SBD scenarios, see [10] and [34].

Assign probability statements

The second, and most fundamental task in pSBD, is to enhance narratives with probability statements that capture the intuition and intentions of the application developers. For each scenario, each component of the intended interface may be enhanced with one or more probability distributions that describe its interactivity or design. For example, these statistical descriptions may quantify the probability that a particular interface

component is used, the geographic center of a map component, the likely value of a numeric filter widget, or the width of a configurable panel element. In general, the goal of this step is to characterize the interactivity or style of each component as a random variable, and specify the expected value, variance, and distribution of these variables for the actors in each scenario. In this way, a level of ‘agency’ is allowed within each user model, while characterizing general differences that delineate distinct user models.

An essential piece of this framework is specifying the correct distribution to use in the scenario. While any distribution may be used in this process, several appear particularly useful in the context of pSBD. Because of interface constraint, users are typically not exposed to infinite degrees of freedom within an interface. Thus, most continuous distributions with infinite support may be less appropriate than discrete distributions and distributions with truncated support over smaller intervals. The binomial distribution and its special case the Bernoulli appear to have particularly useful applications in pSBD. The probability that a certain interface component is used in a given scenario may be modeled as a Bernoulli distribution, with a single parameter α that describes the probability of use in the scenario. Component dimensions, such as width, can be modeled as a binomial distribution with 100 trials with an α probability of success, corresponding to the expected width in the scenario. The number of times users invoke a specific feature can be characterized using a Poisson distribution. While more complex distributions, particularly continuous distributions, may be helpful in characterizing complex usage patterns, they have not been tried at the present time.

Because the probability-enhanced scenarios are, like traditional SBD scenarios, a way to facilitate communication between project stakeholders, they should be collectively refined by developers, designers, domain experts, and other stakeholders. During this processes, the expectation of each distribution may be modified according to team consensus. In some cases, new distributions may be proposed to model components in alternate ways, such as modeling an event as a Poisson distribution rather than a binomial distribution. This exercise alone may have significant benefit to the development of the application, as it will provide a common, rigorous ground on which to negotiate envisioned tool use. To facilitate communication, some of the visualizations introduced below may be produced on the fly to dynamically reflect the negotiated product.

Communication and negotiation with the target users can be done in several ways. Informal interviews with key members of each stakeholder may be the best approach. Other social science data collection methods, including formal interviews or focus groups may also be effective. While an online survey would capture the input of more potential users, it may be more effective to limit feedback to a few key stakeholders. The input of real users may be captured through interaction logging and used to refine the distributions during the data assimilation phase discussion below.

Visualization

Once the distributions have been specified and finalized, the resulting user model is available for visualization and inference. An initial analysis step is to generate probability probability-generated configurations (PGCs) by drawing at random from each distribution in the pSBD scenario. Using a scripting language such as R [??] or python [??] independent draws from a probability distribution with given parameters can be easily simulated. A PGC is considered complete when each component distribution has been drawn from. Therefore, a complete PGC represents an interface configuration representative of the corresponding scenario.

A set of complete PGCs can then be used to produce visualizations regarding component use or design among groups or to highlight intra-group variability. Density curves, that plot the density of a parameter distribution, or bar charts, which plot the frequency of boolean outcome, can be effective in communication and negotiation. Principle components analysis and its associated plots can be effective method of succinctly describing multivariate differences among scenarios, as represented by interface components. Finally, wireframes with each component overlaid with the corresponding probability density function, may prove useful in gaining a holistic view of how an interface might look under multiple scenarios. These visualization are developed further in the case study below.

Integration with real usage data

If the application in question does have an early version release ready for user tests (alpha- or beta- stage), the pSBD scenarios and observations of real user interactions and configurations may be used in conjunction to test similarities and refine the existing distributions. Traditional k-means cluster or fuzzy c-means clusters can be helpful in determining whether real usage data is likely to be distinctly different than an existing pSBD scenario. Traditional clustering delineates crisp boundaries among two or more clusters, while fuzzy c-means uses fuzzy logic to assign probabilities of membership to each cluster to each point in a dataset. While both may prove effective, fuzzy c-means is conceptually more appropriate, as it allows cluster to overlap, as they might in the real world. For example, a graduate student user may exhibit characteristics of the pSBD scenarios developed for researcher and student personas.

Once a clustering method has been selected, a clustering rule, such as the silhouette method, can be used to assess the optimal amount of clusters in a dataset. The silhouette method works by quantifying the agreement within a cluster (cohesion) and different among clusters (separation)[35]. The silhouette statistic is calculated by taking the difference between the cohesion coefficient and the separation coefficient. For a given k clusters, the closer this statistic is to 1, the better the points are clusters. The optimal number of clusters – that which maximizes within cluster homogeneity while maximizes between cluster variance – can be found by trying different values of k , and selecting the k with the highest average silhouette statistic value.

Data assimilation

Finally, the real usage patterns can be assimilated into the existing pSBD distributions using Bayes’ rule to update current knowledge about expected usage. Formally, Bayes rule $P(\theta|y) \propto P(y|\theta)P(\theta)$, states that the posterior distribution of a parameter ($P(\theta|y)$) is proportional to a likelihood function ($P(y|\theta)$) times a prior ($P(\theta)$). In other words, the posterior, which quantifies updated inference about an unobservable parameter’s distribution, is related to the likelihood of the observed data, given the model, and any prior knowledge about the parameter’s distribution. Bayes rule can be applied an arbitrary number of times; in each successive iteration the posterior becomes the prior, representing our current best guess at the unobservable parameter, human intent.

To apply Bayes rule, both a prior and likelihood distribution are required. It is helpful to choose conjugate priors so that the posterior may be calculated analytically in closed form. While other distributions are feasible, the solution must be done numerically with sophisticated sampling algorithms. A common conjugate prior distribution for a binomial dataset is a beta distribution. As the prior to a binomially-distributed likelihood in a data assimilation context, the Beta prior represents current knowledge about how the probability of success is distributed, with uncertainty. This distribution has two shape parameters α and β and has support on the interval $[0, 1]$.

The first step in the assimilation is to choose the appropriate parameters for the prior distribution. Specifying the shape parameters of the beta prior may seem arbitrary, so it may be more efficient to calculate these parameters based on the expected value and effective sample size of the distribution. Specifically, the expected value of a beta distribution is given as $m = \frac{\alpha}{\alpha+\beta}$ and the effective sample size of the distribution is given as $\nu = \alpha + \beta$. Therefore, $\alpha = \nu * m$ and $\beta = \nu(1 - m)$. Thus, it is possible to work in terms that are directly related to how we expect the random variable to behave and our current degree of belief in the distribution. Furthermore, by working with ν and m , the terms are more equivalent to parameters of a beta distribution, making comparisons with the likelihood function easier.

To apply Bayes’ theorem, we multiply the likelihood function by the prior function to obtain the posterior that represents a weighted combination of the two. If, as above, we assimilate a binomial likelihood (Y successes in N trials) with a beta prior (α and β), our posterior distribution shape parameters are $\alpha = Y + (\nu * m) - 1$ and $\beta = N - Y + (\nu * (1 - m)) - 1$. The posterior, likelihood, and prior can be effectively visualized using density curves to illustrate the differences between the distributions.

Case Study

Introduction

In this section, I introduce a simple yet illustrative case study demonstrating the utility of the method described here. Consider an interface with two components, the *map panel* and *information panel*. The information panel can be opened and closed via a user interaction as well as be resized on demand by the user. The user cannot control the dimensions or presence of the information panel, but, as screen size is fixed, it takes pixels that would otherwise belong to the main panel, if it is in the open position. The interface is illustrated for both the information panel open state and the information panel closed state in Fig. 1. Imagine that the map panel contains an interactive map that can be browsed by the user, with points describing the spatiotemporal position of a geographic phenomenon overlaid on the map. In this example, the geographic phenomenon is murder events in the city of Chicago. The information panel has additional information about each occurrence of the phenomenon, and additional information about each point can be accessed and displayed in the panel when the user clicks on the corresponding point on the map. The attributes associated with each event include fields like weapon, time of day, day of week, and whether a subject was arrested in connection with the crime.

The developers of this interface envision two potential user groups, novices and researchers. A key difference between novices and researchers is that they are unlikely to dig as deeply into the details of each murder as their researcher peers. With that in mind, the application developers envisioned two simple scenarios to describe the use of each of these user groups (Box 1).

Table 1: Narrative scenarios

Scenario: Novice

User one is a student at the University of Chicago, enrolled in an introductory geospatial analysis class. She has a personal motivation to understand the patterns in murders, because she lives in Chicago. She has also been tasked with identifying the salient features of the spatiotemporal distribution of the murders for her classes. She is unlikely to investigate the details of each crime; rather, she is likely to browse the map itself to determine if she can recognize patterns in the spatial distribution.

Scenario: Researcher

User two is a criminology professor at the University of California, Berkeley. She is interested in a recent spike in murders, but has no firsthand experience with Chicago. She is particularly interested generating hypotheses about murder weapons in different parts of the city. She is likely to make extensive use of the information panel. Furthermore, because her interest is primarily in the attributes associated with each crime, she is less interested in the map component. She will likely open the information panel to dominate the visual layout of the application to focus her attention there.

Assigning Probabilities

From these basic narrative scenarios, we can infer potentially differing patterns of use (Table 2). Indeed, we can determine, from these scenarios, how the prototypical research user’s interface might be displayed and how it might differ from that of the prototypical novice user. At this point, we may begin to assign probabilities to the scenarios. In this case study, we will consider only two statistical statements for each scenario. Specifically, we will specify the probability of use of the information panel and the probable width of the information panel. Since the novice user is, from the scenario, unlikely to investigate the details of

each crime, we can assign her a rather low probability of using the information panel. Using a Bernoulli distribution, we might assign the novice user’s $p(\text{infoPanel}) = 0.25$, $X \sim (1, 0.25)$. This probability means that, as designers, our best guest suggests that novices will use the info panel approximately one in four sessions with the interface. Moreover, since novices are unlikely to be interested in a particular element of the information panel, we may assume that the width of this panel may be less than half of the screen, perhaps 30% of the screen, when it is open. Specifically, this can be modeled as a binomial distribution with 100 trials, with each trial corresponding to a percentage width on the page ($X \sim B(100, 0.3)$).

We can complete a similar exercise for the researcher model. In this case, the researcher is likely to rely extensive on the information panel. Indeed, from the scenario, it appears that such research may rely entirely on the information from the information panel. Thus, we assign the probability of use of the information panel as ($p(\text{infoPanel}) = 0.9$, $X \sim B(1, 0.9)$). Therefore, for researchers, we expect nine in ten user sessions to employ the information panel. Furthermore, our scenario specifies that the researcher is likely to dominate the visual layout with the information panel component. Therefore, the assigned probability statement for panel width is $X \sim B(100, 0.65)$, corresponding to an expectation of 65% width.

Table 2: Probability Statements

Parameter	Distribution	Expectation
<i>Novice</i>		
info panel	$X \sim B(1, 0.25)$	0.25
width	$X \sim B(100, 0.3)$	0.3
<i>Researcher</i>		
info panel	$X \sim B(1, 0.9)$	0.9
width	$X \sim B(100, 0.65)$	0.65

Generating PGCs and Visualizing Them

Using a scripting language, it is possible to generate random draws from these distributions, such as using the distribution functions in R [???]. In this way, it is possible to have thousands of PGCs that represent the developer’s intuition of how users will interact with the components of the software. From these draws, it is possible to create some visualizations that can be helpful in communicating this intended use. Density plots (Figure 2) can be helpful in communicating the variation between and within a user model. Similarly, for the Bernoulli distribution describing boolean values, bar graphs showing relative frequency of each group for each outcome can be compelling. Both of these visualizations can be helpful in inter-design team communication and communicating with stakeholders, because they give a graphical representation of intended use, rather than weak qualifiers like ‘more’ or ‘less’. These representations can be particularly helpful when negotiating between team members to arrive at an agreement for intended use. An additional and equivalent method of visualization is to overlay the PGCs probability densities over top of the wireframe of the interface (Figure 3).

Another important visualization metric is the principle components analysis (PCA) plot with loadings. A PCA is a dimensionality reduction procedure the derives new orthogonal dimensions that describe the variance in the dataset. In this case study, there are only two dimensions, so a PCA is not appropriate. However, consider a situation where there are not two dimensions of each PGC but 20 or 50 or 100. In this larger case, it would be difficult to assess the usage of each component separately using densities or bar charts. Here, a PCA would be appropriate. The routine would derive new dimensions. A plot of these new dimensions would thus be easier to digest. The correlations between the derived dimensions and the original dimensions, known as the PCA loadings, are informative in this setting, by describing the agreement between the derived axes and the original axes. In this case, we can examine the loadings the determine how each of the original variables contributes to the new axes, and therefore, which of the original variables are the most important in explaining the most variance in the original dataset.

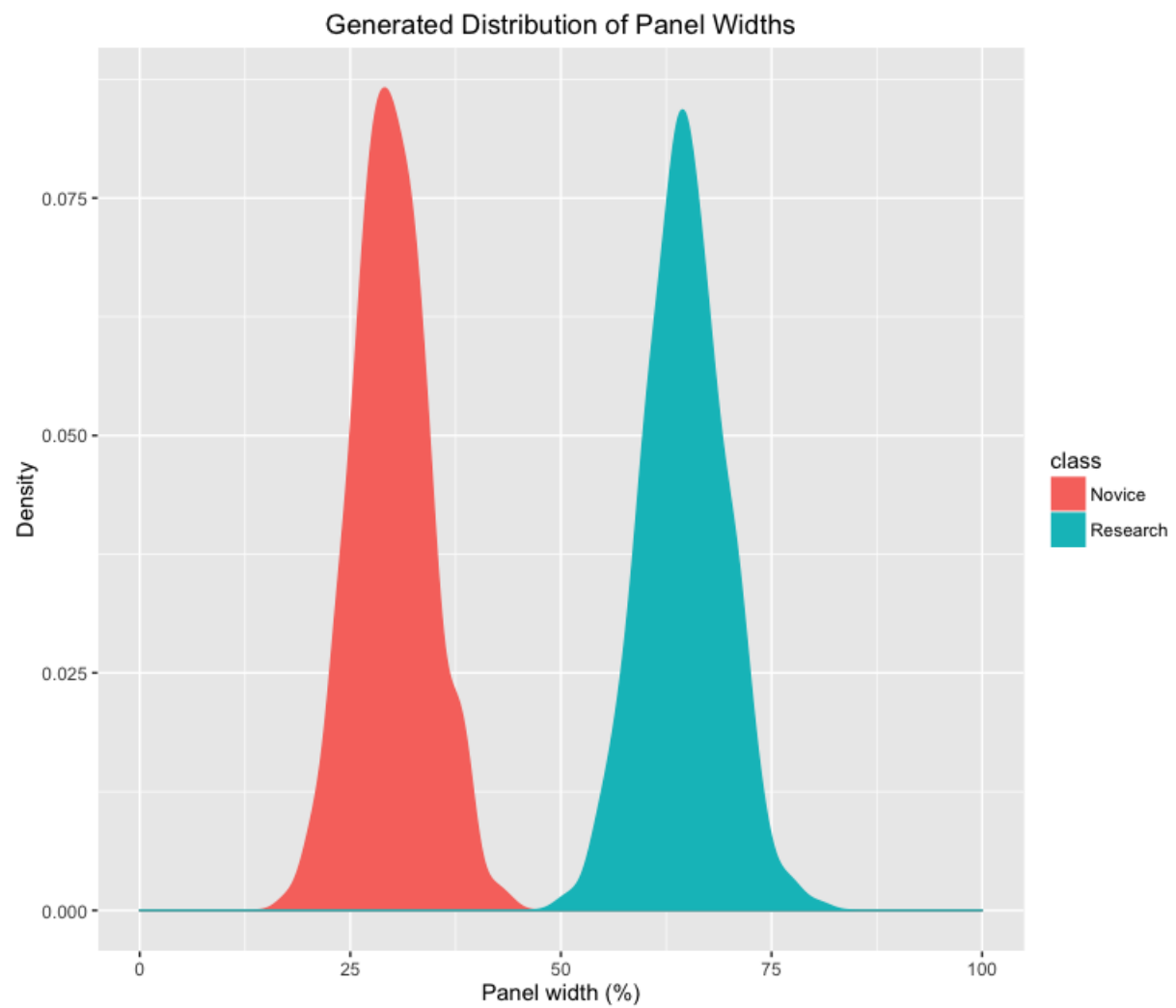


Figure 1: A density plot describing expected panel widths from each user model in the hypothetical case study

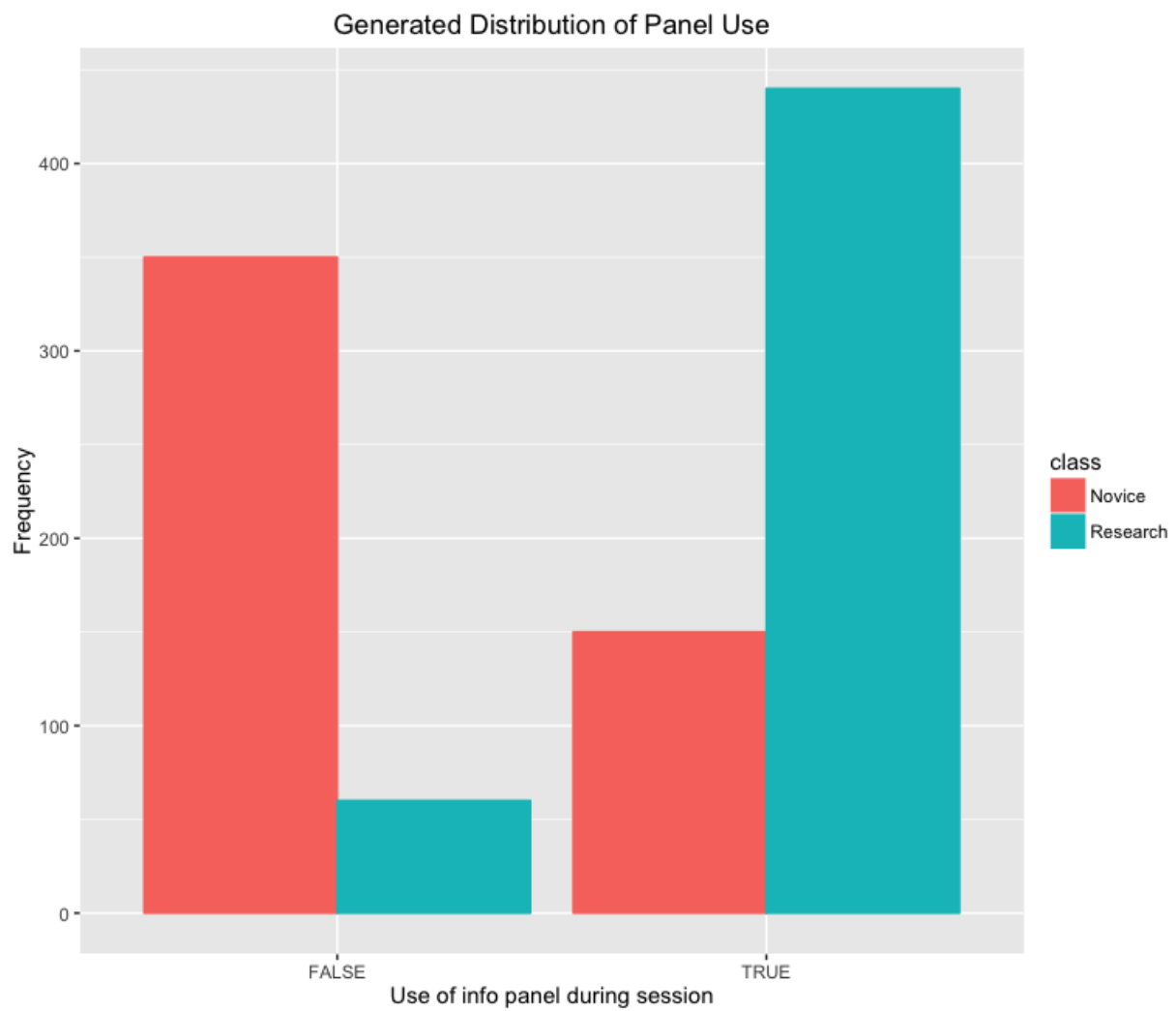


Figure 2: A barchart showing relative frequency for use of the information panel in the hypothetical case study

Probability distributions characterizing the use of Panel A

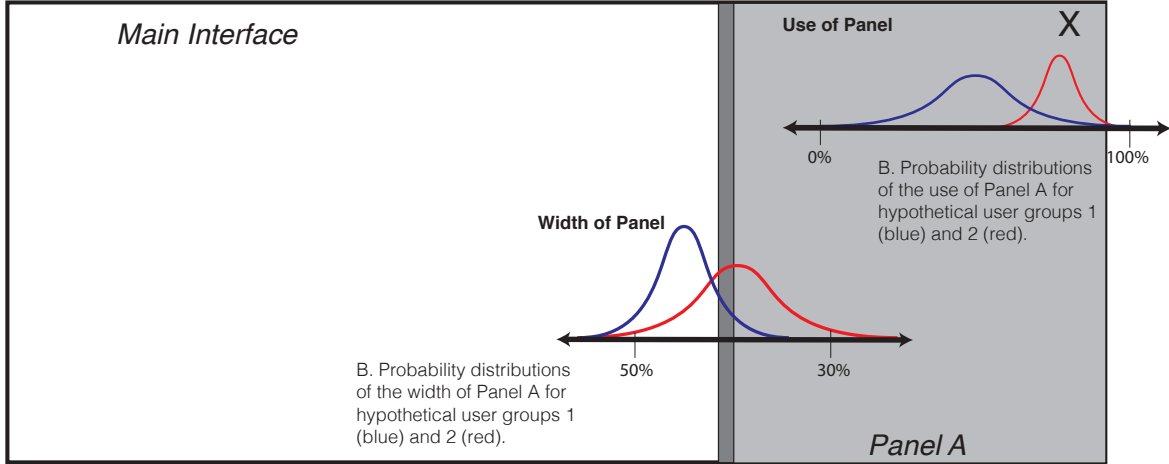


Figure 3: PDFs from the pSBD process overlain on top of the low fidelity wireframe of the interface

Assessing real usage vs. expected usage

While the visualizations outlined above can be helpful in negotiating expected use, the pSBD framework becomes especially valuable when real usage data is available from an in-development or in-production application. In this case study, we will examine a scenario in which we have ‘real’ usage data for our hypothetical application, that has been synthetically generated for this purpose (Figure 4).

In this case, we use the fuzzy clustering in the `fanny` function in the R package `cluster` [???]. Fuzzy clusters are ideal for this analysis, since fuzzy logic allows the clusters to overlap. Traditional clustering analyses typically require crisp boundaries between clusters. Fuzzy clusters may have fuzzy boundaries, and each cluster is assigned a degree of membership to each of k clusters. In this way, the clusters are allowed to overlay, as users in real life might overlap.

As in traditional k-means clusters, it is first important to choose the correct number of clusters for our data. In this case, we can use the silhouette method for determining the optimal number of clusters in a dataset. In this case, we expect there to be either 2 (2 user models, real data fits one of them) or 3 (2 user models, real data is distinct), however, we illustrate the analysis with k between 2 and 8 (Figure 5). After this procedure, it is clear that the optimal number of clusters in this dataset is 2.

Once we have chosen the number of cluster, we can visualize how the real data and the theoretical user model data aligns. Since each point in the dataset has a degree of membership in one or more clusters, we can plot them in cluster membership space to see how they differ from one another (Figure ??). Since we know which data came from the pSBD process and which data came from real usage collection, we can overlay the real class labels on top of this space. In this case, we see that the ‘real’ data appears to be quite similar to the research model developed during pSBD. This would indicate that nearly all of the users using this system are research users and that the scenario was developed quite accurately. In this case, the developers may want to reconsider why scenario one is not being utilized, and perhaps reimagine some design decisions to specifically target novice users.

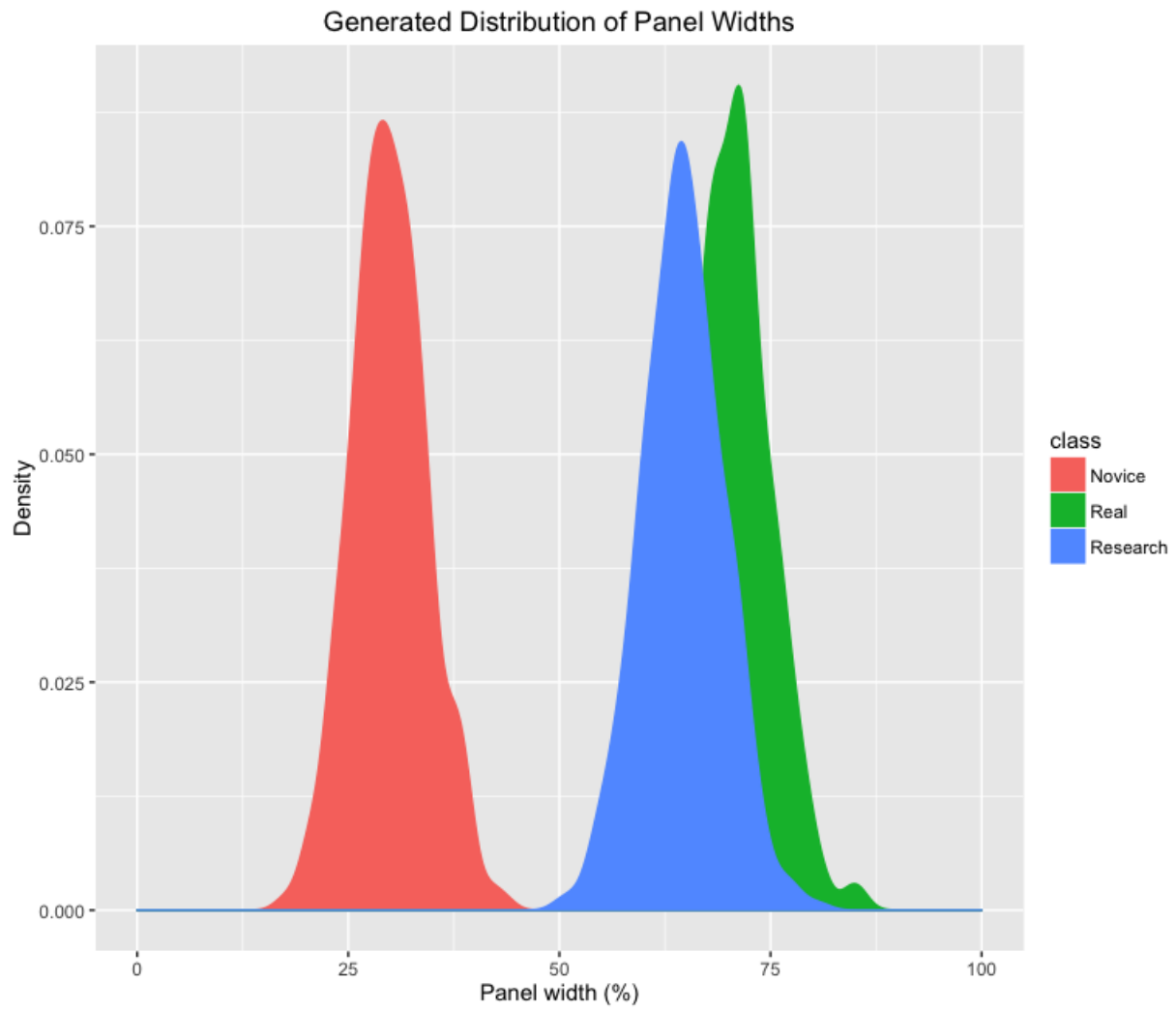


Figure 4: Densities of panel width with simulated real data included

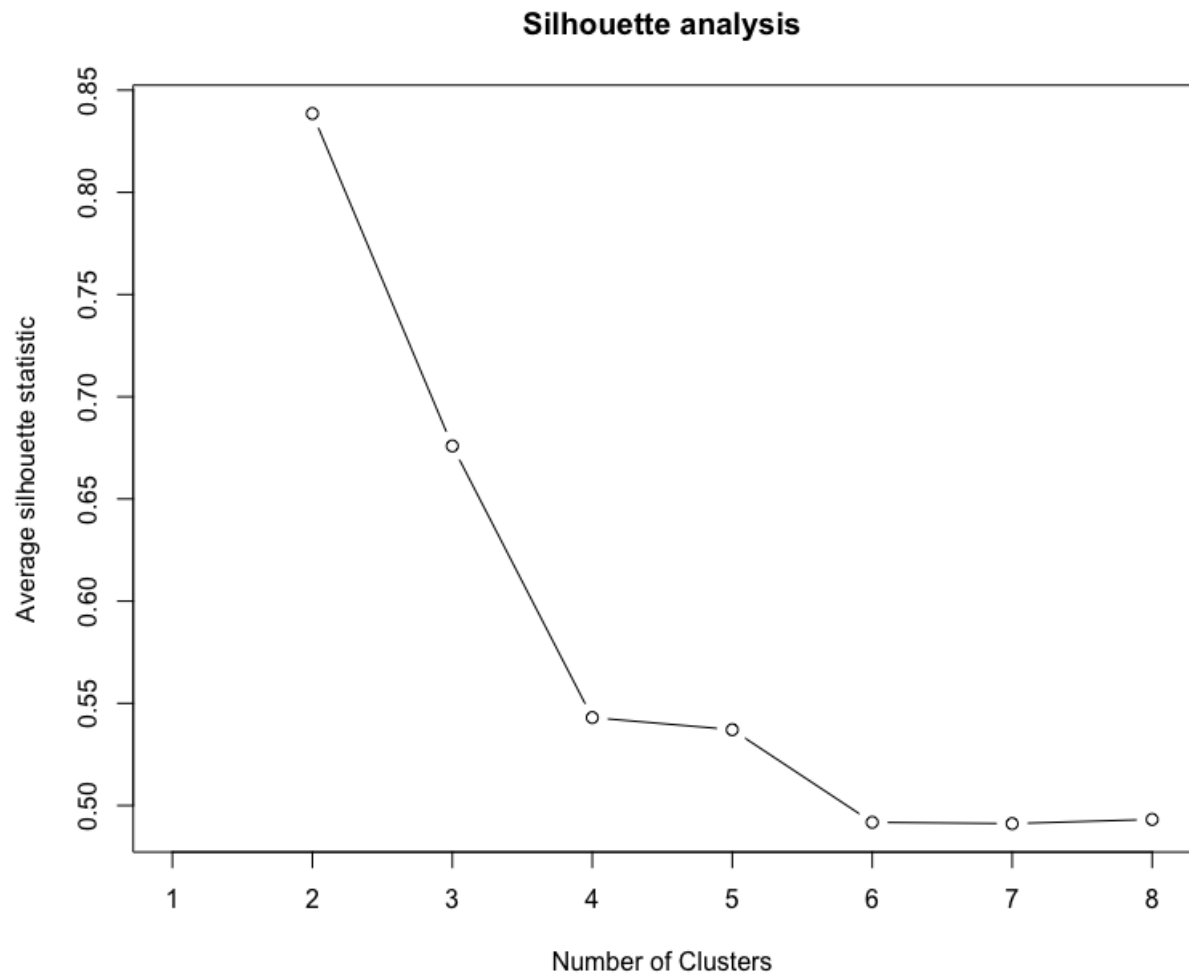
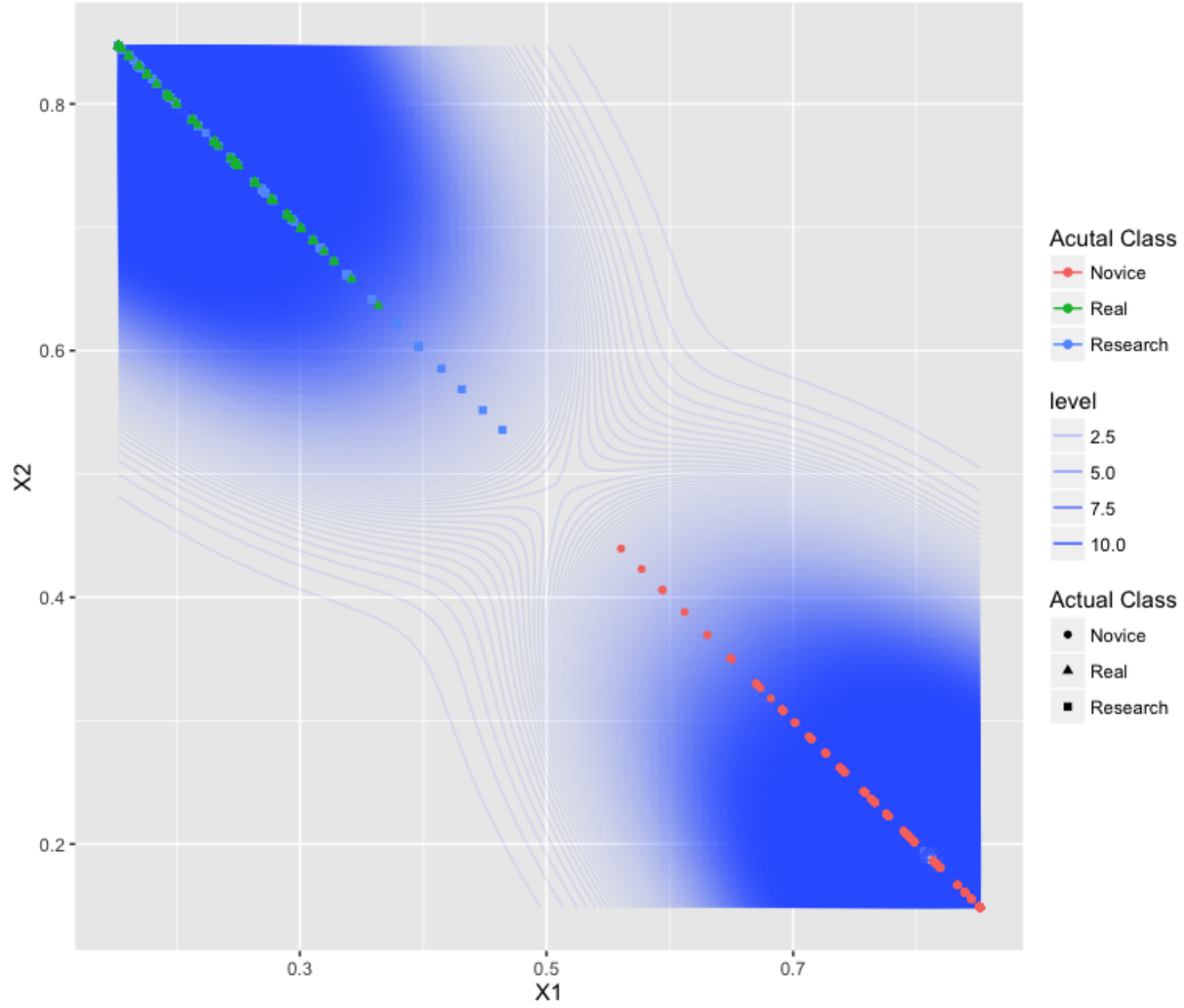


Figure 5: Average silhouette statistic for $k=2$ to $k=8$ clusters. Notice that the maximum silhouette value occurs at $k=2$, indicating an optimal number of clusters of 2

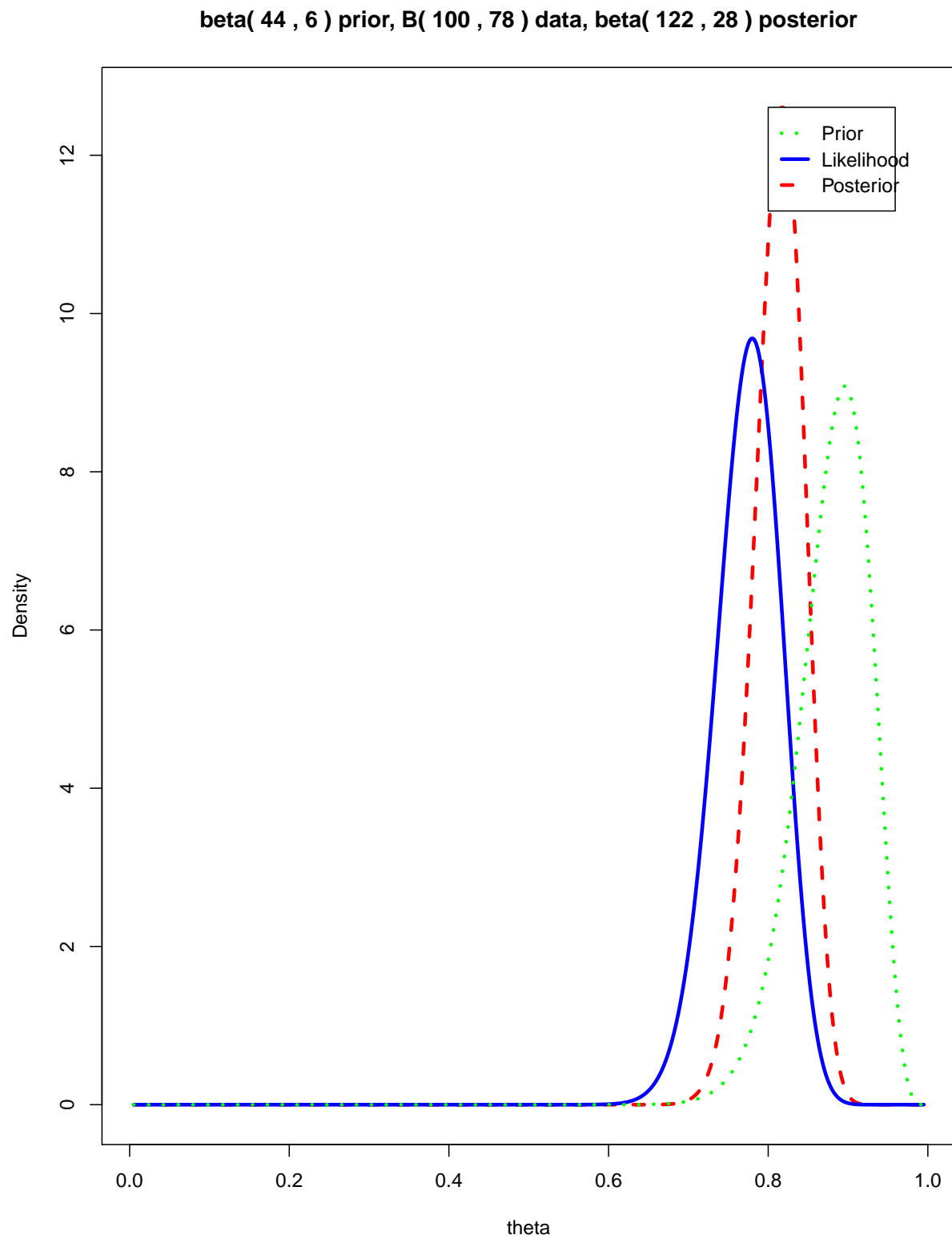


Assimilating real usage data into pSBD scenarios

A final activity that can be done in pursuit of personalized interfaces to assimilate the real usage data into our best understanding of how our users utilize the interface. In this case, we will consider an example for the usage of the information panel on the hypothetical interface.

Consider a situation where we are weakly confident in the developer's quantification of the scenario parameters. In this case, we will only consider the research distribution, as we have effectively shown above that the new data is more than likely to correspond with our intended idea of how researchers use the system. Specifically, we could set our effective sample size, ν to, for example, 50. This would correspond to the level of certainty we would have if we observed fifty distinct users behaving in the prescribed way. With $\nu = 50$ and using the expectation given in Table 2 ($m = 0.9$), we obtain values for $\alpha = 44$ and $\beta = 6$. From the 'real' data we observed previously, we observed 78 users utilize the information panel out of 100 independent events. Using Bayes Theorem, we can multiply the likelihood function of the data by the prior function, to obtain a posterior distribution that includes a weighted combination of the two. Because of the conjugate distributions we can obtain the new shape parameters of the posterior distribution as $\alpha = Y + (n * m) - 1 = 122$ and $\beta = N - Y + (n * (1 - m)) - 1 = 28$. Using the equations shown above, we can solve for the effective sample size and expectation of the new distribution ($\nu = 150$, $m = 0.813$). These parameters are now a weighted combination of the prior knowledge and new observations. Therefore, given both, our new best estimate of

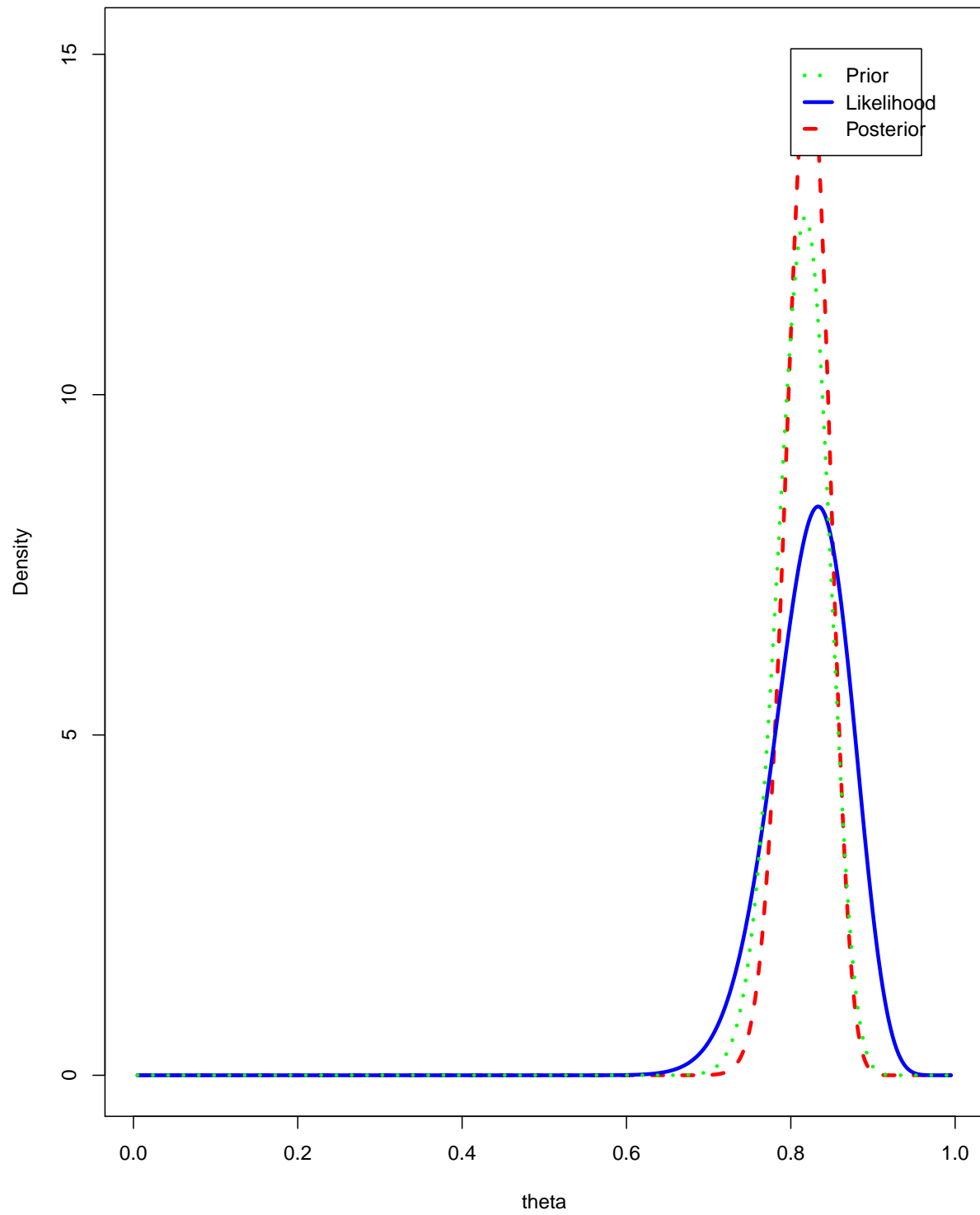
the probability of a researcher using the information panel is approximately 0.81 (Figure ??).



We can repeat this process an arbitrary number of times. Each time, we take our posterior, representing our

current best guess of the distribution, and turn it into our prior distribution. We then combine it with new observations as they are observed to have a running assimilation of new data. For example, if we observe an additional 60 system uses with 50 of those using the information panel, we can assimilate those into the posterior from the prior iteration ($\alpha = 122$, $\beta = 28$) and obtain a new posterior (Figure ??).

beta(122 , 28) prior, B(60 , 50) data, beta(172 , 38) posterior



Discussion

Advantages of pSBD over traditional SBD

By introducing visuals into the design process, pSBD promotes visual model to efficiently present abstract thoughts and ideas about intended UX, replacing the weakly communicative qualitative language that is present in traditional SBD [???].

These tools can be helpful early in the development process by helping designers to improve smart defaults or refine interaction strategies for different groups.

These updated knowledge distributions can be used to update the look, feel, and interactivity of the interface on the fly. For example, if it appears that users always utilize the information panel, the interface can be configured to automatically open the information panel as the default configuration. Similarly, if the information panel is always reduced in screen size, the default width can be specified to reflect the current posterior mean width.

This data assimilation technique also allows for user preferences, as a whole, to change through time. For example, should users eventually begin to use the information panel less often, the posterior distribution would reflect this changing usage. If properly configured, the interface could automatically adjust its interactivity to reflect that users use the the panel less often, and therefore disable it by default.

limitations

In the present study, PGCs are generated from random draws from multiple independent probability statements. Future work could focus on deriving PGCs from a joint distribution that conditions each component's status on all others. For example, consider conditioning the width of the panel on whether or not it is open in this particular PGC ($P(\text{width} \mid \text{open})$), rather than drawing from both the width distribution ($P(\text{width})$) and the isOpen distribution ($P(\text{open})$).

Future research goals

While the current study introduced this technique in a conceptual way with a hypothetical case study, a forthcoming study will present the results of a formal implementation of pSBD on a real application. Several key questions to be addressed in a real application of SBD include: (1) how accurate are developer's intuitions for the appropriate scenario probability statements? (2) how are the distributions best negotiated between developers and stakeholders? and (3) do the visualizations provided by pSBD positively influence the application development process? Another worthy research target would be to develop a Bayesian belief network for each component. As noted above, the current framework is limited by drawing from independent probability distributions. If an efficient method for constructing a Bayesian network of conditional probabilities was developed, pSBD could make better predictions by conditioning each probability on other components in the interface.

Conclusion

In this paper, I introduced probabilistic scenario-based design, a new method for underpinning traditional narrative scenarios during the development process with statistical distributions. pSBD maintains the flexibility and low-costs of traditional SBD, but introduces a new set of visual and statistical tools for application developers to test usage patterns and assimilate new knowledge. By leveraging Bayesian data assimilation, the process can formally incorporate developer and stakeholder intuition with real usage patterns, providing a better estimate of a latent variable that cannot be observed: human behavior. Clustering methods

can be used to determine whether real usage patterns fall within existing usage categories, or whether entirely new categories exist. Several novel visualization types were introduced and are useful throughout the process for communicating between developers, stakeholders, and to understand real usage patterns in conjunction with developer intuition.

#References

- [1] Hassan Q. Demystifying Cloud Computing. *CrossTalk* 2011:16–21.
- [2] Mosco V. To the Cloud: Big Data in a Turbulent World. *Popular Communication* 2014;12:271–69. doi:10.1080/15405702.2014.960779.
- [3] Roth RE. Interactive maps: What we know and what we need to know. *Journal of Spatial Information Science* 2013:1–57. doi:10.5311/josis.2013.6.105.
- [4] Adomavicius G, Tuzhilin A. Context-Aware Recommender Systems. In: *Recommender systems handbook*, Boston, MA: Springer US; 2015, pp. 191–226. doi:10.1007/978-1-4899-7637-6_6.
- [5] Lallé S, Toker D, Conati C, Carenini G. Prediction of Users’ Learning Curves for Adaptation while Using an Information Visualization. In: *The 20th international conference*, New York, New York, USA: ACM Press; 2015, pp. 357–68. doi:10.1145/2678025.2701376.
- [6] Isbell CL, Pierce JS. An IP continuum for adaptive interface design. *Proc of HCI International* 2005.
- [7] Lieberman H. User interface goals, AI opportunities. *AI Magazine* 2009.
- [8] G Smith Linden B.; York J. Amazon.com Recommendations: Item-to-item Collaborative Filtering n.d.
- [9] Mulvenna MD, Anand SS, Büchner AG. Personalization on the Net using Web mining: introduction. *Communications of the ACM* 2000;43:122–5. doi:10.1145/345124.345165.
- [10] Rosson MB. Scenario-Based Design. In: Jacko J, Sears A, editors. *The human-computer interaction handbook fundamentals, evolving technology and emerging applications*, CRC Press; 2002, pp. 1032–50. doi:10.1201/b11963-56.
- [11] Carroll JM. Five reasons for scenario-based design. *Interacting with Computers* 2000;13:43–60. doi:10.1016/S0953-5438(00)00023-0.
- [12] Carroll JM, Rosson MB, Chin G, Koenemann J. Requirements development in scenario-based design. *IEEE Transactions on Software Engineering* 1998;24:1156–70. doi:10.1109/32.738344.
- [13] Go K, Carroll JM. The blind men and the elephant: Views of scenario-based system design. *Interactions* 2004;11:44–53. doi:10.1145/1029036.1029037.
- [14] Kazman R, Abowd G, Bass L, Clements P. Scenario-based analysis of software architecture. *IEEE Software* 1996;13:47–55. doi:10.1109/52.542294. [url_ctx_fmt=info:ofi/fmt:kev:mtx:ctx&rft_val_fmt=info:ofi/fmt:kev:mtx:journal&rft_id=info:doi/10.1109/52.542294](#)
- [15] Rosson MB, Carroll JM. *Scenario-Based Design*. 2005.
- [16] Cunha B, Pereira JP, Gomes S, Pereira I, Santos JM. Using personas for supporting user modeling on scheduling systems. *Proceedings of IEEE HIS*; 2014.
- [17] Zukerman I, Albrecht DW. Predictive statistical models for user modeling. *User Modeling and User-Adapted Interaction* 2001.
- [18] Young S. Cognitive User Interfaces. *IEEE Signal Processing Magazine* 2010;27:128–40. doi:10.1109/MSP.2010.935874.
- [19] Yannakakis GN, Spronck P, Loiacono D. Player modeling. *Dagstuhl Follow- ...* 2013. doi:10.4230/DFU.Vol6.12191.45.
- [20] Liu J, Wong CK, Hui KK. An adaptive user interface based on personalized learning. *IEEE Intelligent Systems* 2003;18:52–7. doi:10.1109/MIS.2003.1193657. [url_ctx_fmt=info:ofi/fmt:kev:mtx:ctx&rft_val_fmt=info:ofi/fmt:kev:mtx:journal&rft_id=info:doi/10.1109/MIS.2003.1193657](#)
- [21] Conati C, Gertner A, Vanlehn K. Using Bayesian networks to manage uncertainty in student modeling. *User Modeling and User-Adapted ...* 2002.
- [22] Horvitz EJ, Breese JS, Heckerman D, Hovel D, Rommelse K. The Lumiere Project: Bayesian User

Modeling for Inferring the Goals and Needs of Software Users. ArXivorg 2013.

[23] Honold F, Bercher P, Richter F, Nothdurft F, Geier T, Barth R, et al. Companion-Technology: Towards User- and Situation-Adaptive Functionality of Technical Systems. In: 2014 international conference on intelligent environments (ie), IEEE; 2014, pp. 378–81. doi:10.1109/IE.2014.60.

[24] Wikle CK, Berliner LM. A Bayesian tutorial for data assimilation. *Physica D: Nonlinear Phenomena* 2007;230:1–16. doi:10.1016/j.physd.2006.09.017.

[25] Fischer G. User Modeling in Human-Computer Interaction. *User Modeling and User-Adapted Interaction* 2001;11:65–86. doi:10.1023/A:1011145532042.

[26] Fan X, Wang J. BayesHeart. In: The 20th international conference, New York, New York, USA: ACM Press; 2015, pp. 405–16. doi:10.1145/2678025.2701364.

[27] Horvitz E. Principles of mixed-initiative user interfaces. New York, New York, USA: ACM; 1999. doi:10.1145/302979.303030.

[28] A Bayesian Approach to Data Assimilation 2005:1–20.

[29] Airoldi EM. Getting Started in Probabilistic Graphical Models. *PLoS Computational Biology* 2007;3:e252–5. doi:10.1371/journal.pcbi.0030252.

[30] Dawson A, Paciorek CJ, McLachlan JS, Goring S, Williams JW, Jackson ST. Quantifying pollen-vegetation relationships to reconstruct ancient forests using 19th-century forest composition and pollen data 2016. doi:10.1101/039073.

[31] Ellison AM. Bayesian inference in ecology. *Ecology Letters* 2004;7:509–20. doi:10.1111/j.1461-0248.2004.00603.x.

[32] Ho SYW, Phillips MJ. Accounting for Calibration Uncertainty in Phylogenetic Estimation of Evolutionary Divergence Times. *Systematic Biology* 2009;58:367–80. doi:10.1093/sysbio/syp035.

[33] Roth RE, Hart D, Mead R, Quinn C. Wireframing for Interactive & Web-based Geographic Visualization: Designing the NOAA Lake Level Viewer. *Cartography and Geographic Information Science* 2015:1–28. doi:10.1080/15230406.2016.1171166.

[34] Carroll JM. Five Reasons for Scenario-Based Design 1998:1–12.

[35] Rousseeuw PJ. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics* 1987;20:53–65. doi:10.1016/0377-0427(87)90125-7.