Probabilistic Scenario-Based Design

Operationalizing probability-based user models in the user-centered design process

Scott Sherwin Farley

May 4, 2017

Constructing hypothetical scenarios and user narratives is a common technique for communicating the envisioned user experience (UX) of a tool, often referred to as scenario based design. Using this approach, application developers, designers, and stakeholders rapidly build stories of expected use of a tool to promote grounded discussion about the tool's UX. This strategy is effective, cheap, flexible, and simple to implement. However, most narrative scenarios are 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 (pSBD) for underpinning narrative scenarios with probability statements. Using this technique, tool developers modify traditional scenarios to include formal statistical distribution that describe the expected UX of the interface by its target audience. This approach offers at least three advantages that complement traditional scenario-based design. First, it offers the potential for novel visualizations of usage patterns. These visualizations provide alternate and more concrete view of the tool's intended UX that can improve communication within the design team. Second, once an early release of the interface is released, pSBD enables formal statistical testing of usage patterns, allowing designers to confirm whether with real patterns meat those expected by UX designers. Finally, pSBD provides a mechanism of smartly improving interface interactivity and design through the assimilation of new observations. This process incorporates both usage data and developer input, allowing the interface to adapt to fit its users, while consuming less data than traditional artificial intelligence approaches to adaptive interfaces. The utility and process of pSBD are illustrated with an synthetic case study based around a web-based interactive mapping tool.

Keywords: Scenario Based Design, Probability, Data Assimilation, User Centered Design, User Interface, User Experience, Visualization, Design, Bayesian Methods

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 for the Web. These apps encourage a two-way communication model that facilitates the production of user-generated content and social interactions among users. Often, these applications serve multiple user groups, with different interests, motivation, or skills [1]. While it is possible to simultaneously support multiple user groups through carefully engineered design decisions, 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 [2–5]. Typically, these approaches improve the tool's user experience (UX) by exposing the user to less information and visual stimuli, allowing cognitive function to become more focused on specific tasks. However, 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 [6]. Such large amounts of real usage data may not be available for early stage applications still under active development. Moreover, there is often no direct linkage between the personalization algorithms and the development process, during which the development team works to design and create the UX.

During a User-Centered Design (UCD) process, 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) [7,8]. Scenarios provide a common vocabulary for communication between stakeholders by describing how hypothetical users may interact with a tool [9]. SBD is used in a wide variety of fields and is not limited to development of software systems [10,11]. Within software development, it is often used to inform the requirements of a proposed tool during the negotiation phase 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 [9]. By design, the scenarios produced during SBD are informal [7]. If a scenario is formalized, it changes in both form and purpose from an illustrative device for communication to a rigorous document describing the functional requirements of the proposed tool [10]. Requirements documents are may be more concerned with the feasibility of the tool than the utility and usability by the target users.

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 one or more probability statements that describe the interactivity and design characteristics 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 the component. In more complex cases, distributions can be used to specify 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 usage patterns, even before the interface has undergone extensive usage. Rather than informally constructed narratives that rely on relaxed language, pSBD scenarios are 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, customization algorithms 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 illustrate three potential advantages of operationalizing the pSBD approach. First, this method would introduce novel visualizations for showing, in a concise manner, how UX would differ among multiple scenarios or target user groups. Second, pSBD enables developers to determine if real usage patterns conform with the expectations of the development team. This is particularly useful during design iterations, allowing team members to alter the interface to meet the needs of multiple stakeholders. Finally, pSBD is amenable to Bayesian data assimilation, where observations of actual usage are integrated into the developers' probability estimates, producing a statistically robust combination of the two. Using this combined knowledge source, a smart interface could adapt its design in an intelligent way to meet the needs of its users.

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. I conclude with a discussion of the advantages of pSBD over traditional SBD and outline future research that could be done to strengthen the pSBD framework.

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 UXs instill positive feelings of success and competency, allowing the interface to recede into the background as users focus on their work, exploration, or pleasure [12]. In many sophisticated apps, AI and user modeling can play important roles in helping the interface to fade away, 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 contact with items the user is unlikely to be interested in and focus attention on tasks that the user is most 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 consumers to 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 [13]. 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 [14,15]. While user modeling need not be quantitative, statistical user models allow an application to anticipate behavior, including goals, actions, and preferences [14]. Models can be constructed as top-down, in which an expert-based, theoretical understanding of user preferences is 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 are hybrids that combine aspects of these two categories [16].

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 which is 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 [17]. AUIs can adapt to the needs of different users for a variety of tasks [4]. AUIs have commonly been implemented in the context of intelligent tutoring and online educations systems, where a user model is used in tracking how a student processes towards an educational goal [18,19]. Typically, AUIs work by identifying membership in a user group based on a series of interaction events, which requires tracking all user interaction sequences [17], potentially creating a large volume of data to store and manage.

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 [20]. Such systems would identify the appropriate user model then assemble the user interface components most suitable for the identified context of use [20]. Clustering methods can be used to identify groups of similar users in a data-driven way.

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 in a variety of contexts; however, while the details may differ between implementations of SBD, all are aimed at concretely describing the expected use of a tool early in its development [21]. SBD narratives generally contain four elements: (1) a setting, (2) an actor with personal motivation and skills, (3) background 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 [8,10,22]. Typically, actors execute a sequence of actions and events that lead to some outcome [21]. Scenarios can be expressed in a variety of ways, such as through text, videos, mockups, or storyboards.

It is important to note that scenarios are inherently 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 [22], evoking reflection in the context of design [8]. 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 [9,22]. 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 [8]. 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 [13], 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 [23]. In classical statistics, it is difficult to take prior knowledge into account when testing hypotheses, and statistical experiments often demand large sample sizes to generate robust results. Moreover, it is difficult to use the results from one experiment to predict the outcome of a future experiment [24]. The Bayesian paradigm provides a coherent approach for combining information from new and existing sources in a probabilistic framework [25]. Bayesian inference allows 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 [23]. 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 [25].

Bayesian belief networks (BBNs) are a 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 estimated 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 [26], and 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 [19,27]. BBNs have been used in a variety of contexts related to user modeling, including in Microsoft Office Assistant [19], educational tools and interactive tutoring applications [18,19], and health-related smartphone apps [28].

Data assimilation involves fusing observations and prior knowledge together in a statistical framework to obtain an estimate of the distribution of the true state of the underlying process [25]. In a Bayesian context, this is accomplished by using Bayes Theorem to obtain a posterior distribution from a likelihood distribution and prior distribution. Data assimilation is often used in a spatiotemporal context for numerical weather and climate models [25,29], as well as in ecological and phylogenetic modeling [30–32], among other fields.

Method

pSBD is an iterative process that may involve designers, developers, and stakeholders. In this section, I outline the key steps taken during the pSBD process.

Prerequisites

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. The visualization techniques described below can be implemented with wireframes alone. 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 is required, so that real user configurations 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 describing the tool's intended 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 [10], to clearly describe the expected 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 [21] and [8].

Assign probability statements

The second phase of pSBD is to enhance narratives with probability statements that capture the intuition and expectations of the application developers. For each scenario, components of the intended interface may be supplemented with one or more probability distributions that describe its interactivity or design. For example, these descriptions may quantify the probability that a particular interface component is used in a scenario, 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 among distinct user models.

An essential piece of this framework is specifying the correct distribution to use in the scenario. While any statistical distribution may be used in this process, several appear particularly useful in the context of pSBD. Because well-designed interfaces typically enforce constraints on interactivity, users are not exposed to infinite degrees freedom in their browsing. Therefore, most continuous distributions with infinite support may be inappropriate. Instead, discrete distributions and those distributions with truncated support over smaller intervals should be used. The binomial distribution and its special case the Bernoulli distribution (n=1) appear to have particularly useful applications in pSBD. The probability that a certain interface component is used in a given scenario can 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 assessed 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 process, the mean or variance 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 in an alternative way. 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 these discussions, some of the visualizations described below may be produced on-the-fly to dynamically reflect the negotiated UX.

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 a data assimilation phase.

Visualization

Once the distributions have been specified and finalized, the resulting user model is available for visualization, analysis, and inference. To perform these tasks, it is first necessary to draw, at random, from the distributions described in the previous phase to generate a collection of potential parameter values that are representititve of the usage patterns in the scenario. Using a scripting language such as R [34] or python [35] independent draws from a probability distribution with given parameters can be easily simulated. A compete probability generated configuration (PGC) is formed when all distributions in a scenario have been drawn from. For visualization purposes, it is useful to draw 100s or 1000s of PCGs.

A set of complete PGCs can be used to visualize differences among groups or to highlight intra-group variability. Density curves, plotting the probability mass of a parameter distribution, or bar charts, which plot the frequency of boolean outcome, can be effective in communicating and negotiating in the early phases of interface development. Principle components analysis and its associated plots can be effective method of succinctly describing multivariate differences among scenarios. 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 discussed further in the case study below.

Integration with real usage data

If the application under study has an early version release ready for user tests (alpha- or beta- stage), the pSBD scenarios can be used to test similarities with observations of real interface usage. Traditional k-means clustering [36,37] or fuzzy c-means clusters [38] can be helpful in determining whether real usage data is distinct from an existing pSBD scenario. Traditional clustering delineates crisp boundaries among $k \geq 2$ clusters, while fuzzy clustering assigns degrees of similarity to a cluster prototype for each point in the dataset. Fuzzy clustering may allow groups to overlap and points to exhibit qualities of multiple clusters, making it more appropriate for modeling users who may display characteristics of multiple theoretical use cases [39]. For example, a graduate student user may exhibit characteristics of pSBD scenarios developed for researcher and student use.

Once a clustering method has been selected, the optimal number of clusters in the dataset can be determined by using a rule to quantify the separation among clusters. One such method, the silhouette analysis, works by calculating the agreement within a cluster (cohesion) and separation among clusters [40]. Each point in the dataset is assigned a silhouette value, describing the how well it is clustered, with a measure of 1 describing perfect clustering. The optimal k clusters can be found empirically by trying different numbers of k and selecting the □ that maximizes the average silhouette statistic across all points in the dataset.

Once the best-fitting number of clusters in the combined real and pSBD dataset has been found, the clusters can be interpreted to yield information about how real patterns relate to the expectations of the developers. If k is equal to the number of scenarios developed during pSBD, the real usage data fits well within the expected use cases. If k is larger than the number of scenarios, the scenarios are either not capable of describing real usage, or, there is a distinct group of use cases that was not described in the scenarios. Further analysis, including visualization, can be performed to determined how the points are clustered, and where real usage differs from the expectation.

Data assimilation

Real usage patterns can also be assimilated into the existing pSBD distributions using Bayes' Theorem to provide an updated estimate of the interaction patterns with the interface. Formally, Bayes Theorem,

$$P(\theta|y) \propto P(y|\theta)P(\theta)$$

states that the posterior distribution of a parameter ($P(\theta|y)$) is proportional to a likelihood function ($P(y|\theta)$) times the prior distribution of the parameter ($P(\theta)$). In other words, an updated inference about the unobservable parameter's distribution is related to the likelihood of the observed data, given the model, and the analyst's prior knowledge about the parameter's distribution. As additional data becomes available, the theorem can be applied an arbitrary number of times; in each successive iteration the posterior becomes the prior, representing the current knowledge of the state of the unobservable parameter – in this case, how a given user will interact with the software system.

During the pSBD design process, information sufficient for forming the prior distribution was given, making this

Box 1: Assimilation of Binomial Data with a Beta Prior

Many of the distributions used in pSBD are binomial, describing a number of successes in a total number of events. The data follows the model

$$X \sim Binom(n, p)$$

where n is the total number of trials and p is the probability of success for each trial. To use binomial data in an assimilation setting, a prior distribution that describes the uncertainty around p is required. The beta distribution provides an easily interpretable, analytically tractable means of doing this. The beta distribution has continuous support on the interval [0, 1], and is defined as:

$$p \sim Beta(\alpha, \beta)$$

where α and β are parameters that control the shape of the distribution. Some algebra can be used to show that α and β can be defined in terms of the expected value (ν) and effective sample size (m) of the distribution.

$$\nu = \alpha + \beta$$

$$m = \frac{\alpha}{\alpha + \beta}$$

$$\alpha = \nu * m$$

$$\beta = \nu(1 - m)$$

It is much more intuitive, when working with binomial proportions, to think and characterize the prior in terms of the probability of success and uncertainty in that parameter, rather than the abstract shape parameters themselves.

In the assimilation, the likelihood for the observed data (Y successes in n trials) is combined with the chosen beta prior to form the posterior. Because the beta and binomial distributions are natural conjugates, the posterior will also be a beta distribution, and can be summarized in terms of α, β, ν or m. Furthermore, because the distributions play well together, simple algebra can be used to perform the assimilation:

$$\alpha_{posterior} = Y + (n * m) - 1$$

$$\beta_{posterior} = N - Y + (n * (1 - m)) - 1$$

$$\nu_{posterior} = \alpha_{posterior} + \beta_{posterior}$$

$$m_{posterior} = \frac{\alpha_{posterior}}{\alpha_{posterior} + \beta_{posterior}}$$

process relatively straightforward. The parameters of the posterior distribution can be found analytically if a conjugate prior, one that results in a posterior of the same family (e.g., exponential), is chosen. While nonconjugate priors do not pose a structural problem in the analysis, they can hinder direct interpretation [23] and often require a numerical solution using complex sampling algorithms such as the Gibbs sampling and Markov Chain Monte Carlo (MCMC) integration [41]. The details of an analytical application of Bayes' Theorem to a beta prior and binomial likelihood is shown in Box 1.

Case Study

Introduction

In this section, I introduce a simple yet illustrative case study to demonstrate the process and utility of pSBD. The data for this case study was generated synthetically. Moreover, while negotiations over the appropriate distributions to use to describe each scenario are an essential portion of the pSBD process, they are not considered in detail here.

Consider a web-based interactive application whose purpose is to support understanding of the spatiotemporal and multivariate attribute patterns of crime events in Chicago (loosely based on [42] and [43]). The developers of this application wish to support a rich interactive map, encouraging users to browse the spatial patterns of crime, while providing details, including crime type, date, and time, for each incidents on demand.

The proposed interface design includes two components, a central *main map* container that features a 'slippy map' that enables users to pan and zoom through space, with crime incidents overlaid as icons, symbolized according to incident type. A secondary panel, the *information panel* provides the additional details about each individual crime. The contents of the information panel are updated each time a user clicks on a point symbol in the main map. The information panel can be resized or closed via user interaction on the corresponding resize and toggle buttons. Since screen real estate is fixed, if the panel is resized, the panel is dragged to a width of $\gamma\%$ and the map width is automatically resized to $100 - \gamma\%$. Similarly, it the information panel is closed, the map is resized to take up 100% of

Scenario: Novice

User one is a first-year student at the University of Chicago. She has a personal motivation to visualize the crime patterns, because she was born and raised in Chicago. Moreover, in an introductory course, she has been tasked with identifying one or more interesting patterns in the distribution of the crimes, worthy of future investigation. With these priorities, she is unlikely to investigate the details of each crime; rather, she is more likely to browse the map itself to determine if she can recognize patterns in the spatial distribution. She is likely to focus her attention on crimes near the locations in which she lives and studies.

Scenario: Researcher

User two is a criminology professor at the University of California, Berkeley. She is interested in a recent spike in murders in Chicago she heard about in news sources, and would like to generate new hypotheses about the method and weapons of murders in the city. As she lives in California, she is not familiar with the city firsthand, but is an expert at recognizing spatial patterns in criminal incidents. She is likely to make extensive use of the attribute data associated with each crime. Focusing her attention on these attributes will cause her visual layouts to be domianted by the attribute panel, rather than the map itself.

Table 1: Narrative scenarios for the research and novice user scenarios outlined in the case study.

the page width. The default configuration includes only the map, with the information panel in the closed position.

The developers of this interface envision two potential user groups, novices and researchers. A key difference between novices and researchers is that novices are not expected to dig deeply into the details of each crime. To support a user-centered design process, the application developers developed two scenarios to describe the intended UX of the tool for each of these user groups with the help of police, neighborhood groups, and other stakeholders (Table 1).

Assigning Probabilities

Clear differences in the UX are expected between the two user groups. Indeed, it is possible to imagine, from these scenarios, how the prototypical user in each user group might interact with the tool and how their interface might be styled and displayed. pSBD probability statements for each user group are developed for formalize these differences. In this case study, two attributes regarding the interactivity and design of the interface will be considered. Specifically, the probability of using the information panel (*P(infoPanel)*) and the width of the information panel (*width*) will be considered for each user group (Table 2).

From the scenario, it is possible to infer that the novice user is unlikely to investigate the attribute data of each individual crime, and is more likely to use the main map interface for the majority of her time with the tool. Thus, she is assigned a low *P(infoPanel)*. Moreover, if the panel is open, it is likely to be rather small in comparison to the map element, indicating a small panel *width*. Specifically, the use of the information panel is modeled as the Bernoulli distribution $P(infoPanel) = 0.25, X \sim Binom(1, 0.25)$, and the width of the information component as a binomial distribution $width = 0.30, X \sim Binom(100, 0.3)$.

In the researcher's case, extensive use of the information panel and its attribute data is likely. Much of the academic research made possible from this interface requires an in-depth understanding of each incident, in addition to its spatial position. Therefore, the probability of a researcher using the information panel is high, and the use of this component is modeled as $P(infoPanel) = 0.9, X \sim B(1,0.9)$. Moreover, the researcher is likely to change the layout of the interface to expand the information panel component, further demonstrating the importance of this component to the research being done. A statement about the width of the information panel can be made to reflect this: $width = 0.65, X \sim B(100,0.65)$.

Table 2: pSBD probability statements for the case study scenarios.

Novice		
Parameter	*Distribution*	*Expectation*
$P(infoPanel)$	$X \sim Binom(1,0.25)$	0.25
$width$	$X \sim Binom(100,0.3)$	0.3
Researcher		
Parameter	*Distribution*	*Expectation*
$P(infoPanel)$	$X \sim Binom(1,0.9)$	0.9
$width$	$X \sim Binom(100,0.65)$	0.65

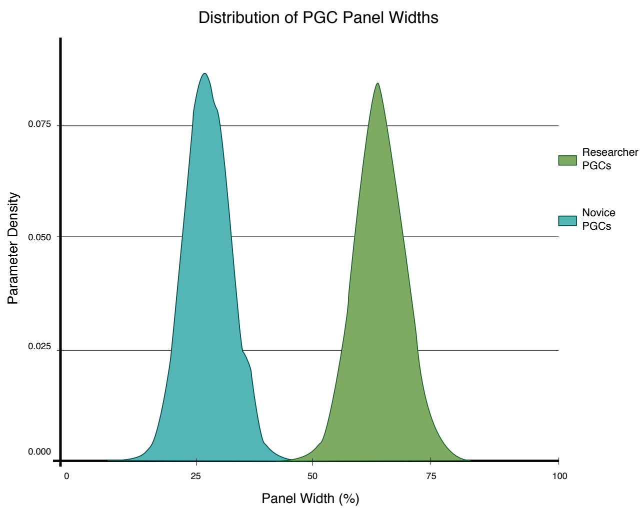

Figure 1: A density plot showing the distribution of the information panel width parameter for researcher PGCs (green) and novice PGCs (blue).

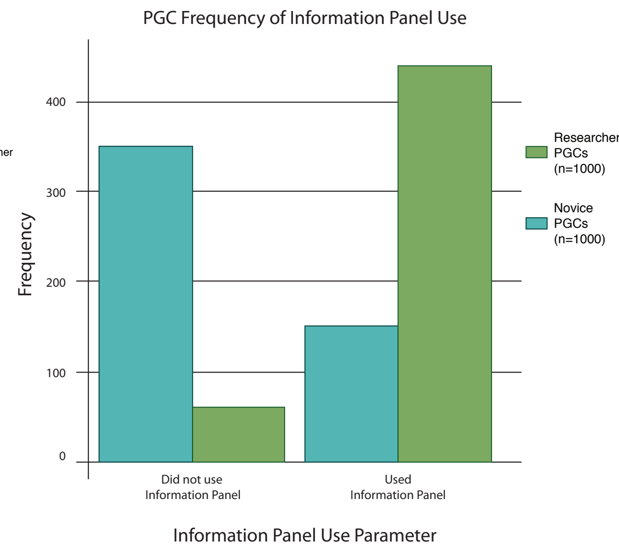

Figure 2: A bar chart showing the relative frequency of information panel use the researcher PGCs (green) and the novice PGCs (blue).

Visualization of intended UX

Draws from the distributions in Table 2 are generated using R [34], using the built-in distribution functions. Specifically, 1000 independent draws from each distribution are made to create 1000 complete PGCs for each scenario. The PGCs are then visualized in three ways. First, the densities of the width parameter for the information panel component are plotted for each user scenario (Figure 1). Second, the frequency of each scenarios' use of the information panel are graphed as a bar chart (Figure 2). These two visualization types are helpful in communicating the intended UX between design team members. By allowing a concrete visual representation of how the intended users will interact with the tool, UX designers can have an informed and grounded conversation that does not rely on weak qualitative language like 'more' or 'less'. Moreover, these plots can be produced and updated rapidly, allowing efficient visualization of the progress of UX negotiations within the design team. A third potential visualization includes the densities of each parameter overlaid on a copy of the wireframe, allowing interpretation of the probabilities in conjunction with the visual design of the interface (Figure 3).

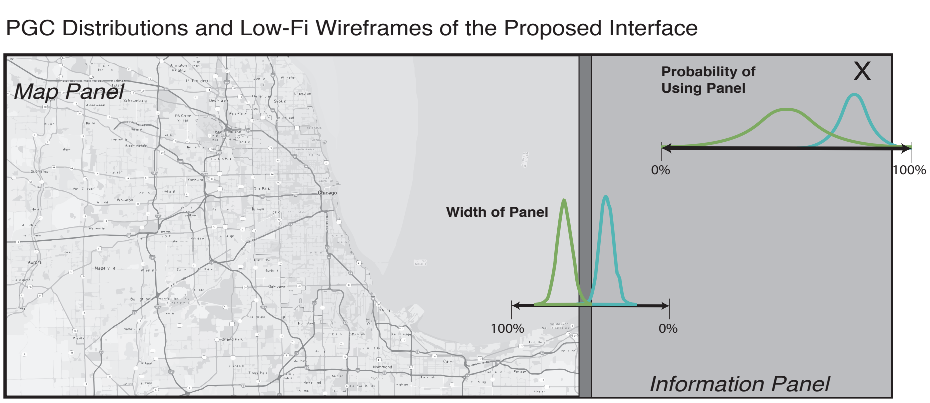

Figure 3: A low fidelity wireframe of the interface with parameter distributions overlaid.

This composite visualization may be appropriate for communicating with domain experts and other stakeholders, as it gives a holistic view of both the intended design and UX of the tool.

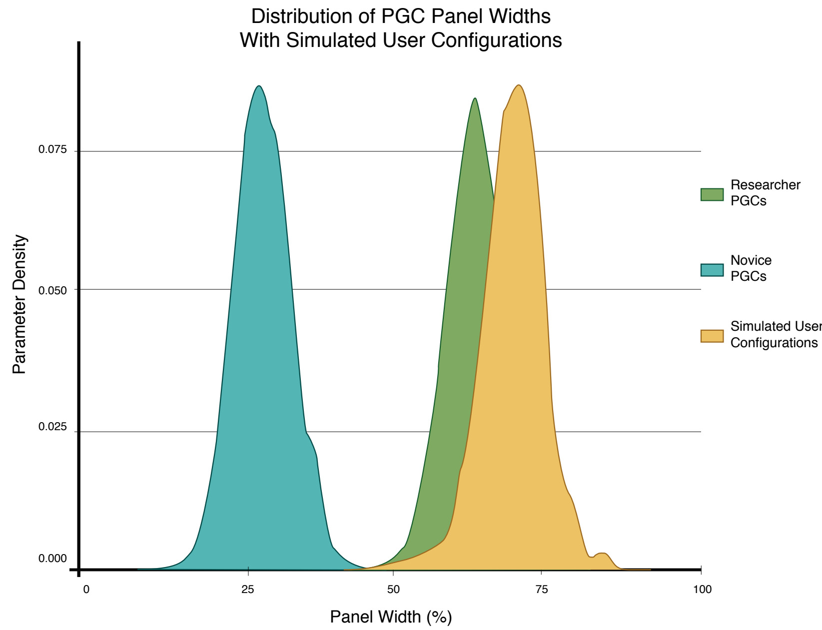

Figure 4: Panel width distributions as in Fig 1, with simulated 'real' usage data overlaid in yellow.

Assessing real usage vs. expected usage

In this hypothetical case study, the real vs. expected usage analysis will be illustrated with a 'real' dataset of simulated observations of behavior that differ from the scenario models (Figure 4).

A fuzzy c-means algorithm is applied to the combined real and pSBD dataset to determine the optimal number of clusters in the dataset ($\hat{C}$). If $\hat{C}$ is two, the real data falls within the expected usage of one of the two existing pSBD scenarios. If it is three, the real data is distinct from the envisioned usage, and it forms a coherent cluster. If $\hat{C} > 3$, the real data is distinct from the envisioned usage, forming multiple clusters.

The clusters are generated using the fanny function in the R package cluster [44]. The silhouette method of choosing the optimal number of clusters is implemented in the silhouette function, also in the cluster package. This empirical analysis determines $\hat{C}$ to be 2, indicating that the real usage falls within the parameters of expected usage envisioned by designers (Figure 5).

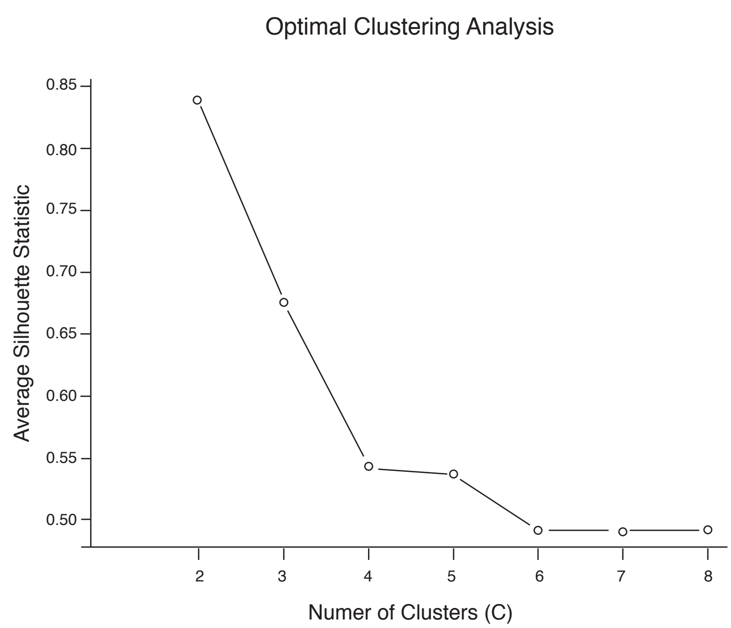

Figure 5: Average silhouette statistic for C=2 to C=8 clusters. The optimal clustering occurs that the silhouette statistic maximum at C=2 clusters.

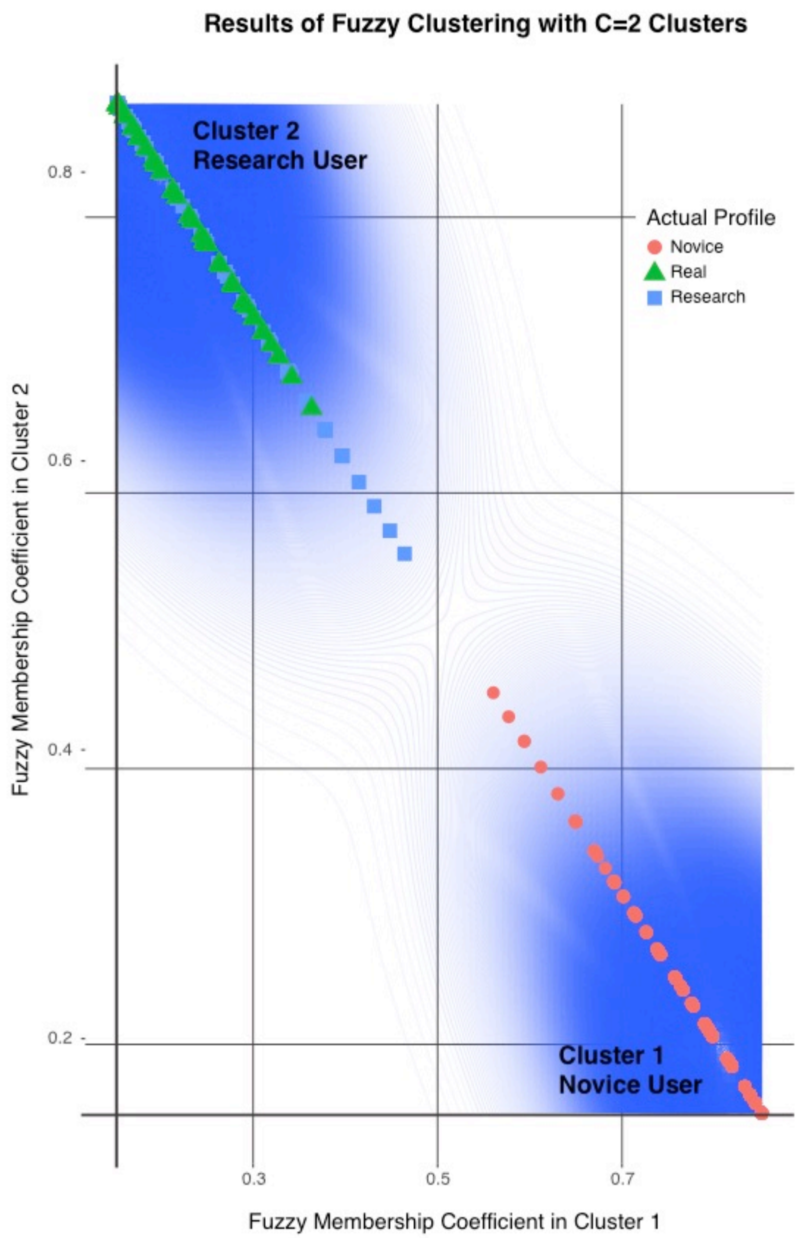

Figure 6: Results of fuzzy clustering for real and pSBD configurations. The known actual class labels are shown for each data point in red (novice PGC), blue (research PGC), and green (real usage). Notice that the real usage patterns align closely in cluster space with the research PGCs.

Once Ĉ is chosen, each point in the real and PGC datasets can be visualized in terms of its degree of membership in one of the two clusters (Figure 6). If the points are clustered in the top-left and bottom-right of the graph, each point has a high degree of membership in its respective cluster, indicating strong resemblance to the cluster prototype. Points in the middle of the graph space, on the other hand, are only weakly related to one of the two clusters. In this case, it appears that the novice and research PGC points are strongly clustered at either end of the membership spectrum and that the real usage data strongly resembles the envisioned researcher profile. At this point, the developers may want to reconsider why novice users are not using the interface in the envisioned way or implement additional measures to entice novice users to use the system.

.

Assimilating real usage data into pSBD scenarios

Finally, the real usage data is integrated with the PGCs using Bayesian data assimilation to improve the interface's default settings. In this example, the data from the previous section will be assimilated to refine the expectation of probability that a research user will utilize the information panel.

Consider a situation in which application developers are weakly confident in their quantification of the *P(infoPanel)* parameter. A beta-distribution, specified in terms of effective sample size and expected value, can be used to quantify the developers' degree of belief about this parameter in terms that are direct comparable with the real usage data. Specifically, because the developers are only weakly confident in their assessment of expected behavior, they may

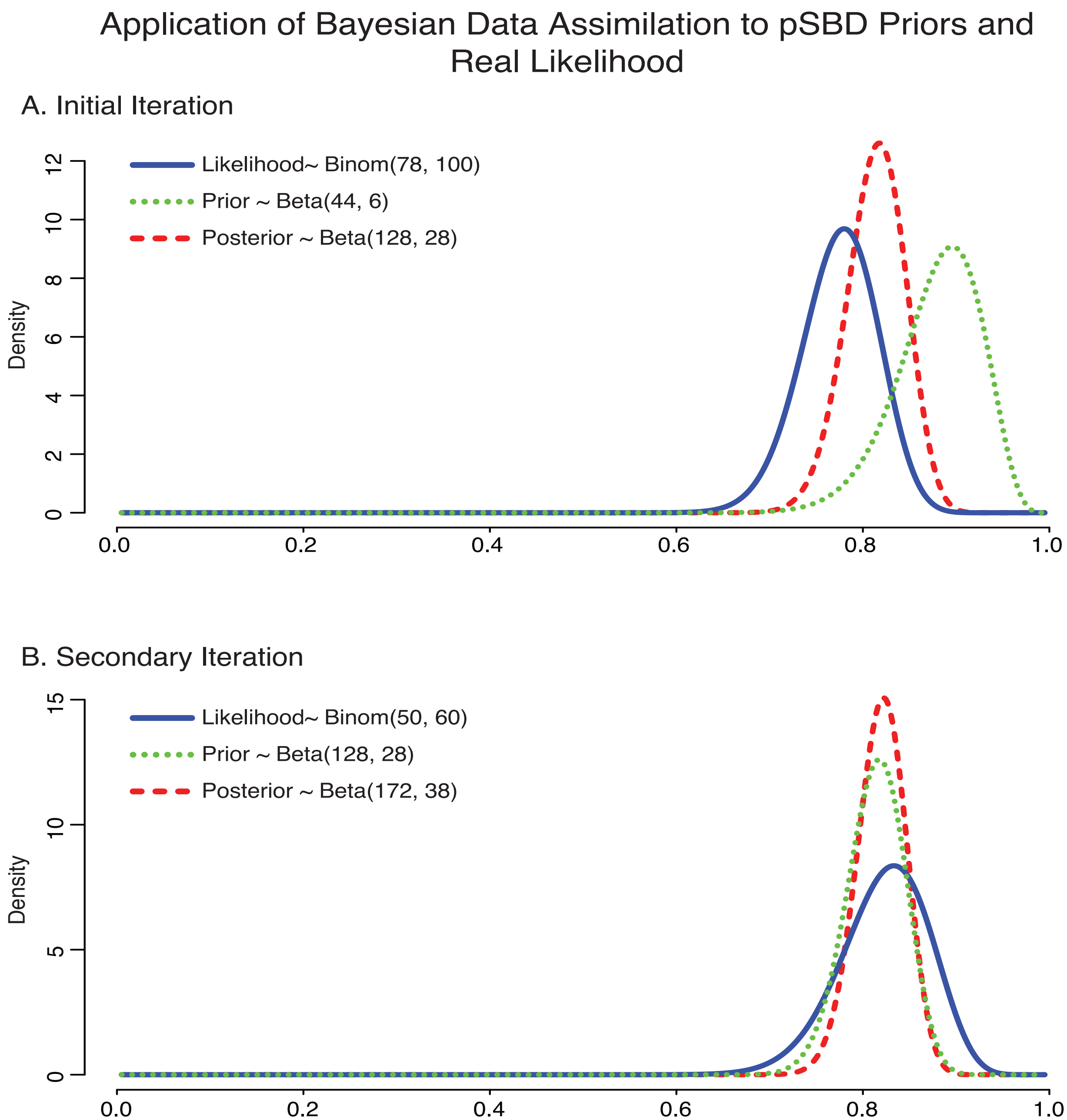

Figure 7: Illustration of the data assimilation processed discussed in the case study. During the first iteration, the observed data is assimilated with the pSBD scenario. During the secondary iteration, new observed data is assimilated with the posterior distribution of the initial iteration. Notice the high degree of confidence in the secondary posterior.

determine the effective sample size of their estimate to be $\nu = 50$. This would correspond to a degree of belief in their scenario that approximately equals fifty observations of that behavior. Taking the expectation from Table 2 of $m = 0.9$, the shape parameters of the beta distribution are found to be $\alpha = 44$ and $\beta = 6$. In the real data described above, there were 78 uses of the information panel out of a sample of 100 observations of user behavior. The likelihood distribution for this data, then, is $X \sim Bern(0.78)$. Applying Bayes' Theorem, the shape parameters of the posterior distribution are shown to be $\alpha = Y + (n * m) - 1 = 122$ and $\beta = N - Y + (n * (1 - m)) - 1 = 28$. To interpret these new shape parameters, α and β are converted back into terms of effective sample size and expected value. In this case, the new expected value of the posterior is $m = 0.813$ and the new effective sample size is $\nu = 150$. Note that this sample size is the effective sample size of the prior ($\nu = 50$) plus the number of observations in the data ($N = 100$). Moreover, the expectation of the posterior is a weighted average of the expected value of the likelihood (0.78) and the prior (0.90). Taken together, this information suggests that a future research has an 81% chance of using the information panel (Figure 7A).

As new observations of the use of the interface are observed as real users interact with the system, data assimilation can be repeated an arbitrary number of times. Each time, the posterior from the previous iteration of the assimilation is taken as the prior for the current iteration. For example, if an additional 60 observations of interface usage are recorded, of which 50 use the information panel, those observation can be assimilated to obtain new behavior estimates ($\alpha = 172$, $\beta = 8$, Figure 7B). The new estimates forecast that a future user has an 82% chance of using the information panel, with decreased uncertainty surrounding that estimate.

.

Discussion

Advantages of pSBD over traditional SBD

The case study described above demonstrates the visual and analytical power of the pSBD process. One of the most important advantages of pSBD over traditional SBD is the ability to rapidly generate visualizations of intended UX for the proposed system, even before development has started. Traditional SBD often uses qualitative language to communicate the differences among scenarios. By introducing visuals into the design process, pSBD promotes visual communication in the presentation of abstract thoughts and ideas about the intended UX, replacing the weak language with a concrete visual model [45]. Throughout science and industry, tools that support visual analytics have been a priority in new tool development, attempting to leverage the strengths of both human and computerized data processing. Broadly, these tools foster constructive evaluations and improvements in our mental models of physical processes that can improve both human decision making [46]. Visualizations of complex and unstructured data, such as the qualitative descriptions described in SBD, can offload cognitive processes to perceptive processes, allowing new features to emerge that are easily picked up by the visual system [47,48]. Therefore, designers using pSBD may uncover unexpected patterns within and among different use case scenarios.

A second advantage of pSBD is that it allows the development team to determine if an early stage application is truly serving its target user groups. Applications that are developed using a user centered approach typically involve many iterations of feedback from both domain experts and target users. During this processes, priorities for what the intended interface should accomplish are identified. Developers and designers should ensure that these priorities are actually met by the delivered application. pSBD allows developers to assess whether these goals are being met, and if not, how the real users actually use the system. Moreover, pSBD can be used with an application very early in the design process, and does not require expensive and time consuming methods of gathering user feedback traditionally employed during iterative user centered design.

pSBD can also be helpful in refining the application's look, feel, and interactivity on the fly to improve the UX for target user groups. The interface can be configured to adapt to the updated posterior distributions to take into account both the developer's intuition and real observation data on a scenario-by-scenario basis. Effective default designs for each component are an important part of any interactive application, and these defaults can be smartly tailored in real time to the preferences of the application users. If it appears that real usage is contrary to the developers' expectation component, the design can be smartly adapted without developer input. Moreover, as data assimilation is performed, the interface can automatically change to create an interface that best suits its target users. Furthermore, an interface built on pSBD assimilation can evolve with its user base through time. Should preferences and usage patterns be modified as the application ages, the change in interaction strategy will emerge from the data and can be automatically incorporated. This is an advantage over some AI-based personalization systems that require an assumption of static preferences through time.

Future Research Priorities

While the current study introduced pSBD 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. The current pSBD framework is limited by drawing from independent probability distributions. Draws from a conditional distribution (e.g., parameter $A = x$ if and only if parameter $B = y$), though more difficult to construct, would allow more powerful statements to be constructed. An efficient method for describing BBN design state-spaces would allow conditional probability to be incorporated into pSBD.

Conclusion

In this paper, I introduced probabilistic scenario-based design, a new method for underpinning traditional narrative scenarios in a user-centered design process with statistical distributions in an operational way. 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 thoughts. 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 also described and are useful for communication between developers and stakeholders.

References

[1] 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.

[2] 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.

[3] 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.

[4] Isbell CL, Pierce JS. An IP continuum for adaptive interface design. Proceedings of HCI International Conference 2005.

[5] Benson BJ, Bond BJ, Hamilton MP, Monson RK, Han R. Perspectives on next-generation technology for environmental sensor networks. Frontiers in Ecology and the Environment 2010;8:193–200. doi:10.1890/080130.

[6] Linden G, Smith B, York J. Amazon. com recommendations: Item-to-item collaborative filtering. IEEE Internet Computing 2003;7:76–80.

[7] Rosson MB, Carroll JM. Scenario based design. Human-Computer Interaction Boca Raton, FL 2009:145–62.

[8] Carroll JM. Five reasons for scenario-based design. Interacting with Computers 2000;13:43–60. doi:10.1016/S0953-5438(00)00023-0.

[9] 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.

[10] 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.

[11] Kazman R, Abowd G, Bass L, Clements P. Scenario-based analysis of software architecture. IEEE Software 1996;13:47–55.

[12] Shneiderman B. Designing the user interface: Strategies for effective human-computer interaction. Pearson Education India; 2010.

[13] Cunha B, Pereira JP, Gomes S, Pereira I, Santos JM. Using personas for supporting user modeling on scheduling systems. Proceedings of IEEE HIS; 2014.

[14] Zukerman I, Albrecht DW. Predictive statistical models for user modeling. User Modeling and User-Adapted Interaction 2001;11:5–18.

[15] Young S. Cognitive User Interfaces. IEEE Signal Processing Magazine 2010;27:128–40. doi:10.1109/MSP.2010.935874.

[16] Yannakakis GN, Spronck P, Loiacono D, André E. Player modeling 2013;6.

[17] Liu J, Wong CK, Hui KK. An adaptive user interface based on personalized learning. IEEE Intelligent Systems 2003;18:52–7.

[18] Conati C, Gertner A, Vanlehn K. Using Bayesian networks to manage uncertainty in student modeling. User Modeling and User-Adapted … 2002.

[19] Horvitz E, Breese J, Heckerman D, Hovel D, Rommelse K. The lumiere project: Bayesian user modeling for inferring the goals and needs of software users. In:. Proceedings of the fourteenth conference on uncertainty in artificial intelligence, Morgan Kaufmann Publishers Inc. 1998, pp. 256–65.

[20] 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.
[21] Rosson MB, Carroll JM. Scenario-based design. In:. The human-computer interaction handbook, L. Erlbaum Associates Inc. 2002, pp. 1032–50.
[22] Rosson MB, Carroll JM. Scenario-Based Design. 2005.
[23] Gelman A, Carlin JB, Stern HS. Bayesian data analysis 2014;2.
[24] Ellison AM. An Introduction to Bayesian Inference for Ecological Research and Environmental Decision-Making. Ecological Applications 1996;6:1036–46. doi:10.2307/2269588.
[25] 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.
[26] Cheng J, Bell DA, Liu W. An algorithm for Bayesian belief network construction from data. In:. Proceedings of ai & stat'97, 1997.
[27] Fischer G. User Modeling in Human-Computer Interaction. User Modeling and User-Adapted Interaction 2001;11:65–86. doi:10.1023/A:1011145532042.
[28] Fan X, Wang J. Bayesheart: A probabilistic approach for robust, low-latency heart rate monitoring on camera phones. In:. Proceedings of the 20th international conference on intelligent user interfaces, ACM; 2015, pp. 405–16.
[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] R Core Team. R: A language and environment for statistical computing. Vienna, Austria: R Foundation for Statistical Computing; 2017.
[35] Rossum G. Python reference manual. Amsterdam, The Netherlands, The Netherlands: CWI (Centre for Mathematics; Computer Science); 1995.
[36] Hartigan JA, Wong MA. Algorithm as 136: A k-means clustering algorithm. Journal of the Royal Statistical Society Series C (Applied Statistics) 1979;28:100–8.
[37] Kanungo T, Mount DM, Netanyahu NS. An efficient k-means clustering algorithm: Analysis and implementation. IEEE Transactions on … 2002;24:881–92.
[38] Bezdek JC, Ehrlich R, Full W. FCM: The fuzzy c-means clustering algorithm. Computers & Geosciences 1984;10:191–203.
[39] Song Q, Shepperd M. Mining web browsing patterns for E-commerce. Computers in Industry 2006;57:622–30. doi:10.1016/j.compind.2005.11.006.
[40] 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.

[41] Carlin BP, Polson NG. Inference for nonconjugate Bayesian Models using the Gibbs sampler. Canadian Journal of Statistics 1991;19:399–405. doi:10.2307/3315430.
[42] Department CP. Clear map crime incidents 2017. http://gis.chicagopolice.org/clearmap/startpage.htm# (accessed May 6, 2017).
[43] Ali T. Where shootings have occurred in chicago since 2010 (map) 2016. https://www.dnainfo.com/chicago/20150707/downtown/where-shootings-have-occurred-chicago-since-2010-map (accessed May 6, 2017).
[44] Maechler M, Rousseeuw P, Struyf A, Hubert M, Hornik K. Cluster: Cluster analysis basics and extensions. 2016.
[45] Cyrs TE. Visual Thinking: Let Them See What You Are Saying. New Directions for Teaching and Learning 1997;1997:27–32. doi:10.1002/tl.7104.
[46] Keim D, Andrienko G, Fekete J-D, Görg C, Kohlhammer J, Melançon G. Visual Analytics: Definition, Process, and Challenges. In:. Information visualization, Berlin, Heidelberg: Springer Berlin Heidelberg; 2008, pp. 154–75. doi:10.1007/978-3-540-70956-5_7.
[47] Pomerantz JR, Pristach EA. Emergent features, attention, and perceptual glue in visual form perception. Journal of Experimental Psychology: Human Perception and Performance 1989;15:635.
[48] Hegarty M. The cognitive science of visual-spatial displays: Implications for design. Topics in Cognitive Science 2011;3:446–74.