



JOSEPH KRAMER, SUNIL NORONHA, AND JOHN VERGO

A USER-CENTERED DESIGN APPROACH to Personalization

The key to successful design is grounding the choice of features and tools upon value to the end user.

Personalization is a toolbox of technologies and application features used in the design of an end-user experience. Features classified as “personalization” are wide-ranging, from simple display of the end-user’s name on a Web page, to complex catalog navigation and product customization based on deep models of users’ needs and behaviors. Similarly, personalization technologies range from commonplace use of databases, cookies, and dynamic page generation, to esoteric pattern matching and machine-learning algorithms, rule-based inferencing, and data mining.

As with any other disorganized assortment of tools—especially fascinating new tools—designers are often drawn into the trap of trying to find uses for the tools, and deploying the coolest new features, forgetting their primary focus should be on providing value to the end user. So we often hear people ask, “Is this recommendation engine (or technology X) worth its fancy price, or is it just the fad of the

moment? Would personalization feature Y provide real value to my customers? How important is it relative to other capabilities that I could implement? How should I use it, that is, when and where in my overall design does it fit?”

This article presents a framework for analyzing the role of personalization in the design of any user experience. In a nutshell, we advocate using a six-

step user-centered design (UCD) process we developed over the course of designing object-oriented interfaces for e-commerce applications. This participatory design process generates each design through explicit elicitation and modeling of users' goals, beliefs, and behaviors, and ensures the resulting personalized system delivers genuine value to the user.

The first step in the UCD process is determining the target user segments of the experience being designed, for example, the "most-valued customers." This step is often guided by the market research processes that define the total pool of potential customers, segment the customers based on their wants and needs, and evaluate each market segment's attractiveness [2]. However, it has significant technical implications for the personalization capabilities that will have to be supported.

An important implication pertains to automated identification of the target user segments: How do we detect if a given user belongs to a target segment?

who the target users will be, such that their identification by the system can be practicably automated.

We use standard task analysis methods to learn the impetus for users' actions ("triggers"), methods of completing the tasks ("processes, use cases"), and the ultimate intention of the user ("goal") [1]. Goal-decomposition graphs can be used to map high-level goals to specific subgoals and to the tasks that achieve those subgoals. Further detail on the task flows is captured through activity diagrams. This step must be done with enough detail to clearly understand the different triggers, processes, and goals employed by different subtypes of users within the target user group, since a given trigger can result in different goals for different users, and a given goal may be accomplished through different tasks.

The branch points in the goal trees and activity flows are key control points for personalizing the user experience. For example, the task of "buying a new computer," for an expert may represent the sub-

—DESIGNERS ARE OFTEN DRAWN INTO THE TRAP OF TRYING TO—
FIND USES FOR THE TOOLS, AND DEPLOYING THE COOLEST NEW
FEATURES, FORGETTING THEIR PRIMARY FOCUS SHOULD
—BE ON PROVIDING VALUE TO THE END USER.—

For example, marketing may have concluded that one of the most profitable customer types is the "technically savvy small-business owner who currently owns a home PC." Even if we knew how to design the perfect experience for this type of user, we would not be able to deliver the experience unless we could identify that a given user fits this profile. In other words, we would need to design ways to assess the technical knowledge of the user, their business size, and so on. This process forms a set of key personalization requirements. Market research is typically conducted via large, rather intrusive, questionnaires; administering a similar quiz to a user on a Web site is generally impractical. Further, these diagnostic questions may not be easily ported to different media.

An alternative way to deal with this problem is to exchange the technical design challenge for a market analysis challenge: Given a set of user profile elements that we know we can feasibly acquire, base the user segmentation analysis solely on this set. At the end of this step, all the stakeholders, including the business and technical teams, should be in agreement as to

goal "Let me build my ideal machine," which can be designed as a detailed configuration. However, a novice may express a subgoal as "Help me understand which one of your standard offerings makes the most sense for me" which can be better designed through needs-based recommendations.

Goal and task analysis goes a long way toward shaking out the personalization features and tools that are (not) important for a given project because each task can be traced up the goal tree to determine value to the end user. We emphasize a holistic view; task analysis produces one big, structured pile of requirements. Requirements that comprise tailoring of the task based on some user information are classified under the rubric of personalization, but are evaluated no differently from the others. It may turn out that an anticipated feature (for example, a personalized shopping assistant), does not show up as an important feature in the pile, perhaps because users aren't interested in recommendations for the given product space or it conflicts with other, more important, requirements (for example, most users may prefer to step through a complete product hierarchy,

reassured they are not being excluded from some parts of the product catalog). Conversely, it may turn out that an ordinary-looking requirement such as comparing two PCs in an online catalog turns into a challenging new personalization requirement because the user wishes to compare against a PC he or she has at home. Either way, much of the confusion regarding the value of different personalization features is cleared away by the end of this step.

Exploring the Blue Sky

Following the task analysis, it is useful to conjecture with your target users what would be the ultimate desirable set of triggers, processes, and goals. This step has been called “blue sky” as it attempts to derive the triggers, goals, and the desired intermediate processes without worrying about any real-world constraints [2].

The primary purpose of a blue-sky exercise is to provoke the creation of novel (and more *valuable*) task flows. Often, the tasks identified in the task-analysis phase are derived from users’ current practices, which are overly constrained by assumptions about what can be implemented. Thus, the task of “shopping for groceries” may lead to a Web site design that mimics a physical grocery store, which is far from optimal. It takes blue-sky thinking to generate an alternative task flow in which “not shopping at all” is considered an implementation of this task, that is, having the grocery store predict one’s routine grocery needs, and manage one’s refrigerator. As in this example, and most others (for example, supplying the perfect clothes, assembling the perfect PC), blue-sky task flows require considerable knowledge of the user and are likely to provoke exploitation of new personalization technologies.

To design for the user, we need to base our designs on the user’s mental models of the domain (language, concepts, beliefs, among others). Therefore, the fourth step in our UCD process is the elicitation and documentation of the user’s object model. This step is achieved by extracting noun phrases and verbs from the use cases and defining them with your users. Subsequently, the users iterate on the conceptual relationships that form their mental model of the domain. The results of this analysis are captured in artifacts such as cognitive maps; our preferred technique is object diagrams.

Ideally, everyone would speak the same language, and there would be a single model of the application’s content (object model). However, the fact is that different users use different object definitions and have different gaps in their understanding (for example, different levels of knowledge about the

attributes of objects and their relationships). Designers must deal with the differing object definitions by refining the terminology used during the experience, or by educating the user. For example, we found some users define workstations as networked machines, while others define them as high-performance PCs. The preferred way to deal with this is to come up with more specific product categories, which capture the different capabilities and shades of meaning (“networked PC”), ensuring these product names match most users’ object definitions. The other gap that can exist is disparity in understanding that results from population differences (for example, technically knowledgeable vs. eager novices). This can be mitigated by delivering different levels and modes of education to different users (personalization of help facilities, information detail, mode of presentation of information).

On a completely different note, the object model is key to mass customization, which is the most powerful form of personalization. Mass customization is based on assembly of components, and judicious componentization is key to managing manufacturing costs. The components and assemblies are reflected in the object model as artifacts the user selects and puts together. If there is a gap between what a user expects as a configurable component and what is actually supplied, there is a major ease-of-use cost (and subsequent economic loss). During the design of the componentization and the assembly process, designers must elicit the mental constructs (a.k.a. object model) most meaningful to the user (for example, base product configurations defined via distinctive value proposition statements) and map them to manufacturable systems.

While the previous steps have focused primarily on creating models of different facets of the user (goals, tasks flows, mental models), the fifth step of the process combines all of them into an end-to-end analysis of each potential user experience. We step through every flow in each task, determining the information (object properties and relationships) that must be accessed to complete each action, and documenting it via interaction diagrams. Visual treatment rules are then applied to design the “views” (renderings of the accessible information of each object) the user ultimately sees.

Since different user types have different information needs, this step results in identification of multiple views, which are used as the basis for personalization. Also, the interaction sequences help pinpoint the kind of persistent state information

—PERSONALIZATION IS CURRENTLY BEST VIEWED—
AS AN EVOLVING SET OF TOOLS. AS CRAFT-PERSONS, OUR
JOB IS TO UNDERSTAND THE CAPABILITIES, LIMITATIONS, AND
—END-USER COSTS OF THE TOOLS WE BRANDISH.—

(the context, for example, the status of the user's transactions, tasks, and so on) that must be maintained by the system as the user moves through the experience. Some of these interaction sequences may be long-running (spanning weeks, months, or longer), for example, relating current browsing for PC accessories to last year's purchase of a desktop machine, and pose significant challenges to designing the interaction. Modeling and managing the state of the long-term relationship with the user is the Holy Grail of the one-to-one enterprise [4].

As always, we iterate through several designs, evolving from a low-fidelity prototype to a fully functional system, improving through testing and participatory design. It is worth noting that in testing systems incorporating personalization, special attention must be given to privacy concerns. Privacy and trust are issues that should be dealt with right from the beginning of the UCD process, but it has been our experience that despite one's best attempts to do so, there are always surprises that come out during testing. Innocuous questions about when the user wishes to have a PC available for delivery (motivated by the designers' knowledge that users are keenly interested in the availability of products) backfire when users interpret it as an insidious marketing ploy to bombard them with marketing information until the date they need the machine. Questions about the user's budget, intended to filter the product space to recommend only PCs that fall within the users desired price range, trigger a suspicion that this will affect the recommended products' prices in other ways. Beyond rewording questions correctly, an important design technique that must be superimposed on the use of any personalization feature is to ensure that users clearly understand why the question is being asked, and how it directly fits in with their goals. This means the consequences of the user's actions must be transparent to the user; the model underlying the feature should be simple enough to engender trust. During testing it is necessary to explicitly probe and measure how well users understand the benefits, and that the total value is adequate to justify the inclusion of the feature in the system.

Conclusion

Current methods of software system design do not adequately capture the needs and values of end users. We have presented a UCD methodology that, at its core, maintains a strong focus on bringing value to the end user. From our perspective, personalization takes a back seat to end-user value. We have given examples of how the use of personalization might result in decreased value to the user. A designer who sets out to personalize an existing application, or to build a new personalized application, is poised to make the classic error of putting technology before the needs of the end users. Personalization is currently best viewed as an evolving set of tools. As craft-persons, our job is to understand the capabilities, limitations, and end-user costs of the tools we brandish. We must make good decisions as to when to reach into our personalization toolbox. When we do, testing, measuring, and iterating on our designs, with an unwavering focus on delivering value to the end user, will help insure the success of the applications we build. ■

REFERENCES

1. Collins, D. *Designing Object-Oriented User Interfaces*. Benjamin-Cummings, Redwood City, Calif., 1995.
2. Dayton, T., McFarland, A., and Kramer, J. Bridging user needs to object-oriented GUI prototype via task object design. In L.E. Wood (Ed.), *User Interface Design*. (1998) CRC Press, Boca Raton, Fla., 15–56.
3. Kotler, P., and Armstrong, G. *Principles of Marketing*. Prentice-Hall, Upper Saddle River, N.J., 1999.
4. Peppers, D. and Rogers, M. *Enterprise One to One: Tools for Competing in the Age of Interactivity*. Currency Doubleday, New York, N.Y. 1997.

JOSEPH KRAMER (kramerjo@us.ibm.com) is a senior application development analyst; **SUNIL NORONHA** (noronha@us.ibm.com) is a research staff member; and **JOHN VERGO** (jvergo@us.ibm.com) is a research staff member at IBM T.J. Watson Research Center, Hawthorne, NY.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.