

Reasoning About User Preferences

Tamara Babaian

Bentley College, USA

R

INTRODUCTION

Personalization has been identified as a key task to the success of many modern systems. As Riecken (2000, p. 28) writes in the editorial of the special issue of the *Communications of the ACM* devoted to this subject, “personalization means something different to everyone.” Personalization can take various forms; however, it can be broadly described as the set of mechanisms by which an application is tailored to a particular end user and his or her goal.

Modern systems typically have a large set of features designed to carry out a multitude of tasks, or operate using an enormous wealth of information available on the Internet. The effectiveness with which a system is able to help its user achieve the desired effect, as well as the user’s satisfaction from interacting with the system, depend critically, among other factors, on the user’s ability to identify and use relevant, customizable options. Configuring the system for optimal performance requires that the user specifies his or her individual preferences related to many different tasks. However, the user’s ability to provide this kind of personal information is often greatly impaired by the following drawbacks in the way personalization is implemented.

- Customization is carried out as a separate process that is taken out of context of the task in which such personal information is used, thus obscuring from the user the purpose and advantages of supplying such information.
- The amount of potentially useful personal information is sometimes overwhelming, thus the systems are installed with a set of settings that are considered typical. Further customization has to be initiated by the user. However, inexperienced users rarely take advantage of customization even if they are aware of potential benefits due to the lack of information on available options. As a result, experience demonstrates (Manber, Patel, & Robison, 2000) that many users shy away from customization when they can benefit from it a great deal.

The items above characterize the shortcomings of a typical process of configuring customizable features. On the other hand, there are problems that the developers of

software face in *designing* for personalization. As Pednault (2000) points out, the underlying representation of “the human-side and the technology-side” is the key. However, representations currently in use, at times, lack flexibility to be easily adjustable and reusable. This is largely a consequence of the absence of a rigorous model of what constitutes personalization. The lack of such model results in ad hoc representations used by most systems.

The approach to personalization that we present here is inspired by the view of interfaces as the means of collaboration between a human user and a computer system, articulated by Shieber (1996), in which the collaborative model of human-computer interaction is contrasted to the view of a system as a mere set of tools available to the user. As a theoretical framework, collaboration theory and its existing philosophical and formal mathematical accounts (Bratman, 1992; Cohen & Levesque, 1991; Grosz & Kraus, 1996) can inform both design and usability analysis of systems, and highlight problems that need to be addressed to make interfaces better collaborative partners. Examples of interfaces that have been created following this view have already been built and are described in Babaian, Grosz, & Shieber (2002), Ortiz and Grosz (in press), Rich, Sidner, and Lesh (2001), and Ryall, Marks, and Shieber (1997).

Theories of collaboration postulate:

1. commitment of the parties to a shared goal, and
2. sharing of knowledge and communication in an effort to establish agreement and mutual knowledge of the recipe for completing the task

as the key features of a collaborative activity. Stemming directly from this view, in our approach the collaborator system has the ability to elicit personal information at the time when the user is most motivated to provide it, that is, when the system is processing a task for which such information is critical. The novelty of our approach and its implementation also lies in defining the task of collecting personal information declaratively via informational goals and preconditions on the actions that the system takes in response to a user’s request. This is enabled by

- (a) the use of a knowledge base that stores all gathered preference information, and

- (b) an automated reasoning and planning system that can reason autonomously about knowledge, lack of knowledge, and actions that the system may take to acquire the necessary but missing information.

The system can perform information gathering autonomously (by inspecting available personal information, such as, for example, a person's Internet bookmarks) as well as by direct user querying. This approach to personalization ensures gradual adaptation of the system to the user's preferences. At the same time, the declarative definition of personal information and its relationship to system actions make it easy to fine-tune personalization options, resulting in a more easily adjustable and extendable design.

BACKGROUND

The problem of end-user tailoring, also known as customization of software, is not new (see, for example, Morch, 1997). Recent explosion of the Internet and its ubiquity in our everyday life have created new challenges and opportunities for the advancement of research on this subject, in particular, in the area of customizing information-access interfaces. Numerous works have addressed the issue of information overload and the resulting need for effective information retrieval and presentation of the results that is tailored to the needs of each individual visitor. A thorough review of these works is beyond the scope of this chapter; however, we briefly describe the leading approaches and provide the reader with a set of further references. Availability of logs of Web site usage has provided an excellent opportunity and an exciting domain for technologies such as machine learning and data mining (for a review see Anderson, 2002; Pierrakos, Paliouras, Papatheodorou, & Spyropoulos, 2003). In the artificial-intelligence community, two approaches to automated personalization on the Web have been explored and used most successfully: adaptive Web sites and collaborative filtering. Adaptive Web sites and Web site agents (e.g., Pazzani & Billsus, 1999; Perkowitz & Eltzioni, 2000) attempt to dynamically tailor the layout and the contents of a Web site or suggest a navigation path for each individual user by observing the user's initial interaction with the Web site and matching it to the previously observed behaviors of others. Likewise, collaborative filtering (Amazon.com is probably the most familiar example) is a technique that creates recommendations that are based on the choices of previous users with similar interests or requests.

Recently, many traditional concepts and techniques of artificial intelligence have been applied in the area of

intelligent user interfaces, in particular, to interface personalization (we refer the reader to the *Proceedings of the International Conference on Intelligent User Interfaces*). An article by Weld et al. (2003) surveys a set of research projects aimed at developing representations and methods for user-directed customization and automatic system adaptation of various kinds of application interfaces.

Many applications of machine-learning and data-mining technologies to Web-based computing have been enabled by the availability of logs recording various details of the interaction of millions of users with the Web sites. At the same time, non-Web-based systems (e.g., common desktop editors, spreadsheets, etc.) have also benefited from the emerging culture of personalization and now commonly incorporate a few personalizable features. Nevertheless, the advancement of research in personalization of common desktop applications has been lagging behind, partly due to the absence of detailed data on their actual usage. The method of software customization presented in this article is applicable to a broad set of software tools and not limited to just Web-based systems.

FUTURE TRENDS

Goal-Directed Personalization in Writer's Aid

Writer's Aid (Babaian et al., 2002) is a system that works simultaneously with an author writing a document, helping him or her with identifying and inserting citation keys, and autonomously finding and caching papers and associated bibliographic information from various online sources.

At its core, Writer's Aid contains

- (a) a knowledge base that contains a system's knowledge about the state of the world, and
- (b) an automated planning system.

The planner is given a description of the actions that Writer's Aid can execute and works to automatically combine these actions into a plan that will achieve a specified goal. Each action is described via preconditions that must be true prior to executing the action, and effects that the action brings about. Plan generation is accomplished by representing both goals and actions using a logic-based language and using a specialized reasoning engine that can infer what is true after performing a sequence of actions. For an example, consider the following action of searching a user's personal directories for bibliographic collections.

4 more pages are available in the full version of this document, which may be purchased using the "Add to Cart" button on the publisher's webpage:

www.igi-global.com/chapter/reasoning-user-preferences/14622

Related Content

An E-Commerce Business Model of Peer-to-Peer Interactions among Consumers

Jason Agnew and Birud Sindhav (2009). *Journal of Cases on Information Technology* (pp. 12-21).

www.irma-international.org/article/commerce-business-model-peer-peer/3241

Principles of Advanced Database Integrity Checking

Hendrik Decker (2005). *Encyclopedia of Information Science and Technology, First Edition* (pp. 2297-2302).

www.irma-international.org/chapter/principles-advanced-database-integrity-checking/14602

'Stream of Training' Approach in Project Management Training

David G. Vequist IV and Mark S. Teachout (2010). *Information Resources Management: Concepts, Methodologies, Tools and Applications* (pp. 742-750).

www.irma-international.org/chapter/stream-training-approach-project-management/54513

Agile Project Management in Product Life Cycle

Haluk Altunel (2017). *International Journal of Information Technology Project Management* (pp. 50-63).

www.irma-international.org/article/agile-project-management-in-product-life-cycle/177291

Implementing Virtual Career Counseling and Advising at a Major University

Shane Haberstroh, Shirley Rowe and Stefanie Cisneros (2009). *Journal of Cases on Information Technology* (pp. 31-44).

www.irma-international.org/article/implementing-virtual-career-counseling-advising/4118