The Traits of the Personable

Naren Ramakrishnan

Department of Computer Science,
Virginia Tech, VA 24061, USA.
Email: naren@cs.vt.edu,
WWW: http://people.cs.vt.edu/∼ramakris

Abstract. Information personalization is fertile ground for application of AI techniques. In this article I relate personalization to the ability to capture partial information in an information-seeking interaction. The specific focus is on personalizing interactions at web sites. Using ideas from partial evaluation and explanation-based generalization, I present a modeling methodology for reasoning about personalization. This approach helps identify seven tiers of 'personable traits' in web sites.

1 Introduction

Web personalization has become so pervasive that, as an enabling technology, it transcends a constantly growing set of applications in electronic commerce, knowledge management, information access, social schemes for decision making, and user interfaces. In some application contexts, personalization has come to occupy such a central role that it is now difficult to imagine a user experience without it. For instance, Riedl [1] estimates that there are at least 23 different types of personalization at Amazon's e-commerce site!

The word 'personalization' lends itself to many individual interpretations, all of which indisputably provide some form of customization. There are broadly two schools of thought. The first adopts the viewpoint that to qualify as personalization research, an approach must employ some form of *user model*, obtained implicitly or explicitly. The notion of user model is itself a rich one, and can range from simple aggregations of usage patterns by analyzing weblogs [2, 3] or transaction records to richer representations of capabilities, interests, and preferences, e.g., see research in adaptive hypermedia [4]. The second school of thought de-emphasizes user models in favor of a flexibility of information access, typically via multiple interaction pathways or dialogs through a site. Here the idea is that by placing fewer constraints on interaction, the user experience can be more personalized, although there is no 'understanding' of the user *per se* by the system. Examples here are faceted browsing interfaces [5] and conversational systems [6].

This chapter grew out of an attempt at trying to answer the question: *What does it mean for a web site to be personable?* Rather than stop at the cliched observation that there are many forms of personalization [7], we are interested in deriving some long lasting attributes of personalization solutions, especially with an eye toward accommodating both schools of thought mentioned above. Of course, this is a difficult goal and we will necessarily make some simplifying assumptions. Nevertheless, the ideas described here are not too abstract as they capture a wide variety of practical personalization situations, referred to here as *traits*.

2 Personalizing Interaction

Let us start with the working assumption that a website is personable if it allows a user's information seeking goals to be met *effectively*. A user's interaction with a web site can be thought of as a dialog between the user and the underlying information system, using the communication facilities afforded by the web site. Thus, when the user clicks on a hyperlink or submits data in a form, information is implicitly communicated from the user to the system. In response, the system presents information back to the user (including opportunities for further user input). Many such dialogs happen in a browsing context.

Consider a hierarchical US Congressional website, where the user progressively makes choices of politician attributes—*state* at the first level, *branch* at the second level, followed by levels for *party*, and *district/seat*—by browsing (see Fig. 1). Imagine how a user would pursue the following tasks:

1. Find the webpage of the Democratic Representative from District 17 of Florida.
2. Find the webpage of each Democratic Senator.

The first task can be satisfied by typical drill-down browsing because it involves supplying responsive information at each level (click 'Florida' first, 'House' next, and so on). Such an interaction where the user merely clicks on presented hyperlinks is called an *in-turn* interaction (see Fig. 1). The word 'in-turn' is drawn from conversational nomenclature and refers to a turn-taking scenario where the website queries for a certain aspect of politician at each turn, and the user makes choices for these aspects in the order in which they are requested. Notice that each hyperlink click, or in-turn input, communicates *partial information* about the desired politician. Achieving the second task by communicating only in-turn information would require a painful series of drill-downs and roll-ups, in order to identify the states that have at least one Democratic Senator, and to aggregate the results. While the user has partial information about the desired politicians, s/he is unable to communicate it by in-turn means. For instance, at the outset the user would like to specify that she is interested in Democratic Senators whereas the website is requesting a specification of state instead.

Out-of-turn interaction is our solution to support flexible communication of partial information not currently requested by the system. One manifestation is to allow the speaking of utterances into the browser. Fig. 2 describes how we can use it to achieve Task 2 above. At the top level of the site, the user is unable to make a choice of state, because s/he is looking for states that have Democratic Senators. S/he thus speaks 'Democrat' out-of-turn, causing some

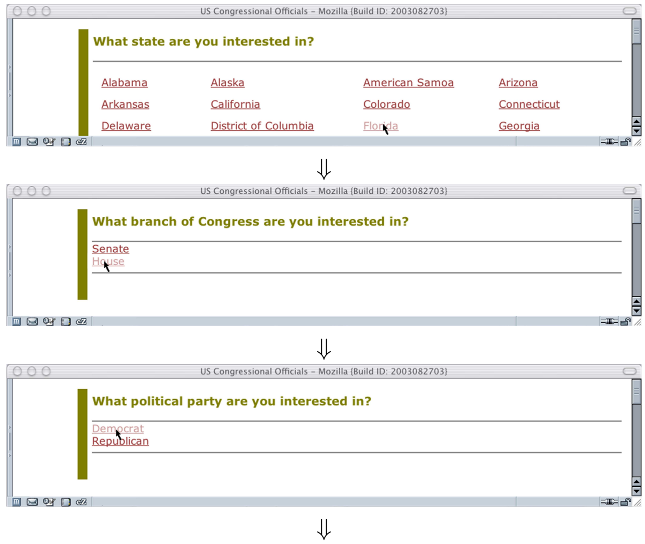

Fig. 1. In-turn interaction with a website.

states to be pruned out (e.g., Alaska). At the second step, the site again solicits state information because this aspect has not yet been communicated by the user. The user speaks 'Senate' out-of-turn, causing further pruning (e.g., of American Samoa), and retaining only regions that have Democratic Senators. At this point, the goal has been achieved (the user notices 31 states satisfying the criteria), and s/he proceeds to browse through the remaining hyperlinks. Notice that these are contextually relevant to the partial information supplied thus far, so that when 'Georgia' is clicked, there is only one choice of seat (Senior) implying that the other Senatorial seat is not occupied by a Democrat.

Out-of-turn interaction should be contrasted with the typical solution adopted in today's websites, namely *faceted browsing* that enumerates all possible dialog options in the site design, i.e., in-turn. Directly supporting all permutations of facets in the browsing structure in this manner results in a cumbersome site design, with a mushrooming of choices at each step. Out-of-turn interaction must also be viewed distinctly from search engines, which are characterized by specification of *complete* information. In this case, the interaction is terminated by returning a flat list of results, which curbs the user-site dialog. OOT interaction *continues* the dialog and situates future dialog choices (e.g., hyperlink options) in the context of previously supplied partial information.

Since out-of-turn interaction is unintrusive, optional, and preserves the closed nature of navigation through the site, it can be interleaved with hyperlink clicks as many times as desired (the stateful implementation of these interactions described below also allows the user to utilize the back-button for backtracking purposes). Such an interaction, with both in-turn and out-of-turn elements, is called a *mixed-initiative interaction* [8, 9]. An interaction with only in-turn inputs, in contrat, can be referred to as a *site-initiated* interaction.

Fig. 2. A web session illustrating the use of out-of-turn interaction in a US congressional site. This progression of interactions shows how the (Democrat, Senate, Georgia, Senior) interaction sequence, which is indescribable by browsing, may be realized. In steps 1 and 2, 'Democrat' and 'Senate' are spoken out-of-turn (resp.) when the systems solicits for state. In step 3, the user clicks 'Georgia' as the state (an in-turn input). The screen at step 4 shows that only the Senior Senator from Georgia is a Democrat, and leads the user to his homepage.

2.1 Representational Approach

Interestingly, both site-initiated and mixed-initiative interactions can be supported in the same dialog programming model! To see how, it is helpful to think of modeling a website as the program of Fig. 3 (left) where the nesting of conditionals reflects the hierarchical hyperlink structure and each program variable denotes a hyperlink label. For an in-turn sequence, the top series of transformations in Fig. 3 depicts what we want to happen. For the interaction of Fig. 2, the bottom series of transformations depicts what we want to happen (For ease of presentation, we are considering only the party, state, and branch of Congress aspects). Notice that both sequences start and end with the same representation, but take different paths.

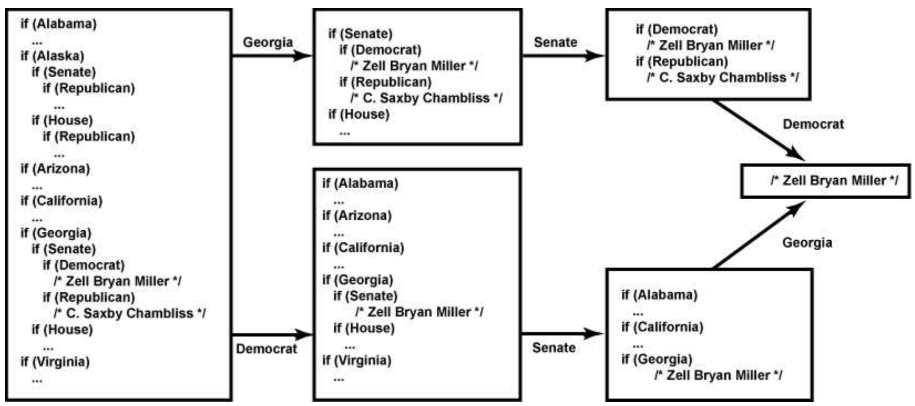

Fig. 3. Staging dialogs using program transformations. The top series of transformations mimic an in-turn dialog with the user specifying (Georgia: Senate: Democrat), in that order. The bottom series of transformationscorrespond to a mixed-initiative dialog where the user specifies (Democrat: Senator: Georgia), in that order.

The first sequence of transformations corresponds to *interpreting* the program in the order in which it is written, i.e., when the user clicks on 'Georgia,' that variable is set to one and all other state variables (e.g., 'Alabama') are set to zero, and the program is interpreted. This leads to a simplified program that now solicits for branch of congress. The second sequence of transformations involves 'jumping ahead' to nested program segments and simplifying them even before outer portions are evaluated. Such a non-sequential evaluation is well known in the programming languages literature to be *partial evaluation* ([10]; see Fig. 4), a technique for specializing programs given some (but not all) of their input. Thus, when the user says 'Democrat' out-of-turn, the program is partially evaluated with this variable set to one (and 'Republican' set to zero). The simplified program continues to solicit for state at the top level, but some states are now removed since the corresponding program segments involve dead-ends. Notice that since PE can be used for interpretation, it can support the first interaction sequence as well.

```c
int pow(int base, int exponent) {
  int prod = 1;
  for (int i = 0; i < exponent; i++)
    prod = prod * base;
  return prod;
}
```

```c
int pow2(int base) {
  return (base * base);
}
```

Fig. 4. Illustration of the partial evaluation technique. A general purpose `power` function written in C (left) and its specialized version (with `exponent` statically set to 2) to handle squares (right). Automatic partial evaluators (e.g., C-Mix) use techniques such as loop unrolling and copy propagation to specialize given programs.

This simple example shows that what is important is a representation of interaction and an expressive operator (PE) for supporting personalization. We say that a representation is *personable* for a user's information-seeking activity if there is a sequence of partial evaluations of the representation that can support the activity.

A realistic dialog model for interacting with websites requires a complete suite of representation and transformation options, for details see [11]. In addition, there are often interesting dependencies underlying attributes that should be harnessed in the personalization system. For instance, if the user says 'Senior seat,' he is referring to a Senator, not a Representative. Saying 'North Dakota' and 'Representative' in the current political landscape defines a unique member of Congress (no party information is needed), and so on. This is very similar to query expansion strategies utilized in information retrieval systems or association rules applied to web site restructuring [2]. For instance, the association rule 'Senior Seat $\Rightarrow$ Senator' holds with confidence 100% in the site structure, immediately suggesting a possible expansion of the input. In [11] we generalize these ideas and present a theory of 'staging transformations' that helps reason about what partial input has been specified thus far, whether it is legal, whether such input can be expanded, and perhaps even remove the need for further interaction. Essentially, we can think of staging transformations as a combination of site transformations and pruning operators, based on partial input. The cited reference further describes robust and scalable XML-based technology for large websites as well as user studies with this approach.

3 More Choices of Representations

Partial evaluation is one way to exploit partial information via a representation. Explanation-based generalization (EBG) is another. Even though they are computationally equivalent [12], we will begin by making a distinction and later show the implications of their equivalence for personalization.

With PE, a user experiences personalization because the site allows him to provide partial information. With EBG, a user experiences personalization because the site knows some partial information about him. EBG is thus best understood here as a technique that incorporates partial information *prior* to a user interaction, whereas PE incorporates partial information *during* a user's interaction.

We introduce EBG by considering a very different form of personalization. Consider a book-reader (Linus) revisiting the amazon.com website; a greeting prompts 'Welcome back Linus.' After Linus selects a book for purchase, the website skips the questions for credit card and shipping address when processing the order. This is presumably because the answers to these parts of the interaction are being reused from a previous session. Admittedly, this is a useful form of personalization.

```
Book Selection:
if (Mystery)
 if (Harry Potter)
    .........
else if (Science)
 if (John Nash)
    .........
Payment:
if (MasterCard)
   .........
else if (Visa)
   .........
Shipping Options:
if (Fedex)
   .........
```

```
Book Selection:
if (Mystery)
 if (Harry Potter)
    .........
else if (Science)
 if (John Nash)
    .........
```

Fig. 5. (left) Default interaction representation experienced by Amazon users. (right) Interaction representation experienced by Linus. Lines such as '**Payment:**' are comments intended to show program structure.

Fig. 5 shows two representations, the default representation seen by Amazon users and the representation experienced by Linus. It is as if the site has performed some 'free' partial evaluations just for Linus! According to our original definition, both representations are personable for Linus's activity but Linus has to provide two extra pieces of information with the representation of Fig. 5 (left). Per EBG terminology, we say that there is a difference between them in terms of *operationality*. Operationality deals with the issue of whether the site should remember Linus's credit card and payment information or whether it should require Linus to supply it during every interaction. This dilemma is actually at the heart of EBG.

3.1 Using EBG

Before we study EBG in more detail, we will make some preliminary observations. The above dilemma is actually a dilemma for the designer of the personalization system and reduces to the problem of identifying templates of interaction for users. A template — such as the returning customer template — defines a starting point for a user interaction and identifies the program variables that can be involved in the interaction. The tradeoff in designing templates is between the partial evaluations performed by the site (in the template) before the interaction begins and the partial evaluations conducted by the user during the interaction.

We can appreciate the difference by considering more users than just Linus. If the design is set up so that the site performs most of the partial evaluations, then a lot of templates will be needed to support all possible users. Each template provides a considerable amount of personalization but every user has to determine the right template for his interactions. A mushrooming of template choices can cause frustrations for users. Conversely, we can attempt to reduce the number of templates but then some users might find that there is no template that *directly* addresses their information-seeking goals. They might then proceed to use a default vanilla template such as Fig. 5 (left) (assuming that it is supported). Such users may be able to satisfy their goals but will experience longer interaction sequences and a not-so-personalized interaction. The trick is to compress many intended scenarios of interaction into a few template structures.

EBG is a systematic way to cluster the space of users and to determine dense regions of repetitive interactions that could be supported. In Amazon, one important distinction is that made between returning customers and new customers. The top-level prompt at the site makes this distinction (this is automated with cookies) and transfers are made to different interaction sequences.

How and why did Amazon decide on these two templates? Why not a distinction such as 'reading for pleasure versus reading for business or education?' Or, 'students versus professionals?' Two issues are important here. First, given a customer, can the right template be determined *easily*? Determining if a customer is a new or returning customer is admittedly easier to automate than determining if the person reads for pleasure! Second, the distinctions used for templating interactions should translate into significantly different interaction sequences. Else, the distinction is useless in practice. In the case of the returning customer, for instance, Amazon can provide more personalized recommendations and exhibit a greater understanding of the customer's preferences and habits. Balancing these considerations is a long-studied problem in EBG; it is interesting that it surfaces in such a natural way in the personalization context.

At this point it should be clear that PE and EBG support different types of personalization. While PE addresses the expressiveness with which a user can supply partial information to the system, EBG addresses the expressiveness by which the system exploits partial information about the user. While with PE we assume that the user provides the partial information in the current visit, EBG requires past navigation experiences to create 'templates' which are then operationalised. Hence EBG is more aligned toward the web mining approach to personlisation [3], involving an offline model building and then an online application of the model.

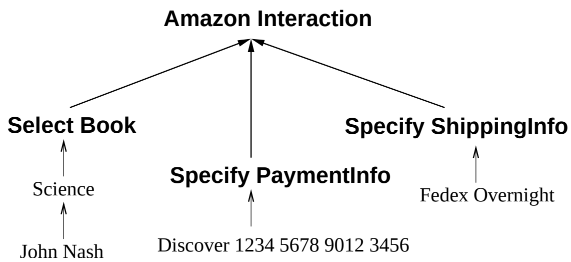

Fig. 6. Explaining a user's interaction as completing an information-seeking task.

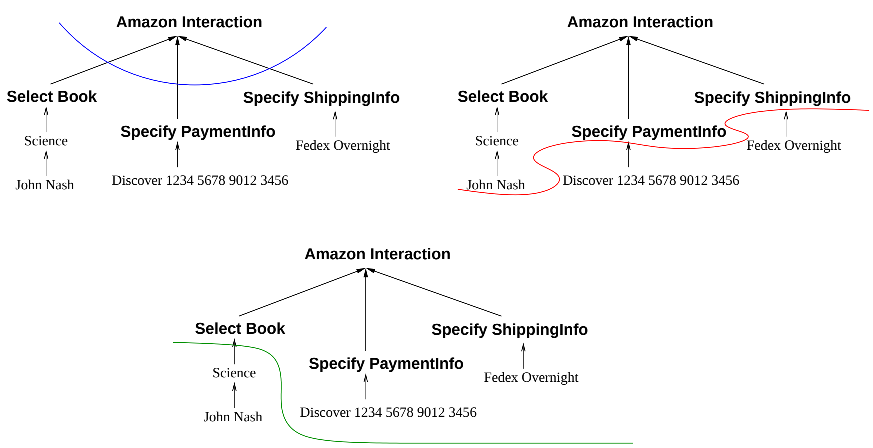

Fig. 7. Different choices of operationality boundaries lead to different templates of personalized interaction.

3.2 Operationality Considerations

EBG is an approach to reason from specific scenarios of interaction to general templates of interaction that should be supported. A user's unpersonalized interaction with a web site is observed and a general template is derived from it. The first step is to use a domain theory to explain the user's interaction. For our purposes, a domain theory captures the site layout, task models, browsing semantics, and their role in information-seeking interactions. Explaining a user's successful interaction at a site with respect to the domain theory will help identify the parts of the interaction that contribute to achieving the personalization objectives. DeJong [13] shows that an explanation can be viewed as a tree where each leaf is a property of the example being explained, each internal node models an inference procedure applied to its children, and the root is the final conclusion supported by the explanation (namely, that the scenario was an example of successful interaction). The explanation tree is used to define a space of personable representations. Searching within this space is the second step in EBG and is called operationalization.

Consider that Linus first used the Amazon site to select a book about John Nash (which he found by browsing through the Science section of the site), paid with his Discover card, and chose Fedex to ship the book. Explaining this interaction of Linus would lead to the proof tree shown in Fig. 6. The tree shows how Linus satisfied the requirements of an Amazon interaction; in this case, by satisfying the requirements for selecting a book, specifying a payment information, and specifying his shipping details. Each of these sub-requirements were in turn satisfied by particular interaction sequences. Operationalization can then be thought of as drawing a cutting plane through the explanation tree. Every node below the plane is too specific to be assumed to be part of all scenarios. The structure above the plane is considered the persistent feature of all usage scenarios and is expressed in the personalization system design. The user is then expected to supply the details of the structure below the plane so that the proof can be completed. Recall that since the proof below the plane is provided by the user's clicks and selections, it can be performed in a mixed-initiative manner.

Fig. 7 shows three ways of drawing a plane through the tree of Fig. 6. The top left really draws the plane at the level of an Amazon interaction, implying that the site will capture no personalization aspects. Every detail is meant to be supplied by the user in his interaction. It is not even assumed, for instance, that the user will buy a book. This gives us the vanilla template that caters to all users. The top right of Fig. 7 draws the cutting plane to include the selection of the book as subsumed by the system, leaving the payment and shipping address to be supplied by the user. This is obviously a very strange notion of operationality! The template resulting from this option would be appropriate only if the same John Nash book is to be purchased over and over again with different credit card and shipment options! The bottom slice of Fig. 7 is probably the reasonable one where the payment and shipping options are subsumed by the system, leaving the user to select the book. It recognizes the fact that in a future interaction, the user is likely to purchase a different book.

Deriving a generalized template of interaction also depends on the class of users it is intended to support. Is the template obtained from Linus supposed to apply only to *his* future interactions or can it be applied to other users as well? Once again, there is a tradeoff. For instance, if we have multiple users in mind then Fig. 7 (top right) no longer looks silly. Implementing this template amounts to creating a 'If you would like to buy the John Nash book, click here to give payment options' link. Contrarily, Fig. 7 (bottom) would be strange here since payment information and shipping details are not transportable across users.

After a template is derived, we have the option of explaining another user's interaction and deriving a new template, if this user's interaction is not well captured by the existing template. As mentioned earlier, we need to be careful about an explosion in the number of templates if this process is repeated. Typically, the default vanilla representation is always retained as one of the templates since there will be many users about whom the site has no prior information.

3.3 Domain Theories for Information-Seeking Interactions

Operationality is thus a matter of utility and an example corresponds to a scenario of interaction. We can evaluate operationality choices by conducting usability studies and determining the coverage of templates; example scenarios of interaction can be obtained by observation and think-aloud protocols. But where do domain theories come from?

While there is significant understanding of information-seeking interactions, there are no large, pertinent, domain theories available for the studies considered here. In [14], we handcrafted a domain theory for reasoning about interactions at the 'Pigments through the Ages' website (http://webexhibits.org/pigments) and used it with EBG to design a personalization system. Pigments is a website that uses pigment analysis catalogs to identify and reveal the palettes of painters in different eras and genres. The domain theory involved an explicit crawl of the site and a 'Background' webpage at the site that outlined a schema for how the website should be used. A group of 10 participants were identified and, after a period of acquaintance, were asked to identify one specific query (or analysis) and use the facilities at the site to answer their query. The exact interaction sequences (including clicked hyperlinks, manual information integration) was recorded for all the participants and then explained using the domain theory. This process revealed that starting from either artists, paintings, or eras, the users systematically browsed through subcategories or compared palettes to arrive at the relevant pigments (used by that artist, in the painting, or in that era, respectively). Furthermore, all pigments share common modes of information seeking, such as browsing through their history of use, procedures for preparation, and technical details of their chemical composition. We hence operationalized the explanation structure(s) as two function invocations in sequence, the first to determine an appropriate pigment category, and the second to browse through the entries in that category by various means. We thus arrived at a single structure in support of all the 10 scenarios. This structure was then evaluated with a set of 15 (different) users who provied 35 scenarios, all except two of which passed our test. The two unrealizable scenarios involved ambiguity of the query that required more contextual information than was modeled in our study.

At the end of this process, there is some optimism that domain theories can be prototyped for certain recurring themes of information-seeking interactions. Besides supporting the construction of explanations, domain theories can help in organizing software codebases for information system design. In other application domains e.g., voice interface design and directory access protocols, this form of codebase organization is already taking place. For instance, commercial speech recognition APIs provide support for task-oriented dialogs (e.g., confirmations, purchase order processing) that make it easy to prototype applications. Such an organization will greatly benefit the study of information personalization.

4 Personable Traits

I have presented two ways to think about personalization; both represent an information-seeking interaction and exploit partial information to deliver a customized experience. Together, they can help capture a variety of personalization scenarios. The EBG viewpoint is more prevalent than the PE viewpoint because the way EBG harnesses partial information lends better to implementation technologies. These observations point us to identifying the expressiveness in which partial information can be utilized by and communicated to an information system.

In Fig. 8, I identify seven tiers of personable traits along such an axis, from most simple to most sophisticated. Alongside each tier is also listed the primary way in which partial information is modeled and harnessed (PE or EBG or both). In reading the following paragraphs, the reader should keep in mind that the presence of EBG is a situation where the site knows something about the user whereas PE captures a situation where the user conveys something to the site. It should also be remarked that many of the personalization solutions surveyed here do not have explicit EBG or PE leanings; it is only our modeling of interaction that permits thinking of them in this manner.

Remembrance

This is an EBG mode of exploiting partial information and refers to the case where simple attributes of a user are remembered, such as credit cards and shipment options. Amazon is a prime example; Citibank Inc. used to provide a toolbar that provided the same functionality. The partial information is thus being exploited in a per-user manner. Web sites that capture and summarize simple form of interaction history (e.g., top 10 visited pages) also fall into this category. Here, explanations from multiple user sessions are operationalized at the leaf level into a single template. This enables a type of personalization that

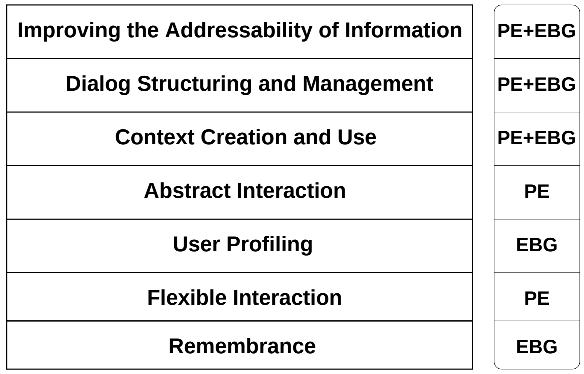

Fig. 8. Seven tiers of personalization, from simplest (bottom) to most sophisticated (top).

is not specific to any user. For an EBG technique that can support this form of specialization, see [15].

Flexible Interaction

This is a PE mode of personalization and supports simple forms of mixed-initiative interaction. The partial information is expected to be supplied by the user and personalization enhances the way in which it can be supplied. A good example is websites that allow the provision of expected, but out-of-turn information, such as in the US Congress application described earlier. Voice-activated systems are more advanced than websites in their support for this type of personalization [16].

User Profiling

Our third tier is another example of EBG and is considerably more involved than remembrance. Here, what the site knows about a user is not restricted to simple attribute-value information but is actually a sophisticated model of prior interactions. For instance, Amazon suggests ‘Since you liked Sense and Sensibility, you will also like Pride and Prejudice.’ A user’s prior interaction is captured and explained. The explanation is operationalized at the level of an internal representation, to be used in a future interaction. This form of personalization has become very popular and many machine learning techniques have been used to induce the internal representation (e.g., to learn a profile of the user). Some of these techniques are now very sophisticated and try to work with many implicit indicators.

Abstract Interaction

Just as user profiling extends remembrance in an EBG mode, abstract interaction extends flexible interaction in a PE mode. Here the partial information that a user can supply is not restricted to values for program variables but can be some abstract property of her interaction. For instance, the user could be interested in movies that featured the lead actor in *Titanic*, but may be unable to frame her partial information as 'movies where Leonardo Di Caprio acted.' I am not aware of any websites that provide such a functionality in any general way. Transformation techniques for supporting such abstract interpretation are also scarce (but see [17, 18]).

Context Creation and Use

This tier of personalization involves both EBG and PE. An example is the shopping basket at Amazon that allows a user to begin an interaction (PE) and save the state of the interaction to be resumed later (EBG). When the user returns to the site, the shopping basket can be checked out by providing the payment and delivery information. The ultimate goal of this tier is to use context creation capabilities to help stage interactions. In many cases such staging naturally breaks down into a context creation phase and a context usage phase.

Dialog Structuring and Management

I have said that EBG and PE utilize partial information in different ways. However, if the operationality boundary is moved down, then information meant to be supplied by the user becomes prior knowledge already known to the site. This shows that 'designing a personalization system' versus 'using a personalizaton system' is quite an artificial distinction. The former just corresponds to choosing a level of operationality (a partial evaluation, of the domain theory), and the latter corresponds to capturing user requests (again, via further partial evaluations, in this case of the template). This argument leads to the equivalence between EBG and PE established in [12]. This tier of personalization removes the distinction between EBG and PE and the interaction resembles more a dialog, with all the associated benefits of a conversational mode. There are not many websites that support such a tier of personalization but this problem has been studied in other delivery mechanisms such as speech technologies [8].

Improving the Addressability of Information

The holy grail of personalization is to provide constructs that improve the addressability of information. Consider how a person can communicate the homepage of, say, the *AI Magazine* to another. One possibility is to specify the URL; in case the reader is unaware, the URL is quite lengthy. Another is to just say "Goto google.com, type *AI Magazine*, and click the 'I'm feeling Lucky' button." The advantage of the latter form of description is that it enhances the addressability of the magazine's webpage, by using terms already familiar to the visitor. This tier of personalization thus involves determining and reasoning about the addressability of information as a fundamental goal, before attempting to deliver personalization. All the previous tiers have made implicit assumptions about addressability. Solutions in this tier take into account various criteria from the user (or learn it automatically from interactions) and use them to define and track addressability constraints. Such information is then used to support personalization. This helps exhibit a deeper understanding of how the user's assumptions of interaction dovetail with his information-seeking goals. The first steps toward understanding addressability have been taken [19]. However, the modeling of interaction here assumes a *complete information* view, rather than partial information.

5 Discussion

My view of personalization is admittedly a very simple one. It only aims to capture the *interaction* aspects underlying a personalized experience and not many others such as quality, speed, and utility. For instance, Amazon's recommender might produce better recommendations than some other bookseller's but if they have the same interaction sequences, then the modeling methodology presented here cannot distinguish between them. The contribution of the methodology is that by focusing solely on modeling interaction, it provides a vocabulary for reasoning about information-seeking. One direction of future work is to prototype software tools to support the types of analyses discussed above (in a manner akin to [20]).

While I have resisted the temptation to unify all meanings of the word 'personalization,' I will hasten to add that EBG and PE are only two ways of harnessing partial information. Any other technique that addresses the capture, modeling, or processing of partial information in the context of interactions will readily find use as the basis for a personalization system. The operative keyword here is, thus, *partial*. A long-term goal is to develop a theory of reasoning about representations of information systems, especially as pertaining to information-seeking [21]. The ideas presented here provide a glimpse into what such a theory might look like.

Acknowledgements

This article is a gift to my mother, who suggested the title. My view of personalization has benefited from interactions with Robert Capra, Saverio Perugini, Manuel Perez, John Riedl, Joseph Konstan, and Ed Fox.

References

1. J. Riedl. Personalization and Privacy. *IEEE Internet Computing*, Vol. 5(6):pages 29–31, Nov-Dec 2001.
2. B. Mobasher, R. Cooley, and J. Srivastava. Automatic Personalization Based on Web Usage Mining. *Communications of the ACM*, Vol. 43(8): pages 142–151, 2000.

3. M.D. Mulvenna, S.S. Anand, and A.G. Buchner. Personalization on the Net using Web Mining. *Communications of the ACM*, Vol. 43(8): pages 122–125, 2000.
4. P. Brusilovsky. Adaptive Hypermedia. *User Modeling and User-Adapted Interaction*, Vol. 11(1-2):pages 87–110, 2001.
5. M. Hearst, A. Elliott, J. English, R.R. Sinha, K. Swearingen, K.-P. Yee. Finding the Flow in Web Site Search. *Communications of the ACM*, Vol. 45(9):pages 42–49, 2002.
6. C.A. Thompson, M.H. Goker, and P. Langley. A Personalized System for Conversational Recommendations. *Journal of Artificial Intelligence Research*, Vol. 21:pages 393–428, 2004.
7. D. Riecken. Personalized Views of Personalization. *Communications of the ACM*, Vol. 43(8):pages 26–28, 2000.
8. J.F. Allen, D.K. Byron, M. Dzikovska, G. Ferguson, L. Galescu, and A. Stent. Towards Conversational Human-Computer Interaction. *AI Magazine*, Vol. 22(4):pages 27–37, Winter 2001.
9. S. Perugini and N. Ramakrishnan. Personalizing Websites with Mixed-Initiative Interaction. *IEEE IT Professional*, Vol. 5(2):pages 9–15, Mar-Apr 2003.
10. N.D. Jones. An Introduction to Partial Evaluation. *ACM Computing Surveys*, Vol. 28(3):pages 480–503, September 1996.
11. M. Narayan, C. Williams, S. Perugini, and N. Ramakrishnan. Staging Transformations for Multimodal Web Interaction Management. In *Proceedings of the Thirteenth International World Wide Web Conference (WWW'2004)*, New York, NY, pages 212-223, May 2004.
12. F. van Harmelen and A. Bundy. Explanation-Based Generalisation = Partial Evaluation. *Artificial Intelligence*, Vol. 36(3):pages 401–412, 1988.
13. G. DeJong. Explanation-Based Learning. In A.B. Tucker, editor, *The Computer Science and Engineering Handbook*, pages 499–520. CRC Press, 1997.
14. N. Ramakrishnan, M.B. Rosson, and J.M. Carroll. Explaining Scenarios for Information Personalization. Technical Report cs.HC/0111007, Computing Research Repository (CoRR), http://xxx.lanl.gov/abs/cs.HC/0111007.
15. N.S. Flann and T.G. Dietterich. A Survey of Explanation-Based Methods for Inductive Learning. *Machine Learning*, Vol. 4:pages 187–266, 1989.
16. N. Ramakrishnan, R. Capra, and M.A. Pérez-Quiñones. Mixed-Initiative Interaction = Mixed Computation. In P. Thiemann, editor, *Proceedings of the ACM SIGPLAN Workshop on Partial Evaluation and Semantics-Based Program Manipulation (PEPM)*, pages 119–130. ACM Press, January 2002.
17. C. Consel and S.-C. Khoo. Parameterized Partial Evaluation. *ACM Transactions on Programming Languages and Systems*, Vol. 15(3):pages 463–493, July 1993.
18. N.D. Jones, C.K. Gomard, and P. Sestoft. *Partial Evaluation and Automatic Program Generation*. Prentice Hall International, June 1993.
19. W. Jones, H. Bruce, and S. Dumais. Keeping Found Things Found on the Web. In *Proceedings of the ACM International Conference on Information and Knowledge Management (CIKM)*, pages 119–126. ACM Press, November 2001.
20. A. Wexelblat and P. Maes. Footprints: History-Rich Tools for Information Foraging. In *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI'99)*, pages 270–277. Pittsburgh, PA, 1999.
21. G. Marchionini. *Information Seeking in Electronic Environments*. Cambridge University Press, 1997.