The Traits of the Personable

Naren Ramakrishnan
Department of Computer Science
Virginia Tech, VA 24061, USA
Email: naren@cs.vt.edu

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

Personalization refers to the automatic adjustment of information content, structure, and presentation tailored to an individual user. It is increasingly employed by commercial sites to help retain customers and reduce information overload. For instance, Riedl [17] estimates that there are at least 23 different types of personalization at Amazon's e-commerce site!

What does it mean for a web site to be *personable*? There are definitely multiple interpretations of 'personalization' [16] and correspondingly many ways of answering this question. Let us start with the working assumption that a site is personable if it allows a user's information seeking goals to be met *effectively*. Obviously, different users have different goals and expectations; thus, a site that is personable for one information-seeking activity might be unpersonable for another. The goal of this paper is to formalize these intuitive notions and describe a modeling methodology for reasoning about personalization.

1.1 Example

I will begin with a simple example and later expand the scope of the discussion to cover more complex cases. 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 two users visiting the web site of a camera retailer:

User 1: I am interested in Nikon models.
User 2: I am looking for an SLR camera.

User 1 thinks of cameras primarily in terms of their manufacturer and hence a web site organization such as shown in Fig. 1 (left) might be appropriate for this user. User 2 thinks of cameras in terms of lens equipment and Fig. 1 (right) might be preferable. Both site designs involve a hierarchical browsing paradigm, with levels corresponding

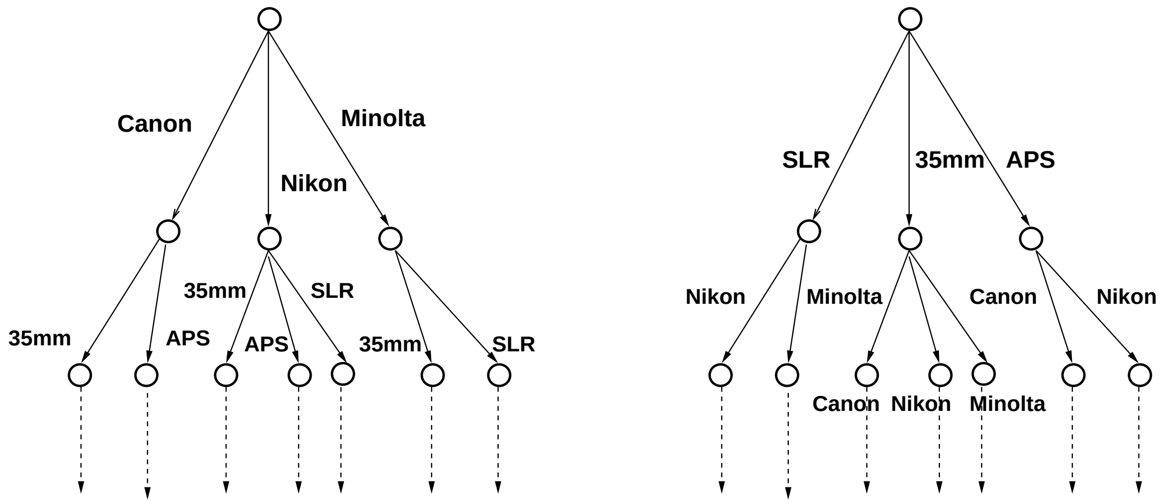

Figure 1: Two organizations of a camera catalog: by maker-type (left) and by type-maker (right). Only two levels are shown for ease of illustration. The nodes are web pages, the edges denote hyperlinks, and labels on edges represent the text anchoring the hyperlinks.

to camera attributes. Only the top two levels are shown in Fig. 1; it is assumed that further levels impose additional classifications and distinctions among camera products.

We say that the site in Fig. 1 (left) is personable for User 1's activity since it allows the specification of choice of model (Nikon) at the outset. Conversely, it is *unpersonable* for User 2's activity since there is no way for this user to specify her interest in an SLR camera at the outset. The only option is to follow all possible model choices, and determine the ones that support SLR features. This might involve pursuing paths that are dead-ends (for instance, no SLR models are available under Canon). In contrast, the organization in Fig. 1 (right) is personable for User 2's activity since it will eliminate Canon as a potential option once the selection of SLR is made at the outset. By a symmetric argument this site is unpersonable for User 1's activity.

Is it possible to have a web site that is personable for both users? The easy solution is to support two different browsing interfaces — browse-by-maker and browse-by-type. The responsibility is then on the user to employ the right interface for his activity.

There are two problems with such a design. The first is the explosion in the scenario possibilities. If cameras are distinguished by n independent attributes, then we have $n!$ possible organizations. Some web sites actually take such an exhaustive approach and support all possible ways of interacting with an information system (e.g., epicurious.com, a web site for organizing recipes, described in [6]). The more fundamental problem with such a design is that it over-specifies the personalization goals by anticipating in advance all the forms of interactions that have to be supported.

Web sites are not traditionally designed for personalization. Enabling true personalization implies supporting a flexible mode of interaction between the user and the underlying information system. I hence adopt the view that personalization is properly approached by studying interactions of users with information systems [10].

2 Personalized Interaction

There is a way by which a site can support both User 1 and User 2, without enumerating the scenario possibilities. The trick lies in recognizing that both users bring different *partial information* to bear upon the information-seeking activity. User 1 knows about manufacturer whereas User 2 knows about lens type. Both attributes are only partial specifications toward the selection of a camera and hence a web site that flexibly allows the specification of partial

`int pow(int base, int exponent) {`
`  int prod = 1;`
`  for (int i=0;i<exponent;i++)`
`    prod = prod * base;`
`  return (prod);`
`}`
	`int pow2(int base) {`
	`  return (base * base);`
	`}`

Figure 2: 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). Such specializations are performed automatically by partial evaluators such as C-Mix.

```
if (Canon)
  if (35mm)
  .........
  else if (APS)
  .........
else if (Nikon)
  if (35mm)
  .........
  else if (APS)
  .........
  else if (SLR)
  .........
else if (Minolta)
  if (35mm)
  .........
  else if (SLR)
  .........
```

```
if (Nikon)
  .........
else if (Minolta)
  .........
```

Figure 3: Using partial evaluation for personalization. (left) Programmatic input to partial evaluator. (right) Specialized program from the partial evaluator, used to create a personalized information space.

information will be personable to both users. Several techniques for exploiting partial information are available. Two are described in this paper — partial evaluation (PE; a technique from the programming languages community [7]) and explanation-based generalization (EBG; a technique from the AI community [3]). Under certain situations [18], EBG and PE can be shown to be essentially the same technique.

2.1 Using Partial Evaluation

Partial evaluation is a technique for specializing computer programs, given some (but not all) of their input. The input to a partial evaluator is a program and some static information about its arguments. The output of the partial evaluator is a specialized version of this program (typically in the same language) where the static information has been used to simplify as many operations as possible. Partial evaluators analyze the given program to determine reducible expressions and use compiler transformations to interpret (evaluate) all such expressions. Given the value of `exponent` in Fig. 2 (left), the loop can be unrolled and the value of the `prod` variable forward propagated to yield the program in Fig. 2 (right). Automatic partial evaluators are available for C, Scheme, PROLOG, and many other languages.

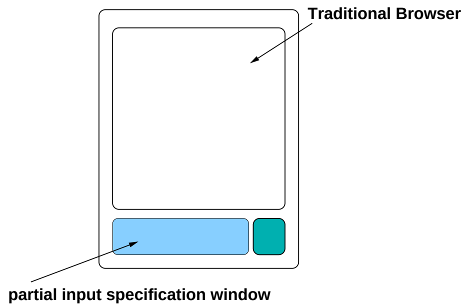

Figure 4: Sketch of an interface for personalized interaction with web sites. The top window of the interface supports a traditional browsing functionality. At any point in the interaction, in addition, the user has the option of supplying personalization aspects out-of-turn (bottom two windows). Such an interface can be implemented as a toolbar in browsers.

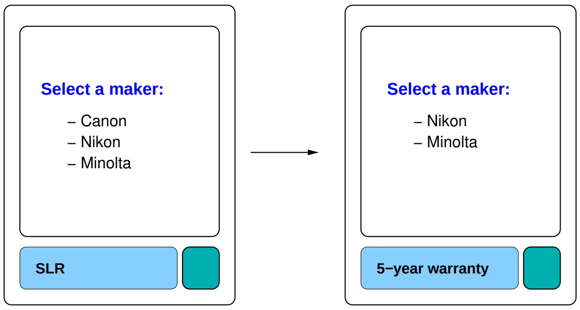

Figure 5: Example of a personalized interaction for User 2. At the beginning of the interaction, User 2 decides to not use any of the presented hyperlinks for camera model. Instead, she uses the toolbar to specify her choice of camera type out-of-turn (left). The results of partial evaluation cause the Canon option to be removed from the model choices (right). Once again, User 2 opts to utilize the toolbar to provide a specification of warranty information (results not shown).

To use partial evaluation for personalization [12], think of a program as representing a dialog script of interaction between a human and an information system. Every program variable is an opportunity for the user to communicate a specification aspect. For instance, we can abstract interaction with the browsing hierarchy in Fig. 1 (left) using the program of Fig. 3 (left). To personalize the site for User 2, we partially evaluate w.r.t. the user's preferences (we set the program variable `SLR` to 1 and all conflicting variables such as `35mm` and `APS` to zero). This simplification, coupled with pruning dead-end branches (like the conditional involving `Canon`) leads to the specialized program of Fig. 3 (right). A personalized web site can be recreated from the simplified program. The act of partial evaluation thus mimics the processing of an out-of-turn input by the user.

This simple example shows that what is important is a representation of interaction and an expressive operator (PE) for supporting personalization. Our earlier statement that Fig. 1 (left) is unpersonable for User 2's activity implicitly assumed that the operator was interpretation, i.e., values for program variables are ascertained from the user in the order in which they were modeled in the program. In reality, the notion of representation of interaction was not a very useful one there. With PE, both sites in Fig. 1 can be made personable for both users. The notion of representation is crucial here, since it drives the partial evaluation.

In human-computer interaction terms, the ability of the user to provide an out-of-turn (but expected) input is recognized as an important facet of *mixed-initiative interaction* [1, 13]. In this view, we can think of the act of clicking on a hyperlink as responding to the initiative taken by the web site (information system). This is a systemdirected mode of interaction. Specifying a variable out-of-turn implies that the user has taken the initiative and made an unsolicited reporting. This is a user-directed mode of interaction. Both these modes of interaction are subsumed by PE, since they only differ in the arrival time of specification aspects. We can thus perform a *sequence* of partial evaluations to support a flexible interaction strategy (see Figs. 4 and 5).

2.2 Representations for Personalization

We now formalize a definition to recognize the importance of a representation:

> *(1)* A representation of interaction $\mathcal{I}$ is personable for an information seeking activity $\mathcal{G}$ if $\mathcal{G}$ can be realized by a sequence of partial evaluations of $\mathcal{I}$.

We have shown that the representation of Fig. 3 (left) is personable for both User 1 and User 2. Let us consider the usefulness of this representation for some additional users:

> **User 3:** I am interested in cameras that allow the setting of shutter speed.
>
> **User 4:** What types of cameras are available?
>
> **User 5:** I am looking for a 35mm Canon.

As it is modeled in Fig. 3 there are no program variables that allow User 3 to describe his preference for shutter speed control. We say that the representation is unpersonable for this user's activity. This should be obvious, since personalization is captured by the assignment of values to program variables. To support all personalization scenarios, we need to ensure that adequate program variables are present in the representation.

Interestingly, the representation of Fig. 3 is unpersonable for Users 4 and 5 as well! User 4's request does involve camera types but this user is not expressing a selection of a type. Rather, he is initiating a dialog to discuss the specification of camera type (which might eventually lead to a selection of type). We can use PE to specify a camera type out-of-turn but we cannot use it to enquire about the types available. In general, this requires the ability to reflect about the operation of a program.

One possibility is to explore other transformation techniques (e.g., slicing) that allow the desired restructuring of the representation. Another is to retain PE as our personalization operator and change the representation to accommodate User 4's request. This is a tricky endeavor due to the optional nature of the request. The representation that achieves this interaction is beyond the scope of this article; it involves using continuations (e.g., via **call/cc** in Scheme) to model the dynamic branching of control, in this case, to discuss the types of cameras available, and subsequently to return from such a dialog. For the reader well versed in programming language implementation, this is akin to designing exception handling capabilities (see Chapter 7 of [5]).

Let us now consider User 5's activity. This activity specifies values for all applicable program variables, corresponding to a *complete evaluation*, rather than a partial evaluation. Complete evaluation implies that every aspect of interaction is determined in advance, removing the need for any interaction! While we can accommodate User 5's request, the representation of Fig. 3 offers no particular advantages for this activity (for instance, the user could have just used the traditional, interpretive-style, browsing interface).

Thus, a representation can be unpersonable because there are *no* program variables available to model user requests. Alternatively, a representation is unpersonable because user requests can be captured only by specifying values for *all* variables. The crucial notion of 'values for *some, but not all* variables' is important to model personalized interaction by partial evaluation. This leads us to the next two definitions:

> *(2)* A representation of interaction $\mathcal{I}$ is unpersonable for an information seeking activity $\mathcal{G}$ if $\mathcal{G}$ cannot be realized by a sequence of partial evaluations of $\mathcal{I}$.

click [here]	if you are interested in Nikon models.
click [here]	if you are interested in SLR cameras.
click [here]	if you are interested in cameras that allow shutter speed control.
click [here]	if you are seeking information about camera types.
click [here]	if you are interested in 35mm Canon models.
click [here]	$\cdots\cdots\cdots$

Figure 6: 'Freezing' interactions for all users results in an unpersonable representation.

> *(3)* A representation of interaction $\mathcal{I}$ is unpersonable for an information seeking activity $\mathcal{G}$ if $\mathcal{G}$ can only be realized as a complete evaluation of $\mathcal{I}$.

Notice that definition *(3)* should be taken as the exception to our earlier definition *(1)*.

2.3 More on Representations

We have thus far fixed the representation and asked the question: 'What are the activities for which a representation of interaction $\mathcal{I}$ is personable?' This is the designer's viewpoint. For a given web site, it allows the designer to determine the target audience that will consider the site personable.

The complementary question is user-driven and asks: 'For my information seeking activity $\mathcal{G}$, what are the personable representations?' This allows the user to study different web site representations and see which ones lead to realization of his information seeking goals. In a realistic deployment of personalization solutions, many user studies will likely be conducted while developing and designing the web site. A participatory design methodology will attempt to address both perspectives.

For instance, we have seen that the representation in Fig. 3 (left) is personable for Users 1 and 2 but unpersonable for Users 3, 4, and 5. Can we have a representation that is personable for all these users? The trivial solution, which is to 'freeze' and support all these situations, actually leads to an unpersonable design! Fig. 6 shows a design that allows all of these users to achieve their information seeking goals. However, there is really only one level of interaction, involving the choice of link to take. All evaluations of the programmatic representation have to be complete evaluations! It can be argued that such an anticipatory design is unsatisfactory since it buckets all users into predefined categories; it is hence not really 'personal' at all!

In practice we try to factor in different users' requirements into our modeling of interaction, without explicit enumeration. If the scenarios requiring complete evaluation are a small fraction of the total number of scenarios, then we can 'excuse' our modeling for being unpersonable for those situations (e.g., User 5). User 3's request can be accommodated by adding additional program variables for shutter speed control. User 4's request cannot be easily accommodated, as stated earlier, without significant revision of our programming style. If such requests will be common, then we might be justified in redesigning the web site to support such interactions.

2.4 Implementation Details

The reader might notice that conditionals in Fig. 3 such as

```
if (Canon)
```

could be replaced by

```
if (manufacturer == Canon)
```

Is one representation better than the other? In the original design, just setting 'Canon' to 1 can model a user's request, but to be semantically correct we had to ensure that conflicting variables such as 'Minolta' and 'Nikon' were set to zero (the `else` clause will trap some of these situations). This introduces some overhead in implementing a personalization system. In the new representation, we can merely set the `manufacturer` variable to 'Canon' but the responsibility is on us to realize that the user was referring to a camera manufacturer. For applications that have polysemous user input, this can be troublesome.

The fundamental problem is learning a mapping from user requests to assignments of values for internal representations (in our case, program variables). This has been long studied in information retrieval in various forms, e.g., query expansion, local context analysis, proximity metrics. In the personalization context, the assignments are also expected to capture the semantics of out-of-turn interactions. There are no solutions that work in general, only guidelines and recommendations.

As a demonstrator of the above ideas, we designed a personalization system for the US Congressional portion of the Project VoteSmart web site (vote-smart.org). The site caters to information about political individuals and users interact with the site by specifying choices of state, party, branch of congress, and seat. There are interesting dependencies underlying these attributes that cannot be captured by a clever representation. 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. It is important to consider such facets in order to deliver a compelling personalized experience. The net effect of such considerations will be that multiple program variables are initialized based on the user's input.

The unspecialized program was represented in an XML notation and modeled by a depth-first crawl of the site. Since VoteSmart coalesces the party, branch, and seat information into a single hyperlink, we conducted a within-page modeling to yield independently addressable program variables for these attributes. The partial evaluator was implemented using XSLT and a demo is available at pipe.cs.vt.edu. Evaluation results are presented in [14].

Additional web applications developed in this manner are described in [12, 15]. Many of them conduct more sophisticated modeling of interaction than just browsing hierarchies; for instance, information integration from multiple web sites, interacting with recommender systems, modeling clickable maps, and representing computed information. All of these can be subsumed in a programmatic modeling; to effect personalization, we just require a meaningful way to view an interaction with an information system as a program and a mapping from user input to program variables. Opportunities to curtail the cost of partial evaluation for large information spaces are also studied in [14]. I refer the reader to these references for more discussion on implementation choices. In this article, I elaborate further on the usefulness of representations.

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 [18], 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.

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.

Fig. 7 shows two representations, the default representation seen by Amazon users and the representation expe-

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)
   .........
```

Figure 7: (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.

rienced 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. 7 (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. 7 (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.

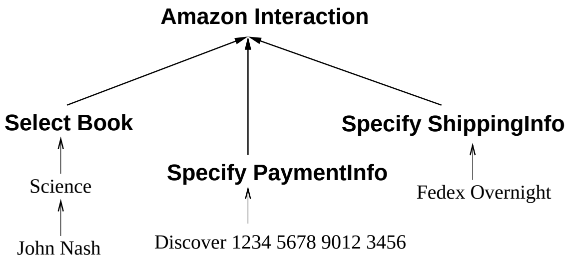

Figure 8: Explaining a user's interaction as completing an information-seeking task.

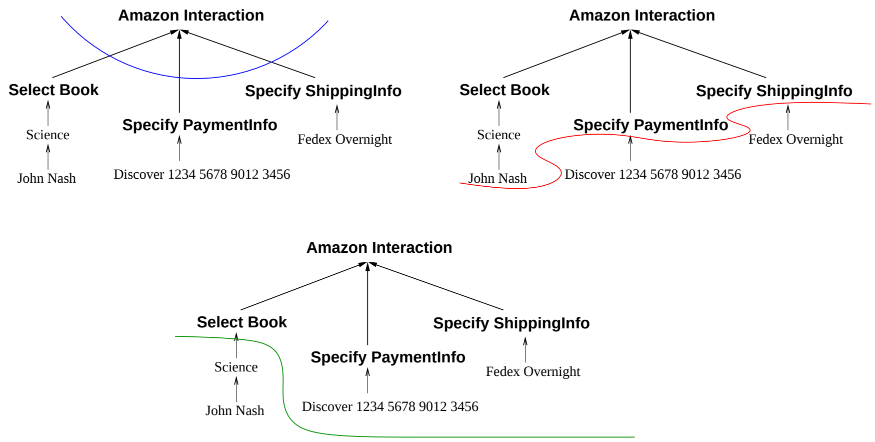

Figure 9: Different choices of operationality boundaries lead to different templates of personalized interaction.

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.

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 [3] 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. 8. 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. 9 shows three ways of drawing a plane through the tree of Fig. 8. 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. 9 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. 9 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. 9 (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. 9 (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 [15], 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. 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

Improving the Addressability of Information	**PE+EBG**
Dialog Structuring and Management	**PE+EBG**
Context Creation and Use	**PE+EBG**
Abstract Interaction	**PE**
User Profiling	**EBG**
Flexible Interaction	**PE**
Remembrance	**EBG**

Figure 10: Seven tiers of personalization, from simplest (bottom) to most sophisticated (top).

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 views of 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. 10, 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 is not specific to any user. For an EBG technique that can support this form of specialization, see [4].

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 camera application described earlier. Voice-activated systems are more advanced than websites in their support for this type of personalization [13].

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 [2, 8]).

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 [18]. 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 [1].

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 *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 [9]. 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 [19]).

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. 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 Rob Capra, Sammy Perugini, Manuel Perez, and Priya Kandhadai.

References

[1] 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.

[2] C. Consel and S.-C. Khoo. Parameterized Partial Evaluation. *ACM Transactions on Programming Languages and Systems*, Vol. 15(3):pages 463–493, July 1993.

[3] G. DeJong. Explanation-Based Learning. In A.B. Tucker, editor, *The Computer Science and Engineering Handbook*, pages 499–520. CRC Press, 1997.

[4] N.S. Flann and T.G. Dietterich. A Survey of Explanation-Based Methods for Inductive Learning. *Machine Learning*, Vol. 4:pages 187–266, 1989.

[5] D.P. Friedman, M. Wand, and C.T. Haynes. *Essentials of Programming Languages*. MIT Press, 2001. Second Edition.

[6] M. Hearst. Next Generation Web Search: Setting Our Sites. *IEEE Data Engineering Bulletin*, Vol. 23(3):pages 38–48, September 2000.

[7] N.D. Jones. An Introduction to Partial Evaluation. *ACM Computing Surveys*, Vol. 28(3):pages 480–503, September 1996.

[8] N.D. Jones, C.K. Gomard, and P. Sestoft. *Partial Evaluation and Automatic Program Generation*. Prentice Hall International, June 1993.

[9] 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.

[10] G. Marchionni. *Information Seeking in Electronic Environments*. Cambridge University Press, 1997.

[11] T. Mitchell. *Machine Learning*. McGraw Hill, 1997.

[12] N. Ramakrishnan. PIPE: Web Personalization by Partial Evaluation. *IEEE Internet Computing*, Vol. 4(6):pages 21–31, Nov-Dec 2000.

[13] 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.

[14] N. Ramakrishnan and S. Perugini. The Partial Evaluation Approach to Information Personalization. *ACM Transactions on Information Systems*, 2001. Communicated for publication. Also available as Technical Report cs.IR/0108003, Computing Research Repository (CoRR), http:// xxx.lanl.gov/abs/cs.IR/0108003.

[15] N. Ramakrishnan, M.B. Rosson, and J.M. Carroll. Explaining Scenarios for Information Personalization. *User Modeling and User-Adapted Interaction*, Feb 2002. Communicated for publication. Also available as Technical Report cs.HC/0111007, Computing Research Repository (CoRR), http://xxx.lanl.gov/ abs/cs.HC/0111007.

[16] D. Riecken. Personalized Views of Personalization. *Communications of the ACM*, Vol. 43(8):pages 26–28, 2000.

[17] J. Riedl. Personalization and Privacy. *IEEE Internet Computing*, Vol. 5(6):pages 29–31, Nov-Dec 2001.

[18] F. van Harmelen and A. Bundy. Explanation-Based Generalisation = Partial Evaluation. *Artificial Intelligence*, Vol. 36(3):pages 401–412, 1988.

[19] 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.