

Modeling Web Site Personalization Strategies

Fabiana Ruas, Wagner Meira Jr., Paulo Araújo and Flávia Ribeiro

e-SPEED - Department of Computer Science
Federal University of Minas Gerais
Belo Horizonte - MG - BRAZIL
faruas, meira, pauloh, flavia@dcc.ufmg.br

Abstract

Personalization is a key factor for differentiating services and retaining customers in World Wide Web sites. On the other hand, designing and implementing an efficient personalization strategy is still a challenge, because of the complexity of the techniques used and the variety of sites and customers, which are always evolving. This paper presents a functional model of personalization strategies that allows not only a simple and concise specification of those strategies, but also their simulation and validation. We demonstrate our model through e-Personal, a framework for estimating the effectiveness of personalization strategies. The framework guides the user through the process of specifying a strategy and estimates its impact based on previous interactions of customers with the site. It is based on our functional model and we illustrate its utilization for designing personalization strategies for a web portal. Our experiments are based on actual logs and show that the proposed framework enhances significantly the personalization process, indicating the goodness of the strategy design, the reliability of input data, and the impact of implementation decisions on the effectiveness of personalized sites.

Keywords: *Personalization, Adaptive Web Sites, World Wide Web*

1 Introduction

There is an increasing competition among World Wide Web sites in all service segments. Although these sites initially focused on correctness and availability, many sites are already working towards differentiating their services so that they can increase their customer population, and keep it despite their competitors. Designing and implementing effective personalized sites arise as an essential component of this service differentiation strategy.

Personalization may be defined as any adaptations in terms of format, process, and nature of resources provided to a customer of a site. It relies on gathering customer information, which may be accomplished employing several mechanisms. Explicit mechanisms depend on customer for gathering information, such as filling a form with his or her personal information. Implicit mechanisms employ cookies or server logs to acquire customer navigational information, to determine the browser being used in order to adapt data, or to determine the customer geographical location based on the client machine IP address. Personalization strategies aim to make a Web site more responsive to the unique and individual needs of each customer, thus enhancing his or her interaction and loyalty, which results in increased site audience and revenue.

However, designing and implementing personalization strategies is still a challenge in most cases, since a significant part of the process is not automated and is performed by experts. This situation arises from several characteristics inherent to the task of personalizing a Web site. The personalization requires some knowledge about the profiles of the site customers in terms of their preferences, but these profiles may change over time, both as a consequence of changes in the customer population and because of evolving customer preferences across time. Techniques such as association rules and neural networks, commonly employed for personalizing sites, demand reasonable expertise and may be costly in terms of computation. Collecting and organizing data for personalization is laborious and error-prone, because these tasks are usually site dependent. Finally, determining which personalization strategies would help to improve the service provided is usually hard, sometimes impossible.

Furthermore, testing and comparing such diverse strategies is even harder. Each strategy may require different data gathering approaches that are sometimes incompatible

and may take time to produce reasonable results. For example, changing a site logging methodology cannot be performed frequently, since it generally requires modifying many configuration options and may invalidate past data. Implementing a new strategy is even more infeasible, since many pieces of software may have to be rewritten. Comparing the results is another hard step which may be time consuming and may not lead to actual improvements. A tool for predicting the efficacy of Web site personalization strategies would be valuable for determining the most effective approach and guiding its implementation.

In this paper we present a functional model of personalization strategies that allows not only a simple and concise specification of those strategies, but also their simulation and validation. By using this model, it is possible to describe the decision process while designing a personalization strategy, subsidizing the entire implementation process. Based on this model, we built a framework for simulating personalization strategies, e-Personal. This framework provides some tools that allow the user (i.e., a personalization designer) to estimate the efficacy of a strategy and embeds the functional model for the sites that allows a simple specification, simulation and validation of strategies. By using the framework, users may describe in detail personalization strategies and test it against actual server logs. During this process, the strategy is evaluated regarding its applicability and impact on the behavior of the customers, providing information for measuring the gains from implementing a strategy.

This paper is organized in 6 sections. This section serves as introduction. The next section discusses some related work and the coverage of the framework in the context of current personalization efforts. Section 3 presents a functional model of Web sites, followed by a formalization of the personalization process using this model in Section 4. One case study based on actual logs is presented in Section 5. The last section draws the conclusions of the paper and describes some future work.

2 Related Work

There are several efforts for personalizing Web sites [10]. In this section we analyze some of these efforts and how the model and the framework are related to them. First of all, we are not aware of any other work that evaluates personalization strategies in advance, thus we discuss how the various personalization strategies fit in our framework. In fact, the lack of tools that guide users in designing and evaluating their personalization strategies before implementing them was our main motivation for developing the model, and later the e-Personal framework.

The personalization is usually based on data such as access logs [9], personal data, and even market data [6].

There are basically two trade-offs that determine how useful some information is: privacy [5, 4] and reliability [7]. Privacy determines which user information can be used or not. Reliability concerns whether we can trust the information for sake of personalization or not.

We may classify personalization efforts regarding several criteria, such as scope, level of automation, implementation resources, and personalization target. The scope may be local or global, where local personalization affects just the user or session being personalized [9] and global personalization affects the structure of the Web site [12]. The level of automation refers to the completely automated [9] application of personalization strategies or not [12], requiring, in this last case, tasks performed manually. We distinguish proprietary and public implementation mechanisms. Proprietary mechanisms include the use of specific tags [13], while public resources employ popular standards such as HTML and XML. The personalization target defines the parts of the site that will be personalized. It may be fixed to some portions [7] or not, so that it is possible to personalize any portion of the site [2].

The basis of the e-Personal framework is the Web site access log that may include not only the pages visited and functions but also user session information. The framework also allows users to specify the structure of the site being personalized and the semantics of the functions provided. Regarding analysis techniques, e-Personal does not impose any restrictions, since the user is supposed to code the personalization strategy using the operators provided. Although it may appear laborious at first glance, the codification of the strategies in advance is very helpful in determining their feasibility and also makes their actual implementation much easier. There are also no restrictions regarding the scope, level of automation, and personalization target, since the strategy is completely specified by the user. Finally, the e-Personal applies to both proprietary and public implementation approaches, since this decision does not affect the strategy.

3 Web Site Personalization

In this section, we are going to present a model for describing the structure of Web sites. This model describes the dynamics of a Web site, whose content is frequently stored in structured or semi-structured databases. In the section that follows, we are going to present a model for specifying personalization strategies, which allows the specification of several variables inherent to the personalization process.

3.1 Functional Modeling of Web Sites

Before describing the personalization model, we are going to present a model that describes the structure of a Web site.

From the functional perspective, Web sites are a set of pages containing links that allow users to access other pages in the site or even in other Web sites. These links comprise not only the target page but also occasional parameters that should be considered while generating the pages.

First we present two concepts that will help us to describe this model: objects and functions. The various pages that compose a site can be characterized by these two concepts. **Objects** (denoted by \mathcal{O}) represent real-world entities such as information and products. Each object, regardless its nature, is characterized by a set of attributes. A value is assigned to each attribute and each object $o \in \mathcal{O}$ has a set of values assigned to attributes that uniquely identify it. Considering a virtual store, a specific book like “*The Shell Seekers*” represents an object.

Classes are sets of objects that are characterized by the same set of attributes. Objects are grouped into classes according to their nature or similarity for a given criterion. We may define a class *Book* that contains this and all other books that are sold in the virtual store.

We also define a **group of objects** as a set of objects from one or more classes. Groups of objects are defined through a logical expression that selects a view from the universe of objects. Examples of expressions are “*the Rosamunde Pilcher books*” or the book “*The Shell Seekers*”. A group of object can be also organized hierarchically, containing other groups of objects. For example, the group of objects “*the Rosamunde Pilcher books*” contains the book “*The Shell Seekers*”.

Functions are actions that can be performed by a customer in a Web site. Each type of page (e.g., browse books or search for books) is generated by a function. Functions (denoted by F) receive two kinds of arguments: function parameter and state. A function performs two actions: (1) generate a page based on the function parameter, and (2) change the state of the customer session.

The function *parameter* is the group of objects used for generating a page. The state is the group of objects that should persist during a customer interaction with the site. In summary, each page in our functional model is uniquely defined by the triple: function, parameter, and state. Consider the action *add the book “The Shell Seekers” to the cart*. In the model, we describe this action by the execution of the function “*add to cart*” on the parameter “*The Shell Seekers*”. This function changes the state of the site, which is represented by all books that were added to the cart during the current user visit.

Based on these definitions, a **page** is the result of a function call $fi \in \mathcal{F}$ having two groups of objects as arguments: $Pi \subset \mathcal{O}$ and $Si \subset \mathcal{O}$. A page can be represented

by a triple $\langle fi, Pi, Si \rangle$ and contains a set of links.

A **link** is a page reference, i.e., it is a reference to a function call. Each link in the page defines not only the function to be invoked when the link is chosen by a customer, but also the parameter to the function. The state, which is also used by the function to generate the requested page, is implicit and maintained by the server.

Pages may have two kinds of links: static and dynamic. Static links are not based on the parameter objects. For example, considering the virtual store, every page has a link to view the *books* in the cart. Dynamic links are generated based on the parameter objects. For example, consider the function *browse books* applied to the group of objects *International books*. The result is a page containing links to selected international books. These links are generated dynamically.

We can summarize the interaction of a customer with a site through the sequence of pages accessed by the customer, which is usually called user **session**. From the starting page, which is generated from empty parameter and state, the user session is a path in the site determined by the sequence of pages followed by the customer. A user session of n pages can be represented by a set of triples $\{ \langle f1, P1, S1 \rangle, \dots, \langle fn, Pn, Sn \rangle \}$.

The generation of each page requested by a customer can be described formally by the expression $f(Pi, Si) = Pi+1; Si+1$. Where f is the page generation function, Pi is the parameter group of object for generating the page, Si is the state group of objects at the time of the page generation, $Pi+1$ is the page generated, and $Si+1$ is the state of the session after executing the function. An abstract implementation of f is presented in Figure 1, where we can see the generic phases for generating a page within the functional model presented. The new state is defined by function *new_state*. This state is used for generating the page links. Pages may have links that don't depend on the group of objects being passed as a parameter. The function generates static links is responsible for them. Dynamic links are generated by *generate links* applied on P . Thus, the result of function f is the set of links that are on the respective page.

```

function f(P; S)
  begin
    Snew = new_state(S)
    links = generate_static_links(Snew)
    foreach object o ∈ P
      links = generate_linkf(o, Snew)
    end
  end

```

Figure 1: Algorithm for generating non-personalized pages

3.2 Personalization Modeling

Without loss of generality, we assume that customers start their interaction with a given Web site by accessing the home (or entry) page of the site. This page, as all other pages provided by the sites, offers content and navigational options. Content comprises all kinds of information provided as text, images, audio, and video. Navigation is supported through hypertext links to other pages. Considering just these two basic components of a page, we distinguish two types of personalization: content and navigation. Content personalization changes the information provided to the customer in terms of both quantity and nature (e.g., amount of detail) aiming to provide a better service to customers. It also includes other types of information such as banner ads inserted in the pages. Navigation personalization changes the formatting (i.e., highlight or not a link), the order (i.e., which links are presented first), and the nature of the links themselves (e.g., pre-defined parameters).

Evaluating the impact of content personalization without a knowledge acquisition model is quite hard, mainly because current technology does not allow extensive verification of the attention and focusing mechanisms employed by customers while verifying the content of a page. On the other hand, verifying the effectiveness of navigation personalization is simple in the sense that we know that a personalized link is effective if the customer clicks on it, requesting the associated page. The model presented in this paper is targeted at supporting intra-site navigation personalization, that is, we intend to personalize the access to internal pages of a given Web site.

In this section, we discuss how the functional model presented supports the modeling of the personalization process. As mentioned, we modeled each page in the site as a list of links to other pages in the site, where each link embeds a function invocation and the objects that serve as parameters for the page to be generated. In the context of the functional model presented, the personalization of Web sites may be performed through two mechanisms:

New functions: The definition of new functions (not originally present in \mathcal{F}) that personalize the resources provided to the customer. A simple example of such function is the creation of an auxiliary page that presents the links of interest for a given customer. In this case, this page is a new function in the sense that there was no such page in the site before it was personalized. This personalization mechanism can be implemented by simply adding more functions to \mathcal{F} .

Weight differentiation: A strategy to personalize existing pages is to assign link weights according to the customer. The weight is a measure that quantifies the importance of a link to a given customer. It might be used for defining the format and order of the links. In this case, discarding specific links can be

accomplished by zeroing out their weights. Weight differentiation is implemented within the model by changing the functions so that they consider characteristics of the customer that invoked them.

In both cases, the personalized functions take into account the following three types of information when generating a personalized page:

History: The history is a set $Hu \subseteq \mathcal{O}$ that contains objects relevant to the customer in previous sessions, such as items purchased or information topics researched.

Session: A session Ss , as mentioned, is a list of triples (function, parameters and state) and records the interactions of a given customer in his/her current visit.

Rules: Personalization rules define when a page must be personalized and how the personalization will transform the page. A rule comprises a condition and an action. The condition specifies the requirements for performing the action. In the model, the condition of a rule is based on the history Hu and on the current user session Ss . The action is represented by a set of triples function, parameter, and weight. These triples are the links that will appear in the personalized page. Thus, for each distinct pair Hu, Ss there is a single rule that may be applied. The set of all rules applicable to the site is represented by R .

The generation of personalized pages may be rewritten as $f(Pi, Si, Hu, Ss, R) = pi+1, Si+1$. Where f is the page generation function, Pi is the parameter object base for generating the page, Si is the current state, Hu is the history, Ss is the session, R is the set of rules, $pi+1$ is the new page, and $Si+1$ is the state of the session after applying the function. The personalized page generation algorithm is presented in Figure 2.

Static links and dynamic links, which are based on P , are generated as in the previous algorithm. The only difference is that we assign a weight equal to one to all links. This weight means these links will appear in the page as if no personalization was employed (same order and same format). Besides, we added a new procedure, *verify-rules*, that verifies all rules $r \in \mathcal{R}$ and returns the set of rules that can be applied to the new page. For each rule, a link is created. Each link is associated with the weight suggested by the rule¹.

4 e-Personal

In this section we present the e-Personal framework, which comprises a methodology and a set of tools and libraries that allow easy specification and estimation of effectiveness of personalization strategies. The framework is built based on the model proposed in the prior section.

¹ Weights are used for ordering and formatting links, but our model doesn't provide a mechanism for performing these tasks.

```

function f(P, S, H, Ss, R)
  begin
    Snew = new state(P, S)
    links = generate static links(Snew)
    foreach objeto o > P
      links = generate linkfo(o, Snew)
    end
    foreach link l > links
      give unitary weight(l, S)
    end
    rules = verify rules(H, Ss)
    foreach rule r > rules
      links = generate link(r)
      give weight(l, S, r)
    end
  end
end

```

Figure 2: Algorithm for generating personalized pages

4.1 Personalization Process

The personalization process using the e-Personal framework is divided into five phases:

Goal definition: Describes the motivation for the personalization in terms of its operational and commercial impact.

Observation: Describes the nature and the acquisition strategy of the data that is used in the personalization process.

Transformation: Determines the set of rules to be used during the personalization. This set of rules is derived from the data collected and should be in accordance to the goal of the personalization.

Simulation: The simulation of the personalization strategy is performed using the e-Personal framework, which is described in the sections that follow.

Application: Once the personalization is calibrated within e-Personal, we implement it in the real site, evaluating its actual performance against the estimated performance and adjusting the simulation model.

All phases but the simulation are common to most personalization processes [11]. In this section we will describe aspects related to the transformation and simulation, which are specially important to the e-Personal implementation.

The simulation of a personalization strategy is performed as a replay of an actual log of a web site that starts employing personalization strategies. The goal is simulating the behavior of those customers that visited the site as if there is a personalization strategy implemented on it. In the transformation phase, the original log is parsed in search of

patterns to describe the navigation and the behavior of the customers, which are inputs for the simulation phase. This phase is discussed in the next section.

4.2 Rules and Personalization

In this section we discuss the starting point of the personalization, the rules. Rules comprise a condition and an action. The condition, as the name says, specifies the requirements for performing the action. In the context of the e-Personal framework, both condition and action are requests. The significance of each rule is quantified through two measures inspired on data mining techniques [1]: support and confidence. Support is the percentage of sessions where both condition and action appear. Confidence is the probability that the action is present in a session, given that the condition is. For sake of simulating personalization strategies, we define three types of rules: personalization rules, 2-request observed rules, and n-request observed rules.

The personalization rules comprise the rules used when we generate a personalized page. Each rule defines the probability that an action is suggested, given that the customer submitted the requests that define the rule condition. It is analog to confidence, but applied to future actions. For each action suggested, a link is added to the personalized page. The personalization rules are stored in a separate file, loaded during simulation initialization, and kept in memory during the simulation.

The n-request observed rules provide information about which requests usually occur in the same session, even if they are not consecutive. These rules are used for determining the probability of a suggestion of a page being accepted by the customer, changing his or her observed behavior. It is also analog to confidence. The selection of these rules is done based on their support, i.e., just the rules with a minimum number of occurrences are considered relevant.

Finally, the 2-request observed rules² are used for identifying all possible links that may be followed from a given page in a non-personalized site. In this case, the algorithm determines the links existing in the pages of the original site. These observed rules are generated by the algorithm presented in Figure 3.

The framework allows the simulation of personalization strategies based on groups of customers. We can assign each rule to groups of customers whose sessions may be personalized using the rule. This feature is useful for

² We used just 2-request rules because of the significant increase in the cost of the simulation process when higher order rules are used, but there are no problems in employing them, since the observed rules may comprise more requests by definition.

taylorizing the personalization strategy for different user profiles. Both customers and their groups are also simulation inputs, which are kept in memory during the simulation.

4.3 Personalization Operators

As mentioned, the simulation of a personalization strategy is performed as a replay of an actual log. The simulation is performed in a per-request basis, that is, for each request submitted by a customer, we verify whether it is possible to personalize the response, perform the personalization, and then estimate the customer behavior given that he or she gets a personalized response. This algorithm is presented in Figure 4.

```
1  begin
2    foreach session
3      foreach request in the session
4        if there is a 2-request rule with request
          as condition and request!next as an action
5          update the rule
6        else
7          create such a rule
8        end
9      foreach next-request after request in
          the session
10       if there is a n-request rule with request
          as condition and next-request
          as an action
11         update the rule
12       else
13         create such a rule
14       end
15     end
16  end
17 end
18 Print the 2-request rules
19 foreach n-request rule
20   if support > minimum
21     print the rule
22   end
23 end
24 end
```

Figure 3: Observed rules generation algorithm

Note that the resulting session may differ from the original one, as a consequence of the estimator, but the customer behavior is based on rules observed previously in the site. Thus, the personalization specification comprises the rules and three operators: verifier, transformer, and estimator.

Verifier: The verifier checks whether there are any rules that are applicable to the request being personalized. Thus, the e-Personal library contains functions that handle sets of rules and determine whether there are rules that match the history and session, which are given as parameters. The verifier input is always the current page and the set of personalization rules. All rules in this set are checked by matching the request that generated the current page against the rule condition. The verifier returns the set of rules that matched. Whenever a rule is returned by the verifier, the transformer is invoked. Otherwise, the tool estimates that the customer will choose the same link of the original session.

Transformer: The transformer determines which links will be included in the page from the personalized links found by the verifier. The transformer input is the page being personalized (in fact the request that generated it), the list of rules found by the verifier, and the maximum number of personalized links that may be inserted in the personalized page. This last parameter is inherent to the personalization strategy and is also an e-Personal input parameter.

The rules are chosen using a roulette strategy. The interval between 0 and 1 is divided proportionally among the rules according to their weights (in the experiments presented in the next section, we used their confidence). The pre-defined number of personalized links are then chosen by selecting randomly one of these sub-intervals. The transformer then returns the actions of the selected rules.

Whenever a personalized link is inserted, the estimator is executed. If no personalized links are inserted, the tool estimates that the customer will follow his or her path observed in the original log.

Estimator: The estimator determines the path that is followed by the customer after the personalized links are inserted in the page. This determination is based on the observed probability of transitions and is determined as follows.

The first task performed by the estimator is to check if the user session should end. The probability of a customer exiting a session is a heavy-tail function [3] of the number of requests already submitted and the original size of the session.

If the customer does not exit, he or she may follow one link from three distinct sets of links:

Original: The links that compose the page before personalization. They are called the original links. These links are determined from the 2-request observed rules.³

Personalized: The personalized links inserted by the transformer. The probability that a customer follows a personalized link comes from the n-request observed link.

Session: The session links are those from the original session that is being personalized. The session links indicate those links that are either in the original or personalized sets and then have a higher probability of being chosen by the customer, since he or she did it before. In practice, we multiply the probability of the personalized links by a factor, the predictability, which is an e-Personal parameter. The predictability represents the probability of predicting in advance a request that will be submitted by the customer. Besides, if any links in the original set of links also occur in the session links, we multiply them by another factor, the navigation inertia, which is also an e-Personal parameter. The navigation inertia represents the probability that the customer does not follow a personalized path, or, even after following it, he or she behaves as in his or her original session, maintaining the same path.

After determining these three sets of links, we assign to each link a probability based on the support of the rule that generated it and on the factors. We employ the roulette mechanism again, assigning each link to a unique interval between 0 and 1, which is proportional to its probability occurrence and draw a random number in the same interval, which indicates the link that is followed by the customer.

Once the operators are defined, we should also define how our strategies are evaluated, as discussed next.

4.4 Evaluating Personalization Strategies

Each personalization strategy is evaluated through most of the process described in Section 4.1. We distinguish three types of metrics: observation, strategy, and operator.

Observation metrics quantify the observation and personalization capacity of a site. The observation capacity measures the level of predictability of the behavior of customers based on their previous interactions. The personalization capacity measures the volatility of personalization rules in terms of the number of new rules that are determined over time. Both metrics may be quantified by using the number of new rules as a function of the number of sessions considered. The capacities are proportional to the

asymptotic values of the respective functions. There are basically two problems that may be easily detected using this approach and arise as lack of knowledge in the context of the personalization process: absence of information or volatile behavior. The absence of information pattern is characterized by a monotonically decreasing function that has a non-zero slope. The volatile behavior function is characterized by an asymptotic value which is above the expected, that is, the number of new rules per session does not decrease, meaning that the customer profile keeps changing.

Strategy metrics are based on the personalization goal and quantify the goodness of the strategies. Regarding the personalization goal, we further divide the strategies into two groups: functional and commercial. Functional strategies try to change the user session profile, by either increasing or reducing his or her session length. For instance, a personalization strategy that aims to increase the session length would increase the banner revenues or promote novel services provided by the site. On the other hand, a session reduction strategy tries to increase the customer satisfaction. Commercial strategies are designed to increase the site revenue, which is achieved by increasing the visit-to-buy ratio. Common commercial strategies are cross-selling and up-selling. The metrics for both cases are quite straightforward. Functional strategies are measured by the changes on the session length, while commercial strategies by revenue increases.

Each operator has its set of metrics. For the verifier we measure two metrics: the verifier hit ratio and the rule popularity. The verifier hit ratio is the percentage of requests for which the verifier found any applicable rules. The rule popularity is the ratio between the number of verifications that selected a given rule by the total number of successful verifications. The transformer is also evaluated through two metrics: transformer hit ratio and personalized link popularity. The transformer hit ratio is the percentage of pages augmented with personalized links. The personalized link popularity is the ratio, for each link, between the number of inclusions in personalized pages and the total number of pages personalized. The estimator is evaluated by the personalized link hit ratio. This metric quantifies how frequently customers follow the personalized path.

5 Case Study

In this section we describe the utilization of e-Personal for evaluating personalization strategies for a very popular type of Web site: a portal. Our case study considered a Brazilian Web portal, which provides services such as free e-mail, free web hosting, search, chat, news and online greeting cards. There are also some thematic channels such as shopping, music, weather, and astrology, among others.

After an extensive analysis of the logs, we concluded

³ Note that there may be links in the actual page that are not considered, but since no customer ever clicked on them (otherwise they are in the 2-request observed rule set), we discard them.

that one of the best opportunities for personalization was to increase the number of channels visited per user session. The same analysis has also shown that most of customers do not visit more than two channels per session, although the portal has fifteen channels. The frequency distribution of the number of channels visited per session (Figure 5 - left graphic) shows clearly this problem, since most of visits access very few channels (less than 3). By checking the popularity of the various channels in Figure 5 - right graph, we can see that about half of the channels concentrate most of the requests, showing the potential for a personalization strategy that promotes those popular channels.

Considering that a significant source of revenue of portals is advertisement, a personalization strategy should also try to increase the number of pages (and thus channels) visited, not only providing personalized access to channels, but also increasing the revenue from advertisement. For sake of personalization, we considered each channel as a different function, since the sole purpose of the portal is to provide content.

As described in Section 4.1, the personalization process starts by defining its goal. In this case, the goal is to increase the diversity of channels visited by each customer, which is a navigational personalization. We may describe the strategy as follows: while the customer browses pages from one channel, we suggest links to related pages in other channels. By analyzing the access logs from the non-personalized site, we found that requests to the various channels may be easily identified, since the URL associated to each request identifies

the channel and also contain customer identification, allowing the determination of user sessions based on intervals between requests from the same session [8].

During the transformation phase, we analyzed the sessions to determine which channels are visited in the same session, as discussed above. We then determine other channels of interest by using an association rule algorithm [1], which populated our observed rule base. The application of the personalization strategy to the site was implemented by adding two personalized links per page.

We then simulated the strategy using the e-Personal framework, which estimated the user sessions in a personalized site. The inputs to the framework were defined as follows:

Functions and Objects: Extracted from the log.

Customers and groups of customers: We did not determine any groups of customers, taking all of them as a single group.

Personalization rules: Derived from the association rules, as mentioned.

2-request observation rules: Extracted from the logs using the algorithm described in Figure 4.2. These rules identify the links that compose the pages prior to personalization.

n-request observation rules: These rules are also extracted using the algorithm presented in Figure 4.2 and are used to estimate the links chosen by the customers after personalization.

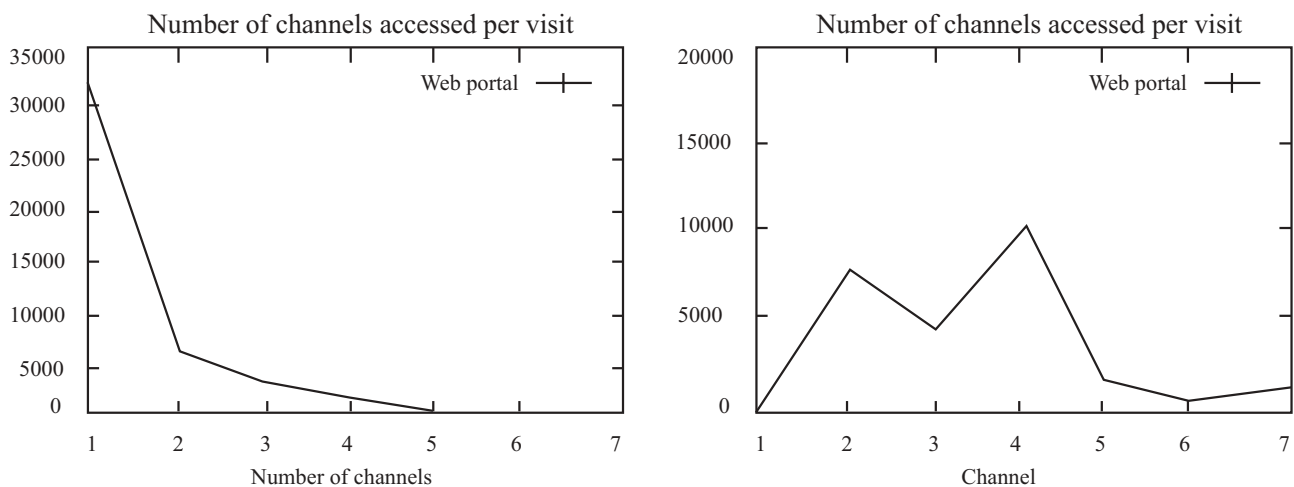


Figure 5: Number of channels accessed per visit and channel popularity

Figure 6 shows the cumulative frequency distribution of rules as a function of its confidence. Notice that more than 90% of the observed rules present a confidence below 60%, affecting the performance of the estimator in determining which link a customer would follow, since estimation is based on rule confidence.

The simulation also takes some parameters:

Predictability: We used 30%, i.e., 30% of the customers will stay in their original session despite the personalization. This parameter should be defined based on the population of site customers, since it reflects the customer's acceptance of novel personalized suggestions.

Navigation Inertia: We used 40%, i.e., 40% of the customers will not click on a personalized link. Similarly to the predictability, this parameter should be also calibrated according to the expectations about the site customers.

After defining the tool input, we performed several experiments. The verifier found rules for 36% of the requests (verifier hit ratio) and never matched 4% of the rules to any request. The remaining rules were verified from 62 to 23,452 times, showing a high variance in the rule popularity. The transformer suggested at least one link per verified request, and the transformer hit ratio is also 36%. The personalized link popularity has also shown a high variance, since the per-link popularity varied from 0.04% to 14%. Finally, 41% of the suggestions were taken by customers.

In order to measure the effectiveness of the strategy, we verified the number of channels accessed before and after personalization. The numbers show that our strategy is quite successful, since the estimated average number of channels visited per session increases from 1.32 to 1.56 (18%). The left graph in Figure 7 compares the frequency distribution of the numbers of channel visited per session before and after the personalization, where we can clearly see that the

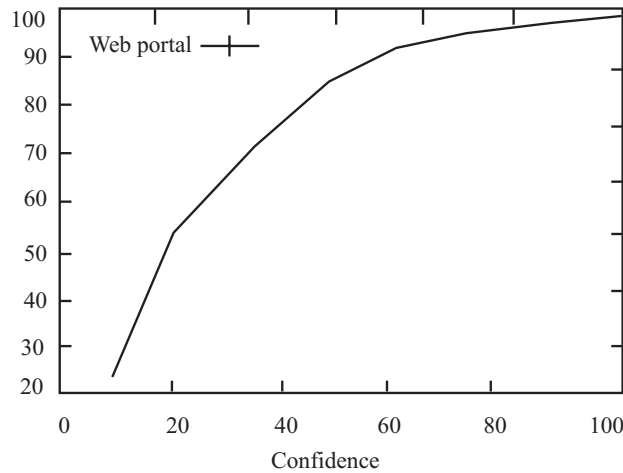


Figure 6: n_request Observed rules

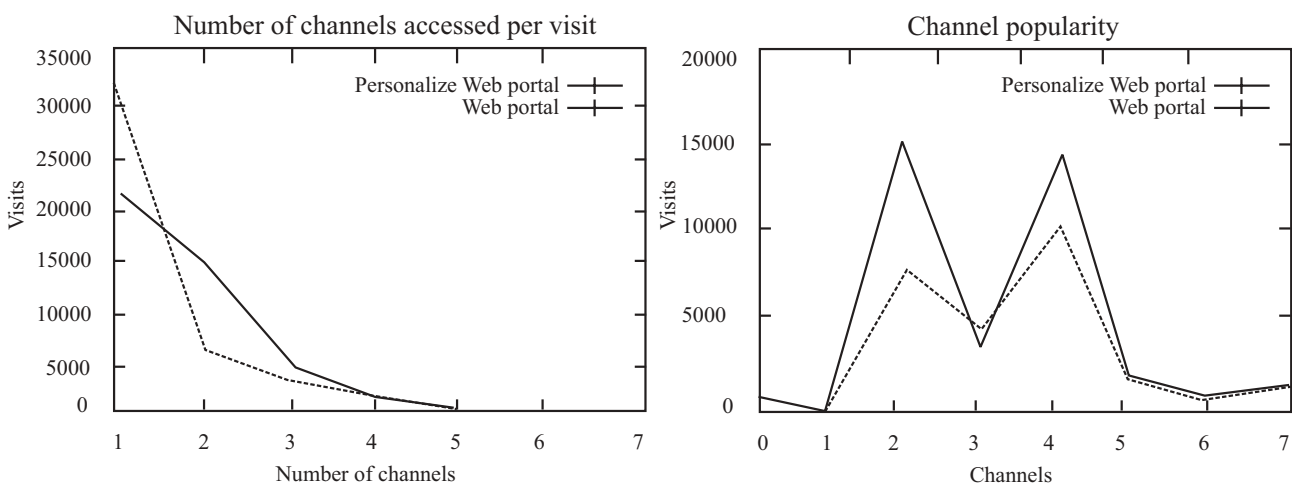


Figure 7: Number of channels accessed per visit and channel popularity

number of visits that accessed just one channel reduced by about one third. In Figure 7 - right graph, we can also verify how the popularity of the various channels changed. In this case, almost all channels got more visits, except those for which almost no personalization rules were generated.

6 Conclusions

In this paper we proposed a model for personalizing Web sites and presented e-Personal, a framework for estimating the effectiveness of personalization strategies. e-Personal helps in designing, implementing, and evaluating strategies based on past behavior of Web site customers. It supports most of the personalization strategies commonly employed and is based on a concise and robust functional model, which makes it applicable to large variety of Web sites, from portals to e-commerce sites. →

We demonstrated the use of e-Personal by discussing personalization strategies for one popular type of Web site, portals. Our personalization strategy is designed to increase the number of channels visited per session, which effectively happened, since that number increased from 1.32 to 1.56 channels per session.

Our next step is to validate our framework implementing the strategies presented and comparing the results with those provided by e-Personal. There are several extensions to the work presented here. For instance, we did not exploit the possibility of generating personalized content per groups of customers, although the framework provides support. Investigating strategies that consider the evolution of the profiles of customers is also planned. Finally, we intend to evaluate other types of personalization techniques, demonstrating the generality of our model.

References

- [1] R. Agrawal, T. Imielinski, and A. Swami. Mining association rules between sets of items in large databases. In *Proceedings of ACM SIGMOD International Conference on Management of Data*, May 1993.
- [2] P. K. Chan. A non-invasive learning approach to building web users profiles. *Workshop on Web Usage Analysis and User Profiles*, 1999.
- [3] A. Feldmann, A. C. Gilbert., W. Willinger, and T. G. Kurtz. The changing nature of network traffic: Scaling phenomena. *Computer Communication Review*, 28(2):5-29, April 1998.
- [4] J. Fink, A. Kobsa, and A. Nill. User-oriented adaptivity and adaptability in avanti project. In *Designing for the Web: Empirical Studies*. Microsoft Usability Group, 1996.
- [5] Y. Fu, K. Sandhu, and M. Shih. Clustering of web users based on access patterns. *Workshop on Web Usage Analysis and User Profiles*, August 1999.
- [6] S. Gomory, R. Hoch, J. Y. Leea, M. Podlaseck, and E. Schonberg. E-commerce intelligence measuring, analyzing and reporting on merchandising effectiveness of online stores. *IBM Institute for Advanced Commerce at <http://www.ibm.com/iac/tech-paper.html>*.
- [7] T. Joachims, D. Freitag, and T. Mitchell. Webwatcher: a tour guide for the world wide web. *15th Int Joint Conf*, pages 770-775, 1997.
- [8] D. Menasc e, V. Almeida, R. Riedi, F. Peligrinelli, R. Fonseca, and W. Meira Jr. In search of invariants for e-business workloads. In *In Proceedings of the 2nd ACM e-Commerce Conference*, pages 56-65, Minneapolis, MN, October 2000.
- [9] D. Murray and K. Durrell. Inferring demographic attributes of anonymous internet users. *Workshop on Web Usage Analysis and User Profiles*, August 1999.
- [10] M. Perkowitz and O. Etzioni. Adaptive web sites: an ai challenge. *Proc. 15th Int. Joint Conf. On Artificial Intelligence*, 1997.
- [11] M. Perkowitz and O. Etzioni. Adaptive web sites: Automatically learning from user access patterns. Technical Report UW-CSE-9703-01, University of Washington, Department of Computer Science and Engineering, March 1997.
- [12] M. Perkowitz and O. Etzioni. Adaptive web sites: Automatically synthesizing web pages. *Proc. 15th Nat. Conf. On Artificial Intelligence*, 1998.
- [13] M. Perkowitz and O. Etzioni. Towards adaptive web sites: Conceptual framework and case study. pages 167-180. *Proc. 8th Int. World Wide Web Conf.*, 1999.

```
1  begin
2    object_loader(objects);
3    function_loader(functions);
4    group_loader(groups);
5    customer_loader(customers);
6    rule_loader(person rules);
7    rule_loader(obs 2 rules);
8    rule_loader(obs n rules);
9    while !feof(log file)
10     old_session = request_loader(log le);
11     new_session  first_request = old_session
12     first_request;
13     current_request = new_session  first_request;
14     while current_request <> NULL
15     list_rules = verier(current_request, person_rules);
16     if list_rules <> NULL
17     person_links = transformer
18     (current_request, list_rules, num_links);
19     if person links <> NULL
20     new_session next_request = estimator
21     (current_request, person_links,
22     old_session, navigation_inertia,
23     predictability, obs_2_rules,
24     obs_n_rules);
25     else
26     new_session  next_request = old
27     session  next_request;
28     end
29     else
30     new_session  next_request =
31     old session  next_request;
32     end
33     current_request = next_session
34     next_request;
35 end
36 end
37 end
```

Figure 4: e-Personal algorithm