An Architecture of e-Butler –

A Consumer-Centric Online Personalization System

Gediminas Adomavicius
Department of Information and Decision Sciences
Carlson School of Management
University of Minnesota
gedas@umn.edu

Alexander Tuzhilin
Department of Information Systems
Stern School of Business
New York University
atuzhili@stern.nyu.edu

This paper presents an architecture of the e-Butler service, i.e., a personalized consumer-centric infomediary delivering a broad range of shopping-related services to consumers. In order to provide these services effectively, e-Butler should be "intelligent." This paper describes how a certain level of intelligence is achieved through the utilization of a broad range of technologies, including data mining, personalization, profiling, recommender systems, data warehousing, Web proxy servers, data monitoring and XML-based technologies, that are integrated into a unique architecture of e-Butler.

---

1. Introduction and Motivations

The rapid growth of online consumers and suppliers of goods and services over the past several years led to an increasing need to match the two groups by processing an enormous amount of information available on the Web and providing value-added services through supplying specialized information that make interactions between the consumers and suppliers easier and more efficient. These types of systems are called *infomediaries* [HS99, GT01]. Priceline.com and Lending Tree constitute examples of infomediaries in the travel and money lending industries. A special type of infomediary provides *personalized* services by taking into the consideration individual preferences and characteristics of the consumers. Most of the work in the personalization area [Cacm97, Paz99, AEK00, Cacm00, SKR01] focuses on *business-centric* approaches since most of the personalization systems were implemented for the companies providing personalized products and services, such as Amazon.com. Moreover, these personalized solutions are provided for a *narrow* range of products and services, such as recommending books, CDs, movies, wines, and newsgroup articles. In contrast to this, relatively little work has been done on *consumer-centric* systems that serve the interests of consumers rather than businesses and that provide personalized solutions across a *broad range* of products and services.

In this paper, we focus on a specific type of consumer-centric infomediary, *e-Butler* [Tuz98], that provides *personalized* online shopping, searching and information gathering services for a *broad range* of products and services. Being a consumer-centric service, it ensures trust of the consumer in various suggestions offered by e-Butler and guarantees that they represent the needs of the customer and not the interests of the businesses "pushing" their offerings on consumers. The e-Butler infomediary provides the following types of services to its customers.

Modify Current and Generate New Browsing Requests. Assume the user generates a new browsing request by specifying a URL of interest. Then the e-Butler service may decide to modify the current URL request by recommending another URL that it thinks should be of more interest to the user. For example, the user may want to visit an online retailer xyz.com in order to buy a shirt. Then e-Butler may recommend to this customer to visit the site uvw.com, because it believes that uvw.com can offer a better value to the consumer for the requested class of products than the site xyz.com does. Moreover, e-Butler can *proactively* recommend (without any prior requests on the part of the user) visiting certain Web sites that it thinks can be of interest to him/her. To do this, the e-Butler service observes user browsing activities and analyses his/her online behavior (e.g., by building user profiles). It also follows various activities on the Web (e.g., sales and promotions) and builds a domain knowledge database, as described below. As a result of knowing the user and following what is "going on" at various Web sites, e-Butler can recommend to the user to visit certain sites that might be of interest to him/her.

Modify Current and Generate New Search Queries. This type of service takes a search query entered by the user and modifies it whenever e-Butler believes that the modified query would better reflect the needs of the user. For example, if the user is searching for walking shoes (i.e., enters the query <"walking shoes">), this type of service may change the query to <*black, "*walking shoes", "*size 10*"> because the user has shoe size 10 and usually buys black shoes. Moreover, e-Butler can generate a *new* search request from "scratch" by observing current user activities, examining his/her profiles, and following other activities on the Web at large. For example, by analyzing the past purchasing history of the user, e-Butler may know that the user may need new sneakers. At the same time, by monitoring various activities on the Web, e-

Butler may know that Nike has a sale on its Web site that might be a good deal for the user. Therefore, it generates a search request on behalf of the user (e.g., find black sneakers of size 10 that are on sale at Nike's Web site) and shows the results to the user the next time he or she uses the service. Some techniques for personalized Web search are already being developed, e.g., [JW02] propose the personalized version of Google's PageRank algorithm that is based on *personalized PageRank vectors* to rank the results returned by a Web search engine according to the individual's notion of importance.

Modify Current and Generate New Purchasing Requests. The e-Butler service can modify various parameters of the purchase and recommend a purchase with modified parameters. These "parameters" are interpreted in a broadest sense and include (a) characteristics of the product, such as color, size, and type; (b) the Web site that sells the product (e.g., the same product can be bought cheaper at a different Web site); (c) billing, shipping and credit card information of the customer (if you use Purple credit card, the product warranty will be doubled); (d) various promotional characteristics of the purchase (such as use coupon X, or buy another pair of shoes and get 50% off). Moreover, e-Butler can monitor various product offerings at different Web sites either using customer-specified or internally built monitoring mechanisms and can generate *new* purchasing requests on behalf of the customer.

To be able to deliver these services and make them valuable to consumers, e-Butler needs to (a) "understand" the consumer, (b) understand the domain of the available services, and (c) be "intelligent" by deploying sophisticated recommendation methods (matching consumers with the most relevant services) that would keep the consumers "happy." To make the e-Butler service "intelligent," it should use various machine learning, data warehousing, personalization,

data monitoring and other "smart" Web technologies, some of which are identified in this paper.

The concept of e-Butler was proposed in [Tuz98], and parts of e-Butler-like capabilities can be found in prior work done in several areas of computer science. In particular, the research on intelligent agents [WJ95, Cacm99b] is relevant to e-Butler because they also provide services assisting customers in shopping or similar activities. However, none of this work provides the range of services described above across a broad range of product categories and delivered with the level of "intelligence" described in this paper. The area of recommender systems [Cacm97, BHK98, Paz99, AEK00, SKR01] is another area of research relevant to e-Butler because providing recommendations to the customers is one of functionalities that comprise the e-Butler service. Moreover, the work on user profiling and user modeling methods [AT01a, Kob01, WPB01] is also relevant because user profiling constitutes one of the functions of e-Butler. Finally, elements of the e-Butler service have been implemented by some commercial Web sites, including MySimon (www.mysimon.com), MyGeek (www.mygeek.com), and Gator (www.gator.com). Most of these services require a plug-in for the customer's browser that monitors customer browsing activities on the Web and reacts at certain points with various offers that vary significantly from one type of service to another. All these services can be viewed as simplified implementations of the e-Butler service. This paper describes an architecture that significantly enhances e-Butler capabilities in comparison to these Web sites.

The main contribution of this paper lies in demonstrating *feasibility* of e-Butler systems by (a) presenting core architecture that is common to the e-Butler systems providing the types of services described above and (b) demonstrating that all the key components of this architecture can be implemented using the *already existing* technologies. Although we demonstrate

feasibility of an e-Butler service in this paper, the true success of such a service depends on how much the customer will be *satisfied* with these services implemented as a part of a working system. Although we have already implemented certain parts of e-Butler, including some profiling and recommendation components [AT01a, AT01b], building a comprehensive real-life e-Butler system constitutes a very large and long-standing effort requiring contributions of many researchers that has not been accomplished yet.

2. An Overall Architecture of the e-Butler System

As shown in Figure 1, infomediaries are usually viewed as interfaces between *users* and certain Web *domains*, such as travel services, loan services, or generic browsing, querying and purchasing services. Users interact with the domain through the interface by making *requests* and are presented with the *results* of the requests retrieved from the domain. The Interface module receives browsing, querying or purchasing requests from the user, sends these requests to the Personalization-Based Intelligence Module for possible personalized modifications to these requests and forwards the modified personalized requests to the domain (some Web sites). The results returned from the domain are forwarded to the user by the Interface module and are also used to update customer profiles.

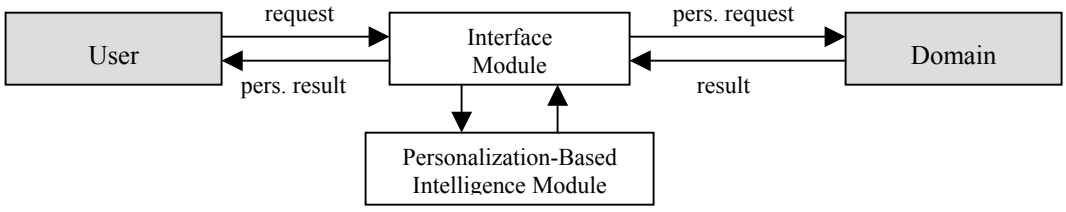

Figure 1. A personalization-based intelligent infomediary.

The personalization aspect of the e-Butler infomediary can be supported via the Personalization-Based Intelligence Module in Figure 1 that (a) modifies, augments, or even automatically generates user requests for services provided by e-Butler and (b) modifies and augments the results of the requests returned to the Interface from the domain. Figure 2 describes the structure of this Intelligence module consisting of five sub-modules that collectively form core components of e-Butler and should be present in one form or another in any comprehensive implementation of e-Butler functionalities described in Section 1. In the rest of this section, we provide a brief overview of these modules and flow of information between them. We describe each of these modules in greater detail in Section 3.

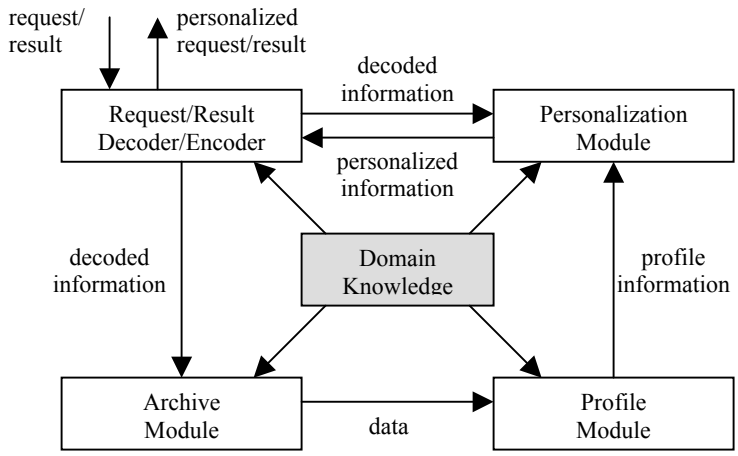

Figure 2. Personalization-based intelligence module for e-Butler.

The formats in which request and result data is transferred over the Internet are usually very different from the formats used by various data analysis algorithms for personalization purposes. Therefore, in order to be able to analyze and then modify user's requests and results retrieved from the Internet (domain), the e-Butler system has to understand and translate them into the data analysis formats and vice versa. In the proposed architecture, the *Request/Result Encoder/Decoder* module implements this capability. Also, in order to determine what is

relevant for each user we need to know what he/she was interested in until now, e.g., which Web sites they frequented, what items they purchased, what search queries they submitted. The *Archive* module in Figure 2 must be able to collect the raw transactional data pertaining to the user and clean, transform, and aggregate it for the subsequent analysis purposes. Moreover, in order to determine what is relevant for each individual user, we must know what their tastes, preferences, and needs are. The *Profile* module implements various methods that take the transactional data stored in the Archive module and generate user profiles that are later used by personalization algorithms. These algorithms are implemented in the *Personalization* module, which can modify and augment the user's requests and results or possibly generate new requests on user's behalf. After receiving the latest request, Personalization module investigates whether there is any way to make it more targeted and personalized based on the information about the user and the domain. And finally, extensive knowledge about the domain is crucial for personalization system effectiveness. The *Domain Knowledge* component contains the domain knowledge base, i.e., various data (e.g., which products are sold in which stores, are they on sale, etc.) and meta-data (e.g., taxonomies, ontologies, attribute descriptions) about the domain.

Figure 2 also presents the main data flows between the modules described above. In the e-Butler system, we identify two main data flows: the *personalization flow* and the *profiling flow*. The personalization flow takes the incoming requests and results (e.g., transmitted as HTTP messages) from the Interface module (see Figure 1) and sends them to Personalization module through the Request/Result Encoder/Decoder, where they are translated into the formats that are understood by personalization algorithms (e.g., relational attribute-value tuples). After the requests and/or results are modified (i.e., personalized) in the Personalization module, the

personalization flow takes them back to the Interface module through the Request/Result Encoder/Decoder. While the personalization flow is responsible for making sure that user's requests and results reach the Personalization module, where they can be augmented, the profiling flow is responsible for making sure that user's transactional activities (i.e., requests and results) are archived and subsequently analyzed for the purpose of learning the up-to-date information about the user's tastes, preferences, and behavior and storing it in the user's profile. The Personalization module can subsequently use this information to augment the user's requests and results according to the user's preferences. More specifically, the profiling flow takes the copies of the incoming request and result data (after Request/Result Encoder/Decoder has translated it) and stores it in the Archive module. This data can be accessed by the Profile module, which analyzes it and uses it to update the user's profile.

As mentioned above, the five modules presented in Figure 2 and described above represent the core functionalities of the e-Butler system that must be present in *any* of its comprehensive implementations. Moreover, in this paper we identify the key components of each of the 5 modules and present them in Section 3 and in Figures 3-5. Therefore, this paper presents the *core* architecture of e-Butler at the level of detail where it is common across the different implementations of the system. However, although the key modules as presented in Figures 1-5 must be the same, implementations of these modules can vary considerably across different e-Butler systems. Moreover, based on a particular implementation of e-Butler, the system designers may also add certain implementation-specific modules to the system. For example, if e-Butler is implemented as a distributed system on several application servers, the *Coordination* module, responsible for distributing and scheduling the workload among the servers, may become a part of the architecture as well. However, in this paper we are focusing

on the core functionalities of e-Butler that should be present in *any* comprehensive e-Butler implementation, and therefore, will focus exclusively on the five modules presented above.

The architecture, presented in this section and summarized in Figure 1 and Figure 2, can be implemented in one of the following ways: (1) on each user's local machine that would contain all the software and data pertaining to the user of that machine; (2) as a client/server architecture: the software and some of the data (e.g., about products and services) are stored on a central server, whereas other (e.g., user-specific) data on the client; (3) on a central server containing the software and the data about all users as well as available products and services.

Among these three options, option (1) is less realistic because of the computationally intensive nature of some of the e-Butler processes and for scalability reasons: we would need to store a copy of all the software and the Domain Knowledge base on each user's computer. Option (2) is also problematic since it would generate a very large amount of network traffic since many of the e-Butler methods are data intensive. Therefore, option (3) provides the only practical approach to building e-Butler. Moreover, this option offers the following additional advantages. First, the collection of the online transactional data is greatly simplified if all the requests go through a central server. Second, communities of practice and various other collaborative capabilities among customers (such as collaborative filtering [BHK98]) can be introduced for the central server model. And finally, privacy preserving services can be provided to the customers using proxy server capabilities [Cacm99a]. Furthermore, hosting the e-Butler service on a third-party server (i.e., independent of any particular supplier) would guarantee the more "objective" personalization that is not biased toward any specific supplier of services, such as an online retailer or a manufacturer. The downside of the central server model is the possible

decrease in the system's response time due to the fact that all users' requests are moving through the central server.

3. Architecture of e-Butler Components

In the previous section we presented an overall architecture of the e-Butler system and identified its key modules as shown in Figure 2. In this section we describe these modules and interactions among them in more detail.

3.1 Request/Result Decoder/Encoder Module

To be able to analyze user's requests and results (usually transmitted over the Internet as HTTP messages), we must be able to decode them by extracting all the relevant information from them. Information extraction on the Web has been addressed extensively in the literature [Car97, BR99, KB00]. While information extraction from HTML documents is a complicated task (mainly because HTML lacks semantic markup [HF99]), this task becomes much more involved when dealing with dynamically generated HTML pages that are not visible to the traditional search engines, i.e., the "hidden" Web [Ber01, IGS01] that by recent estimates comprises 500 billion pages [IGS01].

One solution to this problem supported by many researchers and by the industry lies in the deployment of XML-based technologies that overcome many problems related to parsing static and dynamic HTML pages [GTM99, HF99, Stu00]. One of the main advantages of XML is that the information organized using DTDs (Document Type Definitions) can easily be extracted and validated (both at the syntactic and semantic levels). Also, XML is ideally suited to be a container for data, which would enable us to search for and retrieve product data from online retailers much more effectively and in a more structured manner than for the HTML-based

Web pages [HF99, Stu00]. Moreover, *Web services* can provide standardized application-to-application interactions and exchange of the XML data. For the e-Butler implementation purposes, several Web services technologies are relevant. For instance, instead of HTTP, Simple Object Access Protocol (SOAP) could be used to facilitate the exchange of XML-based messages among applications and, therefore, e-Butlers could use it to gather information from online retailers [GSB+01]. For this to work, the online retailers must provide Web service interfaces using WSDL and UDDI technologies [GSB+01]. Then service users (e.g., e-Butlers) can use registries to search for the relevant services. One of the main issues with XML, however, is that it is very hard for all the online retailers to agree on a common set of standards. To address this issue, an industry-wide consortium, called RosettaNet (www.rosettanet.org), has been created. However, it has yet to develop the set of standards that is adopted by all major "players" in this field.

3.2 Archive Module

Archive module collects and stores copies of all the individual's transactional data that comes through the Encoder/Decoder module. Types of data that can be collected by the e-Butler system may include browsing, querying, and purchasing data pertaining to the user, e.g., timestamps, URL strings, Web page content, query keywords. In particular, browsing data describes which stores the user has been visiting and which items this user has been looking at. Some examples of the collected browsing data may include store id, product id, date, time, duration of a page view, listed price at the viewing time, and other relevant content from the retrieved Web pages. Querying data represents parameters of the search queries that have been issued by the customer. The collected querying data may include the keywords of the search query, product attributes (e.g., color, price, brand) and their respective values (e.g., red, $19.95,

Sony)), date, time, etc. Purchasing data describes what the user purchased, when and in which store. The collected data may include the store id, product id, date, time, price, shipping cost, the payment type, and other relevant information.

The internal structure of the Archive module is presented in Figure 3. This module contains the Transaction Log database, where Request Decoder module places all the users' transactions as soon as they arrive. Before the collected data can be analyzed, it needs to be processed (e.g., cleaned, transformed, aggregated) and stored. Therefore, the Archive module also contains a Data Processor sub-module (as shown in Figure 3) that implements various data pre-processing techniques for the Web log and e-commerce data [Pyl99, Cacm00, SCD+00]. For example, because of the "statelessness" of the HTTP protocol, some of the most commonly used pre-processing techniques in Web data analysis are related to identifying *users* and *sessions* in the clickstream data [SCD+00]. After the data is pre-processed, various data warehousing technologies [Kim96, CD97] are used to store it in a data warehouse as shown in Figure 3

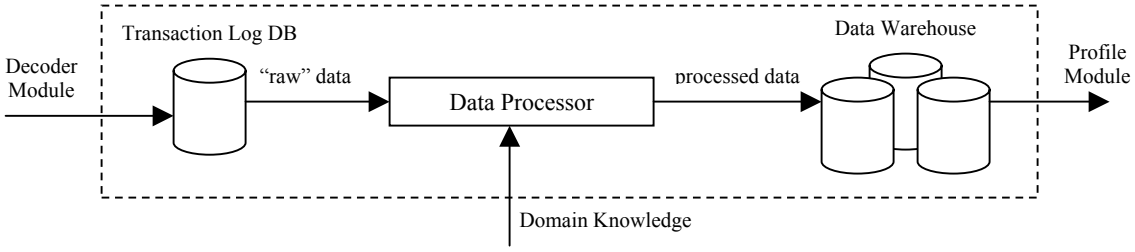

Figure 3. The internal structure of Archive module.

3.3 Profile Module

The Profile module, presented in Figure 4, analyzes the transactional data stored in Archive module and generates, stores and maintains user profiles. It is also responsible for keeping the profiles up-to-date, since tastes, preferences, and needs of users are likely to change over time. The Profile module contains the Archive Monitor, which monitors the changes in data

warehouses of the Archive module (such as arrival of new transactions) using various data monitoring techniques, including data monitoring triggers [TS96] and event monitors [Sno88]. Whenever changes are significant (e.g., enough new data has arrived about a certain user), the Archive Monitor copies the relevant data into the Analysis database, for the subsequent analysis and reevaluation of user's preferences based on the new data.

The data analysis and user profile generation is performed by the Profiler sub-module. Depending on the desired level of personalization, user profiles may range from very simple profiling techniques generating keywords and demographic and psychographic data about the customers to very complex methods building elaborate *user models* [PB97, BMD99, ES99, ZAN99, AEK00, AT01a, WPB01]. Complex profiles (preference models) contain information, indicating how user behaves under different circumstances. There is a substantial amount of research done on how to model user's preferences using various data analysis techniques, including Bayesian models [PB97, CLM+99], decision trees [PB97], association rules [AT01a], clustering [BHK98, OH99], regression models [AEK00, SKK+01], neural networks [PB97], and other user modeling techniques [BMD99, ES99, ZAN99]. For example, the association rule representing the user's purchasing behavior "whenever the user buys a shirt, he usually purchases a tie with it" can be expressed as $shirt \Rightarrow tie$.

Sometimes it is also useful to differentiate between short-term and long-term profiles. Short-term profiles can be maintained only for the current browsing session by analyzing only user's most recent transactions (e.g., what kind of Web sites the user visited in the last half hour). Long-term profiles, on the other hand, may contain user's preferences that are more stable over time (e.g., what kind of Web sites the user visits frequently and periodically). The News Dude

system [BP99] follows this approach and uses different short-term and long-tem profiling techniques to compile a personalized daily news program for individual users.

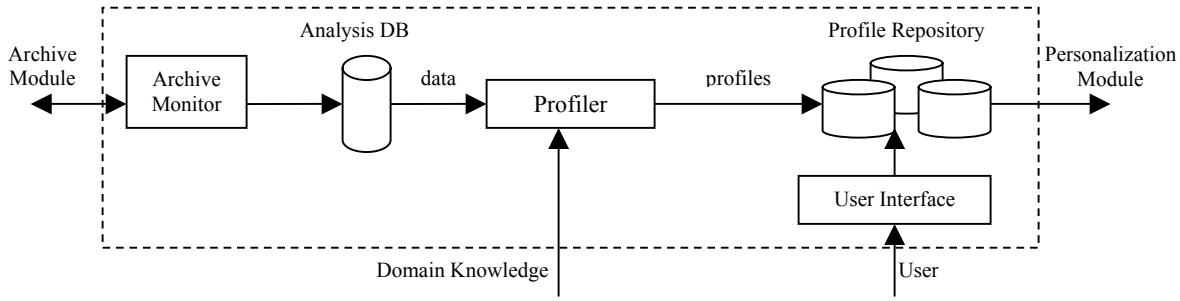

Figure 4. The internal structure of Profile module.

The resulting user profiles are stored in the Profile Repository, as shown in Figure 4. The Profile Repository also stores demographic and other relevant factual data that can be entered directly by the user (e.g., age, gender, income, and marital status). For this purpose, the Profile module also contains a User Interface module, through which a user can inspect and modify all the information in his or her profile. The knowledge representation methods used to store profiles largely depend on the types of profiles being used. Simple factual profiles can be stored as records in a relational database. More sophisticated profiles may require more complex knowledge representation methods including rules and XML-based structures.

3.4 Personalization module

The internal structure of the Personalization module is presented in Figure 5. Its main sub-modules are Request Modifier and Request Generator. The Request Modifier module operates in a synchronous manner by waiting until it receives the information on the current request (e.g., browsing, searching, or purchasing) from Request/Result Encoder/Decoder module described in Section 3.1 and then tries to modify it using the information about user's tastes and preferences. Profile Interface and Domain Knowledge Interface modules provide the interface

capabilities to retrieve the relevant information about the user and the domain, needed to, possibly, modify the request. Because of its synchronous nature, Request Modifier has to be fast to operate in real time. Therefore, it must deploy fast analysis techniques, such as fast pattern matching [AG97] and database querying [Ull90]. For example, the browsing request to see a particular white shirt in the online store XYZ can be augmented to include links to other stores that sell the same shirts at a lower price, by issuing a simple knowledge base query to find all such stores. Also, the purchasing request to buy a certain product using the credit card A can be modified to use the card B in order to not exceed the credit limit of card A.

The Request Generator module is operating in an asynchronous manner: it does not wait for requests, but is constantly monitoring through the Profile Monitor module for any significant changes in user behavior and interests as reflected in user profiles. It also monitors through the Domain Knowledge Monitor module for any significant changes in the domain knowledge base. Significant changes occurring to the profiles and/or domain may trigger new requests generated by the module. For example, if the user recently became interested in hip hop music (i.e., the change in user's preferences is observed – new keyword "hip hop music") and 3 new hip hop albums were just released (i.e., the change in the domain is observed related to the keyword "hip hop music"), the system may issue a certain request on the user's behalf (e.g., email information about the 3 new hip hop albums to the user) whenever it thinks it is appropriate. As with the Archive Monitor located in the Profile Module, similar data monitoring techniques can be used to monitor changes in user profile repository and the domain knowledge base [Sno88, TS96]. As soon as changes are significant enough so that some action can be taken, Request Generator issues a request on behalf of the user. However, before issuing a request the module has to have a high level of confidence that the request to be issued will

indeed be relevant to the user. For example, assume that the user is actively looking to buy a microwave oven and is browsing various stores looking at different microwave models. After a while, the short-term profile of the user (dedicated to tracking user's interests in the current Web session) will reflect the fact, that a significant number of pages visited by this user in the last 30 minutes contain descriptions of microwave ovens with a certain set of characteristics *S*. This change in the short-term profile will be noticed by Profile Monitor. If the user profile also specifies that this user is price sensitive, then the Request Generator can generate the following search request on user's behalf: "find the microwave ovens with characteristics *S* and sort the resulting list by price."

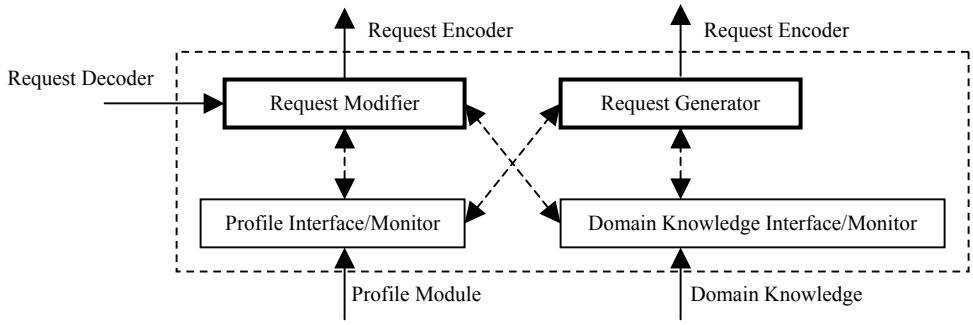

Figure 5. The internal structure of Personalization module.

Request Generator can have different strategies on how to generate new request on behalf of the user. The scenario described above is an example of a *content-based* strategy, where e-Butler tries to find patterns based on the content of Web pages the user has been requesting, in order to predict what user's current needs are. A related work was reported in [DS00] where entropy minimization methods are used in an e-commerce setting to answer the question "when do I have enough information about this individual to make a meaningful offer?" based on users' profile information and transactional information. Another strategy could be a *periodicity*-based strategy that deploys data mining methods, such as the ones reported in

[ORS98], to discover periodic patterns. Such periodic patterns can be stored in user profiles and used for recommendation purposes. For example, we can discover that a user purchases a pair of sneakers approximately every 6 months. Then, if it has been more than 7 months without a purchase of sneakers, e-Butler might consider issuing a search query to find where the sneakers of the user's favorite brand are currently on sale. The research area of recommender systems offers a number of techniques [Cacm97, BHK98, Paz99, SKR01] on how to match users with the most relevant items that could be used in the Request Generator module. While many of these techniques are quite complex and computationally intensive, other techniques, such as [GRG+01, SKK+01], are scalable and could be used not only in the Request Generator, but in the Request Modifier module as well.

3.5 Domain Knowledge module

The Domain Knowledge module contains various data about the stores, product categories, individual products, and other related events. To keep the knowledge base up-to-date, this module has interfaces (e.g., using SOAP protocol [GSB+01] and/or an XML query language, such as XQuery [MCS+01]) to the online retailers, through which it can download the latest information about products, discounts, etc. In addition, the knowledge base must contain meta-data, such as taxonomies, ontologies, attribute descriptions, and business rules [Tiw99, Fen01] that can be used by Personalization and other modules to generate and modify users' requests and results. An example of meta-data that would be useful for e-Butler to have is product hierarchies, e.g., so that the system is aware that "skim milk" is a kind of "milk," "milk" is a "dairy" product, and "dairy" products are "food" products. An example of a useful business rule could be "do not recommend meat to vegetarians". The Domain Knowledge module contains a data and meta-data repository, which is constructed and maintained using various

database, knowledge base [Ull90], and data warehousing [Kim96, CD97] technologies. There is a large body of work done in the AI community on how to represent domain knowledge [SD99, Tiw99, Fen01] that can be leveraged in this module. Meta-data (e.g., taxonomies, business rules, etc.) is usually provided by domain experts; however, it can also be discovered using various data mining methods.

4. Conclusions

In this paper we presented an architecture of the e-Butler service, i.e., a personalization-based consumer-centric infomediary delivering a broad range of shopping-related services to the customers that generates new or modifies existing browsing, querying and purchasing requests. As was argued in [Tuz98], e-Butler is an important service that should help consumers in their daily shopping activities and should simplify their lives. In order to provide these services effectively, e-Butler should be "intelligent." As described in the paper, this is achieved through the utilization of a broad range of "smart" technologies, including data mining, personalization, profiling, recommendation, data warehousing, Web proxy servers, data monitoring and XML-based technologies. The main contribution of this paper lies in demonstrating feasibility of e-Butler by presenting an architecture that integrates all these technologies into a unique system and demonstrating that all the components of e-Butler can be built using already existing methods.

References

[AEK00] Ansari, A., Essegaier, S., and Kohli, R. Internet recommendations systems. *Journal of Marketing Research*, pages 363-375, August 2000.

[AG97] Apostolico, A. and Galil, Z. *Pattern Matching Algorithms*. Oxford University Press, Inc., 1997.

[AT01a] Adomavicius, G. and Tuzhilin, A. Expert-driven validation of rule-based user models in personalization applications. *Data Mining and Knowledge Discovery*, 5(1/2):33-58, 2001.

[AT01b] Adomavicius, G. and Tuzhilin, A. Multidimensional Recommender Systems: A Data Warehousing Approach. In *Proc. of the 2nd Intl. Workshop on Electronic Commerce, Lecture Notes in CS,* vol. 2232, pp. 180-192, 2001.

[Ber01] Bergman, M. The Deep Web: Surfacing Hidden Value. *The Journal of Electronic Publishing*, University of Michigan Press, August 2001.

[BHK98] Breese, J. S., Heckerman, D., and Kadie, C. Empirical analysis of predictive algorithms for collaborative filtering. In *Proc. of the Fourteenth Conference on Uncertainty in Artificial Intelligence*, Madison, Wisconsin, July, 1998.

[BMD99] Bushey, R., Mauney, J., and Deelman, T. The Development of Behavior-Based User Models for a Computer System. In *User Modeling: Proceedings of the Seventh International Conference, UM99*, New York, June 1999.

[BP99] Billsus, D. and Pazzani, M. A Personal News Agent that Talks, Learns and Explains. In *Proceedings of the Third International Conference on Autonomous Agents (Agents '99)*, Seattle, Washington, 1999.

[BR99] Baeza-Yates, R. and Ribeiro-Neto, B. *Modern Information Retrieval*. Addison-Wesley, 1999.

[Cacm97] *Communications of the ACM*, 40(3):56-89, March 1997. Special issue on Recommender Systems.

[Cacm99a] *Communications of the ACM*, 42(2):29-67, February 1999. Special issue on Internet Privacy.

[Cacm99b] *Communications of the ACM*, 42(3):79-114, March 1999. Special issue on Agents in E-Commerce.

[Cacm00] *Communications of the ACM*, 43(8):122-158, August 2000. Special issue on Personalization.

[Car97] Cardie, C. Empirical methods in information extraction. *AI Magazine*, 18(4):65-80, 1997.

[CD97] Chauduri, S. and Dayal, U. An overview of data warehousing and OLAP technology. *ACM SIGMOD Record*, 26(1):65-74, 1997.

[CLM+99] Condliff, M., Lewis, D., Madigan, D., and Posse, C. Bayesian mixed-effects models for recommender systems. In *ACM SIGIR'99 Workshop "Recommender Systems: Analysis and Evaluation"*, August 1999.

[DS00] Dhar, V. and Sundararajan, A. Customer Interaction Patterns in Electronic Commerce: Maximizing Information Liquidity for Adaptive Decision Making. *Proc. of European Conf. on Information Systems*, Vienna, Austria, 2000.

[ES99] Encarnacao, L. and Stoev, S. An Adaptation-Independent Intelligent User Support-System Exploiting Action-Sequence Based User Modeling. In *User Modeling: Proc. of the 7th Intl. Conf., UM99*, New York, June 1999.

[Fen01] Fensel, D. *Ontologies: A Silver Bullet for Knowledge Management and Electronic Commerce*, Springer, 2001.

[GRG+01] Goldberg, K., Roeder, T., Gupta, D., and Perkins, C. Eigentaste: A constant time collaborative filtering algorithm. *Information Retrieval Journal*, 4(2):133-151, July 2001.

[GSB+01] Graham, S., Simeonov, S., Boubez, T., Davis, D., Daniels, G., Nakamura, Y., and Neyama, R. *Building Web Services with Java: Making Sense of XML, SOAP, WSDL, and UDDI*. Sams, 2001

[GT01] Grover, V. and Teng, J. "E-commerce and the information market." *Comm. of the ACM,* 44(4), April 2001.

[GTM99] Glushko, R. J., Tenenbaum, J. M., and Meltzer, B. An XML framework for agent-based e-commerce. *Communications of the ACM*, 42(3):106-114, March 1999.

[HF99] van Harmelen, F. and Fensel, D. Practical Knowledge Representation for the Web. In *IJCAI'99 Workshop on Intelligent Information Integration*, Stockholm, Sweden, July 1999.

[HS99] Hagel, J. and Singer, J. *Net Worth.* Harvard Business School Press, Boston, 1999.

[IGS01] Ipeirotis, P., Gravano, L., and Sahami, L. Probe, Count, and Classify: Categorizing Hidden Web Databases. In *Proceedings of the ACM SIGMOD Conference*, 2001.

[JW02] Jeh, G. and Widom, J. Scaling Personalized Web Search. *Technical Report 2002-12*, Computer Science Department, Stanford University, 2002.

[KB00] Kosala, R. and Blockeel, H. Web Mining Research: A Survey. *SIGKDD Explorations*, 2(1):1-15, 2000.

[Kim96] Kimball, R. *The Data Warehouse Toolkit*. John Wiley & Sons, Inc., 1996.

[Kob01] Kobsa, A. Generic user modeling systems. *User Modeling and User-Adapted Interaction*, 11:49-63, 2001.

[MCS+01] Maharry, D., Cagle, K., Saran, R., Fussell, M., Lopez, L., and Wilper, C. *Early Adopter Xquery*. Wrox Press, Inc., December 2001.

[OH99] O'Conner, M. and Herlocker, J. Clustering items for collaborative filtering. In *ACM SIGIR'99 Workshop "Recommender Systems: Analysis and Evaluation"*, August 1999.

[ORS98] Ozden, B., Ramaswamy, S. and Silberschatz, A. Cyclic Association Rules. In *Proc. of the 14th International Conf. on Data Engineering*. February, 1998.

[Paz99] Pazzani, M. A framework for collaborative, content-based and demographic filtering. *Artificial Intelligence Review*, pp. 393-408, December 1999.

[PB97] Pazzani, M. and Billsus, D. Learning and revising user profiles: the identification of interesting Web sites. *Machine Learning*, 27(3):313-331, 1997.

[Pyl99] Pyle, D. *Data preparation for data mining*. Morgan Kaufmann Publishers, Inc., San Francisco, California, 1999.

[SKK+01] Sarwar, B., Karypis, G., Konstan, J., and Riedl, J. Item-based Collaborative Filtering Recommendation Algorithms. In *Proc. of the 10th International World Wide Web Conf. (WWW10)*, Hong Kong, China, May 2001.

[SKR01] Schafer, J. B., Konstan, J. A., and Riedl, J. E-commerce recommendation applications. *Data Mining and Knowledge Discovery*, 5(1/2):115-153, 2001.

[SCD+00] Srivastava, J., Cooley, R., Deshpande, M., and Tan, P.-N. Web usage mining: discovery and applications of usage patterns from Web data. *SIGKDD Explorations*, 1(2):12-23, January 2000.

[SD99] Sowa, J. and Dietz, D. *Knowledge Representation: Logical, Philosophical, and Computational Foundations*, Brooks/Cole Publishing Co., 1999.

[Sno88] Snodgrass, R. A relational approach to monitoring complex systems. *ACM Transactions on Computer Systems.* 6(2):157-196, May 1988.

[Stu00] Sturm, J. Developing XML Solutions. Microsoft Press, 2000.

[Tiw99] Tiwana, A. *The Knowledge Management Toolkit: Practical Techniques for Building a Knowledge Management System*, Prentice Hall, 1999.

[TS96] Tuzhilin, A. and Silberschatz, A. A belief-driven discovery framework based on data monitoring and triggering. *Technical Report IS-96-26*, Stern School of Business, New York University, 1996.

[Tuz98] Tuzhilin, A. The E-butler service, or has the age of electronic personal decision making assistants arrived? *Working paper IS-98-16*, Stern School, NYU, 1998.

[Ull90] Ullman, J. *Principles of Database and Knowledge-Base Systems, Vols. 1 and 2*. W. H. Freeman Company, 1990.

[WJ95] Wooldridge, M. and Jennings, N. Intelligent agents: theory and practice. *Knowledge Engineering Review*, 10(2), June 1995.

[WPB01] Webb, G. I., Pazzani, M. J., and Billsus, D. Machine learning for user modeling. *User Modeling and User-Adapted Interaction*, 11(1-2):19-29, 2001.

[ZAN99] Zukerman, I., Albrecht, D., and Nicholson, A. Predicting Users' Requests on the WWW. In *User Modeling: Proceedings of the Seventh International Conference, UM99*, New York, June 1999.