Recommendations for Virtual Universities from Observed User Behavior

A. Geyer-Schulz, M. Hahsler, M. Jahn[1]

[1]Abteilung für Informationswirtschaft
Wirtschaftsuniversität Wien, A-1090 Wien, Austria

Abstract: Recently recommender systems started to gain ground in commercial Web-applications. For example, the online-bookseller *amazon.com* recommends his customers books similar to the ones they bought using the analysis of observed purchase behavior of consumers.

In this article we describe a generic architecture for recommender services for information markets which has been implemented in the setting of the Virtual University of the Vienna University of Economics and Business Administration (http://vu.wu-wien.ac.at). The architecture of a recommender service is defined as an agency of interacting software agents. It consists of three layers, namely the meta-data management system, the broker management system and the business-to-customer interface.

1 Introduction

Recommender services for customer relationship management and one-to-one marketing as offered e.g. by Net Perceptions, Inc. (Minneapolis) belong to the hottest e-commerce applications today with an estimated market growth rate of more than 40 percent as reported by Selland et al. (2000). Recommender services are at the heart of business-to-customer e-commerce applications, they **are** the information channels of electronic markets.

In this paper we define a distributed, scalable, and flexible architecture for recommender services for information broker systems as an agency of software agents. See Minsky (1988). The recommendations are based on observed user behavior and experience profiles obtained by self-selection. A prototype of a scientific and educational broker system has been implemented and is currently field-tested in the Virtual University of the Vienna University of Economics and Business Administration. In an educational and scientific environment recommender systems have a considerable potential for improving student/teacher communication, reducing information overload, addressing user heterogeneity, and team-building.

We have structured this paper as follows: In section 2 we present an overview of the architecture of an information broker with recommender services, in sections 3, 4, and 5 we describe each type of software agent required for recommender services in more detail. In section 4 we address the problem of generating recommendations for heterogeneous user groups and we propose a solution based on establishing experience profiles by self-assessment.

2 An Architecture for integrating Recommender Services into an Information Broker

In figure 1 we show an architecture for recommender services as an agency of software agents which consists of three layers, namely the meta-data management system, the broker management system, and the business-to-customer interface. See Russel and Norvig (1995). The interactions between persons, software agents and information stores is represented by arrows, where the direction indicates who starts an activity. A name near an arrow states the nature of the activity, if the arrow is unnamed, it means a simple request for information.

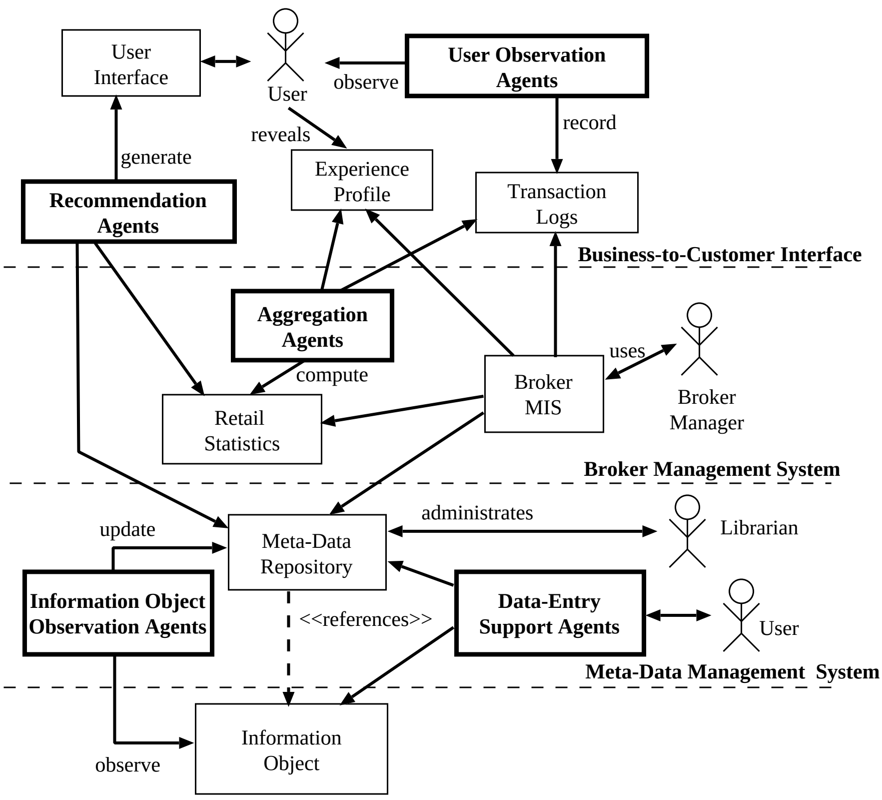

Figure 1: An Architecture for an Information Broker with Recommender Services

On the level of the **meta-data management system** (which can be an instance of a virtual library), every information object (product) is described by its meta-data stored in the repository. Depending on the application area, standards for meta-data may exist, as for example the IEEE P1484.12/D4.0 draft standard for learning object metadata for educational broker systems. See IEEE (2000). However, because the interface to the other layers of this architecture is quite minimal (it requires only a method for retrieving the meta-data by object), the broker management system and the business-to-customer interface are almost completely independent from the database technology used in this layer.

In the meta-data management system we have integrated information object observation agents and data-entry support agents. These are 2 types of software agents which observe, analyze, and classify information objects on the Internet and which update the meta-data periodically. In a virtual university, the main tasks of these agents are to reduce the cost of data entry, administration and maintenance of meta-data, as well as to improve the service quality of the information broker system. Consider, for example, an agent which detects revisions in course material and its application for distributed course-ware versioning control and update systems.

The **broker management system level** and the business-to-customer interface are more tightly coupled. Recommender services are market information services based on observed user behavior. In an information market selecting a recommended information object (e.g. following a link) is considered as a purchase of this information object. Aggregation agents on the broker management system level compute market-baskets, purchase histories and other statistics common in the retail industry (e.g. conditional purchase probabilities from transaction logs and customer experience profiles collected in the business-to-customer interface). Consumer behavior models, diffusion-models and web-mining algorithms are integrated into the broker management information system (broker MIS) which supports the manager of the broker in decision-making about the future development of the information broker system (e.g. decisions on content acquisition, bundling of services, design and placement of user interface elements).

For the recommender agents on the **business-to-customer interface** level the retail statistics provide information on the preferences of users for information objects and for internal broker services which is inferred from observed user behavior recorded in the transaction logs. The user observation agents are also embedded into the business-to-customer interface. Conceptually they observe the behavior of a user and record his transactions. Depending on the degree of anonymity of a user, different methods are applied in order to gain as much information as possible from user transactions. Self-assessment of a user´s experience for the disciplines he is interested in are collected in an experience profile and used to improve the recommender services offered to him. The complete observe - analyze - generate cycle of interacting user observation agents, aggregation agents, and recommendation agents corresponds to to an interactive evolutionary algorithm which evolves for each user a personalized user-interface. The user interface contains several market information services (e.g. favorites of a user, lists of information objects used by users with similar experience profiles, ...).

3 User Observation Agents

User observation agents observe the behavior of a user and record relevant transactions of the user. In the left part of figure 2 the cookie mechanism is used to add session identifiers to the stateless http-protocol for anonymous sessions, in the right part of figure 2 anonymous sessions become associated with a pseudo user id, when the user logs into myVU, the personalized Virtual University environment. What is remarkable about this approach is, that it does not matter, when a user logs into myVU. If he uses a myVU service during a session, the whole session is associated with his pseudo user id.

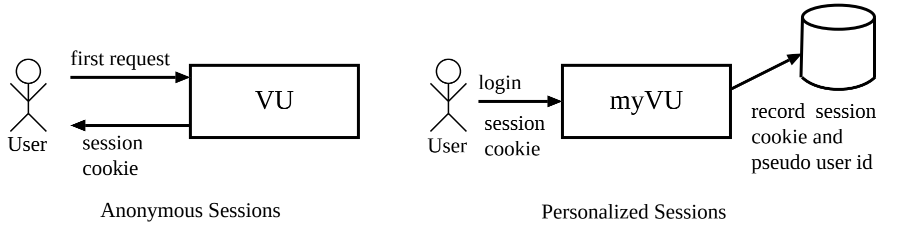

Figure 2: Cookies and Pseudoids

In the following we describe three different user observation agents, namely the VU web transaction log agent, the VU purchase log agent, and the myVU log agent.

The VU web transaction log agent basically is a reconfigured standard web-server with a modified log format shown in figure 3. The entry shown in figure 3 is the record of the purchase of an information object. Line (1) shows the address of the computer used and the date of the purchase. The purchased object is identified in line (2) by the string `file=wu01_698`. Lines (3) and (4) contain the session cookie (`tempvu=16d3f`) and a cookie for myVU authentification (`myvu=myvuf46e...`). The rest is additional information provided by the web server (the time needed to process the request, the referer and the user agent). Of course, a web server records all requests includ-

```
(1)  aipc14.wu-wien.ac.at - - [25/May/2000:16:10:23 +0200]
(2)  "GET /dyn/virlib?&lib=materials/english&file=wu01_698&
(3)  type=stat HTTP/1.1" 302 226 "myvu=myvuf46e35c7706bb432fd08;
(4)  tempvu=16d3f" 0 "http://vu.wu-wien.ac.at/dyn/virlib?
(5)  type=doquery&errors=0&lib=materials&query=
(6)  CATEGORY:='Artificial+Intelligence+AI/Agents'"
(7)  "Mozilla/4.0 (compatible; MSIE 5.0; Windows 95)"
```

Figure 3: VU Web Transaction Logs

ing images, icons and navigation pages. Therefore, the analysis of such transaction log files requires a considerable amount of preprocessing.

The task of preprocessing transaction log files can be considerably simplified by inserting specialized user observation agents into the mediation mechanism of an information broker. The VU purchase log agent which is integrated into the mediation mechanism of the virtual library system which serves as meta-data repository is an example for this approach.

```
[Thu May  4 13:08:27 2000]
   "wu01_533" "tempvu=1478e" "atisrv2.alcatel.at"
[Thu May  4 13:17:24 2000]
   "wu01_8bd" "tempvu=14793" "proxy.luiss.it"
[Thu May  4 13:19:24 2000]
   "wu01_4cf" "tempvu=1479c" "fsztmss01.tu-graz.ac.at"
```

Figure 4: VU Purchase Logs

The VU purchase log agent records purchase incidents as shown in figure 4. A record of a purchase incident consists of a time stamp, the unique identifier of the purchased information object, the session cookie, and the host name of the computer on which the user works with his browser. Note, that this agent eliminates the need for preprocessing and considerably reduces the amount of data stored.

Next, consider the myVU log agent as yet another representative of the species of user observation agents. This agent is integrated into the user-interface elements of the myVU environment. He tracks all actions of a user within myVU and all purchases of information objects of a user in the myVU environment.

Figure 5 shows such a myVU log which contains the purchase history of a user combined with internal broker services accounting. An entry in the purchase history has three elements, namely a time stamp, the record type (`mediate` for a purchase), and the unique information ob-

```
(1) Tue Mar 21 14:05:55 2000 function bookmarks over
(2) Tue Mar 21 14:06:05 2000 mediate   wu01_eff
(3) Tue Mar 21 14:07:13 2000 mediate   wu01_13ab
(4) Tue Mar 21 14:10:33 2000 function bookmarks over
(5) Tue Mar 21 14:10:48 2000 function bookmarks wu_bookmarks
(6) Tue Mar 21 14:10:51 2000 function bookmarks drop_bookmark
   wu01_2b6c
```

Figure 5: myVU Logs

ject id. For an example, see lines 2 and 3 in figure 5. An entry for internal service accounting has at least four elements: time stamp, the record type (`function` for a service), the service name and function with optional arguments. For example, the last line in figure 5 means that the user has deleted the information object `wu01_2b6c` from his personal myVU bookmark list. Again, the myVU log agent considerably reduces the need for preprocessing and he produces a purchase history for information objects and internal broker services.

4 Aggregation Agents

Aggregation agents compute market baskets, purchase histories, ... and estimate e.g. consumer behavior models from the transaction logs generated by the user observation agents periodically. In general, aggregation agents must address the following problems:

- User heterogeneity.
- Discounting the impact of older information.
- Data representation.
- Update period.

User heterogeneity. Aggregation agents must respect differences in the preferences of non-homogeneous user groups. Therefore, identification of user groups is vital for successful aggregation agents. User groups can be identified by several approaches. For example, a-priori segmentation based on demographic user attributes, or cluster techniques on user purchase histories (for a recent survey, see Jain et al. (2000)), or by the principle of self-selection suggested by Shapiro and Varian (1999). For example, in an educational environment the crucial dimension for grouping users is the user's experience in a discipline. In the myVU virtual university environment the principle of self-selection is used for obtaining user groups by incrementally establishing discipline specific user experience profiles.

Discounting the impact of older information. Information objects as well as the preferences of users for them change over time. Aggregation agents discount older observations in the aggregation process in order to take this non-stationarities into account. The challenge is in finding personalized, context-specific discount rates. Too high discount rates lead to fast changes in the user interface which tend to disorient users. Too small discount rates leads to a user interface which is perceived static.

Data representation. Because of the usually large number of information objects in an information broker, aggregation agents must work efficiently on sparse data structures. For example, the representation of empirical conditional cross-selling distributions for information objects in the VU environment is organized as a list of lists which tends to grow with the order of $O(n)$ with the number of items. Representing this as a $n \times n$ matrix leads to $O(n^2)$ growth in the number of items. The data structure must be suitable to support efficient access by the recommendation agents.

Update period. The aggregation process is a time and computationally intensive task. Therefore, aggregation agents usually process observation logs in the background and update statistics only in regular intervals. The time required for each update grows with the size of the transaction logs.

5 Recommendation Agents

Recommendation agents provide recommendations about information objects to the users. They infer information on the preferences of users for information objects and for internal broker services from the statistics generated by the aggregation agents and they generate and bundle appropriate user-interface elements. These agents must cope with the following problems:

- The influence of the user-interface design on purchase behavior.
- Scalability to large numbers of users.

The influence of the user-interface design on purchase behavior. Empirical evidence exists that the design of the user interface has a strong impact on user behavior. For example, Introna and Nissenbaum (2000) discuss the influence of ranking information objects in search engines and the problem of biased search results which is also present for recommendation agents.

Scalability to large numbers of users. The recommendation agents generating the user-interface and its elements have to process all requests of users within acceptable time. The request, the user's experience profile and the statistics provided by the aggregation agents have to be combined and formatted in real-time. For example, in the myVU environment performance is improved by incremental compilation of user interface elements. Scalability can be achieved by using several servers for the interface and distributing the users among them.

6 Links

The reader is welcome to try the anonymous recommender services integrated in the Virtual Univeristy (http://vu.wu-wien.ac.at) of the Vienna University of Economics and Business Administration and to get a myVU account at (http://myvu.wu-wien.ac.at) for experiencing personalized and group recommendations.

7 Acknowledgement

This project is financed by the Jubiläumsfonds of the Austrian National Bank under Grant No. 7925.

References

IEEE (2000): Draft Standard for Learning Object Metadata (IEEE P1484.12/D4.0). IEEE Standards Department, Piscataway, 5 February 2000. http://ltsc.ieee.org/doc/wg12/LOM_WD4.htm

INTRONA, L. and NISSENBAUM, H. (2000): Defining the Web: The politics of search engines, *IEEE Computer*, 33(1), 54–62.

JAIN, A. K., MURTY, M. N. and FLYNN, P. J. (2000): Data Clustering: A Review, *ACM Computing Surveys*, 31(3), 264–323.

MINSKY, M. L. (1988): *The Society of Mind*. Simon & Schuster, New York.

RESNICK, P. and VARIAN, H.R. (1997): Recommender Systems. *Communications of the ACM*, 40(3), 56–58.

RUSSEL, S. and NORVIG, P. (1995): *Artificial Intelligence – A Modern Approach*. Prentice Hall, Upper Saddle River.

SELLAND, C., ANDERSON, H. and the YANKEE GROUP (2000): E-Biz 150: E-Business Winners and Loosers: Trends to Watch. *UpsideToday*, February 18th, 2000.
http://www.upside.com/texis/mvm/ebiz/story?id=38a1ee130

SHAPIRO, S. and VARIAN, H.R. (1999): *Information Rules: A Strategic Guide to the Network Economy*. Harvard Business School Press, Boston.