國立中山大學97學年度博士班招生考試試題

科目：資訊科技論文評述(一)【資管系選考】

Read the attached paper carefully and answer the following questions.

1. Please summarize the paper in English. Your summarization should be about 200 words. (50%)
2. In Section 4.3, a number of notations are defined to illustrate the recommendation chain proposed by the authors. Please describe the meaning of the following notations: (20%)

   A. $(c_{i_{j-1}}, c_{i_j})$

   B. $t(u, v)$

   C. $b(u, v)$

   D. C^2
3. Consider Figure 4. Suppose c_5 finally receives an ad through a recommendation chain $m \rightarrow c_0 \rightarrow c_1 \rightarrow c_5$ and goes shopping using the ad. How many bonus points will each relevant customer receive? (15%)
4. According to the proposed approach, each customer, upon receiving an ad, is free to decide a certain number of bonus points (s)he will keep. If you are the customer, what is your strategy in deciding the retaining bonus points for each incoming ad? Please justify. (15%)

An Anonymous Bonus Point System For Mobile Commerce Based On Word-Of-Mouth Recommendation

Tobias Straub*
tstraub@gkec.tu-darmstadt.de

Andreas Heinemann*
aheine@gkec.tu-darmstadt.de

Department of Computer Science
Darmstadt University of Technology
Wilhelminenstr. 7
D-64283 Darmstadt, Germany

1. MOTIVATION

With the integration of wireless communication technologies like Bluetooth or IEEE 802.11b WiFi into mobile devices (PDAs, mobile phones, etc.), the vision of spontaneous ad-hoc networking becomes feasible.

*The authors' work was supported by the German National Science Foundation (DFG) as part of the PhD program "Enabling Technologies for Electronic Commerce" at the Darmstadt University of Technology.

SAC'04, March 14-17, 2004, Nicosia, Cyprus

A fairly large amount of research has been carried out in the construction of multi-hop ad-hoc networks (MANETs) with the introduction of several kinds of multi-hop routing algorithms (see [14, 9] for a start). Often sensor networks or military applications are named as sample application domains. A key characteristic within these domains is that the individual nodes are strongly related to each other, trust each other and want to accomplish a common goal.

This is very different from ad hoc communication within anonymous groups of humans. Here the following problem exists: Consider the situation in Figure 1 with A, B and C as mobile nodes (i.e. humans equipped with mobile devices).

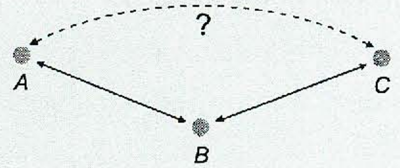


Figure 1: Multihop Communication.

Person A is in communication range to B but not to C who is again also in communication range to B. If A wants to communicate with C, all their traffic has to be routed via B. Having in mind that A, B and C a priori do not know each other, two questions arise:

- What is the incentive for node B to route messages between A and C? Especially, with regard to the fact that battery power is still a limited resource, why would node B donate it to the communication link between A and C?
- Why should node A and C trust and rely on node B for their communication? Node B could easily eavesdrop, manipulate or just reject messages.

As a possible answer to the questions above, we propose a system that deliberately does not support multi-hop communication. In this paper, we describe how to spread digital advertisements (ads) among interested users using a simple one-hop communication protocol. Each user specifies his interests in a profile that is stored on the mobile device. Our communication scheme resembles the way information is spread by word of mouth between human beings, e.g. when recommending something to someone else (see [7] for a thorough discussion of an application scenario).

As an incentive for users to take part in our system, we have designed an anonymous bonus point model that rewards a user who

```
<iWish>
  <advertisement>
    <description>
      <productclass> DVD </productclass>
    </description>
    <due> 2003-12-23 </due>
    <price> EUR 50 </price>
  </advertisement>
</iWish>
```

Figure 2: iWish list.

carries an advertisement on the way down from the vendor to a potential customer.

The paper is structured as follows. In the next section, a brief introduction into the *iClouds* project is given. iClouds devices are used to disseminate information, in this case digital advertisements, between anonymous groups of people and serve as the vital part of the infrastructure for our bonus point system. Section 3 gives an overview of the whole system, names the participants for our proposed bonus point system and describes the different communication steps. The core bonus point model including a precise formal description is formulated in Section 4. Section 5 focuses on the system design, including our design goals, security threats and the technical realization. Our prototype and some runtime measurements are presented in Section 6. We name important related work in Section 7 before we finish our work with a conclusion.

2. THE ICLOUDS PROJECT

In this section, we explain the iClouds project on top of which we deploy our advertisement dissemination and the bonus point model. Please refer to [6, 5] for a more extensive description of the iClouds architecture, how information is stored on the devices, and how it is passed from one device to another.

iClouds devices are small devices with mobile communication support for a maximum of a few 100 meters; one example is a 802.11b-capable PDA. iClouds supports spontaneous one-hop networks of humans where the communication takes place in the user's vicinity. Digital items such as ads are shared and propagated by interest. This exploits the phenomenon of *small worlds* [18], where information is spread by word-of-mouth recommendation.

The two most important data structures found on the iClouds device are the following information lists (*iLists* for short):

- The *iHave list* (information have list) holds all the ads the user wants to pass on.
- In the *iWish list* (information wish list), the user specifies what kind of ads, belonging to a certain product class, he is interested in.

Figure 2 shows an extract of the XML file. There a fictitious user expresses his interest in DVD advertisements that are offered for a price of 50 Euro or less before December, 23rd. Please note that currently we use a fixed taxonomy for our product classes and every product is identified by a unique id[1] (cf. Section 5.2.1). We already addressed the problem of semantics in [6].

Each iClouds device periodically scans its vicinity to check if known nodes are still active and in communication range and also to see if any new nodes have appeared. Information about active nodes is stored in a neighborhood data structure.

In the second stage, the iClouds devices align the stored ads and the ads they are looking for. This is achieved by exchanging iLists. Search entries on the iWish lists are matched against ads on the iHave lists. On a match, ads move from one iHave list to the other.

Digital advertisements in iClouds pass through several nodes, assuming certain conditions are met. First, two nodes must come into communication range, for example about 100 meter using 802.11b WiFi. Second the users share interest in the same ads, expressed through the iLists.

3. COMMUNICATION PATTERN

In this section we describe the *big picture*: the participants and the communication pattern structure.

Our proposed system has three kinds of participants:

- A *merchant* disseminates digital advertisements within its vicinity. For example, there are several fixed nodes located in a merchant's shop. These so-called *information sprinklers*, which are stationary transmitter units, are described in [5]. Customer devices learn about advertisements while their owners are browsing around the shop.
- A *customer* carries an iClouds device. This device collects advertisements and transports or passes the ads to other interested customers. Ideally, some of them come to the shop and buy the advertised good.
- A so-called *mediator* keeps track of the users' accumulated bonus points. In this regard it works similar to a central database where both the merchant and the customer have access to (e.g. via the Internet). Therefore it guarantees reachability to the inherently transient customers (see example below). In addition, the mediator acts as an "anonymizer" to guarantee customers' anonymity. See Section 5.2.3 for details.

Note that our proposed system supports several merchants and customers, but only one mediator is used. Having introduced the participants, we now describe each communication step. All the following refers to the example shown in Figure 3.

1. Customer A visits a merchant. While staying in the shop, his device learns about a lot of advertisements and filters them against the iWish list. These advertisements are stored on the user's device. In the example, customer A learns about a DVD advertisement.
2. Customer A (after leaving the shop) encounters another potential customer B (on the street for instance), who is interested in the ad. B stores the ad and then later passes it on to an interested party C.
3. C itself is taken with the ad, goes to the shop and buys the advertised good. C also passes information to the merchant about how he has learned about the ad – in our example via A and B.
4. The merchant informs the mediator which customers should be rewarded with bonus points.
5. Assuming that there is a standard Internet connection available, A, B and C can download their bonus points from the mediator's server onto their devices. This can happen for example during a sync operation.

[1]Similar to the Amazon Standard Identification Number (ASIN) described on `http://www.wikipedia.org/wiki/ASIN`.

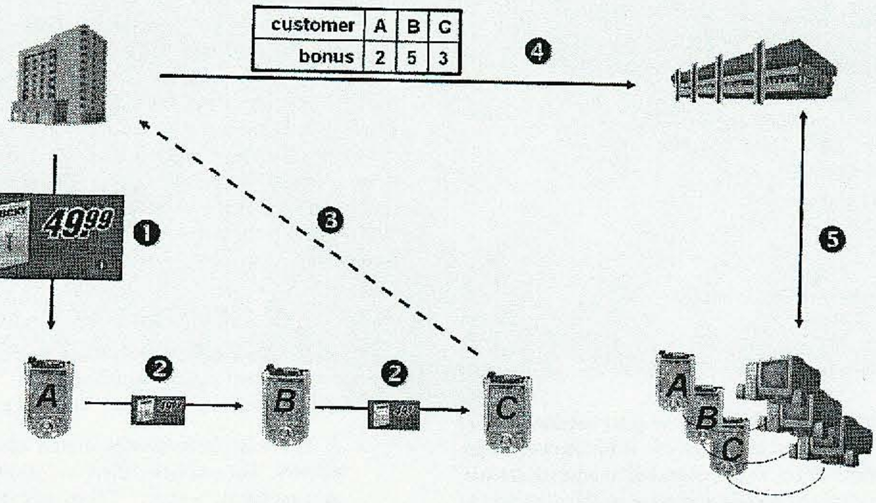


Figure 3: Communication steps in the bonus point model.

4. BONUS POINT MODEL

In this section, we present and formally describe our new bonus point model. First we explain what are the new incentives for participants compared to commonly known bonus point systems. We then introduce the notion of virtual bonus points which leads us to a graph theoretic model. This formalism allows a precise description of the information flow and is a prerequisite for an analysis of the security threats and countermeasures in Section 5. At the end of this section, we briefly discuss possible variants of our model.

4.1 Creating Incentives

Let us first explain the differences of our system compared to common bonus point models. Usually a customer is granted bonus points depending on the price of goods he buys at a certain store, e.g. in the popular PAYBACK system [13]. Thus, from the participating companies' point of view, the idea of such a system is to reward regular customers with small rebates.

Our idea is to disseminate digital advertisements with the help of interested customers following the communication pattern of "word-of-mouth-recommendation" as explained earlier. A merchant will grant bonus points to all people in a *recommendation chain* provided the product in the advertisement is bought. Thus, bonus points are a means of rewarding people not only for buying, but *recommending* or *promoting* products by contributing their system resources.

4.2 Virtual Bonus Points

For technical reasons, we distinguish two types of bonus points. While the advertisement is being distributed among the participants, everybody is allowed to claim a certain share of *virtual* bonus points. These virtual points will only become *real* bonus points if the product is bought by the last person in the chain. Information about preceding participants is always transported along with the ad and, in the end, is sent to the mediator for the purpose of clearing. Virtual bonus points resulting from chains which do not lead to a purchase decay.

4.3 Formal Description

For the sake of simplicity we restrict ourselves to the case of a particular advertisement α, the general case of multiple merchants and advertisements is straight-forward. Relations of participants passing along an advertisement α are modeled by a directed and weighted simple[2] graph $G = G_\alpha = (V, E, b)$.

The set of vertices $V = \{m\} \cup C$ consist of one merchant m and a set of customers $C \subseteq \{c_j : j \in \mathbb{N}\}$. By $E \subseteq V \times C$ we denote the set of edges of G. The mapping $b : E \to \mathbb{N}$ assigns a non-negative weight (number of bonus points) to every edge of G. To describe the timely behavior of our system, we use the mapping $t : E \to \mathbb{R}$ whose values are interpreted as points in time.

For $(v, w) \in E$, we write $b(v, w)$ and $t(v, w)$ in the following as a shorthand for $b((v, w))$ and $t((v, w))$ respectively.

We assume the merchant m is disseminating the ad for a certain period of time staring at the moment $t_0 \in \mathbb{R}$. This dissemination may be typically done by the help of an information sprinkler (see Section 3) broadcasting new offers regularly.

The interpretation of an edge $(v, w) \in E$ is the following: advertisement α was passed along from v to w at the moment $t(v, w) \geq t_0$. At this time the intersection of the communication horizons of v and w was non-empty and, in addition, α matched an entry in w's iWish list.

Because customers can only pass along ads they already have, this imposes a restriction for t, namely

$$(v, w) \in E \cap C^2 \Rightarrow \exists (u, v) \in E \quad t(u, v) < t(v, w). \tag{1}$$

A *recommendation chain* from m to $c_{i_{k-1}}$ (of length k) is described as a sequence $[e_j]_{j=0}^{k-1} = [e_0, \ldots, e_{k-1}]$ of edges $e_0 = (m, c_{i_0})$ and $e_j = (c_{i_{j-1}}, c_{i_j}) \in E$ for $1 \leq j \leq k-1$.

In the beginning the merchant fixes the total number of (virtual) bonus point $b_0 \in \mathbb{N}$ which he will pay for each purchase of the product. These points are shared among the participants in the chain $[e_j]_{j=0}^{k-1}$.

We let each customer c in the chain decide how many of the remaining bonus points to keep. This parameter influences the probability of an ad being passed along over a long distance.[3] For every $c' \in C$ where $(c, c') \in E$, the value $b(c, c') > 0$ denotes how many

[2] A *simple* graph contains neither multiple edges nor loops.

[3] As a matter of fact there is another reason for this design choice which we will explain later.

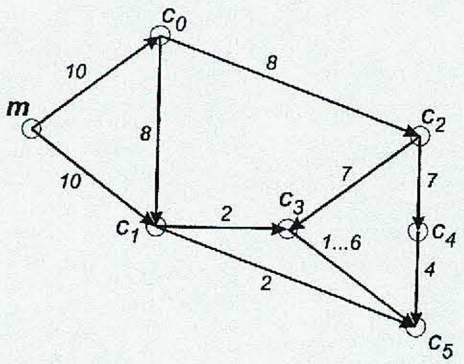


Figure 4: Example of bonus point passing.

bonus points c passes along to c'. A natural restriction on b is that c can only pass along less virtual points than he obtained before. Therefore, condition (1) is modified as follows:

$$(v, w) \in E \cap C^2 \Rightarrow \exists (u, v) \in E$$
$$t(u, v) < t(v, w) \wedge b(u, v) > b(v, w) \qquad (2)$$

Assume the last customer in the chain $(c_{i_{k-1}})$ decides to shop at m's store. Then he gets the product at the price quoted in α and moreover the remaining bonus points $b(e_{k-1})$. The other participants in the chain are granted the same amount of real bonus points as they kept virtual points when passing along the ad. These are precisely $b(e_j) - b(e_{j+1})$ points for customer $c_{i_j}, 0 \leq j \leq k-2$.

Example: We consider a graph with 6 customers, i.e. $V = \{m, c_0, \ldots, c_5\}$ and assume that c_5 will buy the product if he learns from the ad. Using this example we will explain several aspects of our system. Merchant m assigns $b_0 = 10$ bonus points to the product. Figure 4 shows how the ad is then passed along.

- Let us first regard the communication chain $[(m, c_0), (c_0, c_2), (c_2, c_4), (c_4, c_5)]$. Customer c_0 claims 2 points and passes $b(c_0, c_2) = 10 - 2 = 8$ points along. c_2 keeps a single point, i.e. $b(c_2, c_4) = b(c_2, c_3) = 7$ and finally c_4 keeps 3 points.

- If c_5 buys the product, then c_0, c_2, and c_4 are rewarded bonus points for successfully passing along the ad, while buyer c_5 gets the remaining 4 points.

- If c_0 were too greedy and claimed 8 virtual points, the ad would have been decayed from c_4 in the above communication scenario because only one point remained. Therefore, the total number of bonus points is an upper bound for the number of hops. This is much like the TTL (time to live) parameter in IP packages and avoids unnecessary network traffic.

- There is the possibility to learn from an ad via two different communication chains: $[(m, c_1), (c_1, c_5)]$, for instance, is another chain which transports the ad to c_5, however, the number of bonus points for c_5 is less $(b(c_1, c_5) = 2)$.

- The system is time-dependent: Possible values for $b(c_3, c_5)$ are $1, \ldots 6$, depending on the relation of $t(c_1, c_3), t(c_2, c_3)$, and $t(c_3, c_5)$.

4.4 Extensions

It might be interesting to model and study different strategies of keeping and passing along bonus points by game-theoretic methods. We do not go into further details here but discuss some variants and features of our model.

In the example, we implicitly assumed that $b(c, c')$ is constant for a fixed c and arbitrary c' such that $(c, c') \in E$. While this appears to be a natural restriction, participants are allowed to vary the number of points they pass along. We also assumed $b(m, c) = b_0$ for all $c \in C$ such that $(m, c) \in E$. But the merchant may choose different values, e.g. for special offers limited in time.

Each participant c_{i_t} in the recommendation chain $[e_j]_{j=0}^{k-1}$ may define a lower bound for one or more of the values $b(e_t), b(e_t)/b_0$ or even $(b(e_{t-1}) - b(e_t))/b_0$. This is a possibility to express a personal strategy or notion of fairness.

An approach which simplifies the system from the users' point of view is the following: In case a product is bought at the end of a recommendation chain $[(m, c_{i_0}), \ldots, (c_{i_{k-2}}, c_{i_{k-1}})]$, the second-to-last (i.e. $c_{i_{k-2}}$) is rewarded the full bonus b_0 while the others get no bonus at all. This relieves us of storing the values $b(e_j)$.

5. SYSTEM DESIGN

In this section, we explain our system design choices against the background of security issues. The first subsection introduces the relevant protection goals for e-commerce applications which we associate to our scenario in Section 5.2. To solve the arising issues, we use a combination of cryptographic primitives, technical measures and legal practice. A few remarks about efficiency are postponed to Section 6.2 where a prototype is studied.

5.1 Protection Goals

In this section, we briefly list typical security services and protection goals in e-commerce. We consider the simple model of two parties exchanging messages over an potentially insecure medium, e.g. Alice sending an e-mail to Bob.

A data origin *authentication* service provides Bob with assurance that the e-mail came from the claimed originator. Alteration of the e-mail in transit is prevented by a *data integrity* service. *Non-repudiation* is a requirement especially for digital contracts providing the recipient (and other parties) with strong evidence of the origin of a message. It is commonly used in conjunction with an integrity service. A *confidentiality* service protects the contents of the e-mail against disclosure to eavesdroppers.

Anonymity and *pseudonymity*, which is somewhat weaker, are protection goals concerning personal *privacy*. Anonymity means that Bob cannot learn (except with the help of Alice) about who sent him the e-mail, whereas pseudonymity means that Alice uses a unique alias instead of her name which can be mapped to her real name by an authorized party only (e.g. on behalf of a judge).

An important observation about pseudonymity is the following: Bob who knows only the aliases is nevertheless able to link different messages which come from a single person. This is due to the fact that users cannot create new pseudonyms without the help of the authorized party. Removing this constraint, we obtain anonymity.

A somewhat different but not unimportant protection goal is that of *availability* of a system which forms a basis for the other services.

Obviously, the protection goals noted above interfere with each other: For example, combined with pseudonymity, the notion of authenticity refers to the alias, not the real name. In a scenario where anonymity is guaranteed for all parties, confidentiality (concerning

```
<advertisement id="A268">
  <description>
      <productclass> DVD
      </productclass>
      <title> Rocky Collection
      </title>
      <actor> Stallone </actor>
      <info> 5 DVDs [Box Set] </info>
  </description>
  <due> 2003-12-24 </due>
  <price> EUR 49.99 </price>
</advertisement>

<recommendation>
  <ad id="A268" remaining_bonus="10"/>
  <sender> merchant </sender>
  <receiver> customer_1 </receiver>
</recommendation>

<recommendation>
  <ad id="A268" remaining_bonus="2"/>
  <sender> customer_1 </sender>
  <receiver> customer_5 </receiver>
</recommendation>
```

Figure 5: Representation of a recommendation chain.

the identity of the communicating parties, not the message content) comes for free.

5.2 Technical Realization

In this section we link the previously introduced protection goals to our bonus point system. As main issues, we consider integrity with respect to the recommendation chain data, user's privacy, and the problem of unsolicited advertisements. We do not discuss the issue of encrypting recommendations further since this is somewhat counter-intuitive in our system[4] and confidentiality is provided by anonymity to some extent (see the end of the preceding section).

5.2.1 Data Representation

When ads are passed along from one user to another, it is necessary to do the bookkeeping of virtual bonus points. Thus, the relevant data consists of the ad α itself and the recommendation chain $[(m, c_{i_0}), (c_{i_0}, c_{i_1}), \ldots, (c_{i_{k-2}}, c_{i_{k-1}})]$. Customers act both in the role of a *sender* ($c_{i_{t-1}}$) who gives a recommendation and a *receiver* (c_{i_t}) who is next in the recommendation chain.

Each recommendation refers to the same ad by a unique ID and carries information about sender and receiver as well as the number of remaining virtual bonus points. The data structures are sketched in an XML-style in Figure 5 representing the recommendation chain $[(m, c_1), (c_1, c_5)]$ shown in Figure 4.

Ads and recommendations are represented as separate elements encapsulating all the relevant information. Observe that the recommendations are *personalized* since they carry information about the receiver. Compared to Figure 2, a unique `id` attribute has been added to the `advertisement` element. This allows linking the recommendations to a certain ad. An additional purpose of this attribute is to unambiguously identify a merchant.

5.2.2 Integrity Protection

Preventing customers from modifying the recommendation chain $[(m, c_{i_0}), (c_{i_0}, c_{i_1}), \ldots, (c_{i_{t-1}}, c_{i_t})]$ is of vital importance. Without precautions, a malicious user c_{i_t} would be able to increase the actual number of remaining virtual bonus points $b(c_{i_{t-1}}, c_{i_t})$. He could also bypass previous users in the chain before he passes along the ad with a fraudulent recommendation chain, e.g. $[(m, c_{i_t})]$.

A typical way to assure the integrity of data is that of restricting access to it, e.g. granting users only the right to append data to a file. But this is not applicable in our situation since the information travels through the ad-hoc network and does not stay under the control of a single party.

An appropriate standard technique is that of digital signatures. Each user c_j has at least one pair of a private and corresponding public key, say $(\mathrm{priv}_j, \mathrm{pub}_j)$. We will also use the latter one as an alias in our system (see Section 5.2.3 below).

For the sake of simplicity, in this paper we will explain a solution which does not make use of the standard [4] for signatures of XML documents, which is a somewhat heavy-weight method. We store the `advertisement` and the `recommendation` elements in single files each, which can be easily fed to a cryptographic hash function and then digitally signed. This operation is done on the sender's PDA for a recommendation while the ad is exchanged; the merchant can precompute the signature for the ad as it is not personalized. Sender and receiver pass through the following protocol:

(0) The sender, say c_{i_t}, already has ads and the respective recommendation chains from the merchant m as sender in the first `recommendation` element and himself being the receiver in the last `recommendation` element.

(1) The sender publishes his iHave list.

(2) The receiver matches it against his iWish list and demands for a certain ad, say α, to get a recommendation, personalized for his alias.

(3) The sender creates a new `recommendation` structure with a `receiver` element containing the receiver's alias and then signs the corresponding file using priv_{i_t}. The ad itself and all recommendations (and all their signatures files) are then copied to the receiver.

The idea of personalizing the recommendations and then signing it is very similar to the well-known concept of usual public key certificates.[5] It provides some kind of "forward integrity" in a sense that the `recommendation` elements cannot be tampered with later on because then at least one signature becomes invalid. This is similar to a certificate chain where manipulating a single certificate invalidates the whole chain.

Let us have a look at the task of signature verification of recommendation $(c_{i_t}, c_{i_{t+1}})$. It it signed by priv_{i_t}, so pub_{i_t} is needed to verify. But this key can easily be obtained from $(c_{i_{t-1}}, c_{i_t})$ since pub_{i_t} is an alias of c_{i_t} and therefore is stored in the appropriate `receiver` element. We require the merchant have a certificate for his public key which allows users to identify the author of an advertisement (see Section 5.2.4).

5.2.3 User Privacy

Customers give away some personal information by putting advertisement search items onto their iWish list. To face users' concerns about privacy issues and to increase the system's acceptance,

[4]Anyway, the encryption of messages is a conceivable feature.

[5]In fact, it would be possible to implement our system (ab-)using X.509 certificates (see [8]) since Java supports them. Information of the ad element could be stored in extension fields.

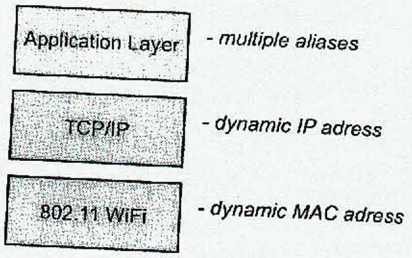


Figure 6: Anonymity w.r.t. the network stack.

we want to prevent other parties from constructing detailed user profiles. Current bonus point systems, for example PAYBACK [13], aim to construct such profiles to trade with and issue bonus points in return.[6]

We can separate three different events, where user privacy must be protected:

(a) While learning new advertisements via the merchant's information sprinkler,

(b) while exchanging ads with other potential customers and

(c) while obtaining (real) bonus points via the mediator.

To protect user privacy during the wireless communication – case (a) and (b) – we dynamically assign network stack IDs that would otherwise be static and therefore exploitable for profile construction. Because our network uses one-hop communication and covers only a limited physical area, we can easily assign dynamic MAC and IP addresses chosen at random from a certain address range. Figure 6 pictures the use of dynamic data with respect to the different network layers.

On the application layer we use public keys as aliases. These public keys are the single content of the `sender` resp. `receiver` elements of the `recommendation` element (this is a slight modification of the content shown in Figure 5).

Each user can change his alias as often he likes to since he can create the corresponding key pair on his own. A paranoid variant would be to use a new key pair for each transaction, a more practical idea is to store a small set of key pairs renewed during a sync operation and choosing one alias at random for each advertisement exchange.

One drawback using this scheme is that a single user may try to increase his bonus points by creating several different aliases to join the recommendation chain multiple times. To avoid this kind of behavior, we deliberately allow users to choose the amount of bonus points they want to keep freely (see Section 4.3).

We now propose a simple solution how customers can get their (real) bonus points paid if a recommendation chain has led to a purchase:

- Assume that customer $c_{i_{k-1}}$ at the end of the recommendation chain $[e_j]_{j=0}^{k-1}$ buys the product α at m's store. The merchant then passes $[e_j]_{j=0}^{k-1}$ and the values $b(e_j), 0 \leq j < k$, on to mediator, but no information about α.
- The mediator serves as a trustworthy party which issues the bonus points on behalf of the merchants participating in the system. These points are temporarily stored together with the respective alias on the mediator's site.
- When customers request bonus points, they have to convince the mediator that their claim is legitimate. Since the mediator knows the public key, it suffices that the customers prove that they know the corresponding private key. This can be done in a challenge-response protocol without revealing the private key (which might still be in use).
- The customers can later on spend their bonus points which can neither be linked to a certain chain nor a product by a merchant.

Note that the user can decide completely on his own whether to uncover his alias or not.[7]

5.2.4 *Unsolicited Advertisements*

An obvious question is how to protect our system against unsolicited ads, since we want merchants to distribute their ads among interested customers. Our customers have to be protected against forged or unwanted ads, because at the least, they will reduce the battery power of the iClouds device.

Let us recall that unsolicited advertisement is a problem with e-mail, because the Internet allows reaching a huge amount of users at virtually no costs and the sender can obscure its authorship.

To circumvent this, we require the merchants using a key pair which is certified by a certification authority (CA); for instance, the mediator may offer this service. Every customer is then able to obtain the authentic public key and to check that an advertisement he learned about is in fact from the alleged merchant.

Merchants use certificates that are issued under a certain CA policy and therefore provide authenticity and non-repudiation, so-called qualified certificates.[8] If a merchant enters unsolicited ads into the system, he can easily be identified and blocked. In addition, by default, all ads that cannot be authenticated are discarded by the device (see Section 6.2 for a discussion about cryptographic operations on a PDA). This approach provides confidence in the advertisements and the promised bonus points to be valid. Furthermore, merchants who fake advertisements or deny the later bonus points redemption take the risk to loose their reputation.

Compared to e-mail, there is only little motivation to spread unsolicited ads since the communication is restricted to a small physical range.

Related to the issue of spam is the legal aspect of *choice of communication*. By checking a button (see prototype Figure 7), the user deliberately expresses his will to collect and pass advertisements.

6. PROTOTYPE

To gain more practical experiences, we have built a first prototype. Since the customer's PDA needs to carry out cryptographic computations, we also give rough runtime estimates.

6.1 User Interface

A screenshot of a customer's PDA is shown in Figure 7. The software runs on several Toshiba Pocket PCs e740 (Windows CE) which are shipped with integrated 802.11b cards. Each network

[6] See `www.bigbrotherawards.de/2000/.com/` for a discussion about the PAYBACK system.

[7] There is a possibility for users to obfuscate the Internet connection to the mediator using the technique of *mix networks* [2]. In this case, the above protocol is modified such that the mediator encrypts the bonus points with the owners RSA public key and stores all these encrypted tokens in a public directory.

[8] In the terminology of the European Commission.

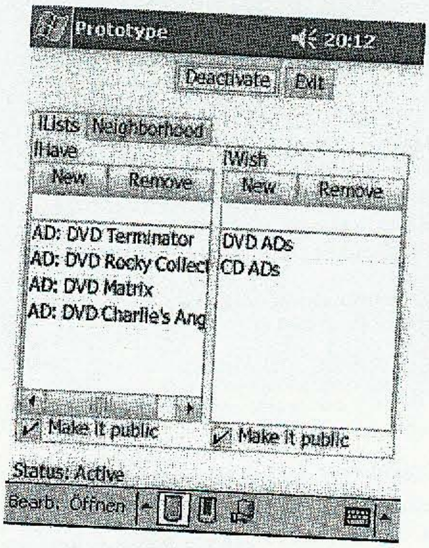


Figure 7: iClouds Pocket PC Prototype.

device runs in 802.11b ad-hoc mode with a statically chosen IP address. For development, we use the Java2 Micro Edition (J2ME).

The screenshot shows the interface of a user who has already collected some advertisements (see the iHave tab). On the iWish tab you can see what advertisements he is interested in: in our example, ads about DVDs and CDs. The checkboxes "Make it public" let the user decide whether to take part in the system or not.

As part of the vendor's implementation, we reuse the customers code base and extend it with administration and communication features, for example the bonus point exchange communication protocol to communicate with the mediator.

6.2 Runtime Measurements

There are some calculations the iClouds devices have to do, so we have taken a closer look at runtime issues to see if our system can be realized efficiently. As signature scheme, we used standard RSA combined with the SHA-1 hash function. We measured the time for generating a key pair, signing and verifying for key lengths from 384 to 1536 bit. Our testbeds consisted of the following components:

- iPAQ Pocket PC (Windows CE 3.0), ARM SA 1110 Proc. 206 MHz, 64 MB RAM,
- CrEme Java VM,
- Cryptix JCE Provider [3].

The timings in Table 1 are averaged over 20 executions of the respective algorithm.

Note that the task of creating a new RSA key pair is in general very time-consuming on a PDA. This can be circumvented easily by using keypairs that were created on a desktop PC in advance. This speeds up the computation by a factor of 20-40. The demand for memory capacity is negligible (about 1-2KB per key pair for the key sizes in question). Also, key pairs can be deleted on the PDA when changing the identity since they are backed up on the PC.

Only the operation of signing data must be completed in real time during the protocol of Section 5.2.2 while the sender and receiver are in communication range. Our experiments shows that this is feasible without doubt. Since signature verification is the simplest task of all, iClouds devices are able to check a recommendation chain on the fly.

The key sizes listed in Table 1 are in part very conservative choices for our scenario. While usually RSA keys of at least 1024 bits are recommended [11], we can use much shorter keys for our application. It is very unlikely that an attacker would try to break the RSA cryptosystem by factoring on his PC because the costs (in terms of CPU usage) would prevail the possible benefit. Also users change their key pairs often and the validity period of offers is limited, so the damage caused by corrupting a single private key is minimal. Because of these lower security requirements, we consider key lengths between 384 and 768 bits as sufficient for signing the recommendations on a PDA. Even if keys can be broken, the merchant does not come to harm since bonus points depend on a concrete purchase.

key bits	operations runtime (sec)		
	key gener.	signature	verification
384	6.28	0.10	0.03
512	12.67	0.18	0.04
640	28.76	0.31	0.06
768	44.79	0.48	0.07
1024	123.54	0.99	0.10
1280	226.45	1.69	0.12
1536	298.91	2.98	0.18

Table 1: Average timings for RSA.

To sign the advertisement, the merchant must use a key pair with a state-of-the-art bit length (at minimum 1024 bits) for which he has a certificate from an appropriate CA. This is due to the fact that the merchant's signature should provide non-repudiation. Verifying such a signature can be done efficiently on the PDA as is shown in Table 1. Doing so, customers can easily learn about the origin of an advertisement and discard unsolicited messages.

7. RELATED WORK

Information sharing among mobile, wireless connected users is also subject of the 7DS architecture [12]. One of the main differences is that nothing is said about incentives for users to contribute their resources, in that case their local cache memories.

The Proem platform [10] supports user collaboration in wireless networks in a more general form and by information sharing only. Again, incentives to join such systems are missing.

Tveit [17] proposes a network of agents, that support product and service recommendation for mobile users. Recommendations are provided by aggregating and filtering individual user input. His ideas may enhance our bonus system in a sense that a customer adds his personal view to the ad that he passes along. Therefore ads about good products will be likely to spread wider and find more customers.

Bonus point models are a natural way to present incentives to join a system and are established in our every day life. Examples are PAYBACK [13] or webmiles [19]. In contrast to the former system, a participant of our model gets bonus points even if he does not make a purchase.

Other examples that wireless communication and collaboration among anonymous humans make sense are the *Shark* system [15] to share knowledge, *Usenet-on-the-fly* [1] for information dissemination and a taxi sharing scenario described in [16].

國立中山大學97學年度博士班招生考試試題

科目：資訊管理論文評述(一)【資管系選考】

1. 如果你的指導教授給你下列的一個論文之研究題目「影響企業引進知識管理系統因素之研究」，請問你會用何種方法來進行(質化 or 量化)?有哪些理論適合此研究?你會如何建立你的研究模式與假說?你的研究設計會是如何(包括抽樣與分析方法)?如何描述此研究的主要貢獻?上述各個議題，請詳細說明(100%)。

Read the attached paper carefully and answer the following questions.

1. In this system architecture, how many types of agents are used? Briefly describe their roles and give an example to illustrate how they work. (15%)
2. Describe how agent communication and agent coordination are implemented in detail to achieve the overall task. (20%)
3. According to the paper, how the intelligent dormitory recognizes individual users? How it provides personalized services? And how it detects and adapts to the user's changing behaviors? (20%)
4. The ISL system uses fuzzy logic-based controllers to provide lifelong learning. Describe how this learning method can be replaced by other methods in detail, and how the lifelong learning can be achieved? (20%)
5. What are the criteria to evaluate a research project like *iDorm*? (15%)
6. How can this ambient-intelligence environment be improved? (10%)

Creating an Ambient-Intelligence Environment Using Embedded Agents

Hani Hagras, Victor Callaghan, Martin Colley, Graham Clarke, Anthony Pounds-Cornish, and Hakan Duman, *University of Essex*

Ambient intelligence is an exciting new information technology paradigm in which people are empowered through a digital environment that is aware of their presence and context and is sensitive, adaptive, and responsive to their needs.[1] Ambient-intelligence environments are characterized by their ubiquity, transparency, and intelligence. In these environments, a multitude of interconnected, invisible embedded systems, seamlessly integrated into the background, surround the user. The system recognizes the people that live in it and programs itself to meet their needs by learning from their behavior.[1]

The Essex intelligent dormitory, iDorm, uses embedded agents to create an ambient-intelligence environment. In a five-and-a-half-day experiment, a user occupied the iDorm, testing its ability to learn user behavior and adapt to user needs. The embedded agent discreetly controls the iDorm according to user preferences.

To realize the ambient-intelligence vision, people must be able to seamlessly and unobtrusively use and configure the computer-based artifacts and systems in their ubiquitous-computing environments without being cognitively overloaded.[1] The user shouldn't have to program each device or connect them together to achieve the required functionality. The complexity associated with the number, varieties, and uses of computer-based artifacts requires that we design a system that lets intelligence disappear into the infrastructure of active spaces (such as buildings, shopping malls, theaters, and homes),[2] automatically learning to carry out everyday tasks based on the users' habitual behavior.

Our work focuses on developing learning and adaptation techniques for embedded agents. We seek to provide online, lifelong, personalized learning of anticipatory adaptive control to realize the ambient-intelligence vision in ubiquitous-computing environments. We developed the Essex intelligent dormitory, or iDorm, as a test bed for this work and an exemplar of this approach.

Intelligent embedded agents

Embedded intelligence refers to including some capacity for reasoning, planning, and learning in an artifact. Embedded computers that contain such an intelligent capability are normally referred to as *embedded agents*[2] and are intrinsic parts of *intelligent artifacts*. These autonomous entities typically have a network connection, thereby facilitating communication and cooperation with other embedded agents to form multi-embedded-agent systems.

Embedded agents in the form of mobile robotic agents can learn and adapt their navigation behaviors online.[3] However, we concentrate on embedded agents in ubiquitous-computing environments that will help us realize the ambient-intelligence vision. Each embedded agent is connected to sensors and effectors, comprising a ubiquitous-computing environment. The agent uses our fuzzy-logic-based *Incremental Synchronous Learning* (ISL) system to learn and predict the user's needs, adjusting the agent controller automatically, nonintrusively, and invisibly on the basis of a wide set of parameters (which is one requirement for ambient intelligence).[4] Thus, we need to modify effectors for environmental variables (such as heat and light) on the basis of a complex, multidimensional input vector. An added control difficulty is that people are essentially nondeterministic and highly individual. Because the embedded agents are located on small embedded computers with limited processor and memory abilities, any learning and adaptation system must deal with these computational limitations.

Most automation systems, which involve minimal intelligence, use mechanisms that generalize actions across a population—for example, setting temperature or loudness to the average of many peoples' needs.

1541-1672/04/$20.00 © 2004 IEEE
Published by the IEEE Computer Society

Related Work

A growing number of research projects are concerned with applying intelligent agents to intelligent inhabited environments and intelligent buildings. In Sweden, Paul Davidsson and Magnus Boman used multiagent principles to control building services.[1] These agents are based on the artificial intelligence thread that decomposes systems by function rather than behavior. Their work does not address issues such as occupant-based learning. In Colorado, Michael Mozer uses a soft-computing approach—neural networks—focusing solely on the intelligent control of lighting within a building.[2] Mozer's system, implemented in a building with a real occupant, achieved a significant energy reduction, although this was sometimes at the expense of the occupant's comfort. Work at the Massachusetts Institute of Technology on the HAL project concentrated on making the room responsive to the occupant by adding intelligent sensors to the user interface.[3] Context-aware systems such as the Aware Home[4] at the Georgia Institute of Technology represent a large body of current research effort but differ from our work in that they are more concerned with time-independent context rather than temporal history or learning, which are central issues in our work.

Other high-profile intelligent-environment projects also exist, such as the Microsoft Smart House, BT's Tele-care, and Cisco's Internet Home.[5] However, most of these industrial projects, including home automation technologies such as Lonworks and X10, are geared toward using networks and remote access with some smart control (mostly simple automation), with sparse use of AI and little emphasis on learning and adaptation to the user's behavior.

References

1. P. Davidsson and M. Boman, "Saving Energy and Providing Value-Added Services in Intelligent Buildings: A MAS Approach," *Proc. 2nd Int'l Symp. Agent Systems and Applications and 4th Int'l Symp. Mobile Agents* (ASA/MA 2000), Springer-Verlag, 2000, pp. 166–177.
2. M. Mozer, "The Neural Network House: An Environment That Adapts to Its Inhabitants," *Proc. Am. Assoc. Artificial Intelligence Spring Symp. Intelligent Environments*, AAAI Press, 1998, pp. 110–114.
3. M. Coen, "Design Principles for Intelligent Environments," *Proc. 15th Nat'l Conf. Artificial Intelligence* (AAAI 98), AAAI Press, 1998, pp. 547–554.
4. G. Abowd et al., "Context-Aware Computing," *IEEE Pervasive Computing*, vol. 1, no. 3, July–Sept. 2002, pp. 22–23.
5. A. Sherwin, "Internet Home Offers a Life of Virtual Luxury," *The Times*, 3 Nov. 1999, p. 10.

However, to achieve the ambient-intelligence vision, any type of intelligence applied to personal artifacts and spaces must be particular to the individual.[2] Furthermore, any agent serving a person must always and immediately carry out any requested action—that is, to achieve the responsive property implied in the ambient-intelligence vision, people must always be in control, subject to overriding safety considerations.[1] The embedded-agent learning technique we've adopted can particularize its actions to individuals and immediately execute user commands. We are testing our embedded agent in the iDorm.

Intelligent inhabited environments and intelligent buildings

Intelligent inhabited environments are spaces such as cars, shopping malls, homes, and even our bodies that respond "thoughtfully" to our needs. Such environments would consist of a multitude of possibly disconnected active spaces providing ubiquitous access to system resources according to the user's current situation. Such environments promise a future where computation will be freely available everywhere, similar to the availability of batteries and power sockets today. These intelligent environments will personalize themselves in response to our presence and behavior.

Intelligent buildings are precursors to such environments.[2] A typical container environment for ubiquitous computing is an intelligent building, possibly a house or office. The heterogeneity, dynamism, and context-awareness in a building make it a good choice to explore ubiquitous-systems design challenges. We view intelligent buildings as computer-based systems, gathering information from various sensors (and other computers) and using intelligent embedded agents to determine various devices' appropriate control actions.[2,3,5] In controlling such systems, we are faced with the imprecision of sensors, the large number of information sources, the lack of adequate models of many of the processes, and the nondeterministic aspects of human behavior. Embedded agents must be able to continuously learn and adapt to the needs of individuals in an intelligent building, while always providing a safe and timely response to any situation.[5] (See the "Related Work" sidebar for other research in this area.)

iDorm

The iDorm (see Figure 1a) is a test bed for ubiquitous-computing environments. We are using the iDorm to test the intelligent-learning and adaptation mechanisms our embedded agent needs with the hopes of realizing the ambient-intelligence vision in ubiquitous-computing environments. As an intelligent dormitory, the iDorm is a multiuse space—that is, it contains areas for varied activities such as sleeping, working, and entertaining—that compares in function to other living or work spaces such as a one-room apartment, hotel room, or office. The iDorm contains the normal mix of furniture found in a study or bedroom, letting the user live comfortably. The furniture (most of which we fitted with embedded sensors) includes a bed, work desk, bedside cabinet, wardrobe, and PC-based work and multimedia entertainment system. The PC contains most office-type programs to support work as well as audio and video services for entertainment (to play music CDs, listen to radio stations using Dolby 5.1 surround sound, and watch television and DVDs).

To make the iDorm as responsive as possible to its occupant's needs, we fitted it with an array of embedded sensors (such as temperature, occupancy, humidity, and light-level sensors) and effectors (such as door actuators, heaters, and blinds).[6] Among these interfaces, we produced the virtual reality system in Fig-

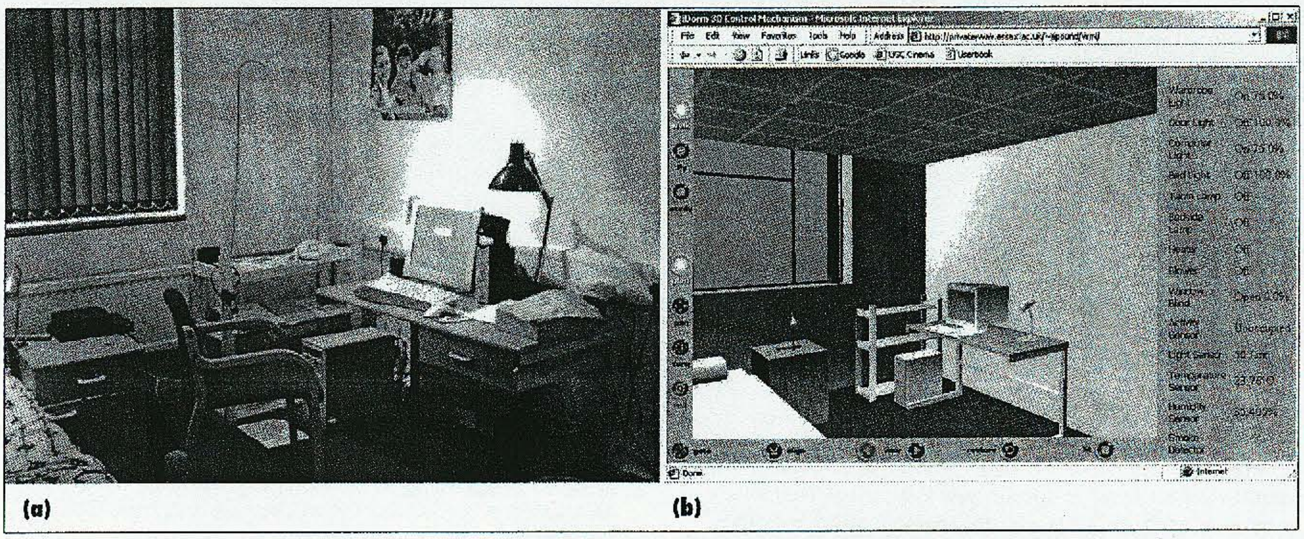


Figure 1. (a) The Essex intelligent dormitory iDorm and (b) the iDorm VRML (Virtual Reality Modeling Language) interface.

ure 1b, which marries the Virtual Reality Modeling Language (VRML) with a Java interface controlling the iDorm. It provides the user with a visualization tool showing the iDorm's current state and enables direct control of the various effectors in the room.

Although the iDorm looks like any other room, the ceiling and walls hide numerous networked embedded devices residing on three different networks: Lonworks, 1-Wire, and IP. They provide the diverse infrastructure present in ubiquitous-computing environments and let us develop network-independent solutions.[6] Because we need to manage access to the devices, gateways between the different networks are critical components in such systems, combining appropriate granularity with security.

Lonworks, Echelon's proprietary network, includes a protocol for automating buildings. Many commercially available sensors and actuators exist for this system. The physical network installed in the iDorm is the Lonworks TP/FP10 network, and Echelon's iLON 1000 Web server provides the gateway to the IP network. This server lets us read and alter the states and values of sensors and actuators via a standard Web browser using HTML forms. Most of the sensors and effectors in the iDorm are connected via a Lonworks network.

The 1-Wire network, developed by Dallas Semiconductor, was designed to connect simple devices over short distances. It offers a range of commercial devices including small temperature sensors, weather stations, ID buttons, and switches. The 1-Wire network is connected to a Tiny Internet Interface board (www.ibutton.com/TINI), which runs an embedded Web server serving the status of the networked devices using a Java servlet. The servlet collects data from the network devices and responds to HTTP requests.

The IP network forms a backbone to interconnect all the networks and other devices, such as the multimedia PC. This PC is the focus for work and entertainment in the iDorm; it also uses the HTTP protocol to display its information as a Web page.

The iDorm's gateway server is a practical implementation of an HTTP server acting as a gateway to each of the room's subnetworks. This shows that by using a hierarchy of gateways, it would be possible to create a scalable architecture across such heterogeneous networks in intelligent inhabited environments and ubiquitous-computing environments.[6] The iDorm gateway server allows a standard interface to all the room's subnetworks by exchanging XML-formatted queries with all the principal computing components. This overcomes many of the practical problems of mixing networks. This gateway server lets the system operate over any standard network such as EIBus or Bluetooth. We could readily develop it to include plug-and-play, letting the system automatically discover and configure devices using intelligent mechanisms.[6] In addition, such a gateway is clearly an ideal point to implement security and data mining associated with the subnetwork. Figure 2 shows the logical network infrastructure in the iDorm.

iDorm's embedded computational artifacts

The iDorm has three types of embedded computational artifacts connected to the network infrastructure. Some of these devices contain agents.

The first type is a physically static computational artifact closely associated with the building. In our case, this artifact contains an agent and thus is termed the *iDorm embedded agent*. This agent receives sensor values through the network, contains the user's learned behavior, and computes the appropriate control actions using the fuzzy ISL system. It then sends them to iDorm effectors across the network. The agent shown in Figure 3a is based on a 68000 Motorola processor with 4 Mbytes of RAM, has an Ethernet network connection, and runs the VxWorks Real Time Operating System.

The agent accesses 11 environmental parameters, some on multifunction appliances:

- Time of day, measured by a clock connected to the 1-Wire network
- Inside room light level, measured by an indoor light sensor connected to the Lonworks network
- Outdoor lighting level, measured by an external weather station connected to the 1-Wire network
- Inside room temperature, measured by sensors connected to the Lonworks and 1-Wire networks
- Outdoor temperature, measured by an external weather station connected to the 1-Wire network

- Whether the user is running the computer's audio entertainment system, sensed by custom code that publishes the activity on the IP network
- Whether the user is lying or sitting on the bed, measured by pressure pads connected to the 1-Wire network
- Whether the user is sitting on the desk chair, measured by a pressure pad connected via a low-power wireless connection to the 1-Wire network
- Whether the window is opened or closed, measured by a reed switch connected to the 1-Wire network
- Whether the user is working, sensed by custom code that publishes the activity on the IP network
- Whether the user is using video entertainment on the computer (either a TV program via WinTV or a DVD using the Winamp program), sensed by custom code that publishes the activity on the IP network

The agent controls nine effectors, which are attached to the Lonworks network:

- A fan heater
- A fan cooler
- A dimmable spotlight above the door
- A dimmable spotlight above the wardrobe
- A dimmable spotlight above the computer
- A dimmable spotlight above the bed
- A desk lamp
- A bedside lamp
- Automatic blind status (open or closed, or at an angle)

Other sensors in the room include a smoke detector, a humidity sensor, activity sensors, and a telephone sensor (to sense whether the phone is on or off the hook) as well as a camera to monitor what happens in the iDorm. It's possible to follow (and control) activities in the iDorm via a live video link over the Internet.

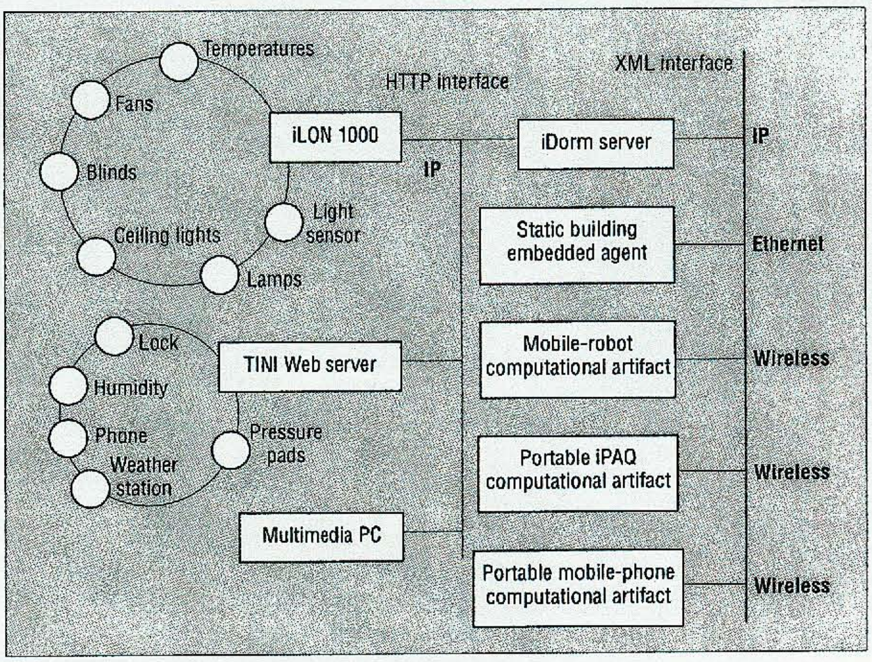
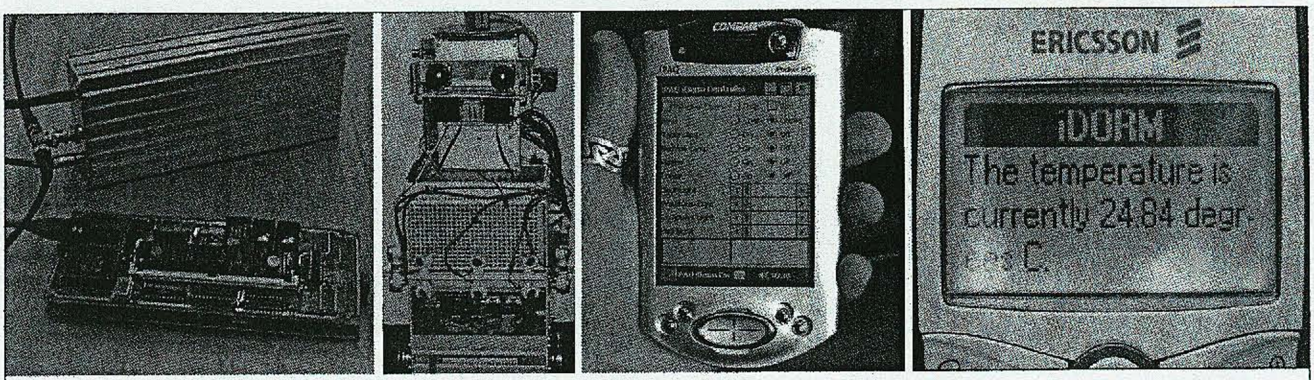


Figure 2. The logical network infrastructure in the iDorm.

The second type of embedded computational artifact is a robotic agent, a physically mobile service robot containing an agent. The robotic agent can learn and adapt robot navigation behaviors online[3] (which is different from the iDorm embedded agent, which seeks to realize ambient intelligence). Figure 3b shows the robot prototype we use in the iDorm. The robot is a servant-gadget for delivering various objects of interest to the iDorm user such as food, drink, and medicine. It has a rich set of sensors (nine ultrasound sensors, two bumpers, and an IR beacon receiver) and actuators (wheels). It uses 68040 Motorola processors and runs the VxWorks Real-Time Operating System.

The robot is equipped with essential behaviors for navigation, such as obstacle avoidance, goal seeking, and edge following. We combined and coordinated these behaviors with a fuzzy coordination module so that the robot could reach a desired location and avoid obstacles. The static embedded agent that controls the iDorm passes and processes the robot's location as an additional input. In



(a) (b) (c) (d)

Figure 3. The iDorm embedded computational artifacts include (a) the static iDorm embedded agent, (b) a mobile service robot, (c) a portable iPAQ (pocket PC) interface, and (d) a portable mobile-phone interface.

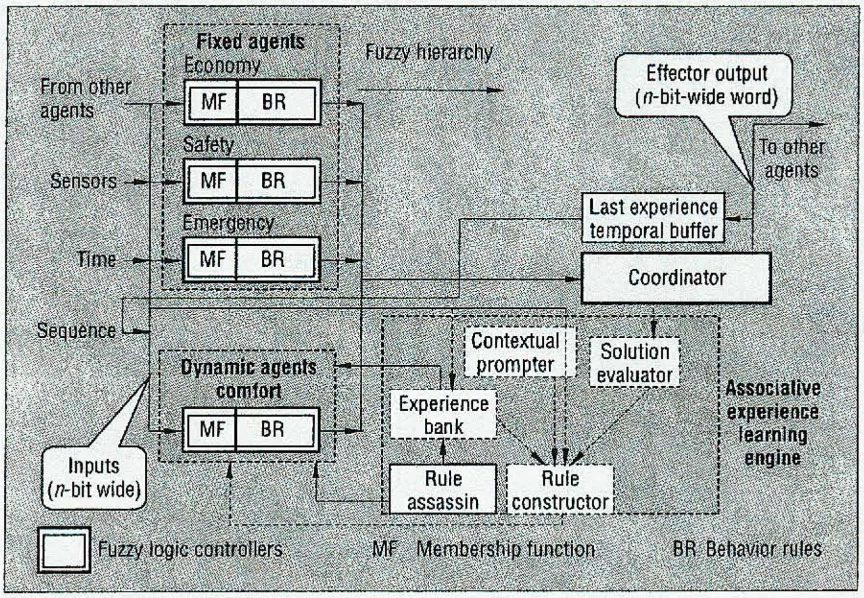


Figure 4. The Incremental Synchronous Learning architecture.

the experimental setup, we use a simplified system in which the robot can go to two locations identified by infrared beacons to pick up objects. After picking up an object, the robot can deliver it to the user and then go to its charging station, which another infrared beacon identifies. The robotic agent sends information about its location to the iDorm agent, and it takes destination instructions from that agent depending on the user's learned behavior. For example, the robot might learn to fetch a newspaper from a specific location whenever it is delivered in the morning.

We implemented the communication between the static embedded agent and the mobile robotic agent via a wireless link. We establish communication by initiating a request from the embedded agent to the mobile agent server. Once the request has been sent, the server passes it to the robotic agent to carry out the task and informs the iDorm embedded agent of the robot's current status. If the task is in progress or not completely finished, the server sends a message indicating that the job is incomplete. Every time the iDorm embedded agent wants to send out a new request, it waits until the robot successfully completes the previously requested job.

The third type of embedded computational artifact is a physically portable computational device. Typically, these are wearable technologies that can monitor and control the iDorm wirelessly. The handheld iPAQ in Figure 3c contains a standard Java process that can access and control the iDorm directly. This forms a type of remote control interface that would be particularly suitable to elderly and disabled users. Because the iPAQ supports Bluetooth wireless networking, it's possible to adjust the environment from anywhere inside and nearby outside the room. It's also possible to interact with the iDorm through mobile phones because the iDorm central server can also support the Wireless Markup Language. Figure 3d shows the mobile-phone wireless application protocol interface, which is a simple extension of the Web interface. Such portable devices can contain agents, but this remains one of our longer-term goals.

We designed the learning mechanism in the embedded agent to learn behaviors relating to different individuals. To achieve this, the embedded agent must be able to distinguish between users in the environment. This is achieved by using an active key button, designed and built by our research team and based on Dallas Semiconductor's 1-Wire protocol. Each user is given an electronic key about the size of a penny. This is mounted onto a key fob and contains a unique identification number inside its 2-Kbyte memory. The user's unique ID number is passed to the iDorm embedded agent so that it can retrieve and update previous rules it learned about that user.

Fuzzy, incremental, synchronous learning technique

In our work, learning is achieved through interaction with the actual environment. We call this *online learning* because adaptive behaviors can't be considered a product of an agent in isolation from the world but can only emerge from a strong coupling of the agent and its environment.[5]

Figure 4 shows the ISL architecture, which forms the learning engine in the iDorm embedded agent. The ISL system aims to provide lifelong learning and adapts by adding, modifying, or deleting rules. It is memory based in that the system can use its previous experiences (held as rules) to narrow down the search space and speed up learning. The embedded agent is an augmented-behavior-based architecture, which uses a set of parallel fuzzy logic controllers (FLCs), each forming a behavior. We use the FLC approach because it's useful when the processes are too complex for analysis by conventional quantitative techniques or when the available sources of information are interpreted qualitatively, imprecisely, or uncertainly[3,5] (this is the case with embedded agents operating in intelligent inhabited environments and ubiquitous-computing environments). For embedded agents, the number of inputs and outputs are usually large, and the desired control behaviors are complex. However, by using a hierarchical assembly of fuzzy controllers, we significantly reduce the number of rules required.[3,5]

In general, we divide the behaviors available to the iDorm embedded agent into fixed and dynamic sets, where the dynamic behaviors are learned from the person's behavior and the fixed behaviors are preprogrammed. We predefined these latter behaviors because they can't easily be learned—for example, the temperature at which water pipes freeze. The fixed behaviors include safety, emergency, and economy behaviors. A safety behavior ensures that the environmental conditions are always at a safe level. An emergency behavior (in case of a fire alarm or another emergency) might open the emergency doors and switch off the main heating and illumination systems. Economy behaviors ensure that energy isn't wasted so that if a room is unoccupied, the heating and illumination will be switched to a sensible minimum value. All these behaviors are fixed but adjustable.

Each dynamic FLC (the comfort behavior in the iDorm case) has one parameter (which is the rule base for each behavior) that we can modify. Also, at the high level, the coordination parameters can be learned.[3,5] Each behavior uses a FLC using a singleton fuzzifier, triangular membership functions, product inference, max-product composition, and height defuzzification. We chose these techniques because of their computational simplicity and real-time considerations. The equation that maps the system input to output is

$$Y_t = \frac{\sum_{p=1}^{M} y_p \prod_{i=1}^{G} \alpha_{Aip}}{\sum_{p=1}^{M} \prod_{i=1}^{G} \alpha_{Aip}}$$

where M is the total number of rules, y_p is the point having maximum membership in the pth rule output fuzzy set, $\Pi\alpha_{Aip}$ is the product of the membership functions for each rule's inputs, and G is the number of inputs.

We use a higher-level coordinator FLC to combine the preferences of different behaviors into a collective preference (giving a two-level behavior hierarchy). (Previous work gives additional information about the fuzzy hierarchical architecture.[3,5])

The ISL works as follows: when new users enter the room, they are identified by the active key button, and the ISL enters an initial monitoring mode where it learns the users' preferences during a nonintrusive cycle. In the experimental setup, we used a 30-minute period, but in reality, this is linked to how quickly and completely we want the initial rule base. For example, in a care home, we might want this rule base to be as complete as possible, and in a hotel, we might want this initialization period to be short to allow fast learning. The rules and preferences learned during the monitoring mode form the basis of the user rules, which are reactivated whenever the user reenters the room. During this initialization period, the system monitors the inputs and the user's action and tries to infer rules from the user's behavior. The user will usually act when a set of environmental conditions (an input vector) is unsatisfactory by altering the output vector (for example, the user needs to turn a light on or adjust the heating). Learning is based on negative reinforcement because users will usually request a change to the environment when they are dissatisfied with it.

After the monitoring period, the ISL enters a control mode in which it uses the rules learned during the monitoring mode to guide its control of the room's effectors. Whenever the user behavior changes, it might need to modify, add, or delete some of the rules in the rule base. Thus, the ISL goes back to the nonintrusive cycle to infer rule base changes—that is, to determine the user's preferences in relation to the specific components of the rules that have failed. The user is essentially unaware of this short cycle; such modifications and adaptations are distributed throughout the lifetime of the environment's use, thus forming a lifelong-learning phase.

As in the case of classifier systems, to preserve system performance, we let the learning mechanism replace a subset of the classifiers (the rules in this case). The worst m classifiers are replaced by m new classifiers.[4] In our case, we change all the consequents of the rules whose consequents were unsatisfactory to the user. We find these rules by finding all the rules firing at this situation

> This marks a significant difference in our method of classifying or managing rules compared to other work: rather than seeking to extract generalized rules, we try to define particularized ones.

whose firing strength is $\Pi\alpha_{Aip} > 0$. We replace these rule consequents by the fuzzy set that has the highest membership of the output membership function. We make this replacement to achieve nonintrusive learning, avoiding direct interaction with the user. The set of learned-consequent fuzzy rules is guided by the contextual prompter, which uses sensory input to guide the learning.

During the nonintrusive monitoring and the lifelong-learning phases, the agent encounters many different situations as both the environment and the user's behavior change. For example, the agent will try to discover the rules needed in each situation guided by the occupant's behavior in response to different temperature and lighting levels inside and outside the room. The learning system consists of different learning episodes; in each situation, the agent will fire only a small number of rules. The model the agent must learn is small, as is the search space. The accent on local models implies the possibility of learning by focusing at each step on only a small part of the search space, thus reducing interaction among partial solutions. The interaction among local models, due to the intersection of neighboring fuzzy sets, means local learning reflects on global performance.[4] Thus, we can have global results from the combination of local models and smooth transition between close models. By doing this, we don't need to learn the complete rule base all at once, only the rules the user needs during the different episodes. This marks a significant difference in our method of classifying or managing rules compared to other work: rather than seeking to extract generalized rules, we try to define particularized ones.

The system has an Experience Bank that stores all the previous occupiers' rule bases. After the initial monitoring phase, the system tries to match the user-derived rules to similar rules stored in the Experience Bank that were learned from other occupiers. The system chooses the rule base that's most similar to the user-monitored actions. By doing this, the system is trying to predict the rules that weren't fired in the initialization session, thus minimizing the learning time as the search starts from the closest rule base rather than starting at random. This action should be satisfactory for the user as the system starts from a similar rule base and then fine-tunes the rules.

Subsequently, the agent operates with rules learned during the monitoring session plus rules that deal with situations uncovered during the monitoring process, which are ported from the most similar user's rule base. All the rules that are constructed and added to the system are symbolized by the Rule Constructor block in Figure 4. The system then operates in the control mode with this rule base until the occupant's behavior indicates that his or her needs have altered; this change is flagged by the Solution Evaluator (that is, the agent is event-driven). The system can then add, modify, or delete rules to satisfy the occupant by briefly reentering the monitoring mode. In this case again, the system finds the rules fired and changes their consequent to the user's action. In this way, the system implements a lifelong-learning strategy.

Because we're dealing with embedded agents with limited computational and memory capabilities, it's difficult to deal with a large number of rules in the rule base. For example, for the iDorm comfort behaviors in our current implementation, a complete rule base contains 62,208 rules. This would lead

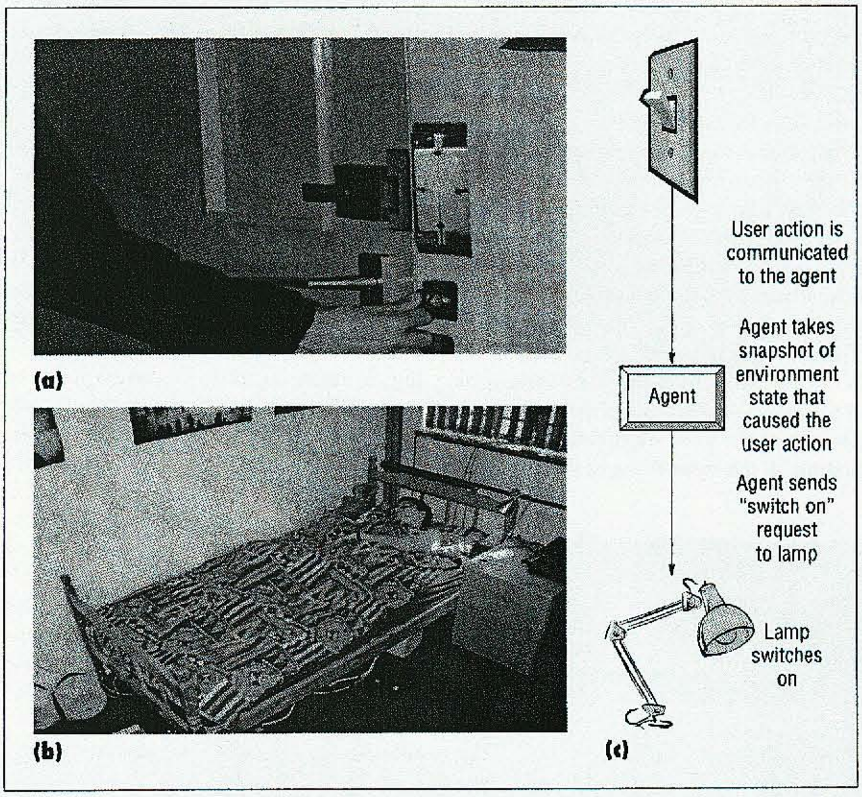


Figure 5. The user is (a) using the iDorm's active lock and (b) sleeping in bed. (c) The agent communication path.

to large memory and processor requirements that are unrealistic in embedded agents. So, we limited the number of stored rules to 450 (in our case, the maximum number the agent can store on the onboard memory without exceeding the memory limit or degrading real-time performance). Each rule will have a measure of importance according to how frequently it's used. In calculating this degree of importance, we also include a most-recent-use measure. When the system reaches the memory limit, the Rule Assassin retains rules according to the priority of highest frequency of use, followed by most recently used. If two rules share the same degree of relative rule frequency recall, the system breaks the tie by eliminating the least recently used rule. This action lets the onboard memory store only the most efficient and frequently used rules and reduces degradation of the embedded agent's real-time performance. However, we can store the rules chosen for replacement with the other rules representing the user behavior in an external hard disk, so that the agent can recall them when needed.

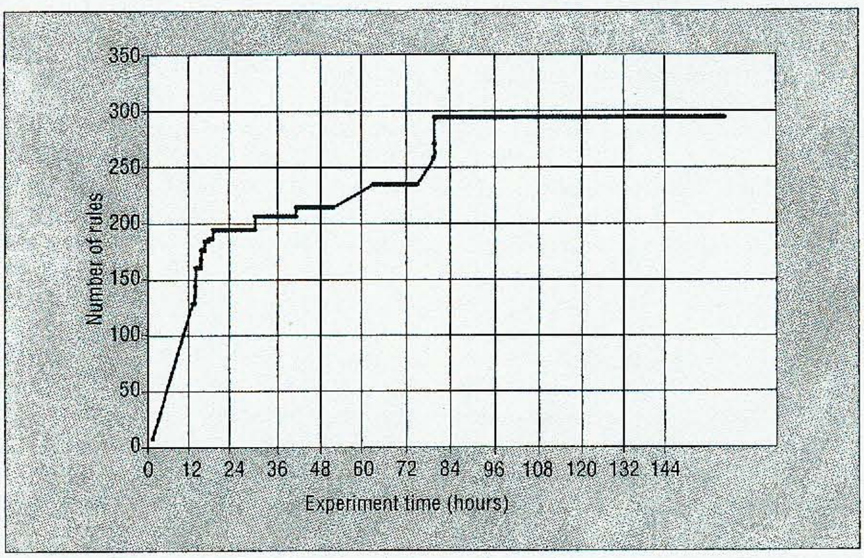


Figure 6. The rules learned plotted against experiment time.

Experimental results

In our experiment, a user occupied the iDorm for five and a half days (132 hours). The system identified the user by his active key button, which operated the active lock (see Figure 5a). In our experiment, the user (shown sleeping in Figure 5b) occupied the iDorm for five and a half days (132 hours). He used the wireless iPAQ to monitor and control the iDorm environment whenever he was dissatisfied with the environment's current state. We recorded a history of the user decisions in a journal. One of our axioms is that "the user is king," by which we mean that an agent always executes the user's instruction immediately, to achieve the responsive property implied in the ambient-intelligence vision, unless safety is compromised. Figure 5c shows this; whenever changes to controls occurred, the iDorm embedded agent received the request, generated a new rule or adjusted a previously learned rule, and allowed the action. We wrote a small parsing tool to convert the text file containing the fuzzy-rule sets into a human-readable format.

At the end of the experiment, we examined the rules in two ways. First, we compared the human-readable rules with the user's journal entries to ensure that the agent had successfully learned the behaviors the user was intending. Second, we compared the number of rules learned over time. We measured the embedded agent's success by monitoring how well it matched the environment to the user's demands. If it did this well, the user intervened less, which resulted in less rule generation over time. If it did this poorly, the user intervened more, which resulted in more rule generation over time. A logging program took a reading of the number of rules learned by the ISL every five minutes for the experiment's duration; it stored each count along with a time stamp.

Figure 6 summarizes the experiment's results. Several sections of the graph are worth noting. In the first 24 hours, starting with an initial rule base of zero, the system learned many rules in a comparatively short period of time. In fact, in the experiment's first nine hours, the agent learned 128 rules—nearly half the total number of rules learned by the end of the experiment. These results are consistent with the agent learning and making incorrect decisions for the user in the initial stage. However, at the end of the first 24 hours, the agent's learning rate (rules/time) dropped drastically. The agent's reactions required less correction by the user because it made useful decisions about the environment based on the user's requirements. This trend of fewer rules learned over time is consistent across the whole experiment. Hence, the level of comfort the user experienced (in relation to the environment state) was high enough for him not to make an environmental change and consequently alter the learning rate.

The second section of the graph (from 60 to 72 hours) shows a sharp increase in the agent's learning rate. This is explained by the user introducing novel activity into his repertoire of behaviors. It shows that our system, which operates in a lifelong learning mode, adapts to user needs.

The third section (from 72 to 132 hours) shows that in the experiment's last two days, the agent didn't generate any new rules. The user didn't intervene with the system because he was generally satisfied with the agent's actions.

The agent learned the 280 rules needed to capture this user's behavior over the 132-hour experiment, which demonstrates that our system can learn effectively using the ISL, and it doesn't need to learn the complete rule base (potentially 62,208 rules in the case of the iDorm). Also, over the experimental period, the agent made a significant reduction in the user's need to intervene. Figure 6 shows that the agent had to learn fewer new rules about the user as the experiment progressed. Because this was one of our criteria for measuring the agent's success, the evidence of the continual reduction in the learning rate leads us to conclude that the agent managed to pick out the user's pertinent behavior over time.

In our previous work, we experimented with different room users.[3] We found that the role of the Experience Bank was important in reducing the time the ISL takes to learn the user's behavior and achieve user satisfaction. This is because it starts learning the user's behavior from the best-matching behavior previously recorded rather than starting from scratch.

Our results suggest that over the experimental period, the embedded agent significantly reduced the user's need to intervene. This not only reduces the complexity of use but can bring significant cost and effort savings

The Authors

Hani Hagras is a senior lecturer in the Department of Computer Science at the University of Essex. His research interests include computational intelligence, notably fuzzy logic, neural networks, and evolutionary computation; ambient intelligence; pervasive computing; and intelligent buildings. He is a member of the IEEE and of the executive team of the IEE's Robotics and Mechatronics Professional Network. He received his PhD in robotics from the University of Essex. Contact him at the Dept. of Computer Science, Univ. of Essex, Wivenhoe Park, Colchester CO4 3SQ, UK; hani@essex.ac.uk.

Victor Callaghan is a senior lecturer in the Department of Computer Science at the University of Essex. He is also head of the Brooker Laboratory for Intelligent Embedded Systems and director of the Inhabited Intelligent Environments Group at Essex. His research addresses methods of integrating intelligence into embedded computers based on soft-computing techniques, notably fuzzy logic, neural networks, and genetic algorithms, aimed at pervasive computing and ambient-intelligence applications. He is a chartered engineer and a member of the BCS and IEE. He received his PhD in computing from the University of Sheffield. Contact him at the Dept. of Computer Science, Univ. of Essex, Wivenhoe Park, Colchester CO4 3SQ, UK; vic@essex.ac.uk.

Martin Colley is a lecturer in the Department of Computer Science at the University of Essex. His primary research interests involve developing parallel and distributed control architectures for robotics and intelligent machines, in particular for autonomous vehicles, and investigating how these architectures could benefit from interaction with their environment. He received his PhD in parallel and distributed computing systems from the University of Essex. Contact him at the Dept. of Computer Science, Univ. of Essex, Wivenhoe Park, Colchester CO4 3SQ, UK; martin@essex.ac.uk.

Graham Clarke is a computer officer in the Department of Computer Science at the University of Essex. His major research interests are in AI, embedded agents for intelligent inhabited environments, and object relations psychoanalysis. He received his MSc in computing applications from the University of North London and his PhD in psychoanalytic studies from the University of Essex. Contact him at the Dept. of Computer Science, Univ. of Essex, Wivenhoe Park, Colchester CO4 3SQ, UK; graham@essex.ac.uk.

Anthony Pounds-Cornish is a senior research assistant with the European Union-funded SOCIAL (Self-Organizing Societies of Connectionist Intelligent Agents Capable of Learning) Project. His main research interests are in computational intelligence, including online evolving of spiking neural networks using genetic algorithms, stigmergic communication, and embedded computing. He received his BSc in computer science from the University of Essex. Contact him at the Dept. of Computer Science, Univ. of Essex, Wivenhoe Park, Colchester CO4 3SQ, UK; apound@essex.ac.uk..

Hakan Duman is a senior research officer with the European Union-funded eGadgets Project. His main research interests are in computational intelligence, including fuzzy logic, neural networks, ubiquitous computing, and embedded agents. He received his BSc in computer science from the University for Applied Sciences in Regensburg, Germany. Contact him at the Dept. of Computer Science, Univ. of Essex, Wivenhoe Park, Colchester CO4 3SQ, UK; hduman@essex.ac.uk.

over the evolving lifetime of products by avoiding expensive programming (and reprogramming). In addition, it empowers ordinary users by letting them use collections of computer-based artifacts to design novel systems to suit their personal tastes without needing to understand the technical complexities or program the systems. The learning curve's shape in Figure 6 suggests that the agent moved increasingly closer to the user's environmental preference even though this preference was never static.

Our experiment also suggests that the agent requires surprisingly few rules to autonomously create a comfortable environment, with diminishing need for user correction. This is important new information: prior to this work, it was unknown whether a tractable rule set would emerge and whether embedded architectures with only megabytes of memory would be able to host agents for such ubiquitous-computing environments.

Our future experimental program includes plans for multiuser and multiroom habitation experiments. We also plan wider deployment of embedded agents (such as personal agents inside wearable technology) and experiments on differing agent granularities. We are building a multiroom version of the iDorm, called iDorm-2, as a preliminary step toward constructing a fully functional apartment (iFlat), which will house visiting researchers and act as a unique ubiquitous-computing test bed. We are currently designing the iFlat for such experimentation from the ground up.

Acknowledgments

We acknowledge the funding support of the European Union's Information Society Technologies Disappearing Computer program and the Korean–United Kingdom Scientific Fund program.

References

1. K. Ducatel et al., *Scenarios for Ambient Intelligence in 2010*, tech. report, Information Society Technologies Advisory Group (ISTAG), Inst. of Prospective Technological Studies (IPTS), 2001.
2. V. Callaghan et al., "Embedding Intelligence: Research Issues for Ubiquitous Computing," *Proc. 1st Equator Interdisciplinary Research Challenge Workshop Ubiquitous Computing in Environments*, Univ. of Nottingham Press, 2001, pp. 110–130.
3. H. Hagras et al., "Online Learning and Adaptation for Intelligent Embedded Agents Operating in Domestic Environments," *Fusion of Soft Computing and Hard Computing for Autonomous Robotic Systems*, C. Zhou, D. Maravall, and D. Ruan, eds., *Physica-Verlag*, vol. 116, Nov. 2002, pp. 293–323.
4. A. Bonarini, "Comparing Reinforcement Learning Algorithms Applied to Crisp and Fuzzy Learning Classifier Systems," *Proc. Genetic and Evolutionary Computation Conf.*, Morgan Kaufmann, 1999, pp. 52–60.
5. H. Hagras et al., "A Hierarchical Fuzzy Genetic Multi-Agent Architecture for Intelligent Buildings Learning, Adaptation and Control," *Int'l J. Information Sciences*, vol. 150, nos. 1–2, Mar. 2003, pp. 33–54.
6. A. Holmes, H. Duman, and A. Pounds-Cornish, "The iDorm: Gateway to Heterogeneous Networking Environments," *Proc. Int'l Test and Evaluation Association (ITEA) Workshop Virtual Home Environments*, ITEA Press, 2002, pp. 30–37.

資訊管理論文評述（二）

Please read the attached paper, entitled "Transaction-driven Personalization: The Moderating Effects of Personality Traits," and answer the following questions in English.

1. Summarize the theme of the paper and describe its major contributions. (40%)
2. Outline the current research method and evaluate the strengths and weaknesses of the current method. (40%)
3. Propose another theory that may also be related to the increasing personalization on the Web. (20%)

13. Transaction-Driven Personalization: The Moderating Effects of Personality Traits

Shuk Ying Ho
The University of Melbourne
suho@unimelb.edu.au

Kar Yan Tam
The Hong Kong University
of Science and Technology
kytam@ust.hk

Michael J. Davern
The University of Melbourne
m.davern@unimelb.edu.au

Abstract

With transaction-driven personalization engines online merchants can use knowledge gained from an individual customer's past transactions to match web content to the customer's individual interests and preferences. Prior research in this area has focused on how to maximize knowledge mined from transaction logs to generate recommendations which are highly similar to the individual's past preferences. However, it remains an empirical question as to whether a recommendation closely matched with previous transactions is most likely to influence choice behavior? In this study, we postulate that a recommendation closely matched with previous transactions may not be the most efficient in biasing an individual. In the consideration and choice process, an individual's personality traits play a pivotal role in moderating the effect of personalized content. Drawing on prior work in marketing, we examine two key personality traits, need for cognition and variety seeking, and explore their effects on choice behavior in the context of transaction-driven personalization. Research hypotheses are tested using 2,294 pre-selected subjects in an online field experiment based on a ring tone download website. Our findings establish that personality traits of an individual moderate content consideration and choice. Theoretical and practical implications of the findings are discussed.

Keywords: Web personalization, personality traits, need for cognition, variety seeking, user behavior

Introduction

As competition intensifies, online retailers have to continuously provide better offerings and unique web experiences to their customers. To remain competitive, online retailers have sought differentiating strategies to attract and retain users (Bakos 1991). One such strategy has been to incorporate a personalization agent to model and adapt to web visitors' objectives and facilitate their navigation/buying process (Albert 2004). Different types of personalization agents are available. For instance, transaction-driven personalization systems customize web layout and content depending on users' preferences as inferred from transaction logs (Reiter and Rubin 1999). In contrast, real time personalization systems adapt the webpages presented based on observations of user click streams in a given session in an attempt to be sensitive to context of interaction (Mulvenna et al. 2000). Alternatively, the personalization systems studied in prior research can be distinguished based on whether they seek to personalize the content (e.g., a set of product offerings) or peripheral cues (e.g, the manner in which the webpage is presented). Figure 1 provides a taxonomy of personalization

2

systems and prior research based on data collection method (clickstream or transaction log) and the nature of the personalization (content or peripheral). In this study we focus on transaction-driven personalization of content (i.e., the upper left quadrant of Figure 1). While prior studies have been conducted in this area, the issue of the effect of personality traits is underinvestigated, despite the importance of personality in purchasing behavior evidenced in the marketing literature (e.g. Chintagunta 1999; Homburg and Giering 2001; van Trijp et al. 1996).

Data Collection by Personalization Agents	Content	Peripheral
Transactions	Customization of web content based on the analysis of user's transactions and search queries in the previous logon sessions. *Examples:* Fan et al. (2000); Smyth and Cotter (2000); Vinaja et al. (2000)	Customization of web layout based on the analysis of user's transactions and search queries in the previous logon sessions. *Examples:* Billsus et al. (2002); Moon (2002)
Click Streams	Customization of web content based on the observation of user's click streams and search queries in the current logon session. *Examples:* Burns and Stollak (1998); Fazlollahi et al. (1996)	Customization of web layout based on the observation of user's click streams and search queries in the current logon session. *Examples:* Andre and Rist (2002)

Nature of Personalization

Figure 1: An Overview of Personalization Systems

Providing an effective personalization service is non-trivial. Although there are many software tools aimed at assisting in personalization (e.g. customer relationship management, data mining, collaborative filtering, and click stream analysis software), Jupiter Research (2003) reported that only 14% of users believe that personalized recommendations on shopping websites lead them to purchase more frequently. In an effort to achieve greater success in personalization, prior research has focused on investigating how to maximize the knowledge from transaction logs to generate a recommendation highly matched with previous transactions. However, it remains an empirical question as to whether *a recommendation closely matched with previous transactions is most likely to influence choice behavior*.

Prior research evidences that personality traits significantly influence choice behavior, (Benbasat and Dexter 1982; Hunt et al. 1989; Nutt 1993; Lu et al. 2001). Recent research in the marketing literature shows that the effect of personalization on consumer behavior may be influenced by personality traits (Andre and Rist 2002; Moon 2002). The context of this prior research was personalization of peripheral cues (e.g. presentation styles and animation), rather than personalization of content (offers and recommendations), and consequently provide insufficient guidance to online retailers seeking to battle fierce price competition through the personalization of offers and recommendations. Nonetheless the prior work suggests that personality traits may prove to be an important moderator of the effect of personalized content (offers and recommendations) on choice behavior. Recommendations that consider both previous transactions and personality traits may thus be the most likely to

3

influence choice behavior. Consequently, this study is aimed at examining personality and personalization of content rather than peripheral cues. In particular, we consider the effects of two of the most widely studied personality traits examined in the prior literature: need for cognition and variety seeking. More generally, this study seeks to address the question: *what is the effect of personality traits on choice behavior in response to personalized offers generated from a transaction-driven agent?*

From a theoretical perspective, the results of this study contribute to the development of a more comprehensive theory of the effects of personalization on consumer choice behavior. From a practical perspective, it offers the potential to improve the success online retailers achieve with personalization agents, providing redress against the disappointing performance reported by Jupiter Research. Pragmatically, we will also provide preliminary evidence demonstrating how relevant personality traits can be derived automatically from observation of behavior, without the necessity of directly questioning the user.

The structure of the paper is as follows. Section 2 presents the theoretical background and the hypotheses. This is followed by an online study in Sections 3 and 4. Section 5 discusses the findings and Section 6 concludes the paper.

Theoretical Development and Hypotheses

Choice Behavior: Consideration Set Theory

We employ Consideration Set Theory (Howard 1989; Howard and Sheth 1969) to structure our examination of the effects of personalization and personality traits on choice behavior (see Finn and Louviere 1990; Gensch 1987; Roberts and Lattin 1997 for more recent analyses). Consideration Set Theory views the choice process as being based on four hierarchical sets: the universal set, the awareness set, the consideration set, and the final choice outcome (see Figure 2).

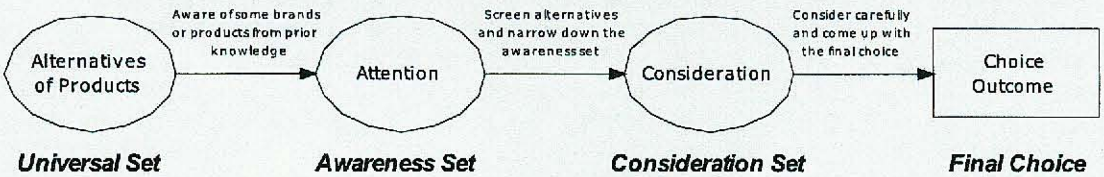


Figure 2: Decision Making Stage

The *universal set* refers to all possible alternatives that could be purchased by users. With limited cognitive resources, users cannot process all alternatives in the universal set. They can only be aware of a few products. A subset of the items forms their *awareness set*. Variables, such as expertise and processing capacity, influence the content of the awareness set. After gathering much information, users identify and select a set of acceptable alternatives from what is available. Their *consideration set* is then formed (Punj and Brookes 2001). This set consists of alternatives that users would consider carefully, and limit their purchases to these alternatives (Roberts and Lattin 1997). The average size of a consideration set may vary (Roberts 1989) reflecting the search costs involved in identifying and selecting the alternatives for inclusion in the consideration set. After eliminating alternatives from their consideration set, users arrive at their *choice outcome* subsequently. That is, they evaluate each alternative in the consideration set carefully to select the best alternative(s).

4

In this study we focus our attention on choice behavior which we define as active evaluation of the alternatives in the consideration selection of a final choice. In the context of the typical transaction-driven personalization system, transaction log data is analyzed to make recommendations to users in an attempt to ensure their consideration set includes items that are likely to be chosen for purchase. Our interest then is on how personality traits influence the active evaluation of alternatives recommended into the consideration set and ultimately the final choice made. Consequently our primary dependent variables of interest are content consideration (whether or not a user actively considers a recommended alternative), and the choice outcome (whether or not a user chooses a recommended alternative).

Personalization: Preference Matching

In characterizing transaction-driven personalization systems that seek to manipulate content (recommendation or offers) the goal is to maximize "preference matching". In this context, preference matching refers to the similarity of personalized content with previous transactions. It is the extent to which recommendations are consistent with previous transactions by an individual or by other "like-minded" people (e.g., those who exhibit substantially similar transaction histories). Prior research has shown that consumers give greater consideration, and ultimately are more likely to choose highly preference-matched personalized content (although the Jupiter Research finding suggests the practical value of this is less significant than desired). Thus as a confirmation of prior work we hypothesize:

<u>Hypothesis H1a:</u> Preference matching will positively influence consideration set size (the number of alternatives actively considered).

<u>Hypothesis H1b:</u> Preference matching will positively influence final choice outcomes.

Our concern here is not simply with differential effects of high versus low preference matching in personalization – an issue which has been subject to substantial prior research. Rather our concern is with how the effect of preference matching is moderated by the personality traits of need for cognition and variety seeking.

Personality Traits and Personalization

We consider two of the most widely studied personality traits in choice behavior: need for cognition and variety seeking. Need for cognition (NFC) is defined as an individual's "tendency to engage in and enjoy thinking" (Cacioppo and Petty 1982, p. 119). Even intuitively it seems obvious that NFC may influence choice behavior as it reflects an individual's propensity to engage in effortful comparison of alternatives within a consideration set, and thus ultimately choice. Likewise variety seeking, defined as a consumer motivation to look for or accept novelty (McAlister and Pessemier 1982) intuitively influences choice behavior as it reflects an individual's desire to look for different items in consideration set, and the propensity to choose something different from what has previously been chosen. We now discuss in detail the effects of each of these personality traits on the Preference Matching-Choice Behavior relationship.

2.3.1 Need for Cognition and Personalization

In general, individuals with high NFC are intrinsically motivated to engage in cognitive endeavors (Cacioppo and Petty 1982; Haugtvedt et al. 1992; Larsen et al. 2004). They search for more information when making decisions (Verplanken 1993); and engage in more effortful processing of persuasive messages (Haugtvedt et al. 1992; Roehm and Sternthal

2001). Compared with individuals with low NFC, they are more open-minded (Cacioppo and Petty 1982), and enjoy more effortful cognitive tasks and develop more complex causal explanations for the behavior of others (Fletcher et al. 1986). Also, they hold attitudes that are more persistent over time and resistant to persuasion attempts (Haugtvedt et al. 1992). Assuming the arguments in a message are valid, high-NFC individuals tend to make greater and longer-lasting attitude changes (Haugtvedt et al. 1992). Thus, we anticipate that high-NFC individuals will actively consider a larger number of alternatives than low-NFC individuals (i.e. they will have a large consideration set). More formally, we hypothesize:

Hypothesis H2a: High NFC individuals will have a larger consideration set size (i.e. actively consider more alternatives) than low NFC individuals.[6]

Since prior work suggests that NFC individuals are more discerning and careful (Haugtvedt et al. 1992; Petty and Cacioppo 1986). This suggests they will be more sensitive to the level of preference matching of personalized content in their choice behavior than low-NFC individuals. Consequently we hypothesize:

Hypothesis H2b: Preference matching will have a greater effect on consideration set size (i.e. the number of alternatives actively considered) for high-NFC individuals as opposed to low-NFC individuals.

The final choice of an individual is heavily influenced by the merits of the alternatives in the consideration set. As high-NFC users are likely to more sensitive to the level of preference matching of the personalized content, we expect the choices of high-NFC users to be more polarized between highly-matched and lowly-matched content than those of low-NFC users. Thus, we postulate:

Hypothesis H2c: Preference matching will have a greater effect on choice outcome for high NFC individuals as opposed to low NFC individuals.

Variety Seeking and Personalization

Variety seeking is a consumer motive to accept novelty (for a review, see McAlister and Pessemier 1982). It has acted as a determinant in many marketing models with the outcome variables, including impulse purchase (van Trijp et al. 1996), purchasing timing (Chintagunta 1999), brand loyalty (Homburg and Giering 2001), brand switching (Chintagunta 1999) and customer satisfaction (Homburg and Giering 2001).

According to the theory of Optimal Stimulation Level (Berlyne 1960; Zuckerman 1979), every individual prefers an ideal level of stimulation, and the level is determined by novelty, change, surprise, ambiguity, uncertainty, complexity, and incongruity that are associated with a stimulus. When a stimulus provides stimulation below the optimal level, the individual feels bored and desire an increased stimulation. This leads to greater exploration. Personalized

[6] We choose to categorize individuals as high and low NFC rather than examining NFC as a continuous variable, as matter of pragmatics. In practice, it is likely to be substantially easier for online retailers to profile an individual as low or high NFC based on observed behavior (i.e., without directly measuring NFC through a questionnaire) than it is to develop such an automated continuous measure.

6

content that deviates from past transactions (i.e. lowly-matched content) thus provide a greater degree of stimulation. Variety seekers have a greater desire for this stimulation and thus are likely to exhibit a greater propensity to explore the lowly matched, different, personalized content. On the contrary, variety non-seekers prefer content similar to their past preferences – they desire highly matched content.[7] Thus we hypothesize:

Hypothesis H3a: Variety seekers will consider much lowly-matched personalized content than highly-matched personalized content, whereas variety non-seekers will consider much highly-matched personalized content than lowly-matched content. (The interaction of variety seeking and preference matching will have negative effect on consideration set size).

Variety seekers prefer stimulation and excitement. Thus, they desire to switch to something different to raise stimulation and excitement. Therefore, we anticipate that they are more likely to make a choice in the context of lowly-matched content, than variety non-seekers who prefer the familiarity of highly-matched content.

Hypothesis H3b: Variety seekers are more likely to choose lowly-matched personalized content than variety non-seekers who prefer highly-matched personalized content as their final choice.

Figure 3 summarizes the research model.

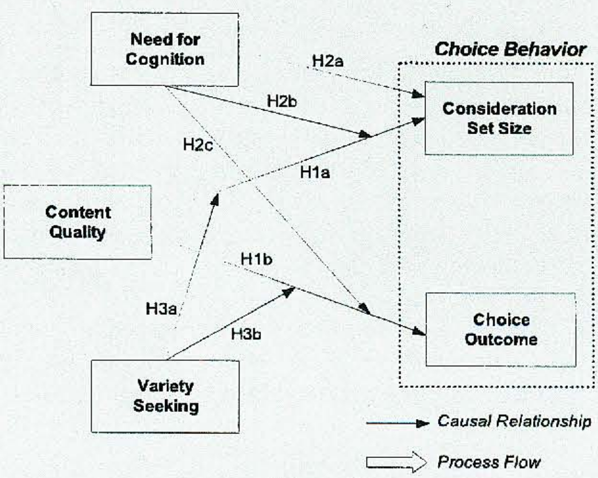


Figure 3: Research Model

[7] Note variety seeking is only relevant to choice behavior in the context of the preference matching of the consideration set, consequently we do not hypothesize any main effect for variety seeking.

7

Research Method

An online field experiment was conducted to test the hypothesized relationships. The experimental design comprised three independent variables and two dependent variables. The independent variables were preference matching which was manipulated using a transaction-driven personalization agent, need for cognition and variety seeking. The dependent variables were consideration set size (the number of alternative actively considered from personalized content) and the final choice outcome. The context for our experiment was shopping for ring tones for mobile phones. Since ring tones were hedonic products, users would be subject to situational and contextual factors which lend themselves to manipulation by a personalization agent.

Participants

We cooperated with a major mobile data services content provider, which offered ring tones, games and other content to mobile users in Hong Kong to conduct the studies. Emails were sent to 40,000 registered users of the content provider to recruit subjects. Subjects were randomly streamed into the two treatments (high vs. low preference matching). Demographic checks were carried out to assess the effectiveness of the random assignment of subjects to treatment conditions. We received 3,267 responses in total. Among them, 2,294 (=70.22%) completed the entire experiment. As a token of appreciation, these subjects were given a free ring tone for their mobile handsets and a chance to join a lucky draw for a special gift. To provide further incentive for subjects to provide accurate responses to questionnaires on personality traits, subjects who successfully completed the experiment received a free individualized personality report. The study lasted for six weeks from mid-November 2003 to early January 2004.

Experimental Procedures

The study was divided into three parts. First, the subjects were asked to fill in a questionnaire about their demographic information and complete a personality test. Second, we asked subjects to indicate their preferences for rhythms and musicians. They chose and ranked their three favorite musicians from a list of 18. Information from the Hong Kong Music Billboard allowed us to determine which songs were popular from the musicians' latest albums. Finally, all participants entered a webpage with 12 ring tone alternatives. The subjects could choose to download only one ring tone free of charge. After they confirmed to download a ring tone, the selected ring tone was sent to the participant's mobile handset via a short message service. A pretest with 56 subjects was conducted to validate the instruments employed and to test the ring tone download system performance. Subjects were able to complete the whole process in 25 minutes. All subjects confirmed that the selection task was smooth.

Manipulation of Preference Matching

To determine the list of ring tones for the experiment, we studied the transaction log provided by the mobile service provider to obtain a list of musicians. This log contained the actual ring tone purchases of 7,858 distinct users. There were 66,795 transactions dated from August 2002 to November 2003. These users downloaded ring tones from 175 distinct musicians. We chose the top 18, who accounted for 32,869 (49.21%) out of 66,795 download transactions. These transactions were conducted by 6,474 (82.39%) distinct users. We then formed a pool of 72 ring tones from 18 musicians (4 ring tones per musician). Most musicians had two ring tones with fast rhythms, and the other two with slow rhythms. The ring tones in the same

8

rhythm category were assigned a recommendation priority based on the information obtained from the Hong Kong Music Billboard.

There were two levels of manipulation of preference matching: highly-matched and lowly-matched. Under the highly-matched condition, the list of recommended ring tones matched with the subject's previous transactions. If the subjects downloaded ring tones from many different musicians, or from the musicians who were not included on the 18-musician list, then we could not determine the best recommendations. In this case, we relied on their responses in the questionnaire (i.e. Step 2 in the experiment) to generate highly-matched recommendations. Under the lowly-matched condition, the list of ring tones was randomly extracted from the ring tone pool. All subjects received a list of 12 ring tones from which to choose.

User Interface Design

All 12 ring tones were presented to a subject on a single page. Under a 1024 × 768 resolution, no page scrolling was needed for viewing the ring tones. Ring tones were presented on the right of the page under Personalized Recommendations, whereas ring tones on the left were Other Offers. There was no other link on the page. The title of the song associated with each ring tone was used as a label and the musician of the song was also indicated. All titles and musicians were labeled in Chinese. For each ring tone, there were two buttons. One labeled as "Trial Listen" and the other as "Download". When a subject pressed on the "Trial Listen" button, an audio file of the selected ring tone was streamed to the subject's client machine. There was no restriction on the number of trial listenings. All mouse clicks of a subject were logged. When a subject pressed the "Download" button of a ring tone, the selected ring tone was sent to the subject's mobile phone.

Measurements

(a) Dependent Variables

Consideration set size was measured by the number of personalized ring tones sampled by the subjects during the field experiment. At the end of field study, all subjects had to choose a ring tone either from the personalized list or from the other list. We recorded their final choice (1=download from personalized list; 0=otherwise).

(b) Need for Cognition

Need for cognition was measured using an instrument taken from Cacioppo et al. (1984). Participants indicated the extent to which each statement was the characteristic of themselves on a 7-point scale anchored by 1 (strongly disagree) and 7 (strongly agree). This scale has been widely used in recent research (e.g. Tormala and Petty 2004; Wheeler et al. 2005). The questions are shown in Appendix I.

(c) Variety Seeking

Variety seeking was measured using an instrument taken from Baumgartner and Steenkamp (1996). This scale has been widely used in past research (e.g. Homburg and Giering 2001; Zhang and Krishnamurthi 2004). All questions were anchored on a 7-point scale, ranging from strongly disagree (1) to strongly agree (7). The questions are shown in Appendix I.

Data Analysis

Validity and Reliability

To assess convergent and discriminant validity, we factor-analyzed the items for the two personality trait instruments. Table 1 shows the results from factor analysis and reveals the item loadings were consistent with two factors and two distinct theoretical constructs. The Cronbach alpha values for need for cognition and variety seeking were 0.88 and 0.76 respectively. Consequently we concluded the instruments were statistically valid and reliable.

Table 1: Factor Analysis

	Comp1	Comp2
VS1	.056	.796
VS2	.028	.814
VS3	-.046	.841
NFC1	.828	.027
NFC2	.881	.018
NFC3	.735	.075
NFC4	.812	-.038
NFC5	.836	-.019

Hypotheses Testing

Table 2 and Table 3 summarize the descriptive statistics for the dependent variables.

Table 2: Descriptive Statistics on the Number of Trial Listens of Personalized Recommendations

	High NFC			Low NFC		
	N	Mean	S.D.	N	Mean	S.D.
Low Preference Matching	533	2.54	2.34	568	1.70	1.70
High Preference Matching	615	3.47	2.74	578	1.92	1.86
Total	1148	3.03	2.60	1146	1.81	1.78

	Variety Non-Seeker			Variety Seeker		
	N	Mean	S.D.	N	Mean	S.D.
Low Preference Matching	560	2.16	2.04	541	2.05	2.11
High Preference Matching	588	2.80	2.59	605	2.66	2.37
Total	1148	2.48	2.35	1146	2.37	2.27

Table 3. Descriptive Statistics on the Number of Downloads of Personalized Recommendations

	Need for Cognition (NFC)				Variety Seeking (VS)			
	High		Low		High		Low	
	N	Mean	N	Mean	N	Mean	N	Mean
Low Preference Matching	533	40.33%	568	47.59%	541	49.91%	560	38.57%
High Preference Matching	615	69.18%	578	68.26%	605	68.78%	588	68.70%
Total	1148	55.79%	1146	58.02%	1146	59.87%	1148	54.00%

We performed a median split (median=4.08, range=-0.15 to 9.07) to classify the 2,294 participants into high-NFC individuals or low-NFC individuals. Similarly, we performed another median split (median=4.62, range=-1.31 to 10.93) to classify the participants into variety seekers or variety non-seekers.

10

(a) Consideration Set Size

We conducted a three-way ANOVA with variety seeking, NFC and preference matching as the explanatory variables for consideration set size. The results confirmed prior research, supporting H1a ($F(2288, 5)=39.28$, $p<0.01$). A significant main effect was also evidenced for NFC supporting H2a ($F(2288, 5)=169.44$, $p<0.01$). A significant interaction effect between NFC and preference matching, was evidenced supporting H2b ($F(2288,5)=15.33$, $p<0.01$). Individuals with high NFC sampled more ring tones (mean=3.03) than individuals with low NFC (mean=1.81) (see Table 2a). As the similarity of personalized recommendations with previous transactions degraded, high-NFC individuals sampled fewer ring tones from a poor list (mean samplings dropping from 3.47 to 2.54). However, the number of samples by low-NFC individuals remained approximately constant (dropping from 1.92 to 1.70). Preference matching clearly exerts a larger influence on high-NFC individuals.

We found that the interaction effect between variety seeking and preference matching was not significant, evidence no support for H3a ($F(2288,5)=0.49$, $p>0.1$)[8] (see Table 2a for details of the mean number of ring tones sampled).

(b) Choice Outcome

Choice outcome was coded as a binary variable (1=download from personalized list; 0=otherwise). A logistic regression was conducted to test the effects from variety seeking, NFC and preference matching on choice outcome. The results confirm the prior research evidencing a positive effect of preference matching, supporting H1b ($\chi2(1)=53.24$, $p<0.01$). A significant interaction effect was found between NFC and preference matching, supporting H2c ($\chi2(1)=3.88$, $p<0.05$). As shown in Table 2b, high-NFC individuals chose more highly-matched personalized offers (69%) and fewer lowly-matched personalized offers (40%). However, low-NFC individuals were less sensitive to preference matching. They took 68% of highly-matched offers and 48% of lowly-matched offers.

Similarly, there was a significant interaction effect between variety seeking and preference matching, supporting H3b ($\chi2(1)=7.07$, $p<0.01$). With lower preference matching, variety seekers were still willing to chose a ring tone from the personalized list (mean=50%), whereas variety non-seekers tended to reject those recommendations (mean=39%).

Discussion and Implications

Our work enhances the understanding of the interaction effects between preference matching and an individual's personality traits, need for cognition (NFC) and variety seeking, on content consideration and decision outcome. Our findings reveal that, in general, individuals take more highly-matched content than lowly-matched content. However, individuals with low NFC or variety seekers will also consider and choose lowly-matched content. Personal traits appear to play a pivotal role in influencing choice behavior in interaction with preference matching. Generating recommendations closely matched with previous

---

[8] Prior to conducting the experiment, we estimated that assuming a medium effect size, a sample size of around 500 could achieve a power of the statistical tests higher than the recommended value of 0.8 (Cohen, 1988). Our sample size was 2,294. Thus, there was no evidence that non-significant result in H2a was due to a small sample size.

11

transactions may not be the best approach. This study contributes to the existing literature in several ways.

First, individuals who have little motivation to exert cognitive effort to evaluate the merits of alternatives tend to rely on recommendations suggested by the personalization agent. These low-NFC individuals sampled fewer personalized ring tones, but also chose more personalized recommendations. They are less able to distinguish highly-matched from lowly-matched recommendations. If the cost of generating a personalized recommendation is high, a firm can consider providing fewer personalized options of average preference matching to low-NFC individuals.

Second, since high-NFC individuals are willing to process more information, more personalized content should be sent to these individuals. Otherwise, they might seek information from other sources and lose interest in the personalized list. Also, since they are more able to distinguish highly-matched recommendations from lowly-matched recommendations, unless the firm is confident that a recommended item matches the interests of a high NFC user, it should not offer the item to the person. That is, by knowing users' NFC, a firm can manipulate the merits of arguments for different users to maximize the pervasiveness of the content.

Third, since variety seekers have a stronger desire to look for stimulation, they are willing to explore personalized recommendations which deviate from their past revealed preferences. However, variety non-seekers prefer the familiarity of content closely-matched with their previous transactions. In addition to understanding the transaction history of individuals, it becomes equally important for online merchants to understand their traits. For variety non-seekers, providing recommendations matching their past preference is the most direct and simple way to influence their decisions. For variety seekers, it may be more effective have the personalization agent provide a list of offers which may not be closely aligned with a user's past preferences. With a higher level of novelty and surprise, variety seekers are willing to take alternatives from this list.

Fourth, the effects of different personality traits at different stages of information processing are different. According to our findings, the effect from variety seeking is not significant at the consideration stage, but is significant at the choice stage. This sheds light on the effectiveness of web personalization and highlights the pivotal role it plays in different information processing stages (i.e. consideration and choice) of a user.

Last, as suggested by Benbasat and Zmud (1999, p.5), the implications of empirical IS research should be "implementable". Thus, a major challenge for web service providers is to understand or derive the personality scores of their customers and prospects from observable web behavior. Unless they are known, strategies that leverage on this finding cannot be realized. The challenge becomes identifying observable web behavior that correlates with NFC and variety seeking scores. To probe further into the topic, we have derived click traits from the web server log and correlated this information with the participants' NFC and variety seeking scores. For example, correlations between NFC and the amount of repetitive listenings on the same ring tone and between NFC and browsing session time are 0.34 and 0.29, respectively. Correlation between variety seeking and the number of distinct sampled ring tones is 0.41. These findings, though preliminary, provide evidence that observable web behavior may form the basis of a scoring index for personality traits and it can serve as an

12

input for personalization systems. Consequently we believe our results can be incorporated into real world systems, beyond our experimental context.

Conclusion and Limitations

We studied the effects of transaction-driven personalization agents on individuals with different personality traits in terms of the user's content consideration and choice outcome. Generally, high-NFC individuals are more willing to explore personalized content, resulting in a great deal of information exploratory activities. And variety seekers are willing to explore the personalized content not totally aligned with their past preference. Thus, apart from data mining of past preferences, personalization agents should also adapt to individuals of different personality traits. Our study sheds light on the significance of personalization to online merchants in offering unique experiences to its users. Increasing investment in mining transaction logs may not generate the best outcome.

There are a variety of ways to extend our work. First, our subjects were invited to an artificial website to ensure that this was their first time to experience this personalized site. And they could experience this website only once. What would happen if the visitors returned to the site? Do variety seekers behave differently in the long run? A longitudinal field study might contribute much to this area of research. Second, our study focused on low-involvement hedonic products (i.e. ring tone). If the involvement of decision is high, would the individuals rely on personalization agents? Would the individuals take the recommendations deviated from their past transactions? With more resources, another field study using another product types can contribute to our knowledge about the effectiveness of personalization agents.

References

Albert, T.C., Goes, P.B., and Gupta, A. "GIST: A Model for Design and Management of Content and Interactivity of Customer-Centric Websites," *MIS Quarterly*, (28:2), 2004, 161-182.

André, E., and Rist, T. "From Adaptive Hypertext to Personalized Web Companions," *Communications of the ACM*, (45:5), 2002, 43-46.

Anonymous (2003), "Jupiter Research Reports That Website 'Personalization' Does Not Always Provide Positive Results", Retrieved May 1, 2005, from http://www.jupitermedia.com/corporate/releases/03.10.14-newjupresearch.html.

Areni, C.S., Ferrell, M.E., and Wilcox, J.B. "The Persuasive Impact of Reported Group Opinions on Individuals Low vs. High in Need for Cognition: Rationalization vs. Biased Elaboration?" *Psychology and Marketing*, (17:10), 2000, 855-875.

Bakos, J.Y. "A Strategic Analysis of Electronic Marketplaces," *MIS Quarterly*, (15:3), 1991, 295-310.

Baumgartner, H. and Steenkamp, E.M. "Exploratory Consumer Buying Behavior: Conceptualization and Measurement," *International Journal of Research in Marketing*, (13), 1996, 121-137.

Benbasat, I. and Dexter, A.S. Individual Differences in the Use of Decision Support Aids, *Journal of Accounting Research*, (20:1), 1982, 1-11.

Benbasat, I. and Zmud, R.W. Empirical Research in Information Systems: The Practice of Relevance, *MIS Quarterly*, (23:1), 1999, 3-16.

Berlyne, D. E. *Conflict, Arousal, and Curiosity*, McGraw-Hill, New York, NY. 1960.

Billsus, D., Brunk, C.A., Evans, C., Gladish, B., and Pazzani, M. "Adaptive Interfaces for Ubiquitous Web Access," *Communications of the ACM*, (45:5), 2002, 34-38

Burns, A.T. and Stollak, M.J. "Development of a Web-Based Intelligent Agent for the Fashion Selection and Purchasing Process via Electronic Commerce," *The Proceedings of Americas Conference on Information Systems*, 1998, 140-142.

Cacioppo, J.T. and Petty, R.E. "The Need for Cognition," *Journal of Personality and Social Psychology*, (42:1), 1982, 116-131.

Cacioppo, J.T., Petty, R.E., and Kao, C.F. "The Efficient Assessment of Need for Cognition," *Journal of Personality Assessment*, (48), 1984, 306-307.

Chintagunta, P.K. "Variety Seeking, Purchase Timing, and The 'Lightning Bolt' Brand Choice Model," *Management Science*, (45:4), 1999, 486-498.

Cohen, J. *Statistical Power Analysis for the Behavioral Sciences*, 2nd Ed. Lawrence Erlbaum, Hillsdale, NJ. 1988.

Fan, W., Gordon, M.D., and Pathak, P. "Personalization of Search Engine Services for Effective Retrieval and Knowledge Management," *The Proceedings of International Conference on Information Systems*, 2000, 20-34.

Fazlollahi, B., Parikh, M., and Vahidov, R. "Intelligent Guidance in Adaptive Decision Support Systems," *The Proceedings of Americas Conference on Information Systems*, August 16 - 18: Phoenix, Arizona. 1996.

Finn, A., and Louviere, J. "Shopping-Center Patronage Models: Fashioning A Consideration Set Segmentation Solution," *Journal of Business Research*, (21), 1990, 259-275.

Fletcher, G.J.O., Danilovis, P., Fernandez, G., Peterson, D., and Reeder, G.D., "Attributional Complexity: An Individual Difference Measure," *Journal of Personality and Social Psychology*, (51:4), 1986, 875–884.

Gensch, D.H. "A Two-Stage Disaggregate Attribute Choice Model," *Marketing Science*, (7:3), 1987, 299-310.

Haugtvedt, C., Petty, R. and Cacioppo, R. "Need for Cognition and Advertising: Understanding the Role of Personality Variables in Consumer Behavior," *Journal of Consumer Psychology*, (1:3), 1992, 239–260.

Homburg, C., and Giering, A. "Personal Characteristics as Moderators of the Relationship between Customer Satisfaction and Loyalty – An Empirical Analysis," *Psychology and Marketing*, (18:1), 2001, 43-66.

Howard, J.A., and Sheth, J.N. *The theory of buyer behavior*. New York, NY: Wiley. 1969.

Howard, J.A. *Consumer behavior in marketing strategy*. Englewood Cliffs, NJ: Prentice-Hall. 1989.

Hunt, R.G., Krzystofiak, F.J., Meindl, J.R., and Yousry, A.M. "Cognitive Style and Decision Making," *Organizational Behavior and Human Decision Making*, (44), 1989. 436-453.

Larsen, V., Wright, N.D., Hergert, T.R. "Advertising Montage: Two Theoretical Perspectives," *Psychology and Marketing*, (21:1), 2004. 1-15.

Lu, H.P., Yu, H.J., and Lu, S.S.K. "The Effects of Cognitive Style and Model Type on DSS Acceptance: An Empirical Study," *European Journal of Operational Research*, (131:3), 2001, 649-663.

Moon, Y. "Personalization and Personality: Some Effects of Customizing Message Style Based on Consumer Personality," *Journal of Consumer Psychology*, (12:4), 2002, 313-326.

McAlister, L., and Pessemier, E. "Variety Seeking Behavior: An Interdisciplinary Review," *Journal of Consumer Research*, (9:3), 1982. 311-322.

Mulvenna, M.D., Anand, S.S., and Buchner, A.G. "Personalization on the Net Using Web Mining," *Communications of the ACM*, (43:8), 2000, 122-125.

Nutt, P.C. "Evaluating MIS Design Principles," *MIS Quarterly*, (10:2), 1986, 139-156.

14

Petty, R.E. and Cacioppo, J.T. (1986), *Communication and Persuasion: Central and Peripheral Routes to Attitude Change*, New York: Springer Verlag.

Punj, G., and Brookes, R. "Decision Constraints and Consideration Set Formation in Consumer Durables," *Psychology and Marketing*, (18:8), 2001, 843-863.

Reiter, M.K. and Rubin, A.D. "Anonymous Web Transactions with Crowds," *Communications of the ACM*, (42:2), 1999, 32-38.

Roberts, J. "A Grounded Model of Consideration Set Size and Composition," *Advances in Consumer Research*, (16), 1989, 749-757.

Roberts, J., and Lattin, J.M. "Consideration: Review of Research and Prospects for Future Insights," *Journal of Marketing Research*, (34:3), 1997, 406-410.

Roehm, M.L. and Sternthal, B., "The Moderating Effect of Knowledge and Resources on the Persuasive Impact of Analogies," *Journal of Consumer Research*, (28:2), 2001, 257-272.

Smyth, B., and Cotter, P. "A Personalized Television Listings Service," *Communications of the ACM*, (43:8), 2000, 107–111

Tormala, Z.L., and Petty, R.E. "Resistance to Persuasion and Attitude Certainty: THE MODERATING ROLE of Elaboration," *Personality and Social Psychology Bulletin*, (30:11), 2004, 1446

van Trijp, H. C. M., Hoyer, W. D. and Inman, J. J. "Why Switch? Product Category-Level Explanations for True Variety-Seeking Behaviour," *Journal of Marketing Research*, (33:August), 1996, 281-292.

Verplanken, B. Need for Cognition and External Information Search: Responses to Time Pressure during Decision-Making, *Journal of Research in Personality*, (27:September), 1993, 238-252.

Vinaja, B.R., Raisinghani, M.S., and Slinkman, C., "Intelligent Agent Based Decision Making on the Internet: An Empirical Study," *The Proceedings of Americas Conference on Information Systems*, 2000, 289-294.

Wheeler, S.C., Petty, R.E. and Bizer, G.Y. Self-Schema Matching and Attitude Change: Situational and Dispositional Determinants of Message Elaboration, *Journal of Consumer Research*, (31:4), 2005, 787-797.

Zhang, J., and Krishnamurthi, L. Customizing Promotions in Online Stores, *Marketing Science*, (23:4), 2004, 561-578.

Zuckerman, M. *Sensation Seeking: Beyond the Optimal Level of Arousal*, Lawrence Erlbaum Associates, Hillsdale, NJ. 1979.

Appendix I

(1) Variety Seeking

VS1. I enjoy taking changes in buying unfamiliar things just to get some variety to me.

VS2. I get bored with buying the same things even if they are good.

VS3. I search around a lot just to find out something more about the latest styles.

(2) Need for Cognition

NFC1. I don't like to have to do a lot of thinking [r][9].

NFC2. I try to avoid situations that require thinking in depth about something [r].

NFC3. I prefer to do something that challenges my thinking abilities rather than something that requires little thought.

NFC4. I prefer complex to simple problems.

[9] Items marked by [r] were reverse coded for statistical analysis.

15

NFC5. Thinking hard and for a long time about something gives me little satisfaction [r].

16