Sma-Sho: Implementation and Evaluation of a Shopping Support Service Using Receipt Log

Akihiro Okushi, Seiki Tokunaga, Shinsuke Matsumoto, Masahide Nakamura
Graduate School of System Informatics, Kobe University, JAPAN
1-1 Rokkodai-cho, Nada-ku, Kobe, Hyogo 657-8501, Japan
Email: {okushi, tokunaga}@ws.cs.kobe-u.ac.jp
{shinsuke, masa-n}@cs.kobe-u.ac.jp

Abstract—We have previously proposed the ReceiptLog service platform, which stores user’s purchase histories from daily receipts for value-added consumer services. As a practical application of the ReceiptLog, we design and implement a shopping support service, called “Sma-Sho”, in this paper. Sma-Sho is designed to cope with two typical problems in daily shopping: duplicate purchase and forgotten purchase. The duplicate purchase is that a user mistakenly buys a product although there is sufficient stock of the same product at home. The forgotten purchase is that a user forgets to buy a product although the product is out of stock. Using the ReceiptLog APIs extensively, Sma-Sho provides useful purchase histories for the user with a mobile terminal, to prevent the above two problems. We also conduct an experimental evaluation with 8 subjects, to show the effectiveness of the proposed method.

I. I NTRODUCTION

The consumer purchase history has been widely used by many companies. Information on when a customer bought which products is crucial for sales strategy planning, promotion, and customer relation management [1] [2].

However, we consider that the purchase history is useful not only for the companies, but also the consumers themselves. Using information extracted from the purchase history, we can implement various services, supporting the consumer’s lives from financial or health aspects. For example, the frequency distribution of purchased foods allows a consumer to review the daily diets. Also, log of the previously purchased price helps a consumer make decision to buy it with the present price.

In our previous research [3] [4], we have proposed and implemented the ReceiptLog service platform (called ReceiptLog, for short). The ReceiptLog stores purchase history obtained from daily receipts of consumers, and provides data and API for value-added application services. The ReceiptLog consists of a receipt scanner, a ReceiptLog database, and a ReceiptLog Web API to retrieve the data from the database.

We have so far registered about 3,300 receipt data during 27 months. We implemented 44 ReceiptLog Web APIs. Since the API is implemented by platform-independent Web services (both in SOAP and REST), it is easy for various applications with different languages to access the API.

As a practical application, we have implemented a social network Web application, called ReciLog [3] [4]. It allows consumers to review their daily receipts like Blog, as well as to share receipts with friends. Currently, our application is ReciLog only, and developing further applications is our challenge.

In this paper, we implemented shopping support service “Sma-Sho” as a practical application using ReceiptLog. Sma-Sho is designed to cope with two typical problems in daily shopping: duplicate purchase and forgotten purchase [5]. The duplicate purchase is that a user mistakenly buys a product although there is sufficient stock of the same product at home. The forgotten purchase is that a user forgets to buy a product although the product is out of stock. We assume that user uses Sma-Sho with a mobile terminal in shopping. Using ReceiptLog Web APIs, Sma-Sho offers functions that user can check the duplicate purchase and the forgotten purchase. Using ReceiptLog Web APIs, Sma-Sho offers user’s purchase day and frequency to user. User can check the duplicate purchase and the forgotten purchase with this information. In addition Sma-Sho offers function to support purchase decision making.

As an evaluation experiment, we got 8 people to use Sma-Sho and then to respond to questionnaire. We confirmed that Sma-Sho enables to prevent two problems, especially, the forgotten purchase.

II. P RELIMINARIES

A. Previous Work: ReceiptLog [3] [4]

Figure 1 shows the overall architecture of the ReceiptLog service platform. It consists of three components: receiptLog service platform, applications, and other lifelog services. In the ReceiptLog service platform (in the middle of the figure), a user first digitizes his/her daily receipts with a receipt scanner. The data is then accumulated in ReceiptLog database. The receipt data in the database can be obtained and searched via ReceiptLog Web API.

Figure 2 shows the data model of ReceiptLog, consisting of the following four entities (i.e., tables).

- user: User’s information is registered in this entity.
A. Overview of “Sma-Sho”

To cope with the duplicate purchase and forgotten purchase, we implement a shopping support service “Sma-Sho” using the ReceiptLog service platform. Sma-Sho is a coined word taken from “Smart Shopping” in full spelling.

Among many potential applications, this paper especially considers shopping support as a practical use of ReceiptLog. The daily receipts are full of shopping information, including product name, shop name, total payment, and date of purchased. Thus, the receipts can characterize shopping trend and preference of individual users. Therefore, the past receipt data can surely help users in the present shopping.

In this paper, we focus on two typical problems in daily shopping: duplicate purchase and forgotten purchase. The duplicate purchase is that a user mistakenly buys a product although there is sufficient stock of the same product at home. For instance, a man bought milk as he thought milk was out of stock. However, he went home and found that milk still remained in refrigerator. The forgotten purchase is that a user forgets to buy a product although the product is out of stock. For instance, a woman forgot to buy facial tissues although the tissues were out of stock. The goal of this paper is to cope with the above two problems by using ReceiptLog extensively. According to an attitude survey on 500 housewives [5], these two problems were reported as very typical mistakes in grocery shopping. Hence, they deserve to be solved from a practical viewpoint.

B. Utilize ReceiptLog in Scene of Shopping

We then explain the applications using ReceiptLog, shown in the upper side of Figure 1. Indeed, there are many potential applications and services for various purposes. They include economy-oriented applications, health-care services, quality-of-life improvement, etc. Our previous “ReciLog” and “Sma-Sho” proposed in this paper are such instances of applications.

Finally, as shown in the bottom of Figure 1, we plan to integrate ReceiptLog with other lifelog services. The mashup would achieve more valuable services than ReceiptLog itself. We expect that there are a wide variety of applications of ReceiptLog, according to intended users, scenes, and service objectives. Thus, it is our great challenge to implement value-added consumer services and applications.

We then explain the applications using ReceiptLog, shown in the upper side of Figure 1. Indeed, there are many potential applications and services for various purposes. They include economy-oriented applications, health-care services, quality-of-life improvement, etc. Our previous “ReciLog” and “Sma-Sho” proposed in this paper are such instances of applications.

Finally, as shown in the bottom of Figure 1, we plan to integrate ReceiptLog with other lifelog services. The mashup would achieve more valuable services than ReceiptLog itself. We expect that there are a wide variety of applications of ReceiptLog, according to intended users, scenes, and service objectives. Thus, it is our great challenge to implement value-added consumer services and applications.

Figure 1. ReceiptLogService

Figure 2. Data model of ReceiptLog Database

- **receipt**: Information of each receipt (date, time, total payment, shop address) is registered.
- **detail**: Details (name, price, expense category) of goods contained in each receipt are registered.
- **category**: Expense category (food, clothing, etc.) of each purchase details goods is registered.

For instance, let us take a receipt “r225” in Figure 2. In this receipt, we can see that user “okushi” purchased for 335 yen at “Hankyu Oasis” in Kobe city on October 26 at 17:52. It can be in the detail table that user “okushi” bought wasabi for 98 yen and sashimi of salmon for 335 yen.

We introduce typical ReceiptLog Web APIs as follows:

* **getReceipts(userID, date)** returns a list of receipts issued for a given user and date.
* **getDetailByTerm(userID, since, until)** returns a list of receipt details of a given user issued between start date and end date.
* **getProductPurchaseHistory(userID, product name)** returns a list of receipt details for a given product, sorted by purchased date.

For instance, if a user calls **getDetail(okushi, r225)**, then two details (wasabi and Sashimi of salmon) are returned.

We then explain the applications using ReceiptLog, shown in the upper side of Figure 1. Indeed, there are many potential applications and services for various purposes. They include economy-oriented applications, health-care services, quality-of-life improvement, etc. Our previous “ReciLog” and “Sma-Sho” proposed in this paper are such instances of applications.

Finally, as shown in the bottom of Figure 1, we plan to integrate ReceiptLog with other lifelog services. The mashup would achieve more valuable services than ReceiptLog itself. We expect that there are a wide variety of applications of ReceiptLog, according to intended users, scenes, and service objectives. Thus, it is our great challenge to implement value-added consumer services and applications.

B. Utilize ReceiptLog in Scene of Shopping

Among many potential applications, this paper especially considers shopping support as a practical use of ReceiptLog. The daily receipts are full of shopping information, including product name, shop name, total payment, and date of purchased. Thus, the receipts can characterize shopping trend and preference of individual users. Therefore, the past receipt data can surely help users in the present shopping.

In this paper, we focus on two typical problems in daily shopping: duplicate purchase and forgotten purchase. The duplicate purchase is that a user mistakenly buys a product although there is sufficient stock of the same product at home. For instance, a man bought milk as he thought milk was out of stock. However, he went home and found that milk still remained in refrigerator. The forgotten purchase is that a user forgets to buy a product although the product is out of stock. For instance, a woman forgot to buy facial tissues although the tissues were out of stock.

The goal of this paper is to cope with the above two problems by using ReceiptLog extensively. According to an attitude survey on 500 housewives [5], these two problems were reported as very typical mistakes in grocery shopping. Hence, they deserve to be solved from a practical viewpoint.

III. Sma-Sho: Shopping Support with Receipt Log

A. Overview of “Sma-Sho”

To cope with the duplicate purchase and forgotten purchase, we implement a shopping support service “Sma-Sho” using the ReceiptLog service platform. Sma-Sho is a coined word taken from “Smart Shopping” in full spelling.

Screenshots of Sma-Sho are shown in Figure 3. A user is assumed to use Sma-Sho with a smart phone during
shopping. The user first logs in the service by filling userID in Figure 3(a). Sma-Sho provides the following three features:

- Prevention of duplicate purchase with “When did I buy it?” and “What did I buy recently?” (Figure 3 (b)(c))
- Prevention of forgotten purchase with “Is there sufficient stock?” (Figure 3 (d))
- Decision making with “how much was it?” and “What did I buy this month of last year?” (Figure 3 (e)(f))

On receiving user’s input, Sma-Sho calls ReceiptLog Web API. It then displays the retrieved purchase history or makes purchase recommendation. Details of each feature are explained in the following subsections.

### B. Prevention of Duplicate Purchase

**Cause of Duplicate Purchase:** Through our investigation with examples, we found that there are two patterns generating the duplicate purchase of a product $p$.

- **Pattern 1:** A user forgets a fact that (s)he bought a product $p$ in the previous shopping.
- **Pattern 2:** A user forgets what (s)he bought recently.
As an example of Pattern 1, let us consider a duplicate purchase of “Wasabi (horseradish)”. Wasabi is often bought as supplementary seasoning of main ingredient, such as Sashimi or Sushi. Since buying wasabi is not the main purpose of shopping, it is easy for the user to forget the fact. As a result, (s)he buys wasabi again although there are sufficient stocks at home.

In this case, he can avoid the duplicate purchase if he is able to know when he bought wasabi recently. Thus, the duplicate purchase of a specific product \( p \) can be prevented by displaying the purchase history of \( p \). The purchase history can be retrieved by calling ReceiptLog API \( \text{getProductPurchaseHistory()} \) described in Section II-A. As a typical example of Pattern 2, we suppose daily grocery shopping where a user wants to reuse remaining foods in the refrigerator. Typically, the user does not remember all the things in the refrigerator, which may cause the duplicate purchase. In this case, he can prevent the duplicate purchase if we is able to know all what (s)he bought recently. The API \( \text{getDetailByTerm()} \), described in Section II-A, can implement the feature to obtain a list of all products recently bought.

Sma-Sho covers the above two patterns by the following two screens.

**“When did I buy it?” screen:** Shown in Figure 3(b). Using this screen, a user can prevents duplicate purchase caused by Pattern 1. The user inputs product name and presses search button. Then, Sma-Sho displays the latest date with a shop name when the user bought the product. If the user bought the same product more than once, the user can search older ones by pressing next button.

To implement the screen, \( \text{getProductPurchaseHistory()} \) is used to obtain the date and shop name. Figure 3(b) shows a result, where user “okushi” search the purchase history of “wasabi”. Sma-Sho says that “okushi” bought a wasabi on 5 February 2012 at “Hankyu Oasis, Mikage” store. Thus, “okushi” can decide whether or not he should buy wasabi now.

**“What did I recently buy?” screen:** Shown in Figure 3(c). In this screen, a user can prevent duplicate purchase caused by Pattern 2. When the user presses a display button, then Sma-Sho displays the latest five products that the user bought. The user can search older history by pressing next button if a user wants to search them. To implement the screen, \( \text{getDetailByTerm()} \) in ReceiptLog Web API is used to obtain the product details bought within past one month. Seeing the product list, the user can recall what he recently bought, which prevents the duplicate purchase.

### C. Prevention of Forgotten Purchase

**Cause of Forgotten Purchase:** We have interviewed several housewives on what products they often forget to buy. They said that they often forget to buy daily supplies, including facial tissues, toilet papers, trash bags, toothbrushes, etc. In addition to them, milk, eggs and mayonnaise are typical goods often forgotten to purchase \cite{5}. We have derived the following properties common for all the above products.

- daily necessities with consumption cycle.
- consumed relatively slowly.
- stocks are stored in place not seen every day.

These products are likely to run out of stock if long time has passed since the user bought them last. Based on this observation, Sma-Sho implements a feature that tells the user goods that are about to run out. Specifically, we implement the feature by the following algorithm.

1. For a target product \( p \) of the forgotten purchase, the user manually configures a list \( N(p) \) of typical name of \( p \) and average consumption cycle \( c(p) \) of \( p \).
2. When requested by a user, Sma-Sho calls \( \text{getProductPurchaseHistory}(n) \) for every product name \( n \) in \( N(p) \), in order to obtain the latest purchased date \( r \).
3. Let today be the date of today. If the elapsed date \( \text{today} - r \) is greater than the consumption cycle \( c(p) \), Sma-Sho says that that \( p \) is likely to be out of stock and recommend the user to buy \( p \).

For instance, we explain a scenario that Sma-Sho recommends user “okushi” to buy facial tissues. Suppose that it is 9 February 2012, today.

1. As typical brand names of facial tissues, “okushi” sets \( N(\text{tissues}) = \{ \text{Kleenex, Scotties, Nepia, Ellemoi, ...} \} \). He also sets the consumption cycle \( c(\text{tissues}) = 100 \) (days) \(^1\).
2. When “okushi” requests, Sma-Sho calls \( \text{getProductPurchaseHistory}(n) \) for every name \( n \) in \( N(\text{tissues}) \). Then, the API returns the latest purchased date 26 November 2011.
3. Sma-Sho then calculates the elapsed days since 26 November 2011 until 9 February 2012. As a result, 106 day is greater than \( c(\text{tissues})(100d) \). So, Sma-Sho says that tissues are likely to run out, and recommends to buy them.

**“Is there sufficient stock?” screen:** The above algorithm is implemented in the screen shown in Figure 3(d). By using this screen, a user can prevent the forgotten purchase. When a user presses the check button, Sma-Sho checks if each of the pre-set products is in stock or not using the algorithm. If the user presses the out-of-stock button, then Sma-Sho displays only products that are out of stock. If the user wants to know the purchase date of a product, the user selects the product from pull-down menu and presses search button.

Figure 3(d) shows a screen where Sma-Sho tells user “okushi” that tissues are likely to run out of stock.

\(^1\)This consumption cycle has been calculated by average amount of consumption of tissues per a year, according to statistics.
D. Supporting Decision Making

We consider the usage of ReceiptLog in shopping other than the above two features. ReceiptLog is a history of shopping that reflects user’s decision and taste. Therefore, we consider to support decision making in shopping.

In general, everybody has experienced an indecisive situation in shopping. Based on our experience and the interview to housewives, we have found that there are typically two kinds of indecisive situations.

- **Case 1**: A user wants a product, however, (s)he cannot decide whether to buy it here for the present price, or not.
- **Case 2**: A user has no idea what to buy in the shopping.

  For instance of Case 1, suppose that a housewife finds her favorite cleanser sold for 250 yen in a new drugstore. She may hesitate to buy it since she does not remember if 250 yen is the reasonable price. In such a case, ReceiptLog can help her by showing her purchase history of the cleanser to compare the price.

  As for Case 2, suppose that a housewife goes to supermarket to buy foods for dinner. However, she cannot come up with the menu of dinner, and has no idea what to buy. In such a case, ReceiptLog may help her by showing what she bought this month of one year ago, as reference information to come up the idea.

Sma-Sho covers the above two cases with the following features.

- **“How much was it?” screen**: Shown in Figure 3(e). Using this screen, Sma-Sho supports users to make decision for Case 1. When a user inputs product name and presses search button, then price, date, and shop name that he bought the product are shown sorted by price. So, the user can compare the price and check if the present price is reasonable.

  In this screen, `getProductPriceRanking()` in ReceiptLog Web API is used to obtain the list of price and shop name of the given product. Figure 3(e) shows that user “okushi” search the lowest price of wasabi. The price was 98 yen on 5 February 2012 at “Hankyu Oasis Mikage”.

- **“What did I buy this month of last year?” screen**: Shown in 3(f). This screen helps users decide for Case 2. When a user presses display button, then Sma-Sho showing the products that the user bought in around the same date of last year. The products would remind the user of what (s)he or her family like, and/or promote a new idea. To implement this screen, `getDetailByTerm()` in ReceiptLog Web API is used.

E. Implementation of Sma-Sho

Sma-Sho has been was deployed in Apache2.2.3 Web server. It was also integrated with ReceiptLog Web API, which had been already deployed as Axis2 Web services (within Tomcat 5.5 server). Sma-Sho is currently on service, experimentally used by interested people and relatives of our research group.

IV. EXPERIMENTAL EVALUATION

A. Experiment Overview

In order to evaluate Sma-Sho, we have conducted a user experiment. Eight master course students participated as subjects of the experiment. Before the experiment, they had already stored their receipts in ReceiptLog Service for one or two years. In the experiment, we asked the subjects to use Sma-Sho as they wanted. After they used Sma-Sho, we collected a questionnaire from every subject. The questionnaire consisted of the following questions. For each of the duplicate purchase and the forgotten purchase, two sets of the same questions were asked conducted.

- Do you think Sma-Sho is useful to prevent the duplicate purchase (or forgotten purchase, in another set)? (1,2,3,4,5)
- How is Sma-Sho useful specifically?
- What is insufficient in Sma-Sho?
- What features do you need more for Sma-Sho?

B. Results

The results of questionnaire are shown in Table I and Figure 4. The average score for the prevention of duplicate purchase was 3.8. Overall, we got positive evaluation. A subject, who marked low score (2), said “I don’t have much experience of duplicate purchase”. The positive comments include that; “it is convenient to use it during shopping”, “it is useful for preventing duplicate purchase of what I don’t buy frequently”, etc. On the other hands, a subject complaint that he couldn’t get expected search result.

The average score for the prevention of the forgotten purchase was 4.1. We received higher score than that of the duplicate purchase, which reflects that Sma-Sho is particularly useful in that context. Subjects proposed some improvement of Sma-Sho, including; “I want to customize the products and their consumption cycles by myself”, and “I want a feature that automatically alerts the out of stock.”

<table>
<thead>
<tr>
<th>Score</th>
<th>duplicate purchase</th>
<th>forgotten purchase</th>
</tr>
</thead>
<tbody>
<tr>
<td>5 (useful)</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>4</td>
<td>2</td>
<td>5</td>
</tr>
<tr>
<td>3</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1 (unuseful)</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Table I

IS SMA-SHO USEFUL TO PREVENT DUPLICATE PURCHASE / FORGOTTEN PURCHASE? (1,2,3,4,5)
C. Discussion

From the result of experience, it was shown that displaying the latest purchase date of a product was effective for preventing duplicate purchase. In the subsequent interview, several subjects said that the duplicate purchase occurs more frequently for fresh foods or seasoning, rather than necessities without consumption limit. Such foods sometimes include milk, bread, vegetables, etc. For these fresh foods, it would be more effective to count not only the latest purchase date but also the best-before date.

As for the prevention of forgotten purchase, subjects thought it convenient that a list of products that were likely to be forgotten was already set in Sma-Sho. Currently, the list of products and their consumption cycle are common for all users, which was determined based on the preliminary survey. However, the products and their consumption cycles may vary from a user to another. So, it would be more practical that every user can make a fine tune for them. Furthermore, in the current implementation, a user has to actively access Sma-Sho to check forgotten purchase. An extension would be that Sma-Sho automatically notifies the user of out of stock. The automatic alert would be more effective, although it would be annoying for some users. We should examine its feasibility based on individual preferences.

A common problem among Sma-Sho as well as ReceiptLog is in the accuracy of product search. Currently, when a user searches receipt data by a product, the user has to specify the product name, but not the product category. Therefore, the user cannot search wasabi if the product has a name that does not contain the string “wasabi”. In addition, even if a user wants to search “salt”, other products like “salty taste chips” are found. To solve this problem, it is essential to use standardized classification of products (e.g., JICFS [6]) as meta-data of each product purchased. This is left for our future work.

V. Conclusion

In this paper, we have implemented a practical application “Sma-Sho” using ReceiptLog. Sma-Sho is designed to cope with two typical problems “duplicate purchase” and “forgotten purchase” in shopping, using purchase history obtained from ReceiptLog extensively. Sma-Sho also implements a feature to support decision making during shopping. We also conducted an experimental evaluation with eight subjects to see the effectiveness of the Sma-Sho.

Our future work is to perform the suggested improvements in the experiment, as noted in Section IV-C. In addition, we want to conduct long-term experiment for investigation how Sma-Sho effects user’s shopping and life style.

ACKNOWLEDGMENTS

This research was partially supported by the Japan Ministry of Education, Science, Sports, and Culture [Grant-in-Aid for Scientific Research(B) (No.23300009), Young Scientists (B) (No.21700077), Research Activity Start-up (No.22800042)], and Hyogo Science and Technology Association.

REFERENCES