Paper:

Design and Evaluation of Consumer-Oriented Reviewing Platform with Receipt Log

Seiki Tokunaga*, **Akihiro Okushi****, **Sachio Saiki***, **Shinsuke Matsumoto***, **and Masahide Nakamura***

*Kobe University
1-1 Rokkodai, Nada, Kobe, Hyogo 657-8501, Japan
E-mail: {tokunaga@ws., masa-n@}cs.kobe-u.ac.jp
**NS Solutions Kansai Corporation
7-20-1 Fukushima, Fukushima Osaka, Osaka 553-0003, Japan
E-mail: okushi@ws.cs.kobe-u.ac.jp

[Received April 3, 2015; accepted August 18, 2015]

This paper develops *ReceiptLogService Platform*, which enables consumers to using their personal purchase receipts, store their receipt logs, and to use the data for various consumer services. The proposed platform consists of three components: *receipt scanner*, *ReceiptLog DB*, and *ReceiptLog API*. The receipt scanner digitizes daily receipts, and the ReceiptLog DB manages the scanned data. The ReceiptLog API provides the receipt log as a service. The API consists of the BasicAPI, which provides fundamental access for the receipt log, whereas the MiningAPI performs a statistical analysis of the receipt log. These APIs are published as Web services, and can used by multiple applications and services for various purposes. We also conduct an experimental evaluation with actual subjects, to confirm the usefulness of services with receipt log.

Keywords: receipt log, web services, database, web API, social network service, shopping support

1. Introduction

By analyzing an enormous quantity of purchase history data, the companies can glean useful information, such as hot-selling or cross-selling items, as well as the consumers' preference. Thus, using purchase history has been key for companies to not only increase their profits but also to improve consumer satisfaction.

We consider that the purchase history should not be limited to business purpose, however. Consumer histories are also useful for the *consumers themselves*. A user can record his purchase history, which we call a *receipt log* from his daily receipts. By collecting a sufficient number of entries from his receipt log, the user can review his purchase history and may find interesting economic information such as the lowest prices of certain goods. Integrating the receipt log with other data sources can yield more sophisticated information. For example, by using body metrics, including weight and BMI with the receipt log, the user may review the relationship between his health and food choice.

Traditionally, daily receipts have been recorded as *bookkeeping*, typically written down on pieces of papers, entered in ledger books, or managed using stand-alone applications. However, the primary purpose of the bookkeeping is to keep track of daily income and expenditures, so this is limited to financial concerns. Beyond conventional bookkeeping, our goal is to extend the use of receipt log to many other value-added services, including daily life improvements and social network services.

The goal of this paper is to provide a consumer-oriented receipt lifelog service. Providing this service would enable consumers not only to review the logs from an economic perspective; the receipt log service would also be useful in the area of shopping. Concretely, we design and implement a service platform (called *ReceiptLogService*) which stores and uses receipt logs for consumers. ReceiptLogService is a Web service that allows external applications to store and retrieve the receipt log easily. Our implementation consists of a receipt scanner, a MySQL database, and ReceiptLogAPI. The receipt data can be accessed via an API such as `getReceipt()` with parameters such as username and date. We have developed two types of ReceiptLogAPI. One is BasicAPI which enables one to call up a receipt. The other is MiningAPI, which supports the statistical analysis of receipts. Since the APIs are invoked by platform-independent and XML-based protocols (e.g. SOAP or REST) based on the concept of Service-Oriented Architecture, it is easy for various applications with different languages to access the API.

Using the APIs, we have also designed and implemented a shopping support service Section 4.1 and social networking service Section 4.5. The aim of the forward shopping support service is to prevent two important issues, namely *duplicate purchases* and *forgotten purchases*. A duplicate purchase refers to when a user mistakenly buys a product even though the still has plenty of it at home. A forgotten purchase refers to when a user forgets to buy a product event though the user has run out of it. Using the ReceiptLog APIs extensively, Sma-Sho, used on a mobile terminal, prevents these two problems by providing the user with useful purchase histories.

ReciLog make is possible to end-users to be able to review their own daily receipts, as if they were in a blog, and share those receipts for comments. We have also conducted a usability experiment with actual subjects, in order to evaluate the advantaged and limitations of the ReciLog and Smas-Sho. As a result, we have confirmed that receipt log is helpful both in terms of reviewing at the economic and sharing aspects.[1]

2. Preliminaries

2.1. Purchase History

Many companies have formed management and sales strategies based on the purchase histories of consumers. In modern systems, the point of sales (POS) system the purchase history is used to store the purchase history in a database [4]. The data items of the purchase histories generally include the date and time of purchase, age of the user, location, and other information. RFM analysis [5] is a well-known method that finds good consumers. The analysis determines the rank of consumers based on *recency*, *frequency*, and *monetization*. Companies use these data to form sales strategies, such as to determine with potential consumers, should receive catalogs and other direct mailings. These analysis are use purchase histories for business purpose; they are not available for use by the consumers themselves; they aren't able to use for consumer themselves. Consumers need a platform for recording their own receipts and a service that records daily receipts so that consumers may be able to put their own purchase histories to use for their own benefit.

2.2. Using Receipts as a Lifelog

A *lifelog* [6, 7] is a social activity to record of various aspects of human-life in terms of events, status, and relationships. The recorded log is used to look back on and improve one's daily life, to discover one's identity and to determine actions. Recently, a variety of *lifelog services*, including tweets, locations, weights, and beat steps, have appeared on the Internet. We can access a wide variety of lifelogging services via API.

A receipt contains a good deal of information; shop name, product name, total, time of purchase. Thus, from the perspective of consumers as well, the receipt provides valuable information. Concretely, the shop name on the receipt shows the consumer has bought something, and the products on the receipt also indicate the consumer's habits. Using the total information, consumers can review their spending. Nowadays, some applications provide functions which store the receipt log. However, they aim to support the keeping of household accounts. Thus, we still hope to implement a platform which unlocks some of the additional potential that the receipt holds.

1. This paper, we have summarized our research group effort [1–3].

2.3. Using Receipt Log While Shopping

We believe that receipt log can be utilized to aid in purchase decisions in various situations. As receipt contain information that characterizes people's shopping habits, including product name, and shop name. Past receipt data can surely is of service in present shopping.

We focus on two typical problems related to daily shopping: *duplicate purchases* and *forgotten purchases*. The duplicate purchases refers to a product that the user mistakenly buys even though there is a sufficient stock of the same product at home. For instance, a man might buy milk because he thinks he has run out of it, but he finds there is still milk in his refrigerator when he arrives home. The forgotten purchases is that a user forgets to buy a product although the product is out of stock. For example, a woman forgot to buy tissue paper although tissue was out of stock. Using purchase history from receipt log, we think it possible to prevent these two problems.

2.4. Potential for Sharing the Receipt Log

We believe it would be beneficial if our receipt log could be shared. One advantage would come from using the shared receipts from an economic perspective. For example, if a user could share his receipts with friends, that would help in various scenes such as while shopping. Another advantage would be that receipts would promote communication. This is a *lifelog*, as we stated before. Moreover, if not only user's purchase histories but also feelings such as "Buying this product made me very happy" could be shared, this could enrich people's lives. However, there is no standard platform that enables us to store and review receipt logs, nor is there a receipt log application to enable sharing within a community.

3. Service Platform for Reviewing Daily Receipts

3.1. Platform Overview

We believe that the recording of daily receipts by consumers can be a consumer-oriented lifelog service that reflects their purchase history. Moreover, integrating the receipt log with other data sources e.g. pictures or body measurements, may enable implementation of sophisticated services.

To facilitate the development of such lifelog services, we have designed and implemented a service platform *ReceiptLogService*, for storing and receipt data. **Fig. 1** shows the architecture of ReceiptLogService. First, each user digitizes daily receipts by using a receipt scanner [8],[2] respectively.

2. In Section 3.2, we explain the receipt scanner in detail. Moreover, we explain detail of the components of *ReceiptLogDB* and *ReceiptLogAPI* in Sections 3.3, 3.4 and 3.5

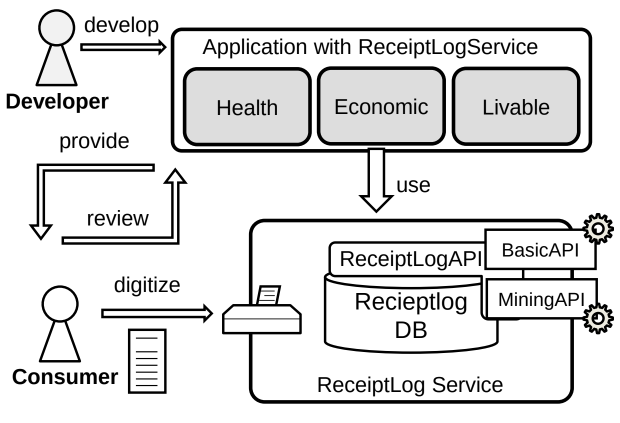
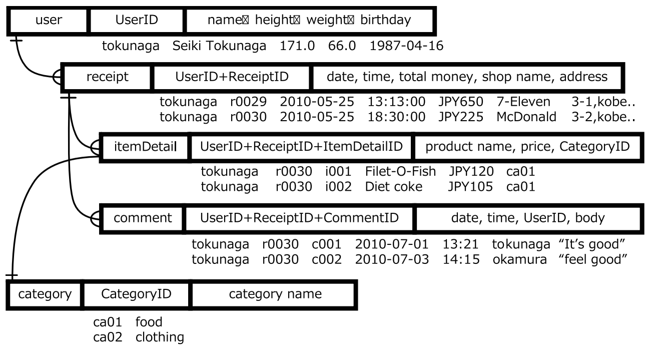


Fig. 1. Architecture of ReceiptLogService.



user
UserID
name⊠ height⊠ weight⊠ birthday
tokunaga Seiki Tokunaga 171.0 66.0 1987-04-16
receipt
UserID+ReceiptID
date, time, total money, shop name, address
tokunaga r0029 2010-05-25 13:13:00 JPY650 7-Eleven 3-1,kobe..
tokunaga r0030 2010-05-25 18:30:00 JPY225 McDonald 3-2,kobe..
itemDetail
UserID+ReceiptID+ItemDetailID
product name, price, CategoryID
tokunaga r0030 i001 Filet-O-Fish JPY120 ca01
tokunaga r0030 i002 Diet coke JPY105 ca01
comment
UserID+ReceiptID+CommentID
date, time, UserID, body
tokunaga r0030 c001 2010-07-01 13:21 tokunaga “It’s good”
tokunaga r0030 c002 2010-07-03 14:15 okamura “feel good”
category
CategoryID
category name
ca01 food
ca02 clothing

Fig. 2. Data schema of ReceiptLogDB.

Then, data items, such as a store name, purchase date, and time, the items, are extracted and stored in a global database, called *ReceiptLogDB*. The data can be accessed using external applications via *ReceiptLogAPI*. The ReceiptLogAPI has been implemented as a Web service so that the receipt log can be accessed from various platforms.

3.2. Receipt Scanner

As a receipt scanner, we use a commercial product *Yasasiku Kakeibo* [8] which includes a receipt scanning device and receipt data management software developed by Media Drive Corporation. First, *Yasasiku Kakeibo* uses a scanning device to read an image of a receipt. Then, the data items on the receipt such as a store name, purchase date and time, and items purchased are extracted from the receipt image by optical character reader. Each receipt data item is stored in ReceiptLogDB.

3.3. ReceiptLogDB

Receipt data items, scanned by the receipt scanner, are stored in ReceiptLogDB. By storing the receipt data in the shared ReceiptLogDB, these data are shared with multiple users. We have designed and implemented the ReceiptLogDB using the following schema.

ReceiptLogDB has five tables. **Fig. 2** depicts an ER diagram of the ReceiptLogDB. A box represents an entity (i.e. table), consisting of an entity name, a primary key, and attributes. Database instances are shown at the bottom of each box. We enumerate instances below each entity to support understanding. A line represents a relationship between entities, where $+\!\!-\!\!\in$ denotes a parent-children relationship, and $+\!\!-\!\!\cdots$ denotes a reference relationship. We show a partly normalized model because of space limitations. The five tables can be described as follows.

- User table: The user table stores personal information, such as the user ID, user name, height, weight of each user.
- Receipt table: The receipt table stores the information from each receipt. Each receipt has details such as product names, which are stored in the itemDetail table as a children-parent relationship. Basically, we assume that a query of ReceiptLogDB will frequently include username, so we set both ReceiptID and UserID as primary keys in order to improve access performance.
- ItemDetail table: The ItemDetail table stores detailed information from each receipt. We added both ReceiptID and UserID as primary keys. Product categories are defined in the category table are described below.
- Category table: The Category table stores product categories, such as food, clothing, daily necessities, hobby and hobby-related items.
- Comment table: This table stores user comments made by other users. The users can make comments on each other’s receipt as they do on a social network postings. Recorded information includes name, date and comment.

3.4. BasicAPI

BasicAPI allows access to the ReceiptLogDB by means of simple and abstract API invocations for ReceiptLogService users. The following, is a lists of ReceiptLogAPIs and their features.

- `getReceipt()` returns a list of receipts for a specified user issued on a specified date.
- `getReceiptDetail()` returns the details of a specified user and receipt.
- `getDetailByTerm()` returns the details of a specified user and term.
- `getUserOutgo()` returns total outgo by a specified user during a specified period.
- `getProductPurchaseHistory()` returns the products in a specified purchase history.
- `getUserComments()` returns the comments of a specified user issued on the date as well as the receipts that elicited the comments.

```
<return type="Receipt">
    <UserID>tokunaga</userID>
    <ReceiptID>r0029</ReceiptID>
    <date>2010-05-25</date>
    <time>13:13:00</time>
    <totalMoney>225</totalMoney>
    <shopName>7-Eleven</shopName>
    <address>3-1, Rokko, Kobe, Japan</address>
</return>
<return type="Receipt">
    <UserID>tokunaga</UserID>
    <ReceiptID>r0030</ReceiptID>
    <date>2010-05-25</date>
    <time>18:30:00</time>
    <totalMoney>650</totalMoney>
    <shopName>McDonald</shopName>
    <address>3-2, Rokko, Kobe, Japan</address>
</return>
```

Fig. 3. Result of `getReceipt(tokunaga, 2010-05-25)`.

Receipt[] getRecency (userid, target, condition, limit)

Parameters:
userid - the ID means the owner of log
target - the analyzed target in the receipt
(i.e. shop-name, total-payment,...)
condition - the condition how to analyze the target
limit - this item shows the number of returns
Returns:
Receipt[] **-** The list of Receipt object which has date, time, shop-name, purchased product, total-payment

Fig. 4. Interface of `getRecency`.

The API can be invoked using standard Web service protocols, based on the concept of Service-Oriented Architecture (i.e. SOAP and REST) [9]. **Fig. 3** shows a REST invocation of `getReceipt()`, which produces four receipts from user `Tokunaga`, receipts issued on 2010-05-25.

3.5. MiningAPI

MiningAPI enables the statistical analysis of receipt log. In this paper, we propose a method for consumers to analyze receipt logs for themselves using RFM analysis in Section 2.1. In analyzing the stored receipt log, we determine how recently the consumer has bought a specific product, how frequently the consumer visits a specific shop, and how much money the consumer spends. Consumers could gain much valuable lifelog service by using RFM analysis. The following is a list of MiningAPI lists.

- `getRecency()` How recency has the consumer gone shopping or bought a product?
- `getFrequency()` How frequently does the consumer go shopping or buy a specific product?
- `getMonetary()` How much mondey does the consumer spend in a shop or on a product?

int getFrequency (userid, target, condition, between)

Parameters:
userid - the ID means the owner of log
target - the target is selected from five column
product name, shop name, area, price, total
condition – specified some query condition(e.g. >=, ==,...)
between - this column specifies the term for the query
Returns:
int **-** returns the frequency which corresponds to the invoked query

Fig. 5. Interface of `getFrequency`.

int getMonetary (userid, target, condition, between)

Parameters:
userid - the ID means the owner of log
target - the target is selected from five column
product name, shop name, area
condition - query that specified some perspectives
between - this specifies the term for the query
Returns:
int **-** returns the monetary which corresponds to the invoked query

Fig. 6. Interface of `getMonetary`.

- `getTimeIntervalReport()` The API shows the timeline report using RFM.
- `getTrend` What is the trend for the user?

Fig. 4 shows the interface of `getRecency()`.

The API was able to access the latest receipt log with perspective target such as product name or store name. This API has targeted information under five data categories; product name, store name, area, price, and total. We therefore call up the recent receipt log by using the five categories above. Using the cond parameter, we can limit the return value of the API. The API responds to the array of receipt object so that the API-user can get some of the recent receipts.

We have also implemented `getFrequency()` and `getMonetary()`, as shown in **Figs. 5** and **6**. Their interfaces are similar to that of `getRecency()`. With the `getFrequency()`, we can understand how frequently the user goes shopping or how frequently the user buys a product. By using the `getMonetary()`, we can learn how much the product costs at the store.

4. ReceiptLog Applications

4.1. Shopping Support Application "Sma-Sho"

"Sma-Sho" is a smart shopping support service that uses the receipt log.

Sma-Sho is designed to cope with two typical problems faced in daily shopping: *duplicate purchases* and *forgotten purchases*. A duplicate purchases refers to when a user mistakenly buys a product even though there is enough at home. A forgotten purchase refers to when a user forgets to buy a product event though he has run out at home. Using the BasicAPIs and MiningAPIs extensively, Sma-Sho provides useful purchase histories so that the users with mobile terminals, can prevent these two problems.

4.2. Prevention of Duplicate Purchases

Cause of Duplicate Purchases: Through our investigation with examples, we found that there are two patterns that result in the duplicate purchase of a product p.

- Pattern 1: A user forgets that she has bought product p on a previous shopping trip.
- Pattern 2: A user forgets what she has bought recently.

As an example of Pattern 1, let us consider a duplicate purchases of "wasabi, or Japanese horseradish." Wasabi is often bought as a seasoning for the main course of a meal, such as sashimi or sushi. Since buying wasabi is typically not the main reason a person goes to the supermarket, it is easy to forget how much there is in the refrigerator. As a result, the shopper may buy wasabi even though there is enough at home.

In this case, the person can avoid making the duplicate purchase if he is able to find out when he last bought wasabi. 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 ReceiptLogAPI `getProductPurchaseHistory()` described in Section 3.4.

As a typical example of Pattern 2, we presuppose that the daily user wants to use up all the foods in the refrigerator when he does his daily grocery shopping. Typically, a user cannot remember everything he has in the refrigerator, and this may result in duplicate purchases. In this case, he can prevent making a duplicate purchases if he is able to know all that he has bought recently. The API `getDetailByTerm()`, described in Section 3.4, can implement the feature to obtain a list of all recent purchases.

Screenshots of Sma-Sho are shown in **Fig. 7**.

A user is assumed to use Sma-Sho with a smartphone while shopping.

The "When did I buy it?" screen is shown in **Fig. 7(b)**. Using this screen, a user can prevent duplicate purchases caused by Pattern 1. The user inputs a product name and presses the search button. Sma-Sho displays the latest purchase date with a store name. If the user has bought the same product more than once, the user can search for previous purchases by pressing the "next" button. The user searches the purchase history of "wasabi." Sma-Sho respond that "okushi" bought wasabi on 5 February 2012 at the store "Hankyu Oasis." The user can decide whether or not he should buy wasabi.

4.3. Explanation of "Sma-Sho"

"Is there sufficient stock?" screen: The above algorithm is implemented in the screen shown in **Fig. 7(d)**. Using this screen, a user can prevent forgotten purchases. When a user presses the "check" button, Sma-Sho uses the algorithm to check whether each of the pre-set products is in stock or not. If the user presses the "out-of-stock button", Sma-Sho displays only products that are out of stock. If the user wants to know the purchase date of a product, he selects the product from the pull-down menu and presses the "search" button.

Figure 7(d) shows a screen that tells user "okushi" that tissues are likely to run out of stock.

Sma-Sho provides the following three features:

- Prevention of duplicate purchases with "When did I buy it?" and "What did I buy recently?" (**Figs. 7(b)**,**(c)**)
- Prevention of forgotten purchases with "Is there sufficient stock?" (**Fig. 7(d)**)
- Decision making with "How much does it cost?" and "What did I buy this month of last year?" (**Figs. 7 (e)**,**(f)**)

On receiving the user's input, Sma-Sho calls ReceiptLogAPI. It then displays the retrieved purchase history or makes the purchase recommendation. Each feature is explained in detail in the following subsections.

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

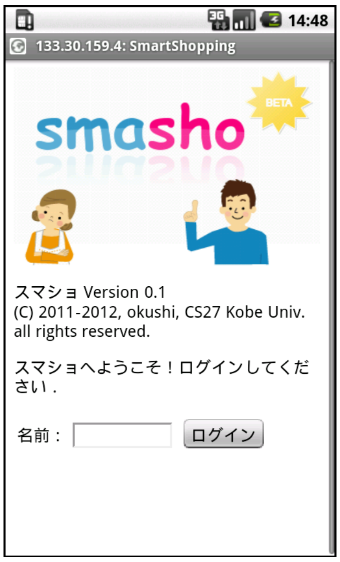
"When did I buy it?" screen: Shown in **Fig. 7(b)**. Using this screen, a user can prevent duplicate purchase caused by Pattern 1. The user inputs product name and presses the search button. Then, Sma-Sho displays the latest date with a shop name when the user bought the product. If the user bought the same product more than once, the user can search older ones by pressing next button.

To implement the screen, `getProductPurchase History()` is used to obtain the date and store name. **Fig. 7(b)** shows the result, of user 'okushi" searches for the previous purchase histories of "wasabi." Sma-Sho says that "okushi" bought 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 **Fig. 7(c)**. In this screen, a user can prevent duplicate purchase caused by Pattern 2. When the user presses a display button, Sma-Sho displays the last five products that the user bought. The user can search previous history by pressing the "next button." To implement the screen, `getDetailByTerm()` in ReceiptLog Web API is used to call up the products bought within the past one month. Seeing the product list, the user can recall what he recently bought, preventing the duplicate purchases.

4.4. Prevention of Forgotten Purchases

Cause of Forgotten Purchases: We asked several housewives what products they often forgot to buy. They

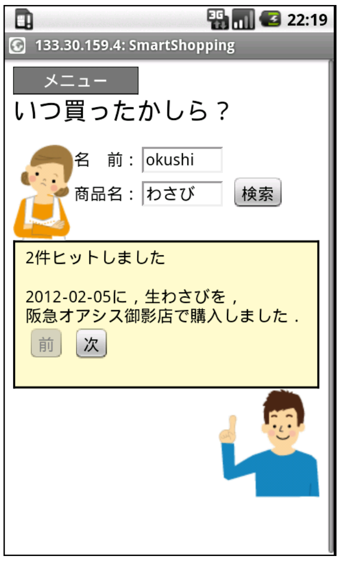

14:48
133.30.159.4: SmartShopping

スマショ Version 0.1
(C) 2011-2012, okushi, CS27 Kobe Univ.
all rights reserved.

スマショへようこそ！ログインしてください．

名前：

(a) “Login screen”


22:19
133.30.159.4: SmartShopping

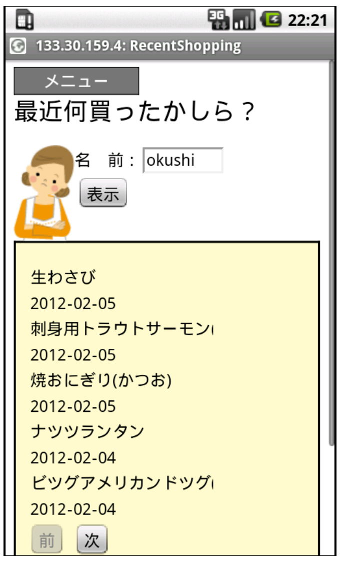
いつ買ったかしら？

名　前：okushi
商品名：わさび

2件ヒットしました

2012-02-05に，生わさびを，
阪急オアシス御影店で購入しました．

(b) “When did I buy it?”

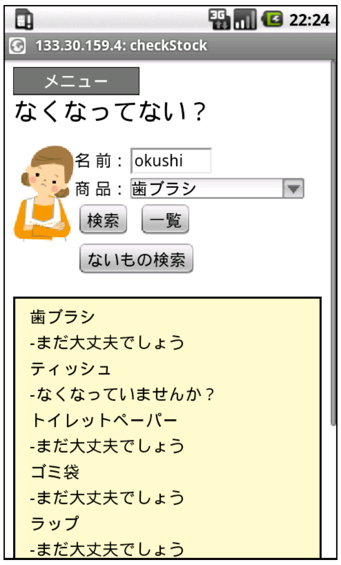

22:21
133.30.159.4: RecentShopping

最近何買ったかしら？

名　前：okushi

生わさび
2012-02-05
刺身用トラウトサーモン
2012-02-05
焼おにぎり(かつお)
2012-02-05
ナツツランタン
2012-02-04
ビッグアメリカンドッグ
2012-02-04

(c) “What did I recently buy?”

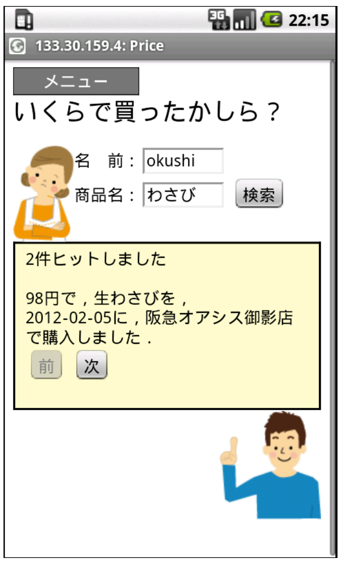

22:24
133.30.159.4: checkStock

なくなってない？

名 前：okushi
商 品：歯ブラシ

歯ブラシ
-まだ大丈夫でしょう
ティッシュ
-なくなっていませんか？
トイレットペーパー
-まだ大丈夫でしょう
ゴミ袋
-まだ大丈夫でしょう
ラップ
-まだ大丈夫でしょう

(d) “Is there sufficient stock?”


22:15
133.30.159.4: Price

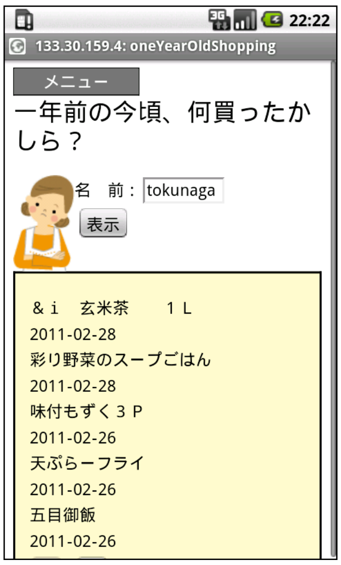
いくらで買ったかしら？

名　前：okushi
商品名：わさび

2件ヒットしました

98円で，生わさびを，
2012-02-05に，阪急オアシス御影店で購入しました．

(e) “How much does it cost?”


22:22
133.30.159.4: oneYearOldShopping

一年前の今頃、何買ったかしら？

名　前：tokunaga

＆ｉ　玄米茶　　１Ｌ
2011-02-28
彩り野菜のスープごはん
2011-02-28
味付もずく３Ｐ
2011-02-26
天ぷらーフライ
2011-02-26
五目御飯
2011-02-26

(f) “What did I buy a year ago this month?”

Fig. 7. Screenshots of “Sma-Sho.”

said that they often forgot to buy daily supplies, including facial tissues, toilet papers, trash bags and toothbrushes. In addition, milk, eggs, and mayonnaise are typical grocery items that are often forgotten [10]. We have derived the following properties common to all the above products.

- Daily necessities with consumption cycles.
- Items consumed relatively slowly.
- Supplies stored away and not seen every day.

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

1. For a target product p in forgotten purchase, the user manually configures a list $N(p)$ of the typical names of p and average consumption cycle $e(p)$ of p.
2. When requested by a user, Sma-Sho calls `getProductPurchaseHistory(`n`)` for every product name n in $N(p)$, to obtain the latest date of purchase$r$.
3. Let *today* be the today’s date. If the elapsed date $today - r$ is greater than the consumption cycle $e(p)$, Sma-Sho says that p is likely to be out of stock and recommends that the user buy p.

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

1. As typical brand names of facial tissues, “okushi”

sets $N(\text{tissues}) = \{\text{Kleenex, Scotties, Nepia, Ellemoi,}\ldots\}$. He also sets the consumption cycle $e(\text{tissues}) = 100$ (days).[3]

2. When "okushi" requests, Sma-Sho calls `getProductPurchaseHistory`(n) for every name n in $N(\text{tissues})$. Then, the API returns the latest purchase date of 26 November 2011.

3. Sma-Sho then calculates the number of days elapsed between 26 November 2011 and 9 February 2012. The result, 106 days, is greater than $e(\text{tissues})(100\text{d})$, so Sma-Sho says that tissues are likely to run out, and recommends he buy them.

"Is there sufficient stock?" screen: The above algorithm is implemented in the screen shown in **Fig. 7(d)**. By using this screen, a user can prevent the forgotten purchase. When a user presses the check button, Sma-Sho checks whether each of the pre-set products is in stock 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 the pull-down menu and presses the search button.

Figure 7(d) shows a screen that Sma-Sho tells user "okushi" that he is likely to run out of tissue.

4.5. Social Network Application: ReciLog

We have also developed a social network Web application, called *ReciLog*, as a client application of ReceiptLogService. ReciLog is constructed uses ReceiptLogAPI and has three main features. The main feature is the *reviewing feature*, which allows consumers to review daily receipts, just like as they would a blog. Therefore, they can look back in time as if reading a diary. Moreover, ReciLog has a *sharing feature*, which enables the users to exchange comments and recommendations about goods and purchases. Also, ReciLog has a *reporting feature (i.e. ranking, trend)*, which summarizes the purchases statistics using table. We developed ReciLog's main screen using BasicAPI. For example, the content of receipts is created with `getReceipt()`. Screenshots of ReciLog can be seen in **Figs. 8(a)** and **(b)**. We have also developed a function for analyzing the receipt log. **Figs. 8(c)** and **(d)** shows screenshots of ReciLog's analysis screen. Concretely, **Fig. 8(a)** shows the reviewing feature, which the user can use to review their receipt logs. The reviewing feature helps users to review their daily receipts easily. A user can add comments to each receipt, to describe or explain of the items, motivation for making the purchase, to give their impressions. ReciLog has a login screen for logging into the service. Once, the user has logged in a blog-like reviewing screen appears **Fig. 8(a)**. Moreover, we have developed the reporting feature **Fig. 8(b)**. Using the feature, users can review their receipt log with graphs and tables. The ReciLog currently supports the visualization of monthly expenses and their breakdown. **Fig. 8(b)** shows a screen summarizing the user "tokunaga's" total expenditures for 2011. Using the reporting feature, the user can easily look back on the receipts for a long time period. Integrating the reporting feature with the sharing feature, we can obtain consumption statistics from the community.

Finally, we present the analysis feature **Figs. 8(c)** and **(d)** in detail. If a user wants to see their trend or ranking of frequency, then he only has to push the button on the screen **Figs. 8(c)** and **(d)**. Using this feature, user can call up data without tallying the enormous quantity number of entries in the receipt log.

5. Experimental Evaluation

5.1. Experiment Overview

In order to evaluate the effectiveness of reviewing the receipt log for users, we conducted two experiments. We surveyed users about the applications which we have introduced. The total number of receipts registered in the RecieptLogService was 1,615, collected over a period of 22 months.

5.2. Experimental Evaluation of Sma-Sho

In order to evaluate Sma-Sho, we have conducted a user experiment in which eight master course students participated. Before the experiment, they had stored their receipts in ReceiptLogService 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 an questionnaire from each subject. The questionnaire consisted of the following questions. Two sets of the same questions were asked, one for duplicate purchases and one for forgotten purchases.

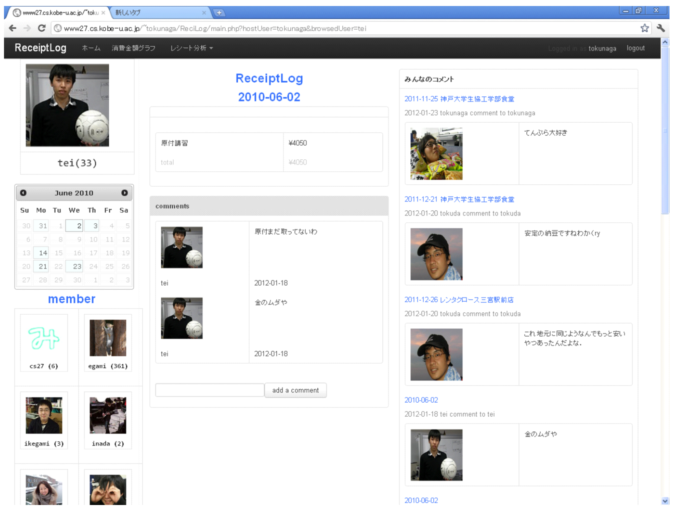
5.3. Result of Sma-Sho's Experiment

- Are the Sma-Sho's functions are useful?
- In what situations do you think Sma-Sho is useful?
- What are the Sma-Sho weak?
- Can you suggest any improvements for Sma-Sho?

The results of the questionnaire are presented in **Table 1**. The average score for the prevention of duplicate purchases was 3.8. Overall, we got a positive evaluation. A subject, who gave it a low score (2) said, "I haven't made many duplicate purchases." Positive comments included that; "It is convenient to use while I'm shopping," "It helps me avoid making duplicate purchases of things I don't buy frequently." On the other hand, one subject complained that he could not get the expected search results.

The average score for the prevention of the forgotten purchases was 4.1. The score was higher than that for the

3. This consumption cycle has been calculated by the average consumption of tissues per year, according to statistics.



ReceiptLog
2010-06-02

原付講習	¥4050
total	¥4050

comments

tei	原付まだ取ってないわ 2012-01-18
tei	金のムダや 2012-01-18

みんなのコメント

2011-11-25 神戸大学生協工学部食堂
2012-01-23 tokunaga comment to tokunaga
てんぷら大好き

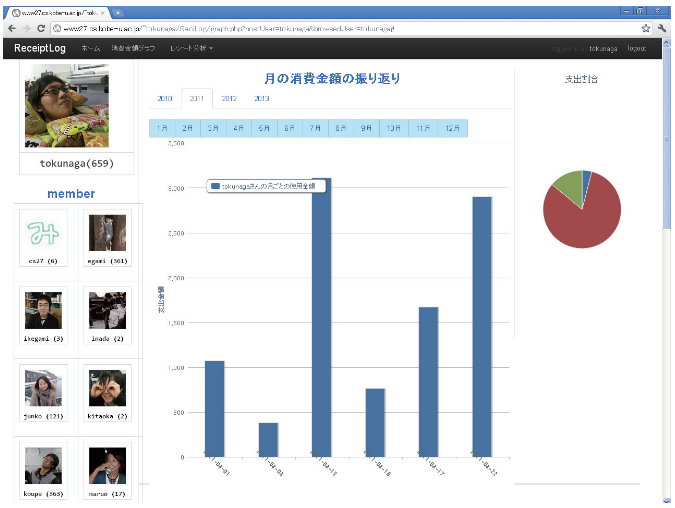
2011-12-21 神戸大学生協工学部食堂
2012-01-20 tokuda comment to tokuda
安定の納豆ですねわか(ry

2011-12-26 レンタクロース三宮駅前店
2012-01-20 tokuda comment to tokuda
これ地元に同じようなんでもっと安いやつあったんだよな.

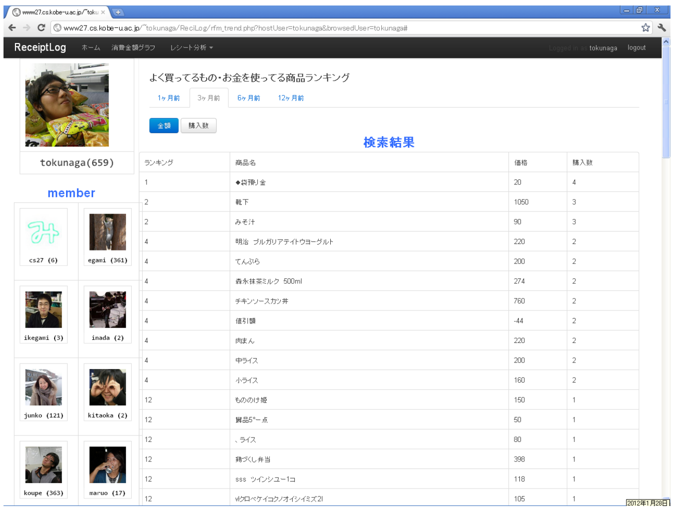
2010-06-02
2012-01-18 tei comment to tei
金のムダや

2010-06-02

(a) Reviewing receipt log and itś comments.



(b) Reviewing user's expenditure.

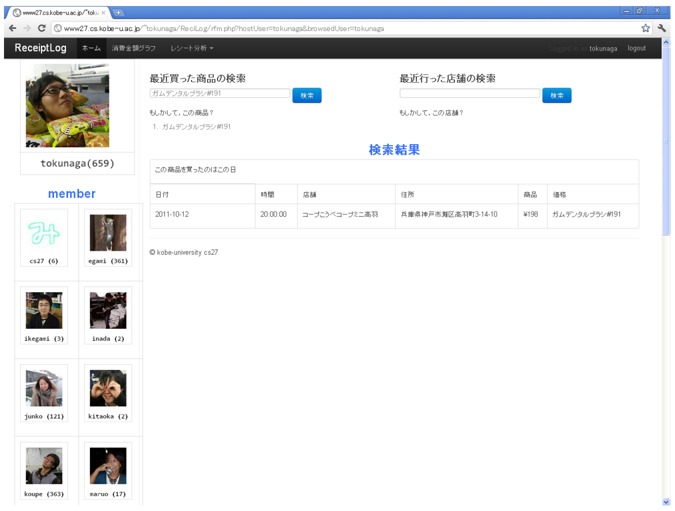


よく買ってるもの・お金を使ってる商品ランキング

検索結果

ランキング	商品名	価格	購入数
1	◆袋預り金	20	4
2	靴下	1050	3
2	みそ汁	90	3
4	明治 ブルガリアテイトウヨーグルト	220	2
4	てんぷら	200	2
4	森永抹茶ミルク 500ml	274	2
4	チキンソースカツ丼	760	2
4	値引額	-44	2
4	肉まん	220	2
4	中ライス	200	2
4	小ライス	160	2
12	もののけ姫	150	1
12	餅品5°ー点	50	1
12	、ライス	80	1
12	鶏づくし弁当	398	1
12	sss ツインシュー1コ	118	1
12	ｲｸﾛﾍﾟｹｲｺｸﾉｵｲｼｲﾐｽﾞ2l	105	1

(c) Ranking of user purchases.



最近買った商品の検索
ガムデンタルブラシ#191
もしかして，この商品？
1. ガムデンタルブラシ#191

最近行った店舗の検索
もしかして，この店舗？

検索結果

この商品を買ったのはこの日

日付	時間	店舗	住所	商品	価格
2011-10-12	20:00:00	コープこうべコープミニ高羽	兵庫県神戸市灘区高羽町3-14-10	¥198	ガムデンタルブラシ#191

© kobe-university cs27

(d) Analysis for user's latest and frequency of items.

Fig. 8. Screenshots of ReciLog.

Table 1. Does Sma-Sho help to prevent duplicate purchases or forgotten purchases? (1, 2, 3, 4, 5)

Score	Duplicate purchase	Forgotten purchase
5 (useful)	3	2
4	2	5
3	2	1
2	1	0
1 (unuseful)	0	0

duplicate purchases, which means that Sma-Sho is particularly useful in that context. Subjects proposed some improvements for Sma-Sho, including; "I want to customize the products and their consumption cycles by myself," and "I want a feature that automatically alerts me to things I'm out of."

5.4. Experimental Evaluation of ReciLog

Secondly, in order to evaluate ReciLog, we also have conducted an experiment, in which subjects actually used the ReciLog to review their own receipt logs. Ten master course students and three college students participated as subjects of the experiment. The questionnaires were asked related to economics and health.

In the experiment, we gave the subjects two tasks. "Task 1" was to evaluate the *self aspect* of ReciLog; where each subject reviewed his receipt log from the aspects of economics and health. "Task 2" was related to the *sharing aspect*, or the subjects' sharing of their receipt logs within the community. We also gave the subjects the opportunity to make free comments.

- Task 1: Please answer the following questionnaire in terms of your personal economics.

1. (E1-1) Is it interesting for you to review your receipt log?
2. (E1-2) Is it useful to review others' receipt logs?

Table 2. Results of ReciLog experiment.

	Economics		Health	
Evaluation	E1-1	E1-2	H2-1	H2-2
5 (Best)	4	2	2	0
4	8	4	5	1
3	0	2	3	6
2	1	2	2	5
1 (Worst)	0	3	1	1

- Task 2: Please answer following questionnaire in terms of your health.

1. (H2-1) Is it useful for you to review your receipt log?
2. (H2-2) Is it useful to review others' receipt logs?

Table 2 presents the results of the experiment. In terms of personal economics task, twelve subjects answered that their receipt log was "very useful" or "useful."

One subject responded "I can review the difference between my perception and actual expenditures.," and another said "I understood what product I had spent the money on." However, the evaluation of Task 2, reviewing others' receipt logs, varied from subject to subject. Positive comments, some subjects included, "It is useful to compare my expenditures to those of other people's in the same age group." On the other hand, negative comments included, "Because of the lack of recommendations, it was too difficult to review other's receipt logs."

Concerning the reviews of the receipt log in terms of health, the scores for Task 1 and Task 2 were 3.28, and 2.53, respectively, which were lower than the score in terms of economics.

Positive comments, included, "In reviewing my own via receipt log, I noticed that I had an unbalanced diet." A subject who likes luxury goods said that "I have spent a lot on luxury items, so I have to cut back." On the other hand, one subject commented, "It is difficult for me to judge good food from bad food by reviewing others' receipt logs."

5.5. Discussion

In reviewing the results of the two experiments, we found that the receipt log helped subjects in reviewing their lifestyles and in making purchase decision. In other words, the users could respond to the following questions with receipt log; "When" did you buy some products, "where" did you buy the products and "how much" did you spend for the products? The answers to these questions can provide many effective ways of reviewing purchases from the viewpoint of personal economics. Also, the MiningAPI we have developed is helpful for applications.

However, evaluation of the sharing the receipt logs varied among the users. This was because the subjects could not understand others' receipt logs just as some subjects said. "Sharing receipt logs was not so interesting, because of the lack of features (e.g., ranking of user's expenditures) in the application." We would therefore like to extend the sharing feature to make reviewing others' receipt logs more interesting. We will try to add new functions, such as the ability to make recommendation.

Moreover, we believe that reviewing the only receipt log has the limitations. Because the receipt log is juste a text-based log, so users feel it is difficult to remember the receipt log, whic makes it less interesting for the users.

To solve this problem, one approach is to "mashup" with other life log source [11]. Mashup is an approach to integrating different life log sources. To adopt this approach, we think that the receipt log has a potential to have a great deal of value for users. For example, a mashup of receipt log and users' weight data, will provide richer service from the aspect of health. Our future work is therefore to make a mashup of receipt log and other life logs.

6. Conclusion

We have developed a consumer-oriented ReceiptLogService platform. Specifically, we have designed and implemented a service platform called ReceiptLogService for recording and reading receipt logs. We also conducted an experimental evaluation with actual subjects, to confirm the usefulness of services with receipt log. As the result, we confirmed the effectiveness of receipt logs for the consumers in terms of economics and sharing aspects.

Acknowledgements
This research was partially supported by the Japan Ministry of Education, Science, Sports, and Culture [Grant-in-Aid for Scientific Research (C) (No.24500079, No.24500258), Scientific Research (B) (No.26280115), Young Scientists (B) (No.26730155)] and Kawanishi Memorial ShinMaywa Education Foundation.

References:

[1] A. Okushi, S. Tokunaga, S. Matsumoto, and M. Nakamura, "SmaSho: Implementation and Evaluation of a Shopping Support Service Using Receipt Log," Asia-Pacific Symp. on Information and Telecommunication Technologies, Nov. 2012.

[2] S. Tokunaga, S. Matsumoto, and M. Nakamura, "Implementation and Evaluation of Consumer-Oriented Lifelog Service Using Daily Receipts," The 13th Int. Conf. on Information Integration and Web-based Applications & Services, pp. 337-340, December 2011.

[3] S. Tokunaga, A. Okushi, S. Saiki, S. Matsumoto, and M. Nakamura, "Consumer-Oriented ReceiptLog Service Platform for Effective Applications," Joint 7th Int. Conf. on Soft Computing and Intelligent Systems and 15th Int. Symp. on Advanced Intelligent Systems, pp. 398-403, December 2014.

[4] F. Chen and T. Ou, "Sales forecasting system based on Gray extreme learning machine with Taguchi method in retail industry," Expert Systems with Applications, Vol.38, No.3, pp. 1336-1345, 2011.

[5] A. M. Hughes, "Strategic Database Marketing," McGraw-Hill Inc., 1994.

[6] G. Bell and J. Gemmell, "Total Recall: How the E-Memory Revolution Will Change Everything," Dutton Adult, Sep. 2009.

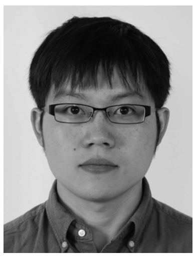
[7] O. H. Kieron, T. Mischa, and S. Nigel, "Lifelogging: Privacy and Empowerment with Memories for Life," Identity in the Information Society, Vol.1, No.5, pp. 155-172, 2008.

[8] MediaDriveInc., "Yasasiku Kakeibo," 2009.

[9] O. Zimmermann, V. Doubrovski, J. Grundler, and K. Hogg, "Service-oriented Architecture and Business Process Choreography in an Order Management Scenario: Rationale, Concepts, Lessons Learned," Companion to the 20th Annual ACM SIGPLAN Conf. on Object-oriented Programming, Systems, Languages, and Applications, OOPSLA'05, pp. 301-312, ACM, 2005.

[10] gooNews, "Danone Japan conducted an attitude survey for housewives. Have you experienced the forgotton purchase?," 2009.

[11] A. Jhingran, "Enterprise Information Mashups: Integrating Information, Simply," Proc. of the 32nd Int. Conf. on Very Large Data Bases, VLDB'06, pp. 3-4, VLDB Endowment, 2006.



Name:
Seiki Tokunaga

Affiliation:
Ph.D. candidate, Kobe University

Address:
1-1 Rokkodai, Nada, Kobe, Hyogo 657-8501, Japan

Brief Biographical History:
2013- Ph.D. candidate, Kobe University

Main Works:
• "How Should Remote Monitoring Sensor Be Accurate?," The 1st Int. Workshop on Reliability of eHealth Information Systems, pp. 31-36, October 2014.

Membership in Academic Societies:
• The Institute of Electrical and Electronics Engineers (IEEE), Student Membership

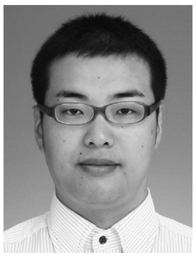
Name:
Akihiro Okushi

Affiliation:
NS Solutions Kansai Corporation

Address:
7-20-1 Fukushima, Fukushima-ku, Osaka-shi, Osaka 553-0003, Japan

Brief Biographical History:
2014- Joined NS Solutions Kansai Corporation

Main Works:
• "Sma-Sho: Implementation and Evaluation of a Shopping Support Service Using Receipt Log," Asia-Pacific Symposium on Information and Telecommunication Technologies, November 2012.



Name:
Sachio Saiki

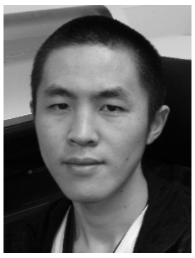
Affiliation:
Assistant Professor, Graduate School of System Informatics, Kobe University

Address:
1-1 Rokkodai, Nada, Kobe, Hyogo 657-8501, Japan

Brief Biographical History:
2009- Joined Kochi University of Technology
2013- Joined Kobe University

Main Works:
• "Dynamically Ratio Controlling Combined Type PtNLMS Algorithm," IEICE Trans. of A (Japanese Edition), Vol.J93-A(10), pp. 677-680, 2010.

Membership in Academic Societies:
• Institute of Electronics, Information and Communication Engineers
• The Institute of Electrical and Electronics Engineers (IEEE)



Name:
Shinsuke Matsumoto

Affiliation:
Assistant Professor, Graduate School of System Informatics, Kobe University

Address:
1-1 Rokkodai, Nada, Kobe, Hyogo 657-8501, Japan

Brief Biographical History:
2010- Received Ph.D. degree in Nara Institute of Science and Technology
2010- Joined Kobe University.

Main Works:
• "Service Oriented Framework for Mining Software Repository," Int. Conf. Software Process and Product Measurement, pp. 13-19, Nov. 2011.

Membership in Academic Societies:
• Institute of Electronics, Information and Communication Engineers
• Information Processing Society of Japan (IPSJ), IPSJ SIG Software Enginnering

Name:
Masahide Nakamura

Affiliation:
Associate Professor, Graduate School of System Informatics, Kobe University

Address:
1-1 Rokkodai, Nada, Kobe, Hyogo 657-8501, Japan

Brief Biographical History:
1999- Joined University of Ottawa
2000- Joined Cybermedia Center at Osaka University
2002- Joined Graduate School of Information Science at Nara Institute of Science and Technology
2007- Joined Graduate School of System Informatics, at Kobe University

Main Works:
• "Considering Impacts and Requirements for Better Understanding of Environment Interactions in Home Network Services," J. of Computer Networks, Vol.57, No.12, pp. 2442-2453, August 2013.

Membership in Academic Societies:
• Information Processing Society of Japan (IPSJ)
• Institute of Electronics, Information and Communication Engineers
• The Institute of Electrical and Electronics Engineers (IEEE)