A Home Context-Aware System with a Mechanism for Personalization of Service Providing

Hiroyuki Yamahara, Takanori Soma, Fumiko Harada, Hideyuki Takada
Ritsumeikan University
1-1-1 Noji-Higashi, Kusatsu, Shiga 525-8577, Japan
{yama, takanori}@de.is.ritsumei.ac.jp, {harada, htakada}@cs.ritsumei.ac.jp

Yukihiro Shimada
GOV Co., Ltd.
134 Minami-cho, Tyudoji, Shimogyo-ku, Kyoto 600-8813, Japan
shimada@go-v.co.jp

Hiromitsu Shimakawa
Ritsumeikan University
1-1-1 Noji-Higashi, Kusatsu, Shiga 525-8577, Japan
simakawa@cs.ritsumei.ac.jp

Abstract

We propose a home context-aware system which has a mechanism for personalization of service activation and context estimation. Personalization of service activation is to realize service activation along user intention. The proposed system can activate a variety of services along user intention by combining a system-active approach and a user-active approach. Because users choose activated services finally, the system can activate services even along user intention which cannot be inferred by computers. Personalization of context estimation is to realize setting values appropriate for each user to parameters used for context estimation in a system-active approach. The proposed system determines values appropriate for each user by utilizing statistical data of test users whose characteristics are similar to each user. Determination of individual values enables stabler context estimation than context estimation with values common to all users.

Keywords - home intelligent service; context awareness; RFID; threshold; behavior

1. Introduction

People would make mistakes in daily activities in homes. They sometimes leave their homes without closing windows and turning off an air conditioner. They go to bed without locking the front door. We are developing a home context-aware system which provides services to prevent users from facing dangers of making mistakes in homes and to make their life more comfortable. Before the users leave their homes, our system can warn them to lock windows and to turn off the air conditioner. The system can also lock windows and turn off the air conditioner instead of them. There have been already technologies which lock doors of house by remote control with a cell phone. However, these technologies cannot prevent dangers, because these technologies are user-active approaches and can be useful only in a case the users get aware of their mistakes by themselves. Moreover, it is difficult for users unfamiliar with computers, such as elderly people, to utilize their cell phones actively.

As an approach for solving these problems, there are studies of context-aware systems which provide services according to user context such as user behavior and user situation. Because the systems estimate user context, the systems can awaken users to something they are not aware of. In addition, because services are provided with a system-active approach, it can provide services to users unfamiliar with computers without any active operations of users themselves. Meanwhile, it is impossible for the systems to estimate every user context accurately. Therefore, this approach is at risk for providing services inappropriate for user situation because of misestimation by the systems. Existing studies of context-aware systems focus on improvement of estimation accuracy of user context, but they do not consider handling of inappropriate services provided based on misestimation.

Not to make users feel dissatisfied, home context-aware systems must have a mechanism for preventing inappropriate service providing based on misestimation. Moreover, the systems are expected to have a mechanism for activating a variety of services along user intention which cannot be inferred by computers. For example, the systems can infer a user leaves his home but cannot infer whether he goes away for a long time or for a short time. If he goes away for a long time, a service for turning off lights and air conditioners is useful. If he goes away for a short time, he may prefer a service for turning off lights but leaving air conditioners as it is. In this way, there can be sometimes different services according to detailed user intention, which cannot be inferred, even in a scene of leaving home. Useful systems should be able to activate a variety of services, which have different details even in a same scene, along detailed user intention. In this paper, to realize such service activation along user intention is referred to as personalization of service activation.

In addition, because home context-aware systems can be introduced into a variety of users who have different characteristics of behavior, context estimation methods of the systems must be stably effective for not some users but as many users as possible. Appropriate values of parameters used as criteria of judgment in the estimation methods vary among users. If values common to all users are used, estimation accuracies on some users become significantly low. It is preferable to determine the values of parameters individually. Values appropriate for each user can be determined by collecting sensor data acquired according to his daily activities for a long period and analyzing the data. However, service providing should be early started after introducing the systems into homes. Therefore, the values must be determined with a small number of data of each user which can be collected in a short period. In this paper, to realize setting values appropriate for each user to parameters used for context estimation is referred to as personalization of context estimation.

The system we previously presented[18] considers a mechanism for preventing inappropriate service providing, but does not enough consider a mechanism for these personalization. In this paper, we propose a home context-aware system with a mechanism for personalization of service activation and personalization of context estimation, which has been improved from the previous system.

In our system, short-range passive RFID tags are installed in a variety of objects such as a doorknob, a wallet, a wristwatch, a refrigerator in homes. A unique ID is stored in each tag. The user wears a ring-type RFID reader on his hand. Using this RFID system, the histories of objects touched by the user in his home are stored in a home server.

The proposed system personalizes service activation by combining a system-active approach and a user-active approach. Services are provided in this system as follows. First, specific user behavior is detected from kinds of touched objects and the order of the objects. Next, with the detection as a trigger, service candidates according to the detected user behavior are narrowed down and they are offered the user. At the same time, the service candidates are respectively mapped to objects around the user. These objects become switches for activating the offered services temporarily. The user can choose the objects mapped with service candidates from objects around him by himself. Finally, services are activated just by user's touching to these objects. In this way, the proposed system provides services along user intention by choosing activated services with a user-active approach from service candidates chosen with a system-active approach.

In addition, the proposed system has a mechanism for determining initial values of individual users for parameters used in a behavior detection method as a mechanism for personalization of context estimation in the system. The system utilizes statistical data on test users, which are acquired before introducing the system into the home of each user. Suppose the system is introduced into a user v. First, a small number of data of v is collected in a short period at the initial stage after introducing the system. Next, the data of v is compared with the data of test users, based on the concept of collaborative filtering. Finally, values appropriate for v are determined from data of test users who have characteristics similar to v.

The proposed system which has a mechanism for personalization of service activation and context estimation has the following advantages.

- By combining a system-active approach and a user-active approach, the system can activate a variety of services along user intention without losing an advantage that the system can awaken users to something they are not aware of.
- Determination of individual initial values for parameters in the detection method enables stable behavior detection by improving detection accuracies of users whose detection accuracies are low with values common to all users.

The remaining part of this paper is organized as follows. Section 2 presents problems on personalization of service activation, and Section 3 presents problems on personalization of context estimation in the proposed system. Section 4 shows the proposed system has a mechanism for personalization of service activation, with the flow of service providing in the system. Section 5 describes a method for determining individual values of parameters in the behavior detection method as a mechanism for personalization of context estimation in our system. Section 6 presents an experimental life space where the proposed system is implemented. In Section 7, we evaluate the usability of the touch-to-object interface for service activation to consider the possibility of a user-active approach which is a core of personalization of service activation. In Section 8, we evaluate accuracy of behavior detection and personalization of context estimation with the proposed system. Section 9 presents challenges for improvement of our system. Finally, we conclude this paper.

2. Problems on Service Activation

2.1. User's Final Decision of Services

The system for supporting daily life is required not to take actions which are against user intention because such actions make discontent. Therefore, it is necessary to improve the accuracy of context estimation for reducing inappropriate services caused by false estimation. However, it is practically difficult to achieve 100 percent accurate estimation along user intention at any given time with computers. Reducing inappropriate services is possible but it is impossible to eliminate such services completely. For example, suppose the system provides a service for turning off lights which a user forgets to turn off before he leaves his home. In a case he goes away for a long time, he wants to turn off lights. But in the case he goes away for a short time or a case a housemate is in his home, he may not want to turn off lights. It is not easy to discriminate such cases whose details are different. To provide services which are along user intention even in such cases, it is preferable that the user finally decides whether or not services should be activated. At the same time, the interface he uses to decide whether he activates services or not must not be complicated. It must be simple and costless so that users unfamiliar with computers can use intuitively without being conscious of computers.

2.2. Service Activation on the Spot

The interface which users use to decide activated services should be available regardless of user position. Suppose the system warns a user that lights in the living room are still on, after putting on his shoes on the front door when leaving his home. It is annoying for the user to go back to the living room, where a home server is located, to send a command for activating a service which turns off the lights to the system. He may rather turn off the lights directly by himself in the living room than activate the service after moving from the front door to the living room. The usefulness of the system which can be used only on a specified place is low. Users should be able to activate services on the spot according to their position.

2.3. Related Works on Service Activation

There are studies of easy-to-use interfaces to activate services with operation of users themselves without automatic activation of systems according to user context.

Nichols et al. have improved the interface of a cell phone used for remote control of home appliances [10]. However, it is difficult for users unfamiliar with computers to operate a cell phone. Tsukada et al. propose remote operation with finger gesture [16]. Although this interface does not make users conscious of computers, users may waste a long time to learn how to operate it because it is not intuitive. Riekki et al. propose an interface for activating services with RFID tags attached by symbol images, with which users can intuitively know the content of activated services [14]. However, users cannot activate services without moving to specified places to touch specified tags, because tags are fixed on specified places and services are fixed on the tags. In addition, although technologies of speech recognition are studied as an interface to activate services with speech of users [4], the technologies are not enough practical at present. More comfortable interfaces are necessary to activate services by users themselves.

3. Problems on Behavior Detection

3.1. Complexity of User Behavior

We address behaviors which can be triggers to provide services as user behavior in this paper. They are detected by observing a sequence of habitual activities of individual user. Suppose the user brushes his teeth, goes to the toilet, picks up a wallet, wears a wristwatch and opens the front door in order. By observing such a sequence of some habitual activities which are taken before leaving home, his leaving home can be detected. Suppose he opens the front door. It is difficult to judge whether he leaves his home only from one activity. He may go out for picking up a newspaper. Similarly in another example, his getting up can be detected by observing a sequence of activities taken right after getting up, such as he goes out of bed, stops an alarm clock, turns on lights and drinks water from the faucet. Note that kinds of habitual activities and habitual order of them depend on individual user.

The user does not always take same activities in same order every time. In the observed sequence of user activities, habitual order relation and non-habitual order relation are mixed. For example, before leaving home, there can be habitual order from "going to the toilet" to "picking up a wallet" and from "picking up a wallet" to "opening the front door" but there may be no habitual order relation between "picking up a wallet" and "wearing a wristwatch". In addition, sometimes rare activities such as "picking up an umbrella” in a rainy day are inserted into the activity sequence. By contraries, part of habitual activities may be also sometimes omitted. From such a complex sequence, it is necessary to extract characteristic kinds and order which represent individual habits indicating user behavior to achieve behavior detection.

Some existing studies propose methods for recognizing user motion such as “walking” and “standing up”[2, 9]. Other studies propose methods for recognizing simple activities such as “brushing teeth”, whose characteristics are similar among users[12, 13, 17]. User behaviors as triggers of service providing, such as “leaving home”, need to be detected by observing a sequence of such activities. Logan et al.[8] and Huỳnh et al.[5] study methods for recognizing behaviors such as “leaving home” called as *high-level activities*. Basically, these existing methods aim to achieve classification or labeling of activities to identify user activity on a certain period of time. Compared with these, behaviors in this paper are not recognized but should be detected as triggers of service providing. Because the behavior handling considered in this paper is handling which is reactive to user behavior, it is a different target from existing methods.

3.2. Deadline for Providing Services

Some services have definite deadlines of providing them, while others have no definite deadline of providing them.

Examples of the former are services provided when the user leaves his home or he goes to bed. Suppose he is warned that he does not have wallet after he leaves his home. He must go back to inside his house to pick up his wallet. The value of the service provided after leaving home is significantly lower than that before his leaving. The deadline of providing services is the instant the user goes out of his house through the front door. Similarly, suppose the user is warned that the front door is not locked when he goes to bed. The deadline of such services provided on going to bed is the instant he goes sleep in his bed. To provide high value services for the user, his behaviors such as leaving home and going to bed must be detected before the deadline.

Examples of the latter are reminder services provided when the user gets up or comes home. Suppose the service, which reminds him one day schedule and what to do on the day, is provided with detection of his getting up as a trigger. Such service can prevent his mistakes. Such reminder services triggered by detection of getting up or coming home have no definite deadline of providing them, nonetheless it is preferable to provide the services within the time the user is doing a series of activities right after getting up or coming home.

Behaviors which can be triggers of service providing must be detected before the deadlines of services.

3.3. Template Matching

In home context-aware systems, data of user behavior is acquired online from sensors. This paper refers to the sensor data as *behavior log*. Generally, behavior of a user is inferred by template matching with behavior logs. The flow of template matching is as follows.

1. A certain amount of behavior logs of the user are collected.
2. A template which represents characteristics of user behavior is created by statistical analysis of the collected behavior logs. The behavior logs used for creating the template are referred to as *sample behavior logs*. The created template is referred to as a *matching template*.
3. User behavior is inferred by checking the degree of conformity when matching current behavior logs, which are acquired online from sensors, with the matching template. Matching is repeated at some kind of specified timing. The behavior logs which are matched with the matching template is referred to as *match-target behavior logs*.

In our system, user behavior is detected based on template matching. The system cannot detect user behavior until the matching template is created after introducing the system into homes. That means users are not provided services. If it takes a long time to create the matching template, users are dissatisfied with waiting for a long time until the start of service providing. Sample behavior logs of individual user need to be collected in a short period to start providing services early. That is, the system should create an initial matching template with a small number of individual sample behavior logs which can be collected in a short period to be accepted by users.

Behavior logs are classified into two types to a matching template. There are *true cases* and *false cases*. Suppose there is a matching template to detect a behavior of leaving home. True cases are behavior logs in scenes where a user is leaving home. False cases are behavior logs in other scenes.

3.4. Setting of Initial Threshold Values

Some parameters should be determined appropriately in a method for behavior detection to achieve high detection accuracy. In particular, the following two threshold values have an impact on the detection accuracy. One is a threshold value used for creating a matching template. The threshold value is used to extract characteristics from the results of statistical analysis on sample behavior logs. If an inappropriate value is set to the threshold, the system cannot extract appropriate characteristics for accurate detection. The other one is a threshold value used for matching match-target behavior logs with matching templates. The threshold value defines threshold of the degree of conformity between them. If the degree of conformity is more than the threshold value, the logs are regarded as conformable logs to the matching template. If an inappropriate value is set to the threshold, the system cannot successfully detect user behavior to be detected or mistakenly detect user behavior not to be detected.

Because kinds of habitual activities and order of them vary with users, appropriate values depend on user behavior. Therefore, threshold values are preferable to be individually determined for each user. If the values are determined after many individual behavior logs are collected, appropriate values are found by simulating template matching with the collected behavior logs. However, the system can use only a small number of individual sample behavior logs to determine initial threshold values because service providing should be early started after introducing the system into homes. It is difficult to determine threshold values appropriate for individual user only with a small number of sample behavior logs. In addition, it is preferable to utilize both true cases and false cases to determine appropriate values. False cases from which true cases are hardly distinguished are particularly required. Even true cases are small when initial threshold values should be determined. It is impossible to use enough false cases useful for determining appropriate threshold values.

In a basic determination method, developers of a system or experts of the system determine the initial values common to all users before introducing the system into homes without sample behavior logs of individual users, because it is difficult to determine initial threshold values only with a small number of individual sample behavior logs. The determination method uses data of test users to determine the common values. Having a system used by some test users on a trial basis, many sample behavior logs of individual test users are collected. These logs include both true cases and false cases. By simulating template matching with the logs individually, the experts analyze relativity between change of detection accuracy and changes in threshold values. Finally, common threshold values are determined so that detection accuracy is averagely high for test users. The values are used as initial threshold values common to all users after introduction of the system. However, because there are some users whose appropriate threshold values are different from the common threshold values, detection accuracy varies with users. Common threshold values cannot achieve high detection accuracy for some users.

It is necessary to improve detection accuracy of some users whose detection accuracy is low with common threshold values, by determining appropriate initial threshold values.

3.5. Related Works on Setting Threshold

There are several approaches to determine appropriate threshold values in a variety of fields. In image processing, a determination method of a threshold used for extracting a specific area from a target image has been proposed [6]. This method can be used only if both parts to be extracted and parts not to be extracted exist together in a recognition target. The issue of behavior detection does not meet such a condition, because behavior detection in this paper considers whether a match-target behavior log conforms to a matching template or not. This approach in image processing cannot be applied to the issue. In other approaches, Support Vector Machines and boosting have been used for text categorization [3, 15], and Hidden Markov Model is used for speech and gesture recognition [1, 11]. These approaches can determine appropriate threshold values under the assumption that the approaches can collect and analyze many samples of a recognition target or many samples of others which have similar characteristics to samples of the recognition target instead. However, initial threshold values must be determined under the constraint of a small number of sample behavior logs for creating a matching template. In addition, because characteristics of user behavior in homes are different among individual users, behavior logs of other people other than a user cannot be used as sample behavior logs of the user. Although these approaches can be used for learning appropriate threshold values after many personal behavior logs have been collected, these approaches cannot be used for determining initial threshold values appropriate for individuals.

In a field of behavior recognition, most existing studies do not discuss how to determine initial threshold values. They use given values or values which are determined by analyzing many behavior logs.

4. Personalization of Service Activation

4.1. System Overview

Figure 1 shows an overview of the proposed system. This system is composed of RFID-tagged objects, a wearable RFID reader and a home server. The RFID reader reads tag-IDs of objects and sends tag-IDs to the server every time the user touches each object. In the server, the histories of touched objects are stored as behavior logs. After introducing the system into individual home, a certain time period is used for collecting behavior logs of individual user. Services are not provided while that period.

With the collected behavior logs as sample behavior logs, the Matching Template Creator creates matching templates which represent characteristics of user behavior. A matching template is created for every behavior such as leaving home and going to bed. For example, in a case

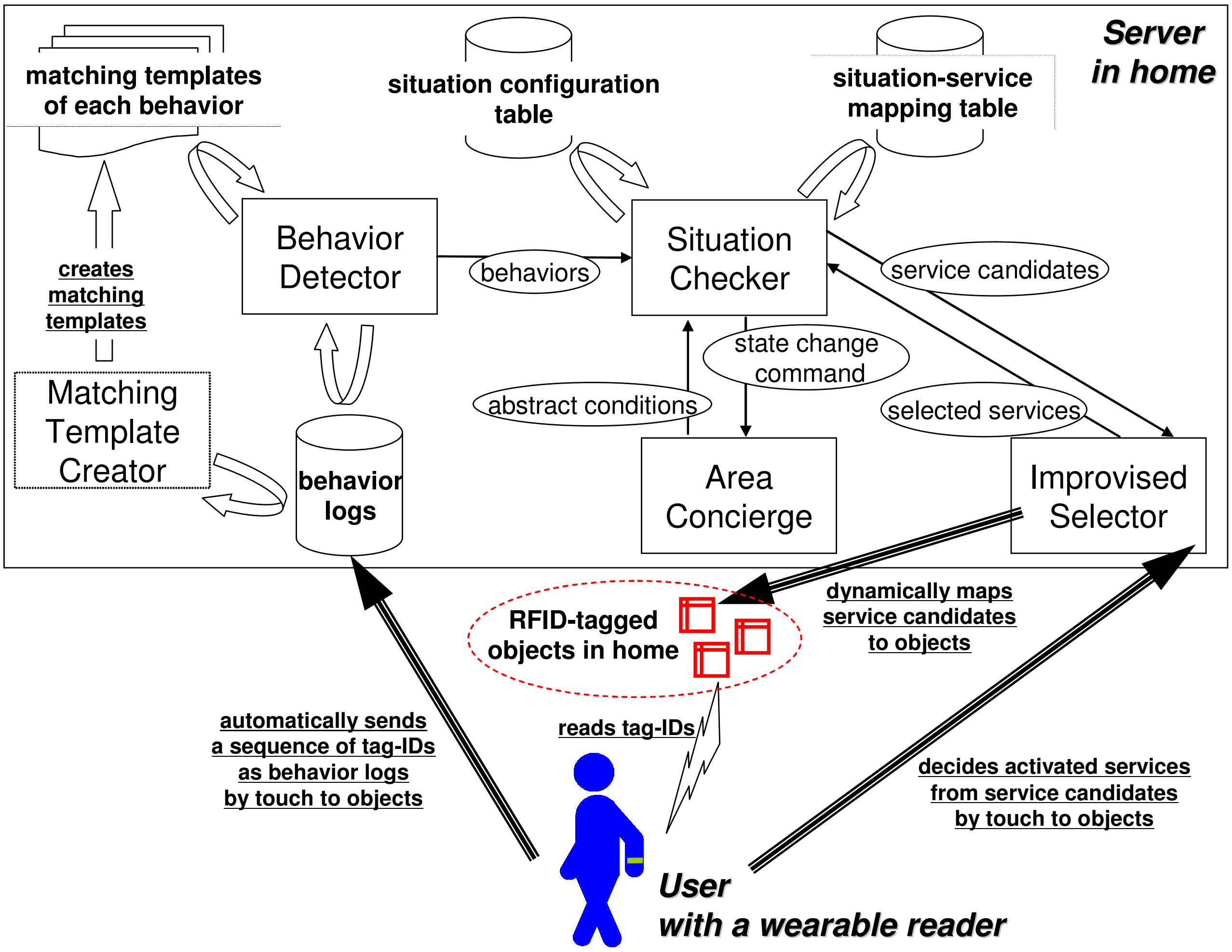

Figure 1. The system with combination of a system-active approach and a user-active approach.

of leaving home, the system shows only behavior logs in which the user touched objects of great relevance to leaving home such as doorknob of the front door and let him select behavior logs of true leaving home. Therefore, the system can exactly use behavior logs of leaving home to create a matching template representing characteristics of leaving home adequately.

After creating matching templates of each behavior, the system starts providing services. First, the Behavior Detector detects behaviors of triggers by matching behavior logs acquired online according to user activities with matching templates every time the user touches something. The Behavior Detector informs the Situation Checker of detected behaviors.

In this system, we refer to a set of conditions to provide services as a $situation$. A situation σ is defined as follows.

$$\sigma = \{b, p, e_1, e_2, ..., e_i, ..., q_1, q_2, ..., q_i, ...\}$$

Here, b is a user behavior to be detected, such as leaving home. p is the user position. e_i denotes the state of an object i which exists in the home, such as "the front door is locked." The variety of the states depends on kinds of sensors combinated to our system. q_i denotes the position of the object i, such as "a wallet is in the bedroom." A variety of situations are defined in the $situation\ configuration\ table$.

Referring to the situation configuration table, the Situation Checker checks whether any situations are happening or not. For example, this means even if the system detects the behavior of going to bed when the front door is already locked, a service for warning of unlocked front door is not provided. The states of conditions except user behavior b are always recognized by the Area Concierge which manages and controls sensors and actuators in cooperation with a home-network. The Area Concierge informs the Situation Checker of changes of the states. If there are situations which are happening, the Situation Checker searches services appropriate for the situations by referring

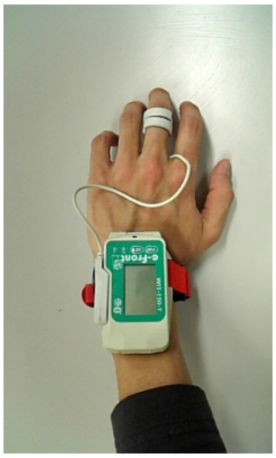

Figure 2. The ring-type RFID reader.

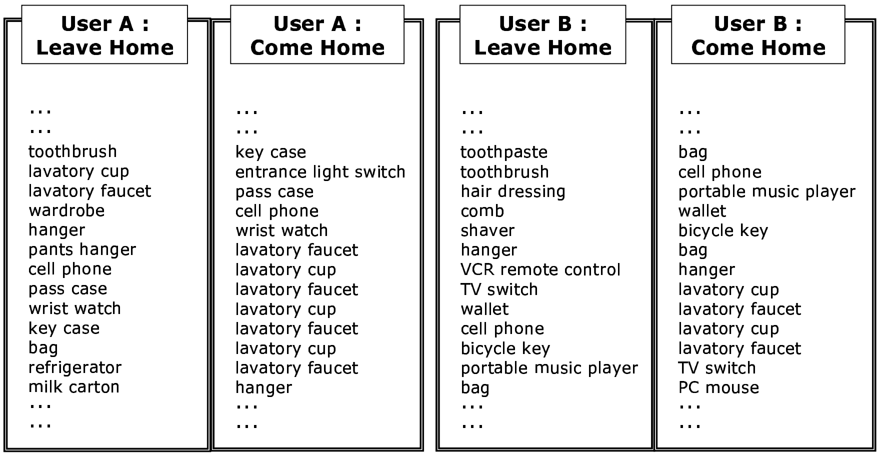

Figure 3. Examples of behavior log.

to *situation-service mapping table* and informs the Improvised Selector of the services as service candidates.

The Improvised Selector offers the user the service candidates for activating and maps each service candidate to one of objects around him. He chooses and can activate desirable services from the service candidates, just by touching to the objects mapped from the desirable services.

Service candidates are selected by a system-active approach which starts the process for providing services without any particular operation of the user. In addition, combining a user-active approach that the user himself finally decides activated services, our system prevents providing inappropriate services and also achieves service providing along user intention which it is not easy to infer. Of course, depending on provided services, our system can automatically activate the services by omitting the final decision of the user.

4.2. Matching Templates

We have studied how to create matching templates and how to detect user behavior with the templates. In Section 4.2 and Section 4.3, we briefly describe our studying detection method[19, 20]. Figure 3 shows actual behavior logs of two users, which have been recorded using a ring-type RFID reader shown in Figure 2. These are parts of behavior logs of two scenes which are before leaving home and after coming home. In the same scene, kinds of habitual activities and the order of them are different among individual users. In addition, comparing each user's log of leaving home to log of coming home, it is found that a user touches different kinds of objects or touches the same objects in a different order in different situations.

We represent characteristics of a sequence of habitual activities with kinds of activities such as "brushing teeth" and the order of them such as "the user wears his clothes after he brushes his teeth". Because the objects touched by the user significantly indicate kinds of activities and the order of them, our system characterizes user behaviors with kinds of touched objects and the order of touched objects.

With sample behavior logs which are histories of touched objects, our system creates a matching template represented by a set of ordered pairs which show the order relation of contact of a user to objects.

The flow to create a matching template of the user is shown in Figure 4 with an example of a matching template in a scene of leaving home. First, w cases of behavior logs of leaving home are collected as sample behavior logs. The deadline to provide services is the instant the user touches a doorknob of the front door in order to go outside house. We must create a matching template with which the system can detect that the user is leaving home by the instant. Therefore, each sample behavior log is a record of t_l minutes just before the user touches a doorknob of the front door. The time length t_l minutes of a sample behavior log is predetermined. If m objects are sequentially touched in a behavior log l, then l is represented as a conjunction $\{o_1, o_2, \dots, o_i, \dots, o_m\}$, where, $o_{i-1} \neq o_i (1 < i \leq m)$. Second, all ordered pairs between two objects are enumerated from all of collected sample behavior logs. If an object o_j is touched after an object o_i is touched, then an ordered pair p is represented as $\{o_i \rightarrow o_j\}$, which includes a case of $o_i = o_j$. For example, ordered pairs enumerated from a behavior log $\{o_1, o_2, o_3\}$ are $p_1 : \{o_1 \rightarrow o_2\}$, $p_2 : \{o_1 \rightarrow o_3\}$, $p_3 : \{o_2 \rightarrow o_3\}$. Next, the occurrence of each ordered pair is counted up as occurrence count. The occurrence count means not the amount of the number of times that each ordered pair occurred in a sample behavior log, but the number of sample behavior logs including each ordered pair. For example, if an ordered pair occurs in all sample behavior

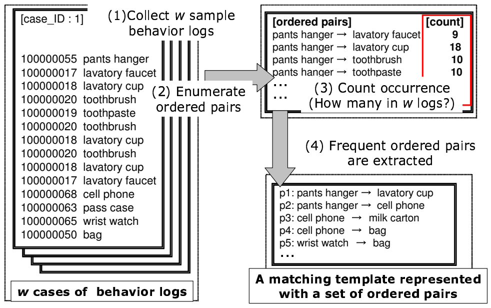

Figure 4. How to create a matching template.

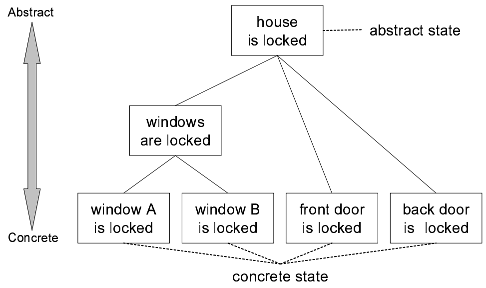

Figure 5. Management of abstract state.

logs, the occurrence count of the ordered pair is w. Finally, ordered pairs where ratio of occurrence count to w is more than e are extracted as a matching template Π. e is a threshold for extracting frequent ordered pairs from enumerated ordered pairs. e is referred to as the $extraction\ threshold$.

Two types of ordered pairs composing a matching template are extracted above. One is typified by $\{toothpaste \rightarrow toothbrush\}$. Both a toothpaste and a toothbrush are touched when the user brushes his teeth. This type of an ordered pair represents kinds of habitual activity. The other is typified by $\{toothpaste \rightarrow pants\ hanger\}$. We cannot guess an activity in which the user touches both of a toothpaste and a pants hanger. This type of ordered pair indicates a habitual order of activities, such as the user wears pants after brushing his teeth habitually. Our system can create a matching template which represents characteristics of user behavior by combining two types of ordered pairs.

4.3. Detection of User Behavior

Matching templates are matched with the current behavior log of time length t_l, which is acquired online, every time the user touches objects. Let a set of ordered pairs in the match-target behavior log be Θ. Our system calculates the degree of conformity c of the match-target behavior log to a matching template Π_b for detecting a behavior b with the following formula.

$$c = |\Theta \cap \Pi_b| / |\Pi_b|$$

Here, d is a threshold of the conformity c. If $c \geq d$ then b is detected. d is referred to as the $detection\ threshold$.

In our system, a matching template is composed of ordered pairs which often occurs before the deadline of service providing. Such ordered pairs are composed of objects which are touched with high probability before the deadline. Therefore, because the conformity c significantly increases as the deadline approaches and it becomes more than the threshold d, b is prone to be detected before the deadline.

4.4. Situation Check

The Area Concierge defines areas such as the entrance of home, a kitchen and a bedroom in homes hierarchically and manages states of objects, position of the user and position of objects in each area. Also, the Area Concierge can check states about weather, earthquake, and so on by cooperation with outside public sensors. As shown in Figure 5, states are hierarchically managed from concrete level to abstract level. At the most concrete level, each state is individually managed with each sensor, for example, "the window A is locked and the window B is locked". The Area Concierge also manages at more abstract levels such as "windows are locked". These hierarchical relations are defined as $concrete\text{-}to\text{-}abstract\ conversion\ rules$. By this way, there is an advantage that it is not necessary to redefine rules at abstract levels when the number of sensors are changed. We only have to redefine part of rules at the most concrete level. This means that our system can be introduced a variety of different homes by customizing only rules at concrete levels along individual user's home because rules at abstract levels can be defined in advance as rules common to all users. Because it is impossible to represent all states in one hierarchical structure, the Area Concierge manages with some kinds of hierarchical structures. In these hierarchical structures, a state at the most abstract level is referred to as an $abstract\ state$. A state at the most concrete level is referred to as a $concrete\ state$. The Area Concierge informs the Situation Checker of an abstract state and concrete states in the following cases.

- a case the Area Concierge received an inquiry about states of something from the Situation Checker
- a case an abstract state changed, which is not a case just concrete states changed

In addition, the Area Concierge changes specified states with actuators when received a command to change states of something from the Situation Checker. For example, when received a command "Lock up the house", the Area

```xml
<SituationConfiguration id="unlocked_unsavedPower_leaveHome"
                    name="unlocked_and_unsavedPower_on_leavingHome">
  <Group logicalOperator="and">
    <Behavior id="LeaveHome" name="LeaveHome"/>
    <Group logicalOperator="or">
      <Condition id="lockExceptEntrance"
              name="locksExceptEntranceAreLocked_true" status="false"/>
      <Condition id="light"
               name="lightsAreOff_true" status="false"/>
      <Condition id="gasValve"
               name="gasValvesAreClosed_true" status="false"/>
      <Condition id="electricAppliance"
              name="electricAppliancesAreOff_true" status="false"/>
    </Group>
  </Group>
</SituationConfiguration>
```

Figure 6. An example of XML description in situation configuration table.

```xml
<sst:SituationService id="unlocked_unsavedPower_leaveHome"
                name="unlocked_and_unsavedPower_on_leavingHome">
  <Announce>Are you leaving now?</Announce>
  <Inquire type="xor" wait="60">
    <Announce>How long?</Announce>
    <Case>
      <Announce>In a case of long time,
               do you lock up house and save powers?</Announce>
      <Service id=" longLeaveHome"
            name="lockupHouse_and_savePower_on_leavingHome_for_longTime">
        <Announce>All locks are locked.</Announce>
        <Announce>All gas valves are closed.</Announce>
        <Announce>All lights are turned off.</Announce>
      </Service>
    </Case>
    <Case>
      <Announce>in a case of short time</Announce>
      <Service id=" shortLeaveHome "
            name="lockupHouse_and_savePower_on_leavingHome_for_shortTime">
        <Announce>All locks are locked.</Announce>
        <Announce>All gas valves are closed.</Announce>
      </Service>
    </Case>
  </Inquire>
</sst:SituationService>
```

Figure 7. An example of XML description in situation-service mapping table.

Concierge locks windows, the front door and all of others which are related to the command, with referring to the hierarchy from top to bottom.

The position information of the user is acquired by connecting the Area Concierge with medium-range RFID readers and other sensors. It is possible to change kinds of sensors and the number of sensors. The position information of objects shows which area the objects exist in. The position information of objects should be updated in response to move of the objects. However, every object is currently linked with a specific area in advance because this part of our system remains in a development stage.

With behavior detection as a trigger, the Situation Checker refers to the situation configuration table, whose example is shown in Figure 6. The Situation Checker checks whether all conditions of situations composed of the detected behaviors are satisfied or not with the information from the Area Concierge. In the situation in Figure 6, conditions of "lockExceptEntrance", "light", "gasValve" and "electricAppliance" are checked. If there are situations whose all conditions are satisfied, the Situation Checker refers to situation-service mapping table for finding services related with the situations. Such services becomes service candidates. Figure 7 shows an example of a situation-service mapping table. In this example, two services can be service candidates.

4.5. Service Activation by Touch-to-Object

The Improvised Selector maps service candidates to objects, which exists in area where the user is, on a one-to-one basis. To decide objects mapped with service candidates, the Improvised Selector makes the user touch objects around him. Because the user decides objects which he can touch easily as switches for choosing services, it is easy for him to understand which objects are mapped with service candidates. Alternatively, the Improvised Selector can automatically decide objects mapped with service candidates if the user wants to omit to decide the objects. Then the Improvised Selector offers contents of service candidates and objects mapped by them. There are some possible ways to offer the user service candidates. Currently, offering is executed by voice announcement. The user chooses desirable services from service candidates by touching to mapped objects. The user's choice is informed to the Situation Checker and the Situation Checker activates the chosen services. If contents of activated services are related with control of actuators such as lights, the Situation Checker sends commands for controlling actuators to the Area Concierge.

Our system can activate meticulous services along user intention and feeling, which cannot be inferred by the system, by defining multiple services or combination of services, which may be provided when detected one behavior, as service candidates in the situation-service mapping table. Suppose the system detects that a user is leaving his home. At that time, it is not easy to exactly infer his intention, whether he goes away for a long time or for a short time, by computers. For example, our system offers him the following two services.

1. The system locks all windows, turns off all lights, and turns off all air conditioners.
2. The system locks all windows, turns off all lights, but leaves air conditioners as it is.

If he goes away for a long time, he wants to activate the first one. If he goes away for a short time, he wants to activate the second one. In this way, our system provides meticulous services along user intention.

In addition, the system can prevent providing inappropriate services based on false detection. Suppose, before a user goes to bed, he brushes his teeth, goes to the toilet and turns the doorknob of the front door for checking that it is locked. At that time, if the system faultily detects that he is

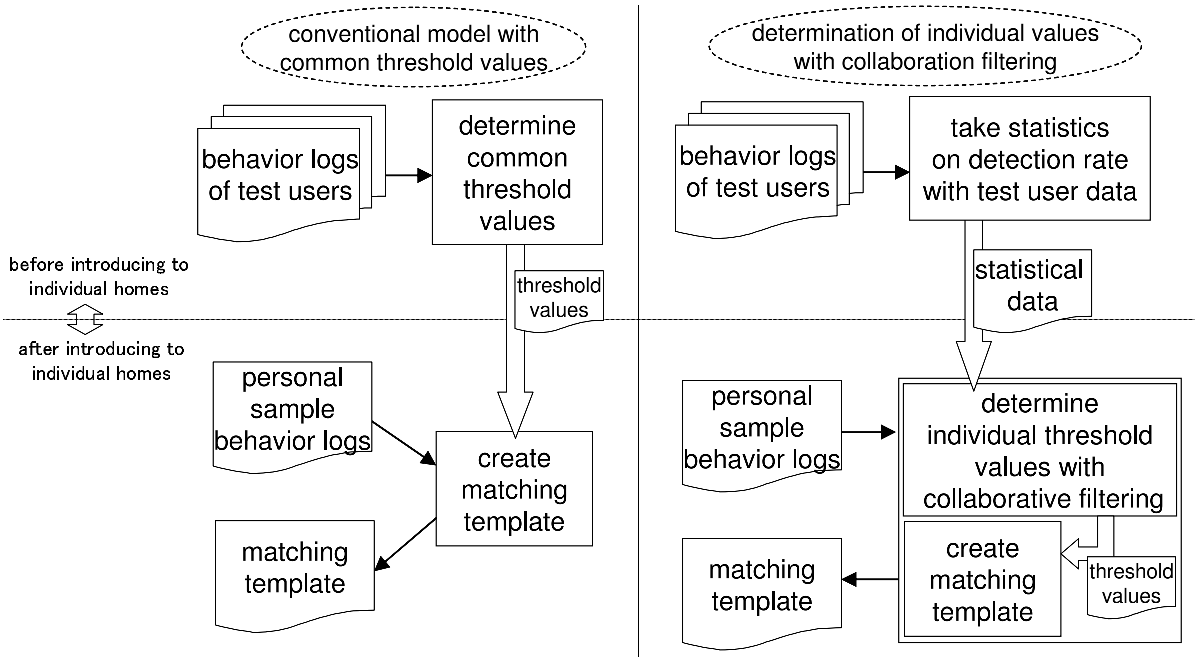

Figure 8. Dynamic determination model with collaborative filtering.

leaving his home and activates a service for turning off an air conditioner, the inappropriate service is not welcome for him. In such a case, our system prevents the inappropriate service if himself finally does not activate the service.

The interface for choosing activated services is expected to be user-friendly so that any users can operate with no stress. In our system, the touch-to-object interface has the following advantages.

- Even users unfamiliar with computers can easily decide whether they activate services or not just by touch-to-object which is an simple and costless operation without being conscious of computers.
- Users can activate services on the spot without moving to specific places because services are dynamically mapped to objects in an area where users are.

5. Personalization of Behavior Detection

5.1. A Model for Utilizing Test User Data

Behavior detection is the most important for context estimation in the proposed system. Our behavior detection method has two parameters whose values should be determined for individual users. They are the extraction threshold and the detection threshold. The proposed system determines initial threshold values dynamically for each user, unlike the conventional model which uses fixed common threshold values. In this paper, a user whose threshold values are determined is referred to as a *target user*. Figure 8 shows the conventional model on the left side, and our model which determines initial threshold values for individuals on the right side. Our model acquires a rule to individually determine the threshold values for each matching template of the target user from the statistics on data of test users. The horizontal center line shows a partition of the two phases for introducing a home context-aware system to the home of the target user. The upper portion is the phase before introducing the system which is referred to as the *development phase*. The lower side is the phase after introducing the system which is referred to as the *operation phase*.

As shown in Figure 8, the conventional model determines common threshold values at the development phase. First, the model collects behavior logs of test users. Next, for every test user, the model repeatedly creates a matching template with the logs, while matching the logs with the matching template based on cross validation. Analyzing the result of detection accuracy on the matching, the model determines the threshold values with which detection accuracy averaged for all test users is the highest. At the operation phase, the model creates an individual matching template with personal behavior logs. However, the threshold values are common irrespective of the target user. In this conventional model, detection accuracy can be low because of differences between common values and the best values of the target user.

To dynamically determine apprppriate threshold values for individuals, it is preferrable to acquire knowledge from personal behavior logs of the target user. However, it is difficult to determine appropriate values only with a small number of personal behavior logs.

Considering the similarity between the target user and each test user, our determination method determines threshold values of the target user, based on the data of test users whose characteristics are similar to characteristics of the target user. To calculate the similarity between the target user and each test user, the method focuses on the average number of ordered pairs composing matching templates of each user. Values of the extraction threshold and the detection threshold are determined by estimating a position of the target user on a feature space, which is composed of behavior detection accuracies and the average number of ordered pairs on every test user, based on an idea of collaborative filtering. As shown in Figure 8, first, our method takes statistics on the average number of ordered pairs of each test user and detection accuracy before introduction of the system. A feature space of this statistical data corresponds to the rule for determinating threshold values. After that, values of two thresholds are determined at the same time by executing collaborative filtering with the statistical data and a small number of personal behavior logs of the target user when a matching template is created after introducing the system into the home of the target user.

5.2. Collaborative Filtering

Collaborative filtering is a process for automatically estimating unknown information of a target user with some known information of him and known information of other users. Here, the informations mean features such as tendency and taste. They have to be able to be collected with a form which can be expressed on the numeric axes. First, the similarity between the target user and each other user is calculated with known information on both the users. Next, unknown information is estimated using known information of other users who are similar to the target user. This estimation is utilized for recommendation or personalization, as used in Amazon.com [7].

Our determination method calculates the similarity between the target user and each test user on a feature of the average number of ordered pairs composing matching templates and estimates behavior detection accuracy of the target user by utilizing statistical data on detection accuracy of test users on every setting of thresholds. The estimation enables to determine threshold values with which detection accuracy is the high.

5.3. Estimation of Initial Threshold Values

With an example of a matching template of leaving home of a target user v, this section describes how to determine threshold values with collaborative filtering. At the development phase, the following steps are executed to calculate detection accuracy of each test user on every setting of two thresholds. Here, w is a given value common to all users.

1. Collect behavior logs of leaving home as true cases and also behavior logs other than leaving home as false cases.
2. Select w true cases as sample behavior logs and create w matching templates with each setting of the extraction threshold value $e = 1/w,\ 2/w,\ ...,\ w/w$, using the w true cases. Here, the number of ordered pairs composing each matching template is recorded.
3. With all settings of the detection threshold d from 0.01 to 1.00, match all true cases and all false cases with the w matching templates.
4. Repeat k times from step 2 to step 3, using new matching templates created with a new combination of w true cases every time.

With these steps, $true\text{-}positive\ rate$ (TPR), $true\text{-}negative\ rate$ (TNR), and $half\ total\ true\ rate$ ($HTTR$) are calculated on every setting of thresholds per combination of w true cases by taking statistics on all results of the matchings. The number of threshold settings is $w \times 100$. Here, we explain these three rates with an example of detection of leaving home. TPR means the rate which our detection method successfully detects each subject's leaving home by the deadline, when matching behavior logs of leaving home with matching templates of leaving home. On the other hand, TNR means the rate which the detection method does not detect their leaving home, when matching behavior logs except leaving home with matching templates of leaving home. It is desirable that both TPR and TNR are high. HTTR is an average of TPR and TNR.

After the above steps, the followings are calculated.

- the averate number of ordered pairs composing k matching templates created on each setting of the extraction threshold e
- the average HTTR value on each combination of the setting of the extraction threshold e and the setting of the detection threshold d

These are statistical data which show characteristics of each test user. Next, threshold values of user v are determined when his matching template is created, by collaborative filtering with the statistical data. Figure 9 shows an example

	number of ordered pairs on each extraction threshold value					detection rate as statistical data calculated with test user data on combination of each extraction threshold value and each detection threshold value													
	e=0.2	e=0.4	e=0.6	e=0.8	e=1.0	e = 0.2			···	e = 0.8							e = 1.0		
						d = 0.01	···	d = 1.00	···	d = 0.01	···	d = 0.62	d = 0.63	d = 0.64	···	d = 1.00	d = 0.01	···	d = 1.00
test user u_1	130	53	15	13	8	53%	···	50%	···	67%	···	74%	74%	74%	···	67%	69%	···	70%
test user u_2	300	133	51	39	33	52%	···	50%	···	61%	···	81%	81%	82%	···	71%	67%	···	76%
test user u_3	211	175	121	97	79	54%	···	50%	···	67%	···	96%	96%	96%	···	71%	72%	···	86%
test user u_4	118	100	88	71	50	58%	···	50%	···	54%	···	86%	86%	87%	···	64%	54%	···	84%
test user u_5	300	177	142	62	22	50%	···	50%	···	55%	···	59%	60%	60%	···	57%	54%	···	55%
test user u_6	129	118	35	31	29	53%	···	50%	···	53%	···	98%	97%	96%	···	65%	54%	···	82%
test user u_7	203	199	164	121	112	53%	···	50%	···	53%	···	99%	99%	99%	···	54%	51%	···	62%
user v	**245**	**184**	**86**	**79**	**64**	$E_{1,1}$	···	$E_{1,100}$	···	$E_{4,1}$	···	$E_{4,62}$	$E_{4,63}$	$E_{4,64}$	···	$E_{4,100}$	$E_{5,1}$	···	$E_{5,100}$

statistical data from test user data

estimated target

data for calculation of user correlation

selected value

Figure 9. Estimation of values for a target user by collaborative filtering with test user data.

of the statistical data and the data of user v, which are used for collaborative filtering. In the example, w is 5. Rows from "test user u_1" to "test user u_7" are the statistical data from above calculation.

Information of how many ordered pairs compose each matching template on each setting of e and information how high detection accuracy is brought with each combination of the setting of two thresholds are obtained from the statistical data. On the other hand, there is only information, obtained from a small number of personal sample behavior logs, of user v when his initial matching template is created. As shown in the bottom row of Figure 9, the determination method utilizes the number of ordered pairs composing each matching template which is created on each setting of e with personal sample behavior logs of user v. At this time, it is unknown how high detection accuracy is brought with each matching template of user v. By collaborative filtering, first, user correlation between user v and each test user is calculated with 5 values of the number of ordered pairs, which are known information of both user v and each test user. The user correlation shows the similarity between users. Second, HTTR values of matching template of user v is estimated with HTTR values of rows from test user u_1 to test user u_7, based on the calculated user correlations. From $E_{1,1}$ to $E_{5,100}$ show the estimated HTTR values. Here, $E_{i,j}$ means the estimated HTTR value on the setting where e = i/w and d = $j/100$. After the estimation, the determination method selects one estimated value from all estimated values as follows.

1. Select the maximum estimated value.

2. If more than two estimated values are selected in the above step, pick up the longest sequence of the maximum estimated values and select the estimated value in the center of the sequence.

3. If more than two estimated values are selected in the above step, select an estimated value on the smallest value of e from the remaining candidates.

Suppose three rows of $\{E_{4,62}, E_{4,63}, E_{4,64}\}$ are the longest sequence of the maximum estimated values in Figure 9. In such a case, $E_{4,63}$ in the center of three values is selected. Finally, because $E_{4,63}$ is the estimated value on e = 0.8 and d = 0.63, these values are determined as threshold values for user v.

6. An Experimental Life Space

6.1. Implementation

We have built an experimental life space by implementing the proposed system. Figure 10 shows the life space. The life space is Japanese style house, where users take off their shoes when they go into the house. The life space is composed of some areas such as entrance, living, kitchen, and dining. Real furniture and electric appliances are equipped. About 1000 passive RFID tags whose frequency is 13.56 MHz are installed in 159 objects in all areas. Figure 11 shows examples of the objects. Behavior logs of users who wear RFID readers on their hand can be collected in this life space. In addition, middle-range RFID readers are installed into this life space as devices for acquiring position information of users.

6.2. Data Collection for Experiments

We have collected behavior logs in the experimental life space to conduct experiments for evaluation of the proposed system, with 8 experimental subjects "A" to "H". Target behaviors to be detected are 4 behaviors which are leaving home, coming home, getting up and going to bed. Previously we had a questionnaire with 17 men and 4 women to decide the target behaviors. As a result, above 4 behaviors

Figure 10. An experimental life space.

Figure 11. Objects installed with RFID tags.

have been selected because user's mistakes are effectively prevented by services triggered by detection of these 4 behaviors.

Prior to collect behavior logs as data for the experiment, we have had a survey questionnaire for 2 weeks to confirm that users have habitual characteristics of 4 behaviors. In the questionnaire, subjects have described complete details of kind of objects they touched and the order of touched objects during 10 minutes before leaving home, after coming home, after getting up and before going to bed. We have confirmed the following things from their description.

- Some activities are interleaved in a 10 minutes sequence of activities.
- Subjects do not finish a sequence of activities within 10 minutes after coming home and after getting up.

Although there are no definite deadlines of providing services after coming home and after getting up, we must decide the deadlines to clear up success or failure of behavior detection in experiments. In view of the above confirmation, we have set the deadline to “10 minutes after subjects open the front door and enter the house” when coming home. Similarly, we have set the deadline to “10 minutes after subjects get out of bed” when getting up.

As experimental data, we have collected behavior logs of 4 behaviors in a database of a server by sensing actual objects which 8 subjects have touched in the experimental life space shown in Figure 10 and Figure 11. We have had subjects touch the objects with being aware of the position

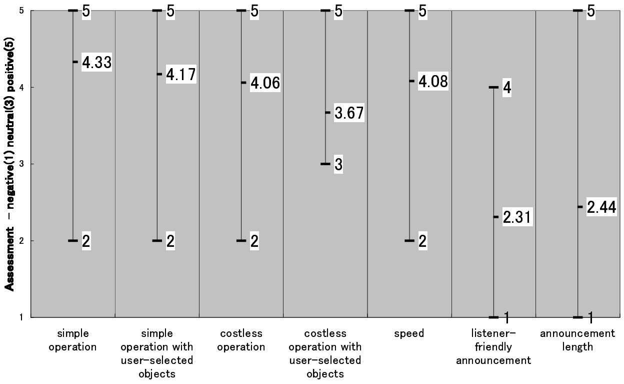

Figure 12. 5-point assessment of the touch-to-object interface.

of tags on the objects. However, we have not forced subjects to keep touching the objects so that tag-IDs are exactly read out. Therefore, though the reading accuracy of the reader is not bad, part of touched objects may be not recorded as behavior logs.

7. Usability of the Touch-to-Object Interface

7.1. Experiment for the Interface

We have conducted an experiment to evaluate the usability of the touch-to-object interface for deciding services to be activated, which is indispensable to actualize service activation by users in a user-active approach of our system. In the experiment, we ask subjects to decide activated services from service candidates by touching actual objects in the above experimental life space before leaving home and before going to bed. The service candidates are offered with voice announcement. The situation configuration table and the situation-service mapping table have been predefined. Each subject experiences both a case where the system automatically decides objects mapped with service candidates and a case he decides the objects by touching some objects around him prior to decision of services to be activated. Subjects answer a questionnaire after above experiences.

7.2. Discussion on Usability of the Interface

Subjects have answered their opinions with free description in the questionnaire. In addition, they have evaluated the usability of the interface with 5 respects which are simple operation for activating services, costless operation for activating services, speed of service activation, listner-friendliness of announcement and lenghth of announcement, based on a 5-point Likert scale[4]. With points between 1 to 5, subjects answer their evaluation on a statement such as "they can be satisfied with the simplicity of operation". The points of answer mean that 1:Strongly disagree, 2:Disagree, 3:Neither agree nor disagree, 4:Agree and 5:Strongly agree. Figure 12 shows the average points of all subjects. On the simple operation and the costless operation, the points of "with user-selected objects" are also shown. The points mean evaluation only in a case subjects decide the objects mapped with service candidates.

As a result, the average points are more than 4 in respect of simple operation, costless operation and speed. Subjects highly valued the touch-to-object interface because they can choose services by simply touching objects without being conscious of computers. Comparing a case where the system decides objects which become switches with a case where subject decides the objects, the average points are lower in the latter case. As a cause of this difference, there is an opinion that it is annoying to touch objects more than once to decide switches and to choose desirable services. However, on the other hand, there are the following opinions from most of subjects.

- If the system automatically decides the objects mapped with service candidates, users sometimes cannot intuitively image the relation between services and objects.
- It is more easy-to-understand to decide the objects by users themselves than the objects are automatically decided.

These opinions indicate that the touch-to-object interface has both advantages and disadvantages. But advantages are more significant for most users, because users who feel annoyed with touching to objects several times can make our system choose the objects automatically. In addition, most subjects valued the following advantage.

- Users can activate services on the spot without moving, because service candidates are dynamically mapped to objects around them.
- Users can comfortably accept services from the system, because they can choose activated services with a simple interface without leaving service choice to computers.

Many opinions from subjects prove the touch-to-object interface has high operability to actualize service activation by users.

Although the touch-to-object interface has received high evaluation, the average points are less than 3 in respect of listener-friendliness of announcement and length of announcement. There is a problem that if all of detailed contents which should be reported to users are announced the length of voice announcement tends to become long. It can be stress for users. Also, there is an opinion that if users do not listen to the voice announcement carefully they may miss the relation between services and objects because the voice announcement is invisible and intangible.

In the future, we will study visualization of relation between services and objects by combinating voice announcement with displaying on an information terminal to resolve annoyance on the dynamic mapping. Display of the relation is expected to make users understand the relation clearly. In addition, it may be effective to report short abstract contents with voice announcement and to display concrete contents for reduction of user stress.

8. Accuracy of Behavior Detection

8.1. How to Calculate Detection Rates

We have conducted experiments to evaluate our method for behavior detection with the collected data described in section 6. In the experiment, detection accuracies of 4 behaviors, which are leaving home, coming home, getting up, and going to bed, are calculated with behavior logs. Detection accuracies are calculated both with threshold values common to all users and with threshold values determined for individuals. The latter results are compared with the former results.

To calculate the accuracies, matching templates are created with part of collected behavior logs and each matching template is repeatedly matched with behavior logs which are not used for creating the template. In the experiments, the ground truth is given by subjects themselves.

Compared with 4 behaviors of our target, subjects touch entirely-different kinds of objects in scenes such as reading books, cooking and having a meal. Our detection method can easily distinguish between behaviors in these scenes and the 4 behaviors. To calculate the accuracies in the experiment, behavior logs which are prone to be faultily detected are adequate as false cases. Therefore, we use behavior logs of a behavior as true cases from target 4 behaviors and those of other 3 behaviors as false cases.

To verify that our detection method detects user behavior by the deadline, we set the time length t_l of sample behavior logs and match-target behavior logs to 10 minutes in experiments. Behavior logs of leaving home are logs of past 10 minutes to the time subjects touch the doorknob of the front door. Behavior logs of going to bed are logs of past 10 minutes to the time subjects lie down on the bed for sleeping. Behavior logs of coming home are logs of 10 minutes from the time subjects touch the doorknob of the front door. Behavior logs of getting up are logs of 10 minutes from the time subjects get out of bed. That is, if the conformity c which is calculated by matching true cases with matching templates is more than the detection threshold d, it means right behaviors are detected by the deadline.

TPR and TNR are calculated as follows. First, statistical data of 8 subjects for collaborative filtering are calculated with their behavior logs, based on the method described in the previous section. Next, the following steps are executed on each subject to calculate TPR and TNR with threshold values determined for individuals. In experiments, each of subjects is considered as a target user and other subjects are considered as test users.

1. Select 5 true cases and create a matching template with the cases, based on the extraction threshold e.
2. Select other 1 true case and match the case with the matching template.
3. Match all of false cases with the matching template, with the detection threshold d.
4. Repeat 100 times from step 1 to step 3, using a new matching template created with a new combination of 5 true cases every time.

Here, TPR is calculated based on cross validation. TNR is calculated by matching all false cases with all created matching templates. The number of sample behavior logs for creating a matching template is set to 5, which can be collected within a week. This is because our study assumes that our system must start providing services to users within a week at the latest since the beginning of use of our system. The values of e and d are determined when each matching template is created in step 1 by collaborative filtering using statistical data of 7 subjects other than the target user whose TPR and TNR are calculated in the above steps. In experiments, if the number of ordered pairs are more than 300, it is calculated as 300 because more than 300 ordered pairs are empirically too many as the number of characteristics of user behavior. Because values of statistical data used for collaborative filtering must be normalized, all values are normalized so that the values are between 0 to 300. From the result of all matchings, TPR, TNR, and HTTR of every subject are calculated on the case with threshold values determined for individuals.

After that, these rates with common threshold values are calculated by similar steps. In this case, e is fixed to 0.8. d are 0.33 in leaving home, 0.31 in coming home, 0.47 in getting up and 0.63 in going to bed. These values have been determined in advance so that detection accuracies are the highest.

8.2. Detection with Common Threshold Values

First, this section shows detection accuracies with common threshold values. We have previously reported experiments on behavior detection with common threshold values[18, 19]. Table 1 shows a result of comparing detection accuracies of our detection method using ordered pairs and detection accuracies of a detection method with matching templates represented as Hidden Markov Model (HMM) which is often used for behavior recognition analyzing time-series patterns by existing methods. TPR and TNR on Equal Error Rate (EER) at which difference between TPR and TNR is the smallest are respectively shown in the table. Each rate is an average rate of all subjects. The accuracies with ordered pairs are higher than those with HMM. There are differences more than 10% about leaving home. Moreover, those are more than 30% about other behaviors. The output probability of HMM falls remarkably in a case that rare activities are inserted into an observed sequence of user activities. It falls also in a case that users change the order of part of activities in the sequence. The differences between accuracies with ordered pairs and accuracies with HMM are proof of that ordered pairs are robuster to such complex user activities than HMM.

Table 1. Detection rate on EER.

	ordered pairs		HMM	
behavior	TPR(%)	TNR(%)	TPR(%)	TNR(%)
leave home	95.25	92.94	79.25	79.17
come home	92.38	95.91	62.00	60.73
get up	85.00	80.45	53.00	56.46
go to bed	80.50	83.50	46.13	46.12

However, detection accuracies of some users are not enough high. Differences between common threshold values and values appropriate for each user affect on the detection accuracies. 4 tables, which are from Table 2 to Table 5, show differences between the common value of the detection threshold and the best value of the detection threshold for each subject. The best values are calculated by analyzing the results. The common value for each behavior is shown in the bottom row of the tables. In addition, TPR and TNR with the common threshold values are shown together in the tables. Differences between common values and the best values of each subject are not relatively big on leaving home and coming home. On the other hand, there are more differences of those values on getting up and going to bed. Accordingly, detection accuracies on getting up and going to bed are overall less than detection accuracies on leaving home and coming home. In addition, there are differences among the best values of subjects. Comparing detection accuracies with differences between common values and the best values of each subject in each table, it is apparent that the more differences bring lower detection accuracies. Detection accuracies of subjects A, G, H in Table 4 and subjects E, H in Table 5 indicate such trend significantly. The detection threshold value directly affects detection accuracy of user behavior. These results show that it is important to

Table 2. Variation of the best value of detection threshold on "leaving home".

subject	TPR (%)	TNR (%)	best value	difference
A	94.00	96.02	28%	5
B	98.00	85.44	40%	7
C	78.00	83.20	46%	13
D	95.00	98.00	23%	10
E	99.00	98.96	34%	1
F	96.00	97.00	28%	5
G	100.00	96.36	35%	2
H	98.00	95.18	36%	3
common threshold value			33%	-

Table 3. Variation of the best value of detection threshold on "coming home".

subject	TPR (%)	TNR (%)	best value	difference
A	89.00	95.93	31%	0
B	99.00	98.12	35%	4
C	81.00	83.37	42%	11
D	98.00	78.40	56%	25
E	93.00	99.60	24%	7
F	99.00	100.00	30%	1
G	100.00	96.80	50%	19
H	100.00	98.27	43%	12
common threshold value			31%	-

Table 4. Variation of the best value of detection threshold on "getting up".

subject	TPR (%)	TNR (%)	best value	difference
A	73.00	99.12	20%	27
B	90.00	96.78	47%	0
C	63.00	84.35	43%	4
D	100.00	99.22	55%	8
E	64.00	87.32	45%	2
F	97.00	99.68	46%	1
G	100.00	74.33	67%	20
H	56.00	83.60	28%	19
common threshold value			47%	-

Table 5. Variation of the best value of detection threshold on "going to bed".

subject	TPR (%)	TNR (%)	best value	difference
A	62.00	85.34	45%	18
B	91.00	71.84	64%	1
C	95.00	96.92	72%	9
D	78.00	94.66	68%	3
E	28.00	91.24	40%	23
F	95.00	99.14	47%	16
G	98.00	99.32	61%	2
H	58.00	100.00	36%	27
common threshold value			63%	-

determine threshold values for individuals.

8.3. Detection with Individual Threshold Values

As results of experiments, detection accuracies with threshold values determined for individuals are shown in 4 tables, which are from Table 6 to Table 9. The tables respectively show the results of leaving home, coming home, getting up, and going to bed.

Behavior detection method must achieve high accuracy stably for behaviors of many users. It is preferable that accuracies of all users are reasonably high rather than that accuracy are very high only for some users and are low for others. As results of experiments, there are some subjects whose TPR or TNR are lower with individual threshold values than those of common threshold values. However, detection accuracies are still high on most of them. They are more than 80%. On the other hand, there are cases where the individual threshold values achieve higher accuracies on some subjects whose accuracies are originally high with common threshold values. Here, these results are not focused on. The following characteristic differences between accuracies with individual threshold values and accuracies with common threshold values are focused on.

- Individual threshold values achieve higher accuracies than low accuracies which are less than 80% with common threshold values.
- Individual threshold values bring lower accuracies, which are less than 80%, than accuracies with common threshold values.

Based on the result of the t-test, the experimental results are evaluated with the idea that difference of more than 5% is a statistically-significant difference between accuracies with individual threshold values and accuracies with common threshold values.

In tables, the differences are shown in parenthesis of each value of TPR and TNR, except the differences which are less than a statistically-significant difference. Positive values mean that individual threshold values have increased detection accuracies. TPR and TNR with the common threshold values are shown in Table 2, Table 3, Table 4 and

Table 6. Detection accuracy of "leaving home" with threshold values estimated by collaborative filtering.

note	subj.	TPR (%)	TNR (%)
	A	95.00	94.46
	B	98.00	82.68
#3	C	96.00 (+18)	57.28 (-25.92)
	D	94.00	91.54
	E	99.00	85.92
	F	90.00	97.44
	G	100.00	95.92
	H	86.00	92.68

Table 7. Detection accuracy of "coming home" with threshold values estimated by collaborative filtering.

note	subj.	TPR (%)	TNR (%)
	A	90.00	97.42
	B	98.00	99.82
	C	79.00	92.60
#1	D	98.00	85.63 (+7.23)
	E	96.00	99.42
	F	98.00	100.00
	G	100.00	92.72
	H	100.00	98.18

Table 8. Detection accuracy of "getting up" with threshold values estimated by collaborative filtering.

note	subj.	TPR (%)	TNR (%)
#1	A	81.00 (+8)	98.07
	B	90.00	92.58
#1	C	78.00 (+15)	82.20
	D	100.00	96.83
	E	62.00	81.23
	F	98.00	99.42
#2	G	100.00	65.72 (-8.62)
#3	H	72.00 (+16)	69.72 (-13.88)

Table 9. Detection accuracy of "going to bed" with threshold values estimated by collaborative filtering.

note	subj.	TPR (%)	TNR (%)
#1	A	69.00 (+7)	80.16
#2	B	95.00	63.94 (-7.9)
	C	96.00	92.48
	D	79.00	84.04
#1	E	48.00 (+20)	84.02
	F	99.00	97.74
	G	100.00	97.62
#1	H	70.00 (+12)	99.10

Table 5. Results are categorized into three groups, #1, #2 and #3. Individual threshold values have achieved higher accuracies in 6 cases of #1. In particular, on TPRs of subject E and H in Table 9, the individual values have improved them +20 and +12 respectively. While, accuracies are lower than that with common threshold values only in 2 cases of #2, which are subject G in Table 8 and subject B in Table 9. In 2 cases of #3, TPR is higher but TNR is lower with individual threshold values. Our determination method is not necessarily effective on these cases. These experimental results show our method can improve detection accuracies of users whose detection accuracies are low with common threshold values, by setting appropriate values to thresholds for individuals.

Because of a property of collaborative filtering, if there is no test user who has strong correlation with the target user at all then the values of the target user are not accurately estimated. It can be a cause of ineffectiveness of our determination method in a few cases.

An additional experiment, which uses all of 8 subjects including an estimated target subject as test users for collaborative filtering, has been conducted. It means that test users include a test user who has likely strong correlation with the target user. The experimental results are shown in Table 10, Table 11, Table 12, and Table 13. As a result, compared to the former experiment, improvement of detection accuracy has been shown in 3 cases. First, TPR of subject C in Table 10 has been improved without decreasing TNR. The difference on TNR of subject G in Table 12 changes from -8.62 to +5.62. In addition, TPR of subject H in Table 13 is improved from +12 to +23. These results indicate that our determination method have a possibility to improve detection accuracy of more users. With these results in mind, to solve above problems, it is important to prepare test users who has strong correlation with the target user by increasing the number of test users and the diversity of test users. To make diverseness of test users, a method for making additional test users artificially is necessary.

9. Challenges for Improvement

Our system can be improved more in the future as follows to personalize the system depending on each user.

The situation configuration table, the service-object

Table 10. Detection accuracy of "leaving home" with threshold values estimated by collaborative filtering which includes a target user in test users.

note	subj.	TPR (%)	TNR (%)
	A	97.00	95.30
	B	98.00	81.84
#1	C	88.00 (+10)	87.68
	D	95.00	98.00
	E	99.00	99.18
	F	95.00	95.76
	G	100.00	95.82
	H	99.00	90.80

Table 11. Detection accuracy of "coming home" with threshold values estimated by collaborative filtering which includes a target user in test users.

note	subj.	TPR (%)	TNR (%)
	A	90.00	97.40
	B	98.00	99.82
	C	80.00	89.72
#1	D	98.00	89.72 (+11.32)
	E	96.00	99.42
	F	98.00	100.00
	G	100.00	97.42
	H	100.00	98.18

Table 12. Detection accuracy of "getting up" with threshold values estimated by collaborative filtering which includes a target user in test users.

note	subj.	TPR (%)	TNR (%)
#1	A	83.00 (+10)	98.83
	B	89.00	94.55
#3	C	79.00 (+16)	79.33 (-5.02)
	D	100.00	97.35
	E	64.00	82.77
	F	99.00	99.90
#1	G	100.00	79.95 (+5.62)
#3	H	74.00 (+18)	65.65 (-17.95)

Table 13. Detection accuracy of "going to bed" with threshold values estimated by collaborative filtering which includes a target user in test users.

note	subj.	TPR (%)	TNR (%)
#1	A	69.00 (+7)	80.16
#2	B	95.00	64.02 (-7.82)
	C	96.00	92.48
#1	D	91.00 (+13)	94.02
#1	E	48.00 (+20)	84.02
	F	99.00	97.74
	G	100.00	97.62
#1	H	81.00 (+23)	100.00

mapping table and the concrete-to-abstract conversion rules are predefined in this system. General contents common to all users are described in these. However, they are not always appropriate for all users. The rules can be redefined, but it is not easy for users unfamiliar with computers to customize these by themselves at present. We must build a method for customizing these easily without complex operation.

This system uses passive RFID system to develop with as few types of sensors and low-cost sensors as possible at present. Later, we will study the possibility of a variety of applications by adding other sensors to this system. For example, we need to add reasonable sensors which achieve acquiring the position information of users and objects more precisely. Checking usefulness and cost of a variety of sensors, we will consider personalization of configuration of sensors combined with our system according to user's budget.

Currently, we have implemented a behavior detection which is appropriate for an application to prevent mistakes and dangers of users. If we implement other behavior detection for different applications in the future, we can extend our system by adding new modules into Matching Template Creator and Behavior Detector. The extentensibility enables our system to personalize kinds of applications depending on each user.

10. Conclusion

In this paper, we have proposed a home context-aware system which has a mechanism for personalization of service activation and personalization of context estimation. The system provides services by combining a system-active approach and a user-active approach. Because users themselves finally choose activated services with a touch-to-object interface in a user-active approach, the system can activate a variety of services also along user intention which cannot be inferred with computers. In a system-active approach, user behavior is detected as a trigger of service providing. The system determine threshold values appropriate for each user in the detection method by utilizing statistical data of test users whose characteristics are similar to each user. The determination enables stable behavior detection. In experiments, we have demonstrated the high possibility of the proposed system.

Acknowledgments

This work was partially supported by Grant-in-Aid for Scientific Research in Japan.

References

[1] T. Asami, K. Iwano, and S. Furui. A stream-weight and threshold estimation method using adaboost for multi-stream speaker verification. In *Proc. the 2006 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP2006)*, volume 5, pages 1081–1084, 2006.

[2] J. Barbič, A. Safonova, J.-Y. Pan, C. Faloutsos, J. K. Hodgins, and N. S. Pollard. Segmenting motion capture data into distinct behaviors. In *Proc. the 2004 conference on Graphics interface*, pages 185–194, 2004.

[3] L. Cai and T. Hofmann. Text categorization by boosting automatically extracted concepts. In *Proc. the 26th Annual International ACM SIGIR Conference on Research and Development in Informaion Retrieval (SIGIR2003)*, pages 182–189, 2003.

[4] T. K. Harris and R. Rosenfeld. A universal speech interface for appliances. In *Proc. the 7th International Conference on Spoken Language Processing (ICSLP 2004)*, pages 249–252, 2004.

[5] T. Huỳnh, U. Blanke, and B. Schiele. Scalable recognition of daily activities with wearable sensors. In *Proc. the 3rd International Symposium on Location- and Context-Awareness (LoCA2007), LNCS 4718*, pages 50–67, 2007.

[6] Y. Kimura, D. Watabe, H. Sai, and O. Nakamura. New threshold setting method for the extraction of facial areas and the recognition of facial expressions. In *Proc. the IEEE Electrical and Computer Engineering, Canadian Conference (CCECE/CCGEI2006)*, pages 1984–1987, 2006.

[7] G. Linden, B. Smith, and J. York. Amazon.com recommendations: Item-to-item collaborative filtering. *IEEE Internet Computing*, 7(1):76–80, 2003.

[8] B. Logan, J. Healey, M. Philipose, E. M. Tapia, and S. Intille. A long-term evaluation of sensing modalities for activity recognition. In *Proc. the 9th International Conference on Ubiquitous Computing (UbiComp2007), LNCS 4717*, pages 483–500, 2007.

[9] D. J. Moore, I. A. Essa, and M. H. HayesIII. Exploiting human actions and object context for recognition tasks. In *Proc. the IEEE International Conference on Computer Vision 1999 (ICCV'99)*, pages 80–86, 1999.

[10] J. Nichols and B. A. Myers. Controlling home and office appliances with smart phones. *IEEE Pervasive Computing*, 5(3):60–67, 2006.

[11] F. Niu and M. Abdel-Mottaleb. HMM-based segmentation and recognition of human activities from video sequences. In *Proc. the 2005 IEEE International Conference on Multimedia and Expo (ICME2005)*, pages 804–807, 2005.

[12] D. J. Patterson, D. Fox, H. A. Kautz, and M. Philipose. Fine-grained activity recognition by aggregating abstract object usage. In *Proc. the 9th IEEE International Symposium on Wearable Computers (ISWC2005)*, pages 44–51, 2005.

[13] M. Perkowitz, M. Philipose, D. J. Patterson, and K. Fishkin. Mining models of human activities from the web. In *Proc. the 13th International World Wide Web Conference (WWW 2004)*, pages 573–582, 2004.

[14] J. Riekki, T. Salminen, and I. Alakärppä. Requesting pervasive services by touching RFID tags. *IEEE Pervasive Computing*, 5(2):40–46, 2006.

[15] J. G. Shanahan and N. Roma. Boosting support vector machines for text classification through parameter-free threshold relaxation. In *Proc. the 12th International Conference on Information and knowledge Management (CIKM2003)*, pages 247–254, 2003.

[16] K. Tsukada and M. Yasumura. UbiFinger: Gesture input device for mobile use. In *Proc. the 5th Asia Pacific Computer Human Interaction (APCHI2002)*, pages 388–400, 2002.

[17] S. Wang, W. Pentney, A.-M. Popescu, T. Choudhury, and M. Philipose. Common sense based joint training of human activity recognizers. In *Proc. the 20th International Joint Conference on Artificial Intelligence (IJCAI2007)*, pages 2237–2242, 2007.

[18] H. Yamahara, T. Soma, F. Harada, H. Takada, Y. Shimada, and H. Shimakawa. Tagged World: an intelligent space providing services by interaction between a user and an environment. In *Proc. the 2nd International Conference on Mobile Ubiquitous Computing, Systems, Services and Technologies (UBICOMM2008)*, pages 333–342, 2008.

[19] H. Yamahara, H. Takada, and H. Shimakawa. Detection of user mode shift in home. In *Proc. the 4th International Symposium on Ubiquitous Computing Systems (UCS2007), LNCS 4836*, pages 166–181, 2007.

[20] H. Yamahara, H. Takada, and H. Shimakawa. An individual behavioral pattern to provide ubiquitous service in intelligent space. *WSEAS TRANSACTIONS on SYSTEMS*, 6(3):562–569, 2007.