

Effective Appliance Selection by Complementary Context Feeding in Smart Home System

Taek Lee, Jiyong Park, Hoh Peter In[†]

Department of Computer Science and Engineering, Korea University,
Seoul, 136-713, Republic of Korea
{comtaek, jayyp, hoh_in}@korea.ac.kr

Abstract. Smart Home System (SHS) is one of popular applications in ubiquitous computing, which provides convenient services for a user with user-friendly intelligent system interfaces. Among them, voice recognition is a popular interface. However, voice command statements given by users are often too unclear and incomplete for the devices in SHS to understand the original user intention. So, the devices become complicated and have no idea about whether to work or not. Therefore, we should make sure the proximate selection for the devices which will be eventually targeted and operated following user intention. In this paper, we propose an effective method to make a decision in electing a promising target device among candidates by taking advantage of complementary context feeding around user environment in SHS even with initial incomplete interface information. The proposed method is based on Bayes theorem using the way of empirical statistic inference.

Keywords: Smart home, Situation-aware computing, HCI, Bayes Theorem

1 Introduction

Ubiquitous computing has a motto to inject intelligent functionality into all of the objects in environment so that they can deliver any service useful and convenient to a user by making the objects collaborating organically with each other. In the similar manner, Smart Home System, which is our prototype application, aims to maximize user convenience by supporting the home appliances in working intelligently satisfying user service requirements. Thus, even for heterogeneous appliances, it should be always guaranteed for them to collaborate and deliver eventually user-friendly services to a user without any confusion in their running conditions. Under our definition, intelligent functionality means that with the minimum interaction costs a user can be served with high satisfaction level originally required. To implement such intelligence, Human Computer Interaction researches, especially multimodal interface area [10], have been getting matured so far. Among them, speech recognition area has been significantly developing since the past decades so that many of commercial products supporting a voice command recognition function have been

[†] Hoh Peter In is the corresponding author.

emerged to market. However, regardless of its attractiveness, voice recognition interface is not so popularly used in practical system implementation. The reason is not because of its technical unstableness but because of the following reasons.

- *Users often use incomplete syntax of voice command statement.* Assuming easy working, users tend to consciously or unconsciously use incorrect voice command structures instead of giving complete syntax of commands to a machine. If such commands are given to SHS, we can not expect the appliances to work correctly. For instance, if a voice command not having any explicit target just like ‘Turn On!’ is given, the system becomes very complicated whether to turn on an audio or to turn on a lamp.
- *Users sometimes use commands not reserved in system manual.* We can not expect that users will always give the system a command statement consisting only of reserved words in system instruction manual.

Therefore, for better understanding, SHS requests more clear and complete interface expression from user. If impossible to expect it from a user, SHS needs other alternative hint information to catch the point of user intention. In this paper, we will take advantage of user contextual information aggregated in running environment to complement the poor primary interface information (i.e. an incomplete voice command). The proposed SHS has ability to train and improve itself by updating a knowledge database. Our proposed method is based on Bayes theory, which plays an important role in understanding user intention with incomplete interface information. When it works, it uses statistics of context (or situation) information stored in the knowledge database.

Basically, intelligent functionality of appliances depends on the system environment deployed. Thus, it is necessary to train and optimize them for user-friendly services. To do so, our proposed SHS identifies the context attributes of situation information relevant to user preferences and then interprets them as a user intention to activate an appliance targeted.

The rest of the paper is organized as follows: Section 2 explains the overview of Smart Home System and Section 3 presents our proposed appliance selection method based on Bayes theorem. Section 4 shows evaluation results of an experiment to which our proposed method was applied. Section 5 introduces related work and the pros and cons of our approach. Finally, we conclude the paper in Section 6.

2 Situation Awareness in Smart Home System

SHS uses user contexts gathered from system environment to complement the incomplete primary interface information (i.e. voice command). To do so, we need to extract more high level of situation information from several of the user-related contexts which are just clues embedding less of implication in their nature. In this section, first of all, we perceive the difference of *context* and *situation*, and then give

an idea about how to utilize their different definitions and apply them to materializing SHS.

2.1 Situation Awareness

Situation is a set of past context attributes and/or actions of individual devices which are relevant to determine future device actions. *Context* is any instantaneous, detectable, and relevant condition of the environment or the device. Situation aware interface definition language (SA-IDL [4], [5]) is a tool to materialize and utilize the situation awareness concept. SA-IDL consists of context tuple, action tuple and derived context as Table 1 [6].

Table 1. SA-IDL Model

$Action := (Time, Device, State, StateValue)$ $Context := (Timestamp, Device, State, StateValue)$ $A := \{x \mid x \text{ is an Action} \}$ $C := \{x \mid x \text{ is a Context} \}$ $DerivedContext := P(C) \rightarrow \{true, false \}$ $Situation := (DerivedContext, P(C), A)$

DerivedContext is a kind of function to determine whether a given context set holds or not in the application environment eventually resulting in true or false. In section 3, we propose a detail method which can be used in implementing the concept of *DerivedContext* function.

2.2 Appliance Selection using Situation Awareness

To complement an incomplete primary interface (i.e. a voice command given by a user), we propose a prototype, Smart Home System (Fig. 1) using not only explicit primary interface information but also context information just like playing a role of additional implicit complementary interfaces.

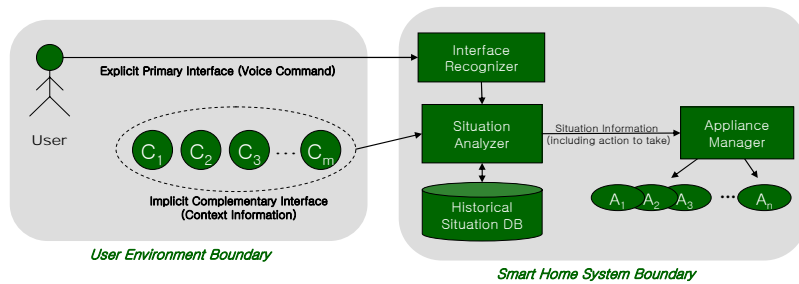


Fig. 1. Overview of Smart Home System

Firstly, Interface Recognizer interprets a given voice command and pass the recognized but not now dependable information to Situation Analyzer. Situation Analyzer produces an Action Tuple including a target appliance of information and its action to take. Situation Analyzer conducts reasoning process with the primary interface information passed by Interface Recognizer, context information gathered from user environment, and statistic information extracted from Historical Situation DB. Appliance Manager receives the Action Tuple from Situation Analyzer. Finally, Appliance Manager distributes a control message to the corresponding appliance supposed to work.

3 The Proposed Appliance Selection Method

We call the appliances being on standby, namely, candidates because they will be potentially designated as targets by an Action Tuple come from Situation Analyzer. Therefore, Situation Analyzer is responsible for making a decision of which candidate must be elected with some justification. In this section, we present a solution approach based on Bayes theorem.

3.1 Selection Problem Interpretation using Bayes Theorem

When primary interface information, a voice command, is given from a user in SHS, appliance candidates likely to be selected are temporary determined. Let us define the n candidates with set $A = \{ a_1, a_2, a_3, \dots, a_n \}$, where the element a_i is a binary random variable whose value is 'on' or 'off'. Hence, $P(a_i = 'on')$ means the probability of that the i th appliance will work in SHS when a voice command is given. Context set C consists of *Context Tuple* elements like $C = \{ c_1, c_2, c_3, \dots, c_m \}$. They can be observed and captured from user environment. Ultimately, in a given selection problem, all of what we have to do for finding a solution is to select a particular a_i making k and the conditional probability $P(a_i = 'on' | C)$ maximized in a solution set S like (1).

$$S = \{ \exists a_i \in A \mid \text{maximize}(k + P(a_i = 'on' | C)) \} \quad (1)$$

The constant k means the average success rate of the voice recognition technology used in SHS. Therefore, we don't try to improve k because it is out of focus in this paper. Therefore, we are just supposed to find a_i making $P(a_i = 'on' | C)$ maximized. For that, we use Bayes theorem, the equation below (2). We newly infer $P(a_i = 'on' | C)$ by considering initial prior probability which can be easily calculated by referring to the empirical knowledge in Historical Situation DB.

$$P(a_i = 'on' | C) = \frac{P(C | a_i = 'on')}{P(C)} \cdot P(a_i = 'on') \quad (2)$$

By the way, if the *Context Tuple* elements in a set C are independently occurred, the equation (2) can be translated to the equation (3) as follows.

$$P(a_i = 'on' | C) = \frac{P(a_i = 'on') \prod_{j=1}^m P(c_j | a_i = 'on')}{P(C)} \quad (3)$$

To find a solution a_i making $P(a_i = 'on' | C)$ maximized, we have to compare all of the right-handed side of the equation (3) for $\forall a_i \in A$. The denominator $P(C)$ is a common factor in the comparison so that it plays a role as a fixed constant and doesn't influence the prioritization procedure. Therefore, the problem we want to solve is conclusively simplified to the question about how to find the a_i making the numerator part of the equation (3) maximized. The answer is explained in the next sub section with taking a practical example.

3.2 Selection Problem Resolution

Table 2 is a piece of training data used in our SHS prototype. Thus, we can assume set A and C are respectively set $A = \{ 'audio', 'heater', 'cooler', 'stove', 'lamp' \}$ and set $C = \{ 'user tracking', 'time', 'temperature' \}$ in Table 2. The more variant and plentiful training data are given, the more Situation Analyzer is trained well and work effectively in inferring some situation.

Table 2. Samples of fifteen records in Historical Situation DB

Tid	Primary Interface	Historical User Context Set associated with Situation			Derived Target & Action Information	
	PI:voice	c ₁ :tracking	c ₂ :time	c ₃ :temperature	Target	Action
1	Turn On!	bed→bed	morning	10℃	a ₁ :audio	play classic
2		door→heater	afternoon	-8℃	a ₂ :heater	run in high
3		bed→kitchen	morning	33℃	a ₄ :stove	-
4		bed→bed	morning	40℃	a ₃ :cooler	run in high
5		door→heater	afternoon	10℃	a ₂ :heater	-
6		door→cooler	afternoon	30℃	a ₁ :audio	play dance
7		door→heater	afternoon	45℃	a ₃ :cooler	-
8		door→cooler	night	57℃	a ₃ :cooler	sleep mode
9		table→bed	night	23℃	a ₁ :audio	play jazz
10		door→cooler	afternoon	29℃	a ₅ :lamp	-
11		bed→bed	morning	-10℃	a ₂ :heater	-
12		door→kitchen	night	5℃	a ₅ :lamp	lamp door
13		bed→kitchen	afternoon	12℃	a ₄ :stove	-
14		table→cooler	morning	38℃	a ₃ :cooler	-
15		door→table	night	20℃	a ₅ :lamp	lamp table

To solve the equation (1) dealing with selecting problem of our interesting target appliance, first of all, we have to identify which contexts are helpful and related to our problem solving and then know the conditional probability $P(C | a_i = 'on')$ before calculating $P(a_i = 'on' | C)$. When calculating $P(C | a_i = 'on')$, if the element values of the set C are categorical attributes, we can calculate $P(C | a_i = 'on')$ by just counting the number of cases holding the condition $a_i = 'on'$. However, if the element values are continuous attributes and they follow the normal distribution, we can utilize the Gaussian distribution PDF whose parameters are μ and σ like the equation (4) below.

$$P(C = c_j | A = a_i) = \frac{1}{\sqrt{2\pi\sigma_{ij}}} \exp \left(-\frac{(c_j - \mu_{ij})^2}{2\sigma_{ij}^2} \right) \quad (4)$$

The variables that we have to know in the equation (4) are just μ_{ij} (the average of the c_j element values for a given particular a_i) and σ_{ij}^2 (the variance of the element values).

For instance, under the assumption of Table 2, if the current context information captured from a user is $C = \{c_1 = 'door \rightarrow cooler', c_2 = 'afternoon', c_3 = '43^\circ C'\}$, each equation (3) value of candidate appliances can be eventually calculated as following Table 3.

Table 3. Conditional probabilities that each appliance will be targeted for a given C

$P(a_1 = 'on' C) = (P(C a_1 = 'on') / P(C)) * P(a_1 = 'on') = 0.000408375 * \alpha * 0.2 = 0.0816 * 10^{-3} * \alpha$
$P(a_2 = 'on' C) = (P(C a_2 = 'on') / P(C)) * P(a_2 = 'on') = 0 * \alpha * 0.2 = 0$
$P(a_3 = 'on' C) = (P(C a_3 = 'on') / P(C)) * P(a_3 = 'on') = 0.00284375 * \alpha * 0.27 = 0.76781 * 10^{-3} * \alpha$
$P(a_4 = 'on' C) = (P(C a_4 = 'on') / P(C)) * P(a_4 = 'on') = 0 * \alpha * 0.13 = 0$
$P(a_5 = 'on' C) = (P(C a_5 = 'on') / P(C)) * P(a_5 = 'on') = 0.00042471 * \alpha * 0.2 = 0.0849 * 10^{-3} * \alpha$

In Table 3, α means $1/P(C)$. If we compare each one with one other probability value, the priority order is as follows.

$$a_3 = 0.76781 * 10^{-3} * \alpha > a_5 = 0.0849 * 10^{-3} * \alpha > a_1 = 0.0816 * 10^{-3} * \alpha > a_2 > a_4$$

Conclusively, it has been discovered that our interesting final solution is a_3 since $a_3 = 0.7678125 * 10^{-3} * \alpha$ has the highest value than the others. The zero cases of a_2 and a_4 happened because no matched situation data was in Table 2 (i.e. Historical Situation DB). To avoid zero probability calculation, we can use the way of Laplace Correction [2]. In the same manner with the calculation procedure for selecting a target so far, inferring an action to take for given contexts can be conducted as well.

Coming back to the system viewpoint of working flow, we can see that the final result inferred is reformed to *Situation Tuple* format as explained in the section 2.1 and then delivered to *Appliance Manager*.

4 Experiment and evaluation of the proposed method

4.1 Experiment Environments

We applied our proposed method, Bayes Theorem Approach (BTA), to implementing Situation Analyzer in Fig. 1. Through the experiment, we aimed at testing the validation of BTA and appealing its effectiveness by comparing it with a popular existing approach, Rule Based Approach (RBA).

In implementing the experiment, for capturing user movement context, we used AR Toolkit [8]. It is very useful to perceive user identity as well as user 3D tracking because it provides developers with different figures of recognizable marks. For implementing voice recognition functionality, we used Microsoft speech SDK 5.1 [9]. Finally, we gathered more context information from the system environment or the Internet, such as time and temperature.

First of all, to test and find the initial success or failure rate (k) of the primary interface, voice command, we asked three users participated in the experiment to speak any possible type of a voice command with conceiving intention to point some different five appliances such as audio, heater, cooler, stove, and lamp in the test space under assumption of fifteen categorical scenarios. Before starting the experiment, they obviously didn't know which type of voice command is desirable and effective to make the intended appliance work correctly. Through the way, they could give a command to SHS without any bias. For evaluation, we used a criterion, the number of successes over all trials, to measure the degree of user satisfaction for a service, which is called success rate.

4.2 Experiment Result and Evaluation

As you can see in Fig. 2, the success rate (k) of the primary interface was 70% because of some incomplete (arbitrary so not valid) command sentences and environmental noise. The next experiment was to reduce the failure rate by implementing and applying RBA to Situation Analyzer. RBA could improve the 66.6% portion of the failure cases (30%) which were not quite recognized when we had used only voice recognition interface. After all, we could improve the original success rate from 70% up to 89.8% by adding 19.8% of success cases in the case of RBA. However, BTA could more improve it up to 94%. The reason why RBA failed was because there were no rules to check some exceptional situations or sensitive values hanging in a threshold boundary of relation operation. For instance, the rule “*if PI='on' AND $c_1=? \rightarrow cooler$ AND $c_3 > 45^\circ C$ then Action=(now, cooler, power, on)*” could not recognize the case Tid 4 in Table 2 as the situation the cooler should be activated following user intention. On the other hand, BTA could work better effectively than RBA not because an exactly matched situation data (just like a rule in RBA) had already existed in Historical Situation DB but because Situation Analyzer referring to Historical Situation DB had been already trained by user preferred similar situation data before that time.

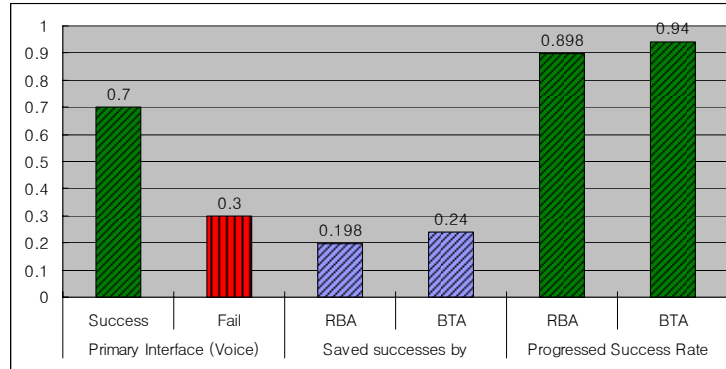


Fig 2. Experiment Results for given scenarios

5 Related Work

Since introducing the paradigm of Ubiquitous Computing, many of applications addressing or utilizing context (or situation) awareness have been emerged. Especially, to solve interoperability problem between devices in their working, interface definition languages were presented in [4], [5], [6]. To make a decision for whether to be a particular situation or not from a given context set, it uses rule based determination (i.e. Rule Based Approach). In many cases, RBA is very useful and easy to abstract and formalize intangible situation concept. So, it has been used in implementing a situation-aware middleware [6]. However, it has still disadvantages. Table 4 is for pros and cons of RBA and BTA which we proposed in this paper.

Table 4. Rule Based Approach vs. Bayes Theorem Approach

	RBA	BTA
Pros	<ul style="list-style-type: none"> • Easy update of a rule set if necessary • Light computation and high performance • Easy implementation 	<ul style="list-style-type: none"> • Flexibility for dynamic user preference and habit change • Self training mechanism by feeding user feedbacks
Cons	<ul style="list-style-type: none"> • Inflexibility against noise input data or a sensitive value in the boundary of a rule threshold • No self-training mechanism 	<ul style="list-style-type: none"> • Need for abundant training data • High computation cost to implement it when having many context attributes
Application conditions	<ul style="list-style-type: none"> • When implementing it on devices having low computation power for lightness and assuming distributed running environment of devices 	<ul style="list-style-type: none"> • When implementing it on a centralized server system having high computation power and storage capacity.

As you can see in Table 4, each approach has its own pros and cons. Therefore, we can not say our proposed BTA is always better. However, if you have enough conditions to implement BTA, it will give you more flexibility in inferring situation information as you can see the evaluation results in the previous section.

6 Conclusion

In home automation system space, the diverse intelligent appliances can suffer from incompleteness and ambiguousness of interface information given by a user. So, the appliances sometimes malfunction against the original user intention. The malfunctions disturb users to satisfy the intelligent functionality appliances could have given to themselves so that they can not avoid in controlling the appliances manually at the end. That is very undesirable to users expected somewhat automatic working. That is why we are still using a legacy controlling system such as pressing buttons or using a remote controller even though we have very matured commercial products having voice recognition functionality in the recent times.

In this paper, we proposed a prototype of Smart Home System architecture and especially a Bayes theorem based algorithm to be adopted by Situation Analyzer. Our proposed study aims at complementing a poor primary user interface with situation information embedding user intention for the purpose of providing a user-friendly service. Our proposed method, of course, can sometimes recommend a wrong advice because it is impossible for a computer system to perfectly predict user intention. However, for permanent improvement, our approach can take the user feedbacks utilized to fatten and evolve the Historical Situation DB. Situation Analyzer using the DB has high flexibility for dynamically changed user preferences.

Despite the advantages of the proposed study, there are still disadvantages which should be improved in the future as follows:

- *Need for reliable training data.* Through diverse case studies, we need to collect high quality of training data before fully operating SHS. Data collecting, analyzing, and managing processes must be simplified and organized well in practical implementation.
- *Heavy computation.* The proposed method requires many multiplication computations of float numbers to calculate and analyze probability values. Therefore it is not so light to implement the method in small devices having low computation power in SHS so that centralized intelligent system architecture is better choice than that of distributing the computation responsibilities to thick client devices. Hence, we have an idea to combine the advantages of the RBA used in SA-IDL and BTA proposed here in order to present a hybrid method not only suitable to a centralized system but also scalable to small devices in the future.

In addition to complementing future works above, we have a plan to test and check computation overhead in implementing the proposed BTA and compare it not with RBA but with other existing probabilistic methods for better improvement of our research.

Acknowledgments. This work was supported by the 2nd Brain Korea 21 Project in 2007.

References

1. Pang-ning Tan, Michael Steinbach, and Vipin Kumar, "Introduction to Data Mining", Pearson Addison Wesley, 2006.
2. Jiawei Han and Micheline Kamber, "Data Mining - concepts and techniques", second edition, MORGAN KAUFMANN, 2006
3. Schilit, B., Adams, and N. Want, R. "Context-Aware Computing Applications", Proceedings of the 1st International Workshop on Mobile Computing Systems and Applications, 1994, pp. 85-90.
4. Heeseo Chae, Taeyoun Kim, Donghyun Lee, and Hoh Peter In, "Conflict Resolution Model Based on Weight in Situation Aware Collaboration System", IEEE 11th International Workshop on future Trend of Distributed Computing System (FTDCS2007), expected publication date - Mar. 2007.
5. Yau, S.S, Yu Wang, Dazhi Huang, In. H. P, "Situation-aware Contract Specification Language for Middleware for Ubiquitous Computing", the proceeding of the Ninth Future Trends of Distributed Computing Systems, May 2003, page:93-99
6. Yau, Y. Wang, and F. Karim, "Developing Situation Awareness in Middleware for UbiComp Environments", Proc. 26th International Computer Software and Applications Conference 2002, page:233-238
7. Yu Wang, "An FSM Model for Situation-Aware Mobile Application Software Systems", Proceedings of the 42nd annual Southeast regional conference, 2004
8. 3D tracking: AR Toolkit, <http://www.hitl.washington.edu/artoolkit/>
9. Microsoft speech SDK 5.1, <http://www.microsoft.com/downloads>
10. Oviatt, S.L. "Multimodal interfaces", In The Human-Computer Interaction Handbook: Fundamentals, Evolving Technologies and Emerging Applications, J. JACKO AND A. SEARS, Eds. Lawrence Erlbaum Assoc., Mahwah, NJ, 2003, chap.14, 286-304.