# Human Adaptive Device Development based on TD method for Smart Home

Chang-Hyun Park*, and Kwee-Bo Sim*

* school of Electrical and Electronics Engineering, Chung-Ang University 221,Heukseok-Dong, Dongjak-Gu, Seoul, Korea
(Tel : +82-2-820-5319; E-mail: kbsim@cau.ac.kr)

**Abstract**: This paper presents that TD method is applied to the human adaptive devices for smart home with context awareness (or recognition) technique. For smart home, the very important problem is how the appliances (or devices) can adapt to user. Since there are many humans to manage home appliances (or devices), managing the appliances automatically is difficult. Moreover, making the users be satisfied by the automatically managed devices is much more difficult. In order to do so, we can use several methods, fuzzy controller, neural network, reinforcement learning, etc. Though the some methods could be used, in this case (in dynamic environment), reinforcement learning is appropriate. Among some reinforcement learning methods, we select the Temporal Difference learning method as a core algorithm for adapting the devices to user. Since this paper assumes the environment is a smart home, we simply explained about the context awareness. Also, we treated with the TD method briefly and implement an example by VC++. Thereafter, we dealt with how the devices can be applied to this problem.

**Keywords:** TD method, Human Adaptive Device, Smart home, Context recognition, Reinforcement learning

## 1. INTRODUCTION

Recent interests are being concentrated on a Ubiquitous environment and smart home. The goal of these issues is to let users feel comfort in the home. So, in order to be like that, many techniques are needed. Firstly, the smart home must identify a specific user and knows user's character (some status the user wants or likes). If it doesn't know user's character, it should be able to learn the user's preference through the trial and error method. Also, in order to learn the user's preference, it should be able to recognize (or aware) a specific context. The context means information about any status. More precisely, Georgia Inst. defined context to be any information that can be used to characterize the situation of an entity, where an entity can be a person, place, or physical or computational object. Also, this information includes something such as a gesture and a relation with human and object [1]. From this reason, Context recognition (or awareness) has been researched. This research is also very important but we will treat with this field simply. Since the function of the context recognition is to search and judge the status of any environment, treating several sensor information and analyzing the collected information is main issue in the context recognition. So, we will explain the method fit to this work. After completing the work, the proposal of this paper is presented. Since the context recognition is completed, several devices (or home appliances) in the smart home should service to user. For good service, it is important to find the user's preference or taste. This paper suggests the temporal difference (TD) method as a method to do that work. The TD learning algorithm is basic to reinforcement learning methods that can learn directly from raw experience of the agent to achieve a goal [Richard S. Sutton]. This characteristic being possible to train online is very effective in this case that system should be adapted to dynamic environment. The proposed system evaluates the performance (or reward) indirectly and the result feed back to the system. After the work has been done iteratively, the system finds the optimal policy. In this process, we explain how the system is composed and what parameters are used for TD method.

## 2. CONTEXT RECOGNITION

Location, identity, time, and activity are the primary context types for characterizing the situation of a particular entity. These context types not only answer the questions of who, what, when, and where, but also act as indices into other sources of contextual information. For example, given a person's identity, we can acquire many pieces of related information such as phone numbers, addresses, relationships to other people in the environment, etc. We suppose that the context widgets for obtaining information of the primary types are implemented. Especially, following figure shows the infrastructure for the context recognition and the infrastructure consists of 3 components [1].

*Widgets: implements the widget abstraction
*Server: responsible for aggregation of context
*Interpreter: responsible for interpretation of context

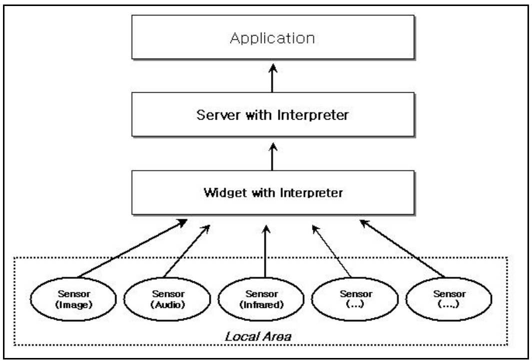


Fig 1. The infrastructure for context recognition

The basement of Fig 1 represents the sensor part for the local area. The dotted box means a part of the whole area and the various sensors collect different types of information from near environment. The function of Widget with Interpreter is the transformation from the raw sensor data to the relevant value and the local context recognition from the value. This is the sub-context recognition and Server with Interpreter merges the contexts from each sensor and interprets the entire context. Thereafter, those are used for an application.

*A. Contextual Situation*

The contextual situation includes surroundings, user's movement, user identification and near objects, etc. In this paper, the modeling of simple home corresponds to the surroundings. Also, we assume that the contextual sensing, context-adaptation, contextual resource discovery and contextual augmentation are implemented as Pascoe proposed [3,4]. The surroundings are composed of the user's room, entrance, kitchen, living room and rest room. Also, there are TV, refrigerator, microwave oven and light switch as the objects for user. There is a set {walk, stand, sit, lay, pick, throw, put, open, close} as user's movement.

**Table 1 Primary Types**

| Primary Types | Objects |
|---|---|
| *Location* | Living Room, User's Room, Entrance, Kitchen, Rest Room |
| *Identity* | User 1,User 2 |
| *Activity* | Walk, Stand, Sit, Lay, Pick, Throw, Put |
| *Time* | Morning, Afternoon, Night |
| *Object* | TV, Refrigerator, Light Switch, Microwave Oven |

Table 1. added one type 'object' to the 4 primary types (Location, Identity, Activity, Time) Anind et al. proposed[1]. Though only 4 types represent a rough action or circumstance, because the meaning of context depends on an object user faces, we need the 'Object' type for detail description. Table 2 shows some of the composed contextual situations based on Table 1. Also, Detected Information is composed of the sequentially collected data.

**Table 2 Imagined Situations**

| Detected Information | Meaning(Inference) |
|---|---|
| ID: User 1<br>Current_Location: Entrance<br>Next_Location: User's Room | *User 1 comes home* |
| ID: User 2<br>Current_Loc: User's Room<br>Next_Loc: Kitchen<br>Object: Refrigerator<br>Activity: Walk to the Refrigerator | *User 2*<br>*wants something to eat* |
| ID: User 1<br>Current_Loc: Living<br>Next_Loc: Rest room | *User 1*<br>*wants to wash or urinate* |

## 3. TEMPORAL-DIFFERENCE LEARNING

This paper uses the TD method for reinforcement learning part. TD learning is a combination of Monte Carlo ideas and dynamic programming (DP) ideas. Like Monte Carlo methods, TD methods can learn directly from raw experience without a model of the environment's dynamics. Like DP, TD methods update estimates based in part on other learned estimates, without waiting for a final outcome. Before explaining the TD method, we present the difference between TD method and other reinforcement learning method. Firstly, TD methods have an advantage over DP (Dynamic Programming) methods in that they do not require a model of the environment because TD method doesn't require the environment model. Secondly, the other advantage of TD methods over Monte Carlo methods is that they are naturally implemented in an on-line, fully incremental fashion. With Monte Carlo methods one must wait until the end of an episode, because only then is the return known, whereas with TD methods one need wait only one time step. Thus, at time t+1 they immediately form a target and make a useful update using the observed reward $r_{t+1}$ and the estimate $V(s_{t+1})$. The simplest TD method is

$$V(s_t) \leftarrow V(s_t) + \alpha[r_{t+1} + \gamma V(s_{t+1}) - V(s_t)]$$

Because the TD method bases its update in part on an existing estimate, we say that it is a bootstrapping method.

Initialize $V(s)$ arbitrarily, $\pi$ to the policy to be evaluated
Repeat (for each episode):
Initialize $s$
Repeat (for each step of episode):
$a$ ← action given by $\pi$ for $s$
Take action $a$ ; observe reward, r, and next state, $s'$
$V(s) \leftarrow V(s) + \alpha[r + \gamma V(s') - V(s)]$
$s \leftarrow s'$
until $s$ is terminal

**Fig 1. The procedural form of TD method**

Above figure specifies TD method completely in procedural form [5,6].

## 4. SYSTEM ARCHITECTURE

The human adaptive system this paper proposes is composed of the context recognition and human adaptive learning.

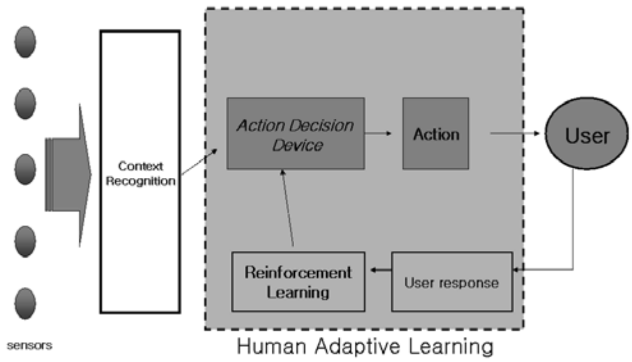


**Fig 2 Human Adaptive System Architecture**

The objective of Human adaptive learning part is that the machine is to be adjusted for a specific user. The Action Decision Device (ADD) decides which action the machine should do under the collected context information. This decision depends on the user's response (reward/penalty). The role of adjusting the factors according to the user's response is specially performed on the reinforcement learning part. Taking the action decided in the ADD causes the change of user's satisfaction extent (that is the performance of the system in this study). Then, the extent of satisfaction is transformed to a value as a reward and the value will be adjusted until the optimal state-action set is found. More precisely, we should define the state and action for a device (or appliance). Truly, home appliances don't have to need too much information. They just require 'current time' and 'user ID'. So, two parameters are used as the state. The Action set should be defined differently according to the function of each device. Here, we treat TV and Light as the device. For TV, action is the TV channel. So, the number of action is same to the number of TV channels. For the Light, action is luminosity.

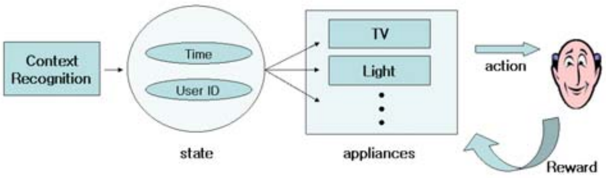


**Fig 3. An example of the system operation.**

As above definition, appliances receive the state information (Time and User ID) and they operate randomly at first time (or developer can set a basic operation according to the operation pattern of common users but we initially select a random action (operation)). Thereafter, if a user satisfies current operation, user will keep the state. Otherwise, user will change the action (of the appliance) manually. Evaluation of performance can be computed in two ways; direct and indirect way. Here, direct way means that the machine can compute the feeling of the user directly. That is, it is possible only when emotion recognition can be perfectly executed. Indirect way means that the machine computes the feeling of the user indirectly. In the case of TV, the performance means the time of having been kept at a specific channel. For example, when a user 'A' comes to living room at PM 8:00, TV automatically is turned on and shows the channel 9. At the time, if user doesn't change the channel, the action will get a reward. Otherwise, the action will get a penalty.

## 5. SIMULATION AND RESULT

Using the TD procedure (Fig 1), we implemented a simple example, a small markov process for generating random walks. This example is presented following figure.

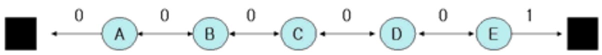


**Fig 4. Random walks**

This example was used by Richard.R.Sutton and since it is easy to simply implement and verify the TD method, we also applied it. In this problem, All episodes start in the center state, C, and proceed either left or right by one state on each step, with equal probability. Episodes terminate either on the extreme left or the extreme right. When an episode terminates on the right a reward of +1 occurs; all other rewards are zero [6]. According to the rule, Values learned by TD method after various numbers of episodes in the random walk example. Following figures presents the result. For reference, Y-axis represents the estimated value and X-axis represents the states (they correspond to the alphabets in the figure 2, respectively.). Also, in this example, $\alpha$ =0.1 (step-size parameter). Figure 3 shows the part of real code for random walk.

```
void CRandomWalkDlg::RW()
{
  //This function has a role of an episode.

        int Position=C;
        int dir,oldPosition=C;
        m_iindx=0;

        while(Position!=E)
        {
                dir=rand()%2;

                if(dir==RIGHT) //direction=Right
                        Position++;

                else //direction=Left
                {
                        if(Position!=A)
                        Position--;

                }
                Vf(oldPosition,Position,0); //Value function
                oldPosition=Position;
                Act[m_iindx]=dir;
                PosDat[m_iindx]=Position; //PosDat-->Position Data
                Reward[m_iindx]=0;
                m_iindx++;
        }

        if(Position==E) //when position becomes E
        {
                Reward[m_iindx]=1;
                Vf(Position,Position,1);
        }

}
```

**Fig 5. VC++ code for random walk**

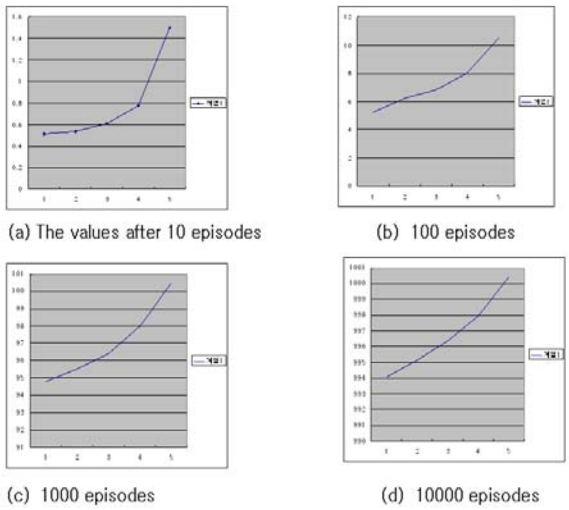


**Figure 6. Values learned by TD method after various numbers of episodes.**

The more episodes are executed, the closer the estimate gets to a linear line. This implementation and simulation verify the TD method works well. This result also reflects that the Values are sorted according to the distance from the goal and direct to direction position can get reward by TD method. This simulation is for random walk problem, but it is corresponded to the problem of our proposed system.

## 6. CONCLUSION

This paper presents that TD method is applied to the human adaptive devices for smart home with context awareness (or recognition) technique. For smart home, the very important problem is how the appliances (or devices) can adapt to user. There are many humans to manage home appliances (or devices). Therefore, managing the appliances automatically is difficult. Moreover, making the users be satisfied by the automatically managed devices is much more difficult. In order to do so, we can use several methods, fuzzy controller, neural network, reinforcement learning, etc. Though the some methods could be used, in this case (in dynamic environment), reinforcement learning is appropriate. Among some reinforcement learning methods, we select the Temporal Difference learning method as a core algorithm for adapting the devices to user. Since this paper assumes the environment is a smart home, we simply explained about the context awareness. Also, we treated with the TD method briefly and implement an example by VC++. Thereafter, we dealt with how the devices can be applied to this problem.

**ACKNOWLEDGMENTS**

This research was supported by the project of Developing SIC(Super Intelligent Chip) and its Applications under the program of Next generation technologies in 2000 of Ministry of Commerce Industry and Energy.

## REFERENCES

[1] A.K.Dey, G.D.Abowd and D.Salber, ”A context-Based Infrastructure for smart environments,” in Proceedings of the 1st International workshop on managing interactions in smart environment (MANSE’99), 1999, Dublin, Ireland, 114-128.

[2] Anind K. Dey and Gregory D. Abowd , “Toward a better understanding of context and context-Awareness,” in Proceedings of the CHI 2000 Workshop on The What, Who, Where, When, and How of Context-Awareness, The Hague, Netherlands, April 1-6, 2000.

[3] Pascoe.J, “Adding Generic Contextual Capabilities to Wearable Computers,” 2nd international Symposium on Wearable Computers, 1998, 92-99

[4] Pascoe.J, Ryan.N.S, Morse. D.R, “Human-Computer-Giraffe Interaction-HCI in the Field,” Workshop on Human Computer Interaction with Mobile Devices, 1998.

[5] M.Mitchell, Machine learning, Singapore,McGraw-Hill, 1997, pp.154-159.

[6] R.S.Sutton and A.G.Barto, Reinforcement Learning, A Bradford Book, Cambridge, 1998, pp.68-69.