Ontology Self-Learning and Service Recommendation in Smart Environment

Jun Qi
Research Associate
School of Computing and mathematics
University of Ulster
Newtownabbey, Northern Ireland

ABSTRACT

In order to be able to provide users with personalized service, the paper analyses the characteristics and attributes of the user, and set up the user personalized ontology model via context in smart environment. Meanwhile, a Bayesian probability algorithm is designed that can automatically adjust according to user's preferences change. Combining with user activity, the user ontology model provides the user with a real-time active service model in the purpose of meeting user's demands, and presents the implementation pseudo code, which shows the usability of user model and algorithm.

Keywords
Ontology self-learning; service recommendation; smart environment

1. INTRODUCTION

With the development of technology and family service robot in smart environment, service object behavior analysis and understanding are increasingly paid close attention. The main purpose is to analyze the service object behavior rule, decide behavior status and needs, and thus provide active service. Unlike fixed place with specific behavior monitoring, there are diversity, heterogeneous and dynamic characteristics in smart home. As such, personalized service recommendation is becoming a cumbersome task.

Recommendation is a specific type of application in knowledge discovery, which can provide active and accurate information for user through a variety of data analysis and intelligent learning technology. Traditional approaches of information acquisition is a "pull" process that user initiatively obtain required information. The "push" approach means user can be received information by system active recommendation.

The principle of personalized recommendation is a process to retrieve matching information based on the user model, and that recommend mutually via searching the user group with similar interest. The process includes personalization and automation. Personalization refers to take different service strategy in terms of different users, so that the users can be provided personalized service content. Automation is the system can actively push the information to users according to their interest. As such, it tends to reduce the searching time for the users as well as improve the efficiency of browsing.

2. RLATED WORK

Currently, there are two main approaches for service recommendation. One is semantic and grammar based approach, such as [1-3]. The approaches take advantage of context, user preference and quality of service (QoS) in smart environment to collect information. The principle of these approaches is in terms of grammar and semantic information and then finds the relationship among them to acquire user's needs for the purpose of service recommendation. The other one is ontology based approach [4-8]. The approaches mainly rely on logic relation and reasoning to recommend services, which use the third party technique, namely ontology to describe concepts and their relationship, and find the potential needs from ontology. As the result of the precise of ontology model that can reflect user's needs, hence it can obtain more effective information from this approach.

Ontology defines concepts relevant to a particular topic or area of interest [9], can centralize the representation of activity definition. It can standardize activity definition, which enables reuse of activity domain knowledge, as well as makes domain assumptions explicit. However, this approach suffers the difficulty of building a complex rule base as it must be manually constructed. Manual construction of rules is expensive, tedious and inflexible. To overcome this drawback, it is important to use automatic methods for building the ontology. Ontology learning refers to extracting ontological elements from input and building ontology from them [10]. Approaches such as [11-14] build concepts and their hierarchy. Others build different types of elements such as Text2Onto [15].

Personalized service recommendation system has been used in different fields and achieved very good effect, but it still has some limitations. First, the system lacks of intelligence and personalization. They exploit to use the same recommendation strategy to all the users, and thus incapable to realize personalized recommendation towards different users. Second, most recommendation system technology is mainly adopted contend based retrieval approach which search services according to keywords rather than semantics, leading to imprecise results. Third, most websites often obtain personalized information from user registration and thus overly dependent on the current user behavior instead of adjusting users' preference via observing and recording their behaviors every time. Finally, efficiency of the real-time is unsatisfying due to the complex recommendation algorithm such as recommendation algorithm based on neural network, genetic algorithm, etc., involving a large number of user data and service information. Therefore, it is necessary to establish a reasonable and effective service model, and apply a more rapid recommendation algorithm for service recommendation.

3. SYSTEM IMPLEMENTATION
3.1 System Structure
In order to provide users with personalized service, as shown in figure 1, four layers is designed in system architecture. The bottom is the physical layer, which contains three main elements: user, environment and device. User refers to any people using applications, environmental context refers to the

context the user in, and application scenario denotes the instance for each user. The second layer of system is the model layer, including user model and context model. The third layer of system is for storage so it can update user model. At the same time, the context is an important factor in model layer, because it is the context for user's specific location and time. This module can collect sensor data and reasoning according to different users.

Personalization layer includes three modules: user model self-learning; reasoning engine and reasoning rules, whose role is transmit user's personalized service to the application layer, and update it according to user's demand. The system also can update user model, then use reasoning engine to infer knowledge as well as perform reasoning results based on semantic ontology. Through the process, the recommendation result can meet the user's personalized preference. A series of rules comprise of the system knowledge base so that provide personalized services according to users' needs. Because in order to adapt to this change, when the user's preferences change, the system will require the ability of self-learning.

On the top is user interaction interface which can provide personalized information according to the context information. Self-learning system monitors user activity, and then adds to user ontology model according to the system's feedback. It tends be more convenient to provide services to user by combining the user information model and context information.
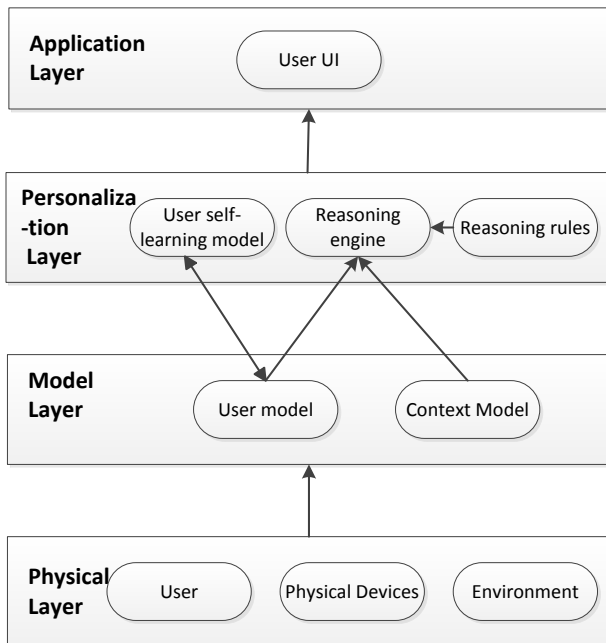
Fig 1: Service recommendation system structure

3.2 Recommendation Algorithm

Update for user ontology knowledge base plays a key role to improve the efficiency of context-aware systems and semantic completeness. The main approach is to construct a domain knowledge base called "class ontology", and thus new semantic information can be added to it. The advantage of the approach not only constantly improves the ontology base, but also enhances the availability of ontology library itself, ontology of learning process as shown in figure 2.
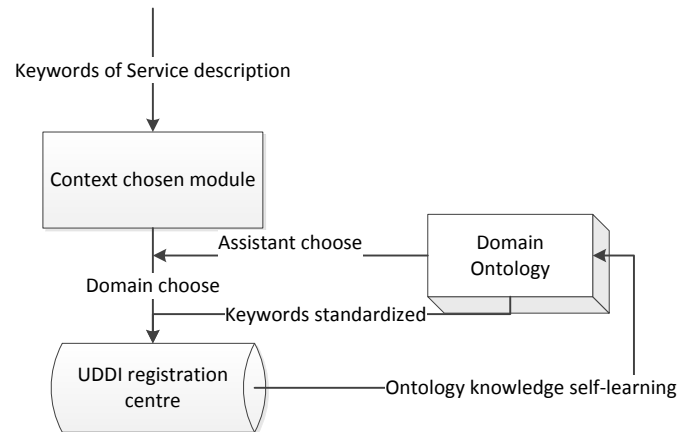
Fig 2: The process of ontology self-learning

When publishing services, domain ontology can assist selection for context field module to determine the service field. Domain ontology, meanwhile, can standardize the concept of service description keywords, which add the new concepts to the original domain ontology, and then complete the semantic integrity of its self-learning function expedition in context awareness.

Hence, it can be seen that the process of user ontology self-learning is a process of continuous extension of knowledge base, as well as addition and completion in the existing domain ontology, which needs to keep consistency between original ontology and ontology self-learning mechanism. In other words, the process of ontology self-learning should avoid redundancy concept definition. The process of expansion of domain ontology is mainly composed of following steps:

Step 1: insert services to be as a new ontology concept (owl class) into current ontology base. As the uniqueness of concepts in the ontology, that is to say it is impossible to be the exact same name for one concept. Thus when the name in the domain ontology is the same as the new service name, namely, < RDFS: label > tags are the same value, the original name of the ontology concepts will be automatically converted to "concept name + existing same concept numbers in ontology. If there is not same concept name, the new concept will be the same as its service name instead of the name of the same name as the service. As such, it can ensure the uniqueness of the service concept in domain ontology.

Step 2: insert new service keywords into domain ontology as a new ontology attribute. There are two situations: (1) due to the uniqueness of this attribute, if current ontology have already had the ontology attributes to be added, the original one would be reused, and a label < RDSF: domain > tag and sub tag < owl: class > would be added. (2) When it is unnecessary to add a new attribute, system add a new ontology attributes to current domain ontology library TAB, described the properties as the name < RDFS: domain > tag.

This paper exploits Bayesian probability algorithm calculates the probability for each service resource, and then checks whether the probability can meet user's needs in terms of user's selection in real situation.

If there are n service resource in context, denoted as $S = \{S_1, S_2, \cdots, S_n\}$, and these n service resource belongs to type m, denoted as $T = \{T_1, T_2, \cdots, T_m\}$, we can extract k of them as the influenced context, denoted as $C = \{C_1, C_2, \cdots, C_k\}$, in which C_I has j value, denoted as $C_i = \{C_{i1}, C_{i2}, \cdots, C_{ij}\}$. Assumed that each note X_I has super note $P(X_i)$ in context service model, so the probability for each note is $p = (X_I \mid P(X_i))$, and the joint probability distribution of service resource set S is:

$$p(s_1, s_2, \cdots, s_n) = \prod_{i=1}^{n} p(s_i \mid P(s_i)) \qquad (1)$$

According to the Bayesian probability rule, each service resource value can also expressed as

$$p(s_i \mid c_1, c_2, \cdots, c_n) = p(s_i \mid P(s_i)) * p(P(s_i \mid P(P(s_i)))) \qquad (2)$$

In the process of service selection, if user adjusts the selection for a long run, it tends to illustrate that the habit of use perhaps changed towards to context service, and thus it should be change the parameter of service $p = (X_I \mid P(X_i))$.

Assumed that current training data set A contains n data, denoted as $A = \{Y_1, Y_2, \cdots, Y_n\}$, and current services set is ω, service note i has sub note k, so the value of super note set j is ω_{ijk}. The new added user preference is:

$y^T = \{C_{11}, C_{21}, \cdots, C_{n2}, S_1, S_2, \cdots, S_m\}$, the new added super note value would be ω_{ijk}^T, in which parameter ω can be expressed as a time period of user preference. The self-learning probability of threshold adjustment algorithm is as follow:

$$\omega_{ijk}^T = \begin{cases} \dfrac{1}{N_{ij}^T} + (1 - \dfrac{1}{N_{ij}^T})\omega_{ijk}^{T-1}, & P(Pa_i^j \mid y^T) = 1 \text{ and } P(Z_i^k \mid y^T) = 1 \\[2mm] (1 - \dfrac{1}{N_{ij}^T})\omega_{ijk}^{T-1}, & P(Pa_i^j \mid y^T) = 1 \text{ and } P(Z_i^k \mid y^T) = 0 \\[2mm] \omega_{ijk}^{T-1} \end{cases}$$

$$(3)$$

Formula 3 illustrates that in the new added information data y^T, if the sub note x_i of number j and k, it will increase the value of ω_{ijk}; if there is only number j of sub note x_i, it will decrease the value of ω_{ijk}, and other situation will keep the same.

User model self-learning algorithm is simple and strong feasibility, as the parameters of the existing and newly added data are taken into account, it can reduce the complexity of threshold adjustment.

4. Implementation of service recommendation

To implement the user ontology self-learning model, the first step is allow users to fill out the interest preference form, creating a static user model, and then it will be change dynamically according to the user choice in actual circumstances of to ensure that the final preference model is the user's actual preferences. System will be reference for users who have established ontology, to provide users with active services, including the following two cases:

If these concepts do not appeared in user ontology model, while the user chose them in practical application, it illustrates that the new services is the user's preferences, so the system will add these new concepts to the concept of user ontology. For example, if the user used to dim the lights when watching TV and then if the system recognizes this activity, it will search it in user model and extract its preferences, and then take initiatively adjust lighting levels for the user. Assumed that there no concept of adjust light brightness in previous user ontology model, then the system will record the user this activity, and through many times of training observation, determine whether it is the real user preferences, if it is, the concept will be added to user ontology model.

In this case there need to set up a concept of time interval and probability threshold value P, if the user does not choose the service for a long period in user ontology model, then increase the probability of the service, until the probability is bigger than the threshold value of P. The system will add the new concepts to ontology repository.

If a user does not choose the concept in the user ontology for a long time, it indicates that the user's preferences tend to changed, so the system will delete the concepts user have not chosen. If the concept to be deleted is sub concept, then delete it directly, while if it is a super concept, and then delete this entire concept and its sub concepts. The pseudo codes of two models are as follow:

AlterUserProfile()

{ For every s   // s is system service concept running automatically

{ select(s);   // if user chooses a service

p1=+w; // then adjust the service's probability}

if (p1>p)   // if probability is bigger than threshold value

{ Interest(s) // user perhaps interested in the service

Insert(c，UserProfile); // put all concepts of service into user model }

If Select(s)=0 // if user does not choose service s

{time(s) ++   // then add time of this service

if (time(e)>t)// if time exceeds the time T

{if (1eaf(c)==1)   // if concept service is a sub concept

delete(c, userprofile); // then delete the concept from user model

else {For every node

if (child(k,c)==l) // if k is k a sub concept of concept c

delete (k, UserProfile);// then delete all the notes of k from user model

delete (c，UserProfile); //and delete its super node}

It can realize the dynamic changes of the user after the correction of user preference model, the system will be updated according to actual condition of user model, so that it tends to meet user's demand in time, greatly improve the accuracy of the service.

5. Bayesian Algorithm Evaluation

The algorithm simulates four users' ontology model. The vector expression that original services provided by the system and the services after the changes user's preference are as follow:

Original service vector:

$$S_{before} = (S_1 = x_1, S_2 = x_2, S_3 = x_3, \cdots S_n = x_n)$$ Service vector after preference changed:

$$S_{after} = (S_1 = y_1, S_2 = y_2, S_3 = y_3, \cdots S_n = y_n)$$ In which x_i

and y_i are the cased that services chosen. The algorithm of the Euclidean distance is exploited to calculate the distance between the two vectors:

$$D = \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2 + (x_3 - y_3)^2 + \cdots + (x_n - y_n)^2} \qquad (4)$$

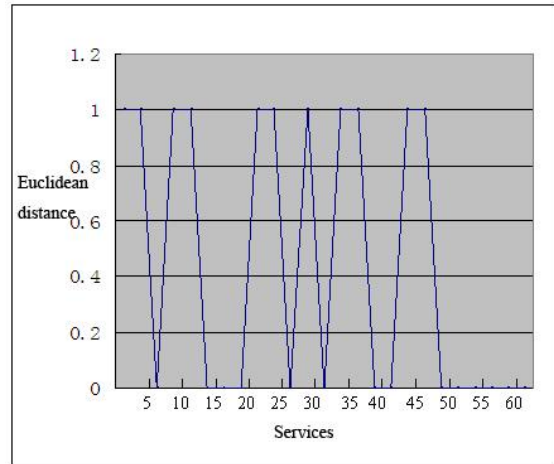Figure 3 shows the results of services provision for four users' preferences changes.
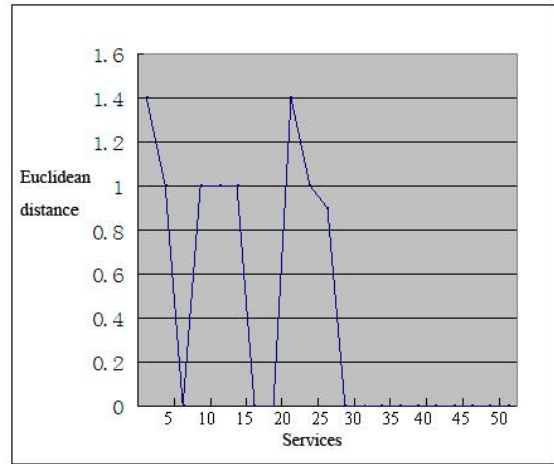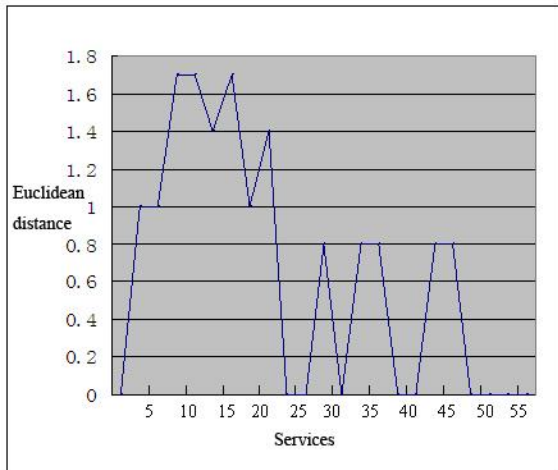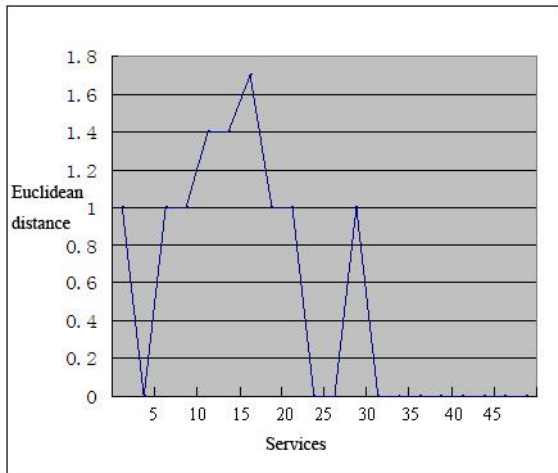

Fig 3: Euclidean distance of user's preference changes

The horizontal axis system is the times of recommended service, and the ordinate demotes the Euclidean distance between the vectors of original services after user's preference change. zero means services original services is the same as the real user preferences, when the distance value becoming larger, it shows that users' preferences changed gradually, and four user preferences are changed, the services recommendation system to perform a self-learning algorithm, this section will provide the service to change so that the active service provided by the tend to change after the user preferences

6. CONCLUSION

The pervasive computing is a new popular research field; its goal is to provide users with transparent service at anytime and anywhere. How to implement knowledge sharing in the pervasive computing environment, reuse and automated reasoning more effectively and conveniently for users is becoming a heated research in recent years. This paper presents a personalized service for users in the smart home. Properties and characteristics of the user is analyses in detail, also established a relatively perfect personalized ontology model. And a new algorithm based on Bayesian probability is implemented on services recommendation that can adjust automatically in terms of user preferences' change. However, the paper only described the functionalities of simple active service. As a huge amount of service source in the real world, it is important to manage the services and improve the efficiency of the services the work in the further.

7. REFERENCES

[1] Liu D, Meng XW, Chen JL. A framework for context-aware service recommendation. In: Proc. of the 10th Int'l Conf. on Advanced Communication Technology (ICACT 2008). Institute of Electrical and Electronics Engineers, 2008.

[2] Lee CO, Lee MY, Han DS, Jung S, Cho J. A framework for personalized healthcare service recommendation. In: Proc. of the 10th Int'l Conf. on e-Health Networking, Applications and Services (HealthCom 2008). Institute of Electrical and Electronics Engineers. 2008.

[3] Chen ZK, Shao Z, Xie ZJ, Huang XD. An attribute-based scheme for service recommendation using association rules and ant colony algorithm. In: Proc. of the Wireless Telecommunications Symp. (WTS). Institute of Electrical and Electronics Engineers, 2010. 6.

[4] Lo CC, Cheng DY, Chen CH. A semantic Web methodology for situation-aware curative food service recommendation system. In: Proc. of the Int'l Conf. on Computer Science and Software Engineering. IEEE Computer Society, 2008.

[5] Liu D, Chen JL, Guo J. An ontology-based LBS system. In: Proc. of the 4th Int'l Conf. on Wireless Communications, Networking and Mobile Computing (WiCOM 2008). IEEE Communications Society, 2008.

[6] Xia HX, Yoshida T. Web service recommendation with ontology-based similarity measure. In: Proc. of the 2nd Int'l Conf. on Innovative Computing, Information and Control (ICICIC 2007). Institute of Electrical and Electronics Engineers, 2007. 412.

[7] Zhang L, Meng XW, Chen JL, Duan K, Peng Y. Personalized service recommendation algorithm. In: Proc. of the 2nd IEEE Int'l Conf. on Computer Science and Information Technology (ICCSIT 2009). Institute of Electrical and Electronics Engineers, 2009. 522−526.

[8] Yuan Y, Cao ZQ, Hou ZG, Tan M. Dynamic programming field based environment learning and path planning for mobile robots. In: Proc. of the 8th World Congress on Intelligent Control and Automation (WCICA 2010). Institute of Electrical and Electronics Engineers, 2010. 883−887.

[9] Karoui, L., Aufaure, M., and Bennacer, N. 2004. Ontology Discovery from Web Pages: Application to Tourism. In ECML/PKDD 2004: Knowledge Discovery and Ontologies KDO-2004.

[10] Shamsfard, M. and Barforoush, A. A. 2003. The state of the art in ontology learning: A framework for comparison. The Knowledge Engineering Review, Vol. 18 No.4 pp. 293-316.

[11] Sabou, M., Wroe, C., Goble, C., and Mishne, G. 2005. Learning Domain Ontologies for Web Service Descriptions: an Experiment in Bioinformatics. In Proceedings of the 14th International World Wide Web Conference (WWW2005), Chiba, Japan.

[12] Cimiano, P., Hotho, A., Staab, S. 2005. Learning Concept Hierarchies from Text Corpora using Formal Concept Analysis. JAIR - Journal of AI Research, Vol. 24, pp. 305- 339.

[13] Davulcu, H., Vadrevu, S., Nagarajan, S., and Ramakrishnan, I. 2003. OntoMiner: Bootstrapping and Populating Ontologies from Domain Specific Web Sites. In IEEE Intelligent Systems, Vol. 18, No. 5, pp. 24-33.

[14] Hazman, M., El-Beltagy, S. R., and Rafea, A. 2009. Ontology Learning from Domain Specific Web Documents. In International Journal of Metadata, Semantics and Ontologies, Vol. 4, No. 1-2, pp: 24 − 33.

[15] Cimiano P., and Vaolker, J. 2005. Text2Onto − A Framework for Ontology Learning and Data-driven Change Discovery. In: Montoyo, A., Munoz, R., Metais, E. Proceedings of the 10th International Conference on Applications of Natural Language to Information Systems (NLDB), Lecture Notes in Computer Science. Alicante, Spain: Springer.