OWL-Based Context-Dependent Task Modeling and Deducing

Hongbo Ni[1], Xingshe Zhou[1], Zhiwen Yu[2], Kejian Miao[1]

1: School of Computer Science, Northwestern Polytechnic University, China
{nihb, zhouxs, kjmiao}@nwpu.edu.cn
2: Academic Center for Computing and Media Studies, Kyoto University, Japan
yu@ccm.media.kyoto-u.ac.jp

Abstract

In the near future, homes are envisioned to be equipped with numerous intelligent communicating devices. Such smart home needs to exhibit highly adaptive behavior to meet the inhabitants changing personal requirements and operational context of environment. To achieve this, smart home application should focus on the inhabitant's goal or task in diverse situation, but not the various complex devices and services. This paper proposes a context-dependent task approach to meet the challenge. The most important component is task model which provides an adequate high-level description of user-oriented tasks and their related contexts, and in such model multiple entities can easily exchange, share and reuse their knowledge. An OWL-based ontology to hierarchically model context-dependent task is presented, which facilitates sharing and reusing of smart space knowledge and logic inferences. The conversion of OWL task ontology specifications to the First-Order Logic (FOL) representations is described. Finally, the performance of FOL rule based deducing in terms of task number, context size and time is evaluated.

1. Introduction

Currently, houses are being networked, bringing the internet to the home and allowing new devices and services. In the future home environment, the user will be overwhelmed by a multitude of devices with complex capabilities, different access network interfaces and different multimedia and control services. In order for smart homes to achieve their promise of significantly improving the lives of families through socially appropriate and timely assistance, they need to sense, anticipate and respond to activities in the home.

Recently, task computing (TC) paradigm has been regarded as a promising way for pervasive computing environments [9, 10]. The key idea behind TC is that the system should take over many low-level management activities of computing resources, so that users can interact in a pervasive computing environment in terms of high-level, user-centric tasks (i.e. WHAT) that they wish to accomplish, rather than on the actual mechanisms (i.e. HOW) to perform those tasks. Another attraction of TC is its ability to manage the tasks in runtime by having the capability to suspend from one environment and resume the same task later in a different environment. This is made possible by the way that a task is often specified independent of both the actual underlying computing services and resources, and its surrounding environmental conditions. The underlying TC software infrastructure achieves this feature by providing the necessary support to maintain and manage task-associated context information in the form of task states (e.g. user preferences) from one environment to another. Despite its promises, however, there are a number of challenges in TC that have yet to be addressed fully (see Related Work section). One of these key challenges lies in the modeling of user-centric tasks, its context information, and how a task can be associated with the underlying service. In this paper, we propose such a task modeling solution and the approach for supporting context-dependent task definition. In particular, we recognize the importance that in a smart pervasive computing environment, the nature of a user's task is closely associated with the context of external physical environment, as well as the particular user's profile.

Our key contributions in this paper include defining a context-dependent task model, proposing a similarity-based matching algorithm for the discovery of *active-tasks* based on prevailing context information, and presenting the detailed design of the context-dependent model and logic-based task deducing scheme. Through performance analysis, we will show a quantitative evaluation for context-dependent deducing in pervasive computing environments.

The rest of this paper is organized as follows. Section 2 summarizes the current related works in TC. In section 3, we describe the design of our context-dependent task model. Section 4 presents FOL-based task deducing to enhance situation awareness and service adaptation. Section 5 describes the prototype implementation and the performance evaluation. Finally Section 6 concludes this

21st International Conference on
Advanced Information Networking and Applications Workshops (AINAW'07)
0-7695-2847-3/07 $20.00 © 2007 IEEE

paper.

2. Related Works

There are a number of related works in pervasive computing towards context-aware. However, most of them attempt to use context-awareness in pervasive environments that focus on the physical aspects of the user context (e.g. number, time, location) and the environment context (e.g. device proximity and lighting condition) [1]. This is despite the fact that many authors have long recognized the importance of using the cognitive aspect of the user context (such as users' goals, preferences and emotional state etc.) [2], to date, very little work has been done to develop such models and apply those models in building context-aware applications [3]. On the other hand, user context modeling has long been adopted for use in: recommendation based systems [4, 5], adaptive information retrieval systems [6], and systems for coaching/teaching users [7, 8]. User preference and user historical information has been well utilized [9]. However, the research in exploiting both the physical and cognitive user context in those fields is still in the early stage. Our proposed modeling solution attempts to capture both these two categories of context information in defining a task.

Earlier research in the TC area has defined task-driven computing [10] and task computing [11], and demonstrated applications in a computing-oriented environment. These earlier works simply treated a task as merely binding together a set of relevant computing applications (called virtual services) in a hierarchical or otherwise fashion, with task defined as the top-level virtual service. The fundamental problem with this approach is that it is too application-centric. Since these applications are only a means to carry out a task, they are not suitable to represent the task itself, or to capture user-related context information. Furthermore, all resources within the task computing are realized as services available to the task computing clients, and it is up to the client to co-ordinate the use, and to monitor these resources. In these systems, the manually configured/defined tasks can then be executed as atomic tasks or can be used for further task compositions. Unfortunately, this is a time-consuming activity and usually means the user needs to be familiar with the environmental conditions and the available computing resources himself. This laborious process could often take longer than executing the intended tasks. This also means the user must have a reasonably in-depth knowledge of how to configure services based on his/her requirements. To solve this bottleneck, we choose instead to look beyond the applications, and focus on the user's goals and prevailing context, using the notion of context-dependent task model, couple this with an automatic TC execution framework. In another word, with our proposed solution, TC can truly be a user-oriented computing model that lets users accomplish complex tasks on-the-fly over an open, dynamic and distributed set of underlying resources/services automatically.

3. Context-Dependent Task Model

The context-dependent task modeling approach uses the abstraction of tasks in order to separate logical relations of relevant items from the services realizing and fulfilling the intended goals. This approach is able to support human requirements and preferences better because of the following reasons:

(1) Task definition models human preferences and requirements better than service-orientation models adopted in earlier works;

(2) Separation of tasks and services would allow for greater flexibility of changing the tasks without changing the services and vice-verse;

(3) It hides the complexity of compositing embedded services in pervasive environment from the users.

As mentioned above, a task is tightly related to the current context, this means that a task is highly context-dependent and this relationship must be captured in the model definition. This interpretation can for example be suitable for the development of a smart home environment, where complex cognitive tasks (e.g. "to relax at home") serving a user's need and preference are to be performed. This is in contrast to the more straight-forward computing task (e.g. "to do a presentation") in the earlier TC work [11]. We elaborate this in the rest of this section.

3.1 Hierarchy of Tasks and Contexts

In this section, we attempt to generalize this notion to cover a wide range of possible task definitions. Generally speaking, a task can be as broadly defined as, *serving the elderly*, or as narrowly defined as *making tea*. Similarly, how different task-specific context information will apply depends on the task itself. The actual scope of the task is therefore defined by its intended use, and thus to generalize, we can further define a set of related tasks and their dependency on context information in a hierarchical manner as seen here in **Fig. 1**.

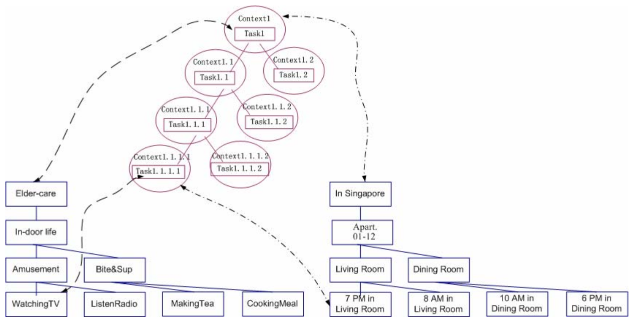

Fig.1 Hierarchy of Contexts and Tasks

To elaborate, in **Fig.1**, Task1 (e.g. "eldercare"), for instance, can be refined by Task1.1 (e.g. "activities at home") and Task1.1 can also be further refined by Task1.1.1 (e.g. "relaxation"). At the lowest level there are real tasks which can not be decomposed, such as Task1.1.1.1 (e.g. "watching TV"). As shown in Fig. 1, three types of tasks exist in the hierarchy: an overall and generic task (root node), composite tasks (intermediate nodes) and atomic tasks (leaf nodes).

On the other hand, the context relevant to individual tasks can be similarly defined using the task hierarchy. Hence, in **Fig. 1**, Context1 (e.g. "in Singapore"), would be relevant to Task1. Similarly, Context 1.1.1.1 (e.g. "7:00 pm in living room") is related to Task1.1.1.1. In summary, whenever a task is decomposed into more objective sub-tasks, the related contexts will similarly be more and more specific with a sub-task automatically inherits the context of its parent tasks.

3.2 Formalization of the Context-Dependent Task Model

We model tasks and their relations on the top of the context-dependent task hierarchy explained above, where each can be further decomposed into a set of sub-tasks (in the case of a composite task), or in the case of atomic task, a set of sequential activities. A task can be described by a union of the following vocabulary:

Task-ID (TI): a unique identifier of a task in a pervasive computing application;

Task-Name (TN): a string to distinguish a task and easy to understand for a user;

Condition (C): a set of preconditions, or context information, that must be met before the task can be performed. The condition is specified in the form of parameters.

Priority (Pr): this field denotes the importance and exigency of a task to further facilitate the execution, suspension and re-scheduling of tasks at runtime. For tasks that have the same priority their relative importance will be determined by the priority of their respective parent-tasks.

Task-Contract (TC): this is a crucial element for our task definition. Task contract has two roles: one is to discovery necessary resources and services for the task; the other is to organize and guide the steps of executing a task. The detail of TC has been further elaborated in [12].

In summary, based on our explanation above, each task is represented as a nested 5-tuple, for example, Task1.1 can be represented as follows:

T1.1= (TI1.1, TN1.1, C1.1, Pr1.1, TC1.1).

4. Construction of Context-Dependent Task Model Using OWL

In this section we present an extensible Context-Dependent Task ONtology (CDTON) for modeling context in pervasive computing environments.

There are various environment types such as homes, offices, workplaces and vehicles, and we do not aim to completely model all contexts in different types of smart spaces. Instead, we define an Upper-Level Context Ontology (ULCO) [13] to provide a set of basic concepts that are common in different environments. Among various entities, we identify 3 classes of real-world objects (i.e., user, location, computing entity) and anther class of conceptual objects (i.e., task) that are most important to characterize a pervasive environment. Therefore, we choose to model these objects as top-level classes in ULCO. For example, given a location, we can acquire related contexts such as noise, weather, the number of people inside, etc.

To keep the context model customizable to a particular pervasive environment, it is intended to complement the classes defined in ULCO. In case that a new application needs additional classes that further specify the existing ones, they can be inherited from the classes of ULCO, forming a so-called Extended Context-Dependent Task Ontology (ECDTON) (see **Fig.2**). In this way, developers can easily build detailed context models for newly-setup smart spaces. Moreover, the use of UCLO can support better interoperability between ECDTONs. Different ECDTONs will be able to interoperate by virtue of shared terms and definitions.

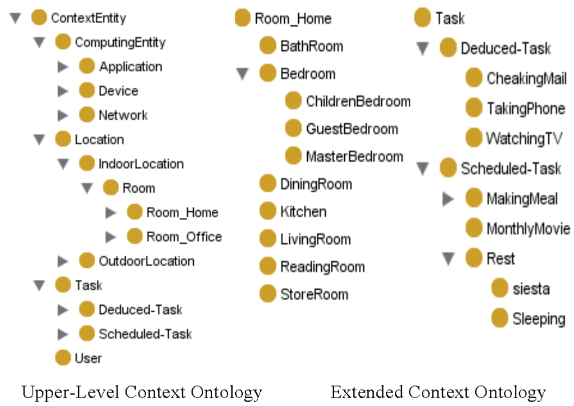

Fig. 2.Ontologies Dedinition in Protege

4.1 Semantic Expression of the Real Entities

With our model, all the entities in real world are represented as ontology instances and associated properties (so-called entity markups) that can be easily interpreted by applications. Real-world entities often originate from diverse sources, leading to dissimilar approaches to generating different markups. Let us take examples of the contexts involved in the smart home scenario. Some of the contexts (e.g., name of a person, gender, and mobile-phone) have relatively slow rates of change. Markups of these contexts are usually generated by users. For example, we provide a JavaScript application that allows users to online create profiles based on the ontology class User. Following example shows the context markup that describes *nihongbo*.

```
<User rdf:ID="nihongbo">
<user_status
rdf:datatype="http://www.w3.org/2001/XMLSchema#strin
g"
>normal</user_status>
<studentOf rdf:resource="#zhouxingshe"/>
<family_name
rdf:datatype="http://www.w3.org/2001/XMLSchema#strin
g"
>Ni</family_name>
<given_name
rdf:datatype="http://www.w3.org/2001/XMLSchema#strin
g"
>Hongbo</given_name>
<gender
rdf:datatype="http://www.w3.org/2001/XMLSchema#strin
g"
>M</gender>
<mobile_phone
rdf:datatype="http://www.w3.org/2001/XMLSchema#strin
g"
>13891866713</mobile_phone>
</User>
```

On the other hand, some other contexts (e.g., location, current time, noise level, door status) are usually provided by hardware or software sources. The marking up of these contexts needs to be performed by automated programs due to the high rates of change. Let consider the RFID indoor location system that tracks users' location by detecting the presence of body-worn tags. When *nihongbo* enters *DiningRoom*, the RFID sensor detects his presence and composes the context markup as described bellow.

```
<User rdf:about="#nihongbo"><locatedIn rdf: about
="#DiningRoom"/> </User>
```

Since each OWL instance has a unique URI, entities markups can link to external definitions through these URIs. Forexample, http://www.dcel.nwpu.edu.cn/SemanticSpace#NiHong-bo refers to the user defined above, and the URI http://www.dcel.nwpu.edu.cn/SemanticSpace#DiningRoom refers to a room that is also defined elsewhere.

4.2 Owl-Based Context-Dependent Task Deducing

When taking a formal approach to model context and task, context can be processed with logical deducing mechanisms to deduce the user's task. To explain the role of task deducing with the context-dependent model, we present a smart home scenario in which the system can help the user with his (her) current task. By defining preference profiles, users can customize the system behaviors to provide smart service supporting the deduced task. For example, when the user is to sleep (*Task*) in the bedroom, incoming calls are forwarded to voice mail box and the drape in the bedroom is closed; when the user is to watch TV (*Task*) in the living room, the blind in the living room is closed and air-condition is turned on if required. Obviously, the context related to a task (so-called high-level context) can not be directly acquired from sensors; it is deduced from sensor-driven, low-level context such as physical location and environmental information.

We choose to implement task deducing by using first-order predicates. The structure of the first-order predicate has three fields - a subject, an object, and a verb. For example, the physical location context "Ni is located in the DiningRoom" can be described as *(Ni, locatedIn, DiningRoom)*. The following is some simple examples showing the Task deducing using OWL rules and related adaptive action expressed in natural language of the system:

Task 1:
Noonbreak (?u locatedIn Bedroom)∧((Time greater Than (currentTime(),13:00:00))∧((TimelessThan(currentTim(), 13:30:00)) ∧ (BedSensor SensorStatus ON)
⇒ (BedroomDoor to be Closed)∧(Bedroom drape to be Closed) ∧(mobile_phone to be silent and forwarded)∧ (…)

Task2:
Makingsuper (?u locatedIn Kitchen)∧(Time TimePeriod Evening) ∧(Kichen lightLevel High)
⇒ (ElectricOven to be ON)∧(Lampblack Device to be ON) ∧(…)

Task3:
WatchingTV(?ulocatedInLivingRoom)∧(SofaSensor SensorStatus ON)∧(TVSet status ON)
⇒ (TV volume o be adjusted according to TimePeriod) ∧LivingRoom drape to be Closed) ∧ (…)

5. Performance Evaluation

In this section, we will present results of our preliminary experiments with OWL-based task deducing. The objectives of these experiments are to conduct a quantitative feasibility study for logic deducing in pervasive computing environments, and provide useful information for the implementation of context-dependent task deducing.

We used our prototype implementation of first-order logic based task deducer to carry out experiments. Task deducer was built using Jena2 Semantic Web Toolkit [14], which supports rule-based inference over OWL/RDF graphs. With the Protégé and MySQL, we have built a pervasive entities database based on the context-dependent task model that simulated a smart home environment. The current version of CDTON contained 167 OWL classes (or 720 triples) that could be regarded as a small-scale context dataset. The experiments were conducted on two Linux Platforms: PA (P4/2.66GHz, 512M RAM) and PB (P3/800MHz, 256M RAM). The task deducer tested is associated with the DL rule set consisting of all 111 axioms entailed by OWL-Lite. We adopted two rule sets including 10 and 20 first-order logic task deducing rules respectively in the experiments.

The results of the experiments are shown in **Fig. 3**. It is obviously that the key factors influencing the performance are the number of tasks and triples besides the hardware configuration. Furthermore, the former (the number of tasks) is more sensitive to the result than the latter (hardware configuration), and the resulting difference of the hardware will be distinct when the triples change to be larger.

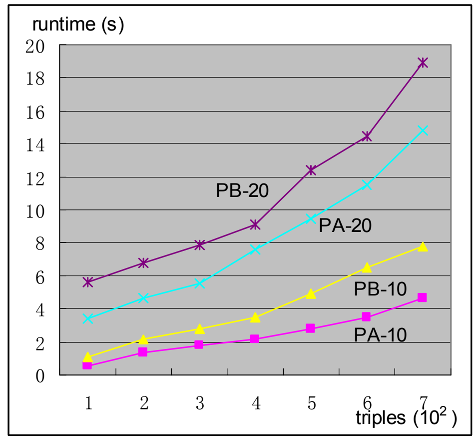

Fig. 3. Performance Result of Task Deducing

6. Conclusions

This paper proposed a context-dependent task model suitable for pervasive computing. The task model focuses on modeling user-centric goals and requirements, and not the numerous complex underlying system services. Based on the context-dependent task model, we developed ontologies for real entities in a smart home by using the OWL. We also presented the context-dependent task deducing by using first-order predicates. For future work, we plan to apply the context-dependent task model in several applications in smart home, and build the prototype on the scalable and standard OSGi [15] platform.

ACKNOWLEDGMENT

This work was partially supported by the High-Tech Program of China (863) (No. 2006AA01Z198) and the Innovation Fund of Northwestern Polytechnical University of China (No. 2006CR13)

7. References

[1] P.Castro and R. Muntz, “Managing Context Data for Smart Spaces,” IEEE Personal Communications, 2000, vol. 7, pp. 44-46.
[2] Paul Prekop and Mark Burnett, “Activities, Context and Ubiquitous Computing, Computer Communications”, 2003, vol. 26, no. 11, pp. 1168-1176.
[3] Heckmann, D., "Introducing Situational Statements as an integrating Data Structure for User Modeling, Context-Awareness and Resource-Adaptive Computing", ABIS Workshop on adaptivity and user modeling in interactive software systems, 2003.

[4] Tiffany Y. Tang and Gordon McCalla, "Smart Recommendation for an Evolving E-Learning System", AIED2003, Vol10, PP. 699-711.

[5] Zhiwen Yu, et al, "Supporting Context-Aware Media Recommendations for Smart Phones", IEEE Pervasive Computing, Vol. 5, No. 3, July-September 2006, pp. 68-75.

[6] Charles Seger Jakobsen and Simon Warthoe, "Adaptive Design Implications for Knowledge Organization and Information Retrieval Systems", Nord I&D, Knowledge and Change, 2004, PP. 58-61.

[7] Kay J., "The um toolkit for cooperative user modeling", User Modeling and User-Adapted Interaction, 1995, Vol 3, pp. 149-196.

[8] Shareef A. F. & Kinshuk, Student Model for Distance Education System in Maldives. In A. Rossett (Ed.) Proceedings of E-Learn 2003 (November 7-11, 2003, Phoenix, Arizona, USA), Norfolk, VA, USA: AACE, pp. 2435-2438.

[9] Zhiwen Yu, Daqing Zhang, Xingshe Zhou, and Changde Li, "User Preference Learning for Multimedia Personalization in Pervasive Computing Environment", in Proc. of 9th Int. Conf. Knowledge-Based Intelligent Information & Engineering Systems (KES'2005), , September 2005, pp. 236-242.

[10] Zhenyu Wang and David Garlan, "Task-driven computing", Technical Report, NO. CMU-CS-00-154, http://www-2.cs.cmu.edu/~aura/docdir/wang00.pdf, Carnegie Mellon University, May 2000.

[11] R. Masuoka, et al., "Task computing - the Semantic Web meets Pervasive Computing", 2nd International Semantic Web Conference, ISWC2003, pp. 866-881.

[12] Hongbo Ni, Xingshe Zhou, Daqing Zhang, Ngoh Lek Heng: Context-Dependent Task Computing in Pervasive Environment. UCS 2006: 119-128

[13] T. Gu, X. H. Wang, H. K. Pung and D. Q. Zhang, "An OWL-based context model in intelligent environments," Communication Networks and Distributed Systems Modeling and Simulation Conference (CNDS'04), January, 2004, San Diego, California.

[14] Jena2 Semantic Web Toolkit: http://www.hpl.hp.com/semweb/jena2.htm.

[15] D. Marples and P. Kriens. "The Open Services Gateway Initiative: An Introductory Overview", IEEE Communications Magazine, Dec. 2001, pp. 110-114