ICAT 2004

A Temporal Planning System for Ubiquitous Service Agents*

Jung-Hwa Choi, Young-Tack Park, Hee-Dong Ko[+]
School of Computing, Soongsil University, Seoul, South Korea
[+]IMRC, Korea Institute of Science and Technology, Seoul, South Korea
cjh79@ailab.ssu.ac.kr, park@computing.ssu.ac.kr, [+]ko@kist.re.kr

Abstract

We argue that representation of temporal relationships (e.g., before, during, etc.) are necessary for planning contexts in ubiquitous environments. Many contexts require a richer notion of time in which actions can overlap and have different durations. This paper proposes a Temporal Planning System (TPS) that represent actions of different duration for temporal planning in ubiquitous environments. Ubiquitous service agent explores the use of Semantic Web language (i.e., OWL) for defining and publishing a context and for reasoning over such service. This paper describes defined planning steps in focusing research of temporal relationships to express dynamic context.

Key words: Ubiquitous System, Planning, Temporal Reasoning, Semantic Web, OWL, Service Agent

1. Introduction

Ubiquitous computing was advocated by Mark Weiser[1] of Xerox PARC in 1988. This research aims at provision of localized context aware information services to user. To offer such information services, an intelligent agent must recognize environmental contexts and perceive environmental changes caused by people geographically apart.

We will describe the detail plan to provide such services and to achieve the goal. There are, however, various limitations for agents to provide the services to all users. First, expressions of events and actions in the real world are not sufficient for agent to understand. Second, the real time device control and agent's detail step control are not clear. Third, there are so many actions that occur simultaneously. To overcome these limits, agents collect contexts that can happen in ubiquitous environment and create event. Let me suggest an instance of an event; "Sally enters the room R213". Contexts are created from this event. Context is expressed as predicated logic. Context of the given example is expressed as `enters (Sally, R213)`. Context is a situational condition associated with a user, such as location, time, room, temperature, lighting conditions, and user roles. Agents understand the event by using represented contexts and infer services by event and then offer it to the user. However, when agents clarify service to relevant device, the following problem can be arise. For example, if agent offers a service like "Projecting the file on the screen", there must be the service, "Turning on the power of the projector" and "Downloading files for presentation" before. Also, these services have order. We are developing a new temporal planning infrastructure for agents in ubiquitous environments called Temporal Planning System (TPS). In this paper, we study on temporal planning that we have developed to support ubiquitous agents in the TPS.

Features of this paper are described as following: First, we research the identical expression for agents being aware of events, defining semantics of various contexts in the real world. This paper makes use of the ontology of semantic web for a definition of context. The semantic web supports functions that agents recognize semantic information connected to web by itself and infer the knowledge of the information searched by user. This system generates the ontology of environment where the service will be provided by using OWL [1] ontology language. Second, this system expresses a semantic context on the basis of ontology. Semantic context adds a time concept to context. Therefore, agents can infer new facts and the order of events from contexts. Third, agents infer the service that user wants by context and planning service action by using TPS framework. The service reasoning, with TPS in the center, will be focused on in this paper.

2. An Overview of Temporal Planning System

In this research, we make awareness of situations as giving meaning on meaningless data sensor creates in the basis of ontology. It is limited to be aware of situation with only one data, but the situation to be inferred can be clear with various sorts of data. Thus we research the Temporal Planning System (TPS) as giving time concept on semantic context.

TPS is a framework for the intelligent service agents for information provision or service provision which users

[1] Mark Weiser web site: http://www.ubiq.com/weiser/

* This research is supported in part by the Ubiquitous Autonomic Computing and Network Project, the Ministry of Science and Technology (MOST) 21st Century Frontier R & D Program in Korea.

need in ubiquitous environment. Service agents can plan the order of composite service or atomic service in the basis of TPS. The general flow, which is based on homecare scenario of TPS, is shown in Fig. 1. It describes the process from context collection from environment to service provision.

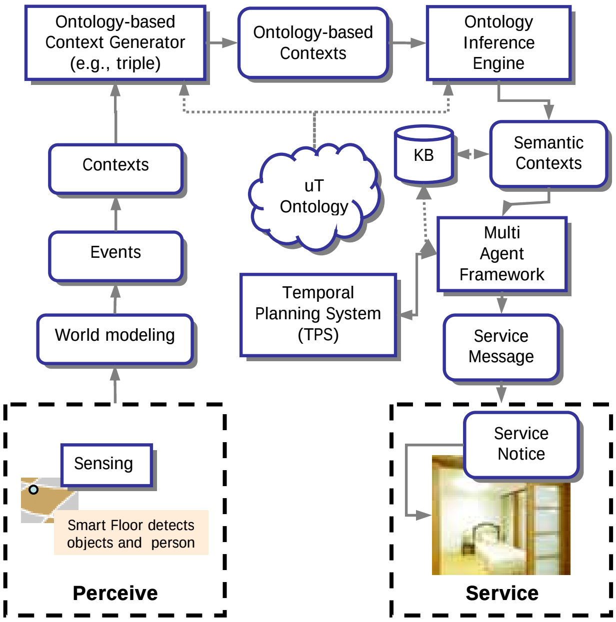

Fig. 1 service reasoning flow in ubiquitous environment

In ubiquitous environment, (i) it collects the data coming through a smart floor sensor and creates ubiquitous contexts. (ii) Context information gets transformed into semantic contexts in the basis of ontology, (iii) and it transforms semantic contexts into triple (i.e., (predicate subject object)) forms so that intelligent service agents can process them. (iv) A main agent transmits the semantic contexts created via communication to service agents which fit to situation information. (v) Service agents partition a composite service into atomic services by adding time concept to the composite service through TPS, and decide order of the atomic service, and (vi) provide a device of ubiquitous environment with the inferred services.

3. Temporal Planning System (TPS)

TPS is an architecture, which enables agents to infer services suitable to users' needs in ubiquitous system, on time and correctly. It is the ubiquitous system that computer resources actively provide users with services as it infers the situations of users and things in ubiquitous environment (e.g., home, office, living rooms, vehicle, etc.). That is, as the TPS is a major framework in ubiquitous system, it reasons services with time for a basis and plans order of services. Our TPS has three main functional components. Three components (i.e., Temporal Reasoning Engine, Service Reasoning Engine, Plan Generator) put inferred facts in KB(Knowledge Base) and then share them. (see Fig.2)

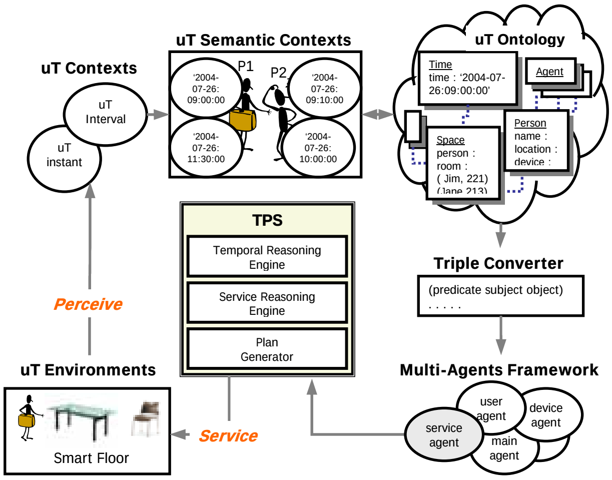

Fig. 2 The TPS-based framework

3.1 Temporal Reasoning Engine

Contexts gathered from ubiquitous environment include temporal relations. In order to reason time and temporal relations, it applies DAML-time ontology[2]. Our system organized a time-ontology by using OWL language with DAML-time ontology for a basis.

DAML-time ontology is composed of temporal relation axioms and set of abstract temporal entities. Our time ontology is defined with Allen's formalization of time [2] and DAML-time ontology for the bases. Temporal relation axioms are represented as a Prolog function with rule-based engine. DAML-time ontology defines the `TemporalEntity` as the highest class. This class has two abstract temporal entity classes as subclasses. They are `Instant` and `Interval`. The `Instant` means a point of time. The Temporal Reasoning Engine expresses the `Instant` as year, month, day, hour, minute and second (e.g., '2004-11-30:09:00:00'). The `Interval` means the gap between two separated Instants (e.g., ('2004-11-30:09:00:00', '2004-12-02:18:00:00')).

This system creates semantic contexts as adding time contexts when an event happened in ubiquitous environment. The scope where the Temporal Reasoning Engine applies is as following: (i) reason activity action in accordance with time (e.g., Sally listens to the music everyday from 7 am. to 8 am.), (ii) reason the situation at the time an event happens. (e.g., When an event, "A doctor entered Sally's room at am. 9:05 on the 1st Sept. 2004." occurs, it applies a time axiom by a treatment schedule semantic contexts which is expressed as `during (checkup (check-up011, interval ('2004-08-27:09:00:00','2004-09-15:11:00:00')))`. Thus one can reason that the doctor visited Sally's room for a medical examination.), and (iii) decides order of the action when two separate events

[2] A DAML ontology of time, http://http://www.cs.rochester.edu/~ferguson/daml/20020830/daml-time-20020830.txt

occurs at the same point of time (Section 4.1).

3.2 Service Reasoning Engine

It is the service in Ubiquitous environment that a device or a network system actively responds to user's action as being included in daily life, and provides services most suitable to the context user wants. Service Reasoning Engine is based on rules. It makes the reasoning rules by activity action inferred via semantic contexts and Temporal Reasoning Engine. The service is inferred via more than one rule. For example, in order to screen a file on a screen with a beam projector, the rule that connects user's personal agent which owns a file is executed first. Next, the rule reasoning the service is executed. The reasoned service is announced to the projector service agent

3.3 Plan Generator

We divide roles of service agent by service functions. Service agents execute composite service or atomic services. Composite services are composed of a number of atomic services.

In this paper, we use OWL-S ontology to giving precise meaning on a service also. OWL-S [3] is available to search and compose services fit to user's need. In case the inferred service has to use various atomic services together, web service composition is necessary [3]. We research into automatic web service composer of STRIPS type. We use OWL-S operator (i.e., input, output, precondition, effect) differing from a traditional STRIP. Therefore, it is possible to connect with Temporal Reasoning Engine and Service Reasoning Engine by changing OWL-S ontology into triple. There are the steps for Web service composition. Change web service written by OWL-S into triple (i.e., predicate logic). Draw the operator from triple. Compose web service chain by using planner based on STRIPS.

4. Planning Services in TPS

This research is for the automatic service reasoning via time-concept in ubiquitous environment and for automatic service control via service planning. In this chapter, we will talk over the process to be controlled after inferring services for home care service.

4.1 Temporal Planning

The reasoning related to time is executed when contexts as following are collected. (i) When Sally lies on a bed on bedtime, (ii) when a projector is operated in order to see a treatment chart, it defines planning axiom and then expresses action. Temporal reasoning can be applied to (i) multi-users perception, (ii) learning of users' patterns, (iii) order definition for services occurred simultaneously, and (iv) order definition for services composed of a number of atomic services.

[3] OWL-S: Semantic markup for Web Services http://www.daml.org/services/owl-s/1.1B/owl-s.pdf

4.1.1 The Curtain Drawing Service

If Sally is lying on a bed and the current time is between 23:00 and 06:00 next day, the service to curtain is executed. At this time, two actions happen simultaneously. It checks if the window is closed before it draws a curtain. If it is closed, it draws a curtain and if not, it executes the action to close the window and to draw the curtain. In case the window is not closed it executes two actions at the same time. These two actions are planned via temporal reasoning. For the control of action order we have defined the temporal logic [4] about axiom and each event (see Fig. 3).

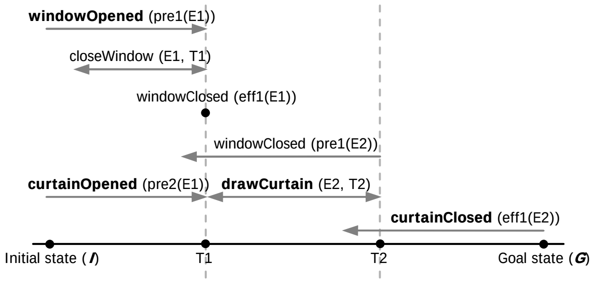

Fig. 3 The temporal logic for two events in the curtain drawing service

`closeWindow` events have a precondition interval (for the window being opened) and an effect moment (for the window being close):

```
Axiom 1. closeWindow: Temporal Logic
∀ e, t. closeWindow(e,t) ⊃ finishes(t,pre1(e))
∧ moment(eff1(e)) ∧ meets(t,eff1(e)).
```

Initial state I is current time, and goal state G is the time when the goal must hold. The goal is simply to have the window close over time G, i.e., curtainClose (eff1(e1)) such that drawCurtain (e1, T).

```
Rule tr1:
drawCurtain(e1, T) :-
  (during(pre1(e1), T); overlaps(pre1(e1), T))
  ∧ meets(T, pre2(e1) ∧ starts(eff1(e1), T).

Rule tr2:
curtainOpened(pre2(e1) :-
  windowClosed(pre1(e1)),
  drawCurtain(e1, T).

Rule tr3:
windowClosed(e2) :-
  ShutWindow(e2, pre1(e1)),
  drawCurtain(e1, T).
```

A sub-goal should be satisfied to meet the goal to curtain (i.e., `Rule tr1`). If the curtain is open, the action to curtain has to be conducted after checking whether the window is closed (i.e., `Rule tr2`). If the window is open, it closes the window and then draws the curtain (i.e., `Rule tr3`). So the goal is fulfilled with the satisfaction of the sub-goals like those.

4.1.2 Projector Screening Service

After Sally enters her room, if another movement is detected at time interval it recognizes as other person exists in the room. It infers that the person is a doctor after it checks Sally's schedule through the personal agent of Sally. Next, it projects Sally's medical records on a screen. The service to project also includes sub-goals. (i) It turns on the power of the projector in the room. (ii) It downloads files for presentation, and (iii) it projects the file on the screen (see Fig. 4).

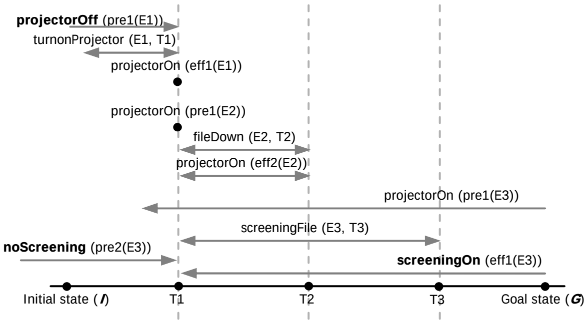

Fig. 4 The temporal logic for three events in projector screening service

Axiom 2. turnonProjector: Temporal Logic

```
∀ e, t. turnonProjector(e,t) ⊃
finishes(t,pre1(e)) ∧ moment(eff1(e)) ∧
meets(t,eff1(e)).
```

Axiom 3. fileDown: Temporal Logic

```
∀ e, t. fileDown(e,t) ⊃ meets(pre1(e), t) ∧
equals(eff1(e), t).
```

Via the temporal logic as above it executes the action by the time sequence. The projector screening service plans the order of the composite service with the Plan Generator Engine. The Plan Generator Engine infers atomic services as following and executes ordinary.

• **Power Checking Service:** It gets device IP (i.e., Input), and checks the power status (e.g., on/off) of this device. Output is the status of this device

• **Searching Chart Service:** It searches medical charts. Beam projector has to be turned on (i.e., prediction) for files downloading. It downloads medical records from device IP (i.e., Input). Output is *fileURI*.

• **Screening Service:** If there is *fileURL* (i.e., pre-condition) that is downloaded by the Searching Chart Service, it presents the files to Input device IP (i.e., Input). Output is *presentationOn*, and Effect is *ChartScreening*.

4.2 Service Reasoning

Service Reasoning detects changes of the space and the user and operates. An example as following can be given that it projects medical records on a screen. In case it is detected that the doctor enters Sally's room, the service to project medical records on a screen is reasoned as a rule like below is executed.

Rule sr1:

```
searchChart(Name, FileURI) :-
  chartNo(No, Name),
  medicalDate(No, CDate),
  before(CDate, currentDate),
  hasFileURI(No, FileURI),
  assert(fileDown(FileURI)).
```

Rule sr2:

```
enters(Doctor, Loc) :-
  searchChart(Name, FileURI),
  devLoc(DevType, Loc),
  hasIP(DevType, DevIP),
  assert(screeningFile(DevIP, FileURI)).
```

By `Rule sr2`, it searches Sally's latest medical records. `Rule sr1` is a sub-goal of `Rule sr2`. By `Rule sr1`, it searches the most recent medical records before today's date. Next, it reasons the service to project a file with the IP of the projector in the room and the URI of the file, and it operates them. It infers services by using rules like those.

5. Conclusion

The purpose of this research is the automatic service reasoning via time-concept and the automatic service control via service planning in ubiquitous environment. TPS proposed in this paper, studies the intelligential agent offering information to user or service at once in ubiquitous environment. It designs ontology through OWL for the agent to understand surrounding context. There are lots of contexts existing in real world that furnishes broad context by inferring unexpressed context by OWL ontology axiom. The agent infers service through gathered context and defines steps of offered service through planning. It also suggests offering intelligential service in focused research of temporal relationships to express dynamic context that is both staring point and end point that exists in ubiquitous environment.

References

1. Sean Bechhofer, Frank van Harmelen, Jim Hendler, Ian Horrocks, Deborah L. McGuinness, Peter F. Patel-Schneider, and Lynn Andrea Stein. "OWL Web Ontology Language Reference", w3c candidate recommendation 18 august 2003 edition, August 2003.
2. Allen, J.F, "Planning as Temporal Reasoning", Proc. Of the Second Int'1 Conf. On Principles of Knowledge Representation and Reasoning, Cambridge, MA April. 1991.
3. Paolucci, M, Sycara, K., Nd Kawamura, T. "Delivering Semantic Web Services." In Proccedings of the 12th international conference on World Wide Web. ACM Press, 2003.
4. Allen, J.F. and G. Ferguson, "Actions and events in interval temporal logic. In Oliveiro Stock (ed.), Spatial and Temporal Reasoning, Kluwer Academic Publishers, pp. 205-245, 1997.