Review - Deployment of Context Aware Application

Mrs.S.Devi

Abstract: Context is any information that can be used to characterize the situation of an entity. An entity is a person, place, or object that is considered relevant to the interaction between a user and an application, including the user and applications themselves. "A system is context-aware if it uses context to provide relevant information and/or services to the user, where relevancy depends on the user's task". It acquires and utilize context from a computer device in order to provide services that are appropriate given the acquired context. It adapts according to location of use, the collection of nearby people and objects and changes to those objects over time.

I. Introduction

Context is any information that can be used to characterize the situation of an entity. An entity is a person, place, or object that is considered relevant to the interaction between a user and an application, including the user and applications themselves. Schilit and Theimer (1994) refer to context as location, identities of nearby people and objects, and changes to those objects.

In a similar definition, Brown, Bovey et al. (1997) define context as location, identities of the people around the user, the time of day, season, temperature, etc. Ryan, Pascoe et al. (1998) define context as the user's location, environment, identity and time. Dey and Abowd's definition (2001) Hull, Neaves et al. (1997) and Pascoe, Ryan et al. (1998) define context-aware computing to be the ability of computing devices to detect and sense, interpret and respond to aspects of a user's local environment and the computing devices themselves.

Schilit, Adams et al. (1994), Brown, Bovey et al. (1997), Davies, Mitchell et al.(1998) define context-aware applications to be applications that dynamically change or adapt their behaviour based on the context of the application and the user. Dey and Abowd's definition (2001), A system is context-aware if it uses context to provide relevant information and / or services to the user, where relevancy depends on the user's task. Almost any information available at the time of an interaction can be seen as context information.

Some examples are:

- Identity
- Spatial information

  e.g. location, orientation, speed, and acceleration
- Temporal information

  e.g. time of the day, date, and season of the year
- Environmental information

  e.g. temperature, air quality, and light or noise level
- Social situation

  e.g. who you are with, and people nearby
- Resources that are nearby

  e.g. accessible devices, and hosts
- Availability of resources

  e.g. battery, display, network, and bandwidth
- Physiological measurements

  e.g. blood pressure, heart rate, respiration rate, muscle activity, and tone of voice
- Activity

  e.g. talking, reading, walking, and running

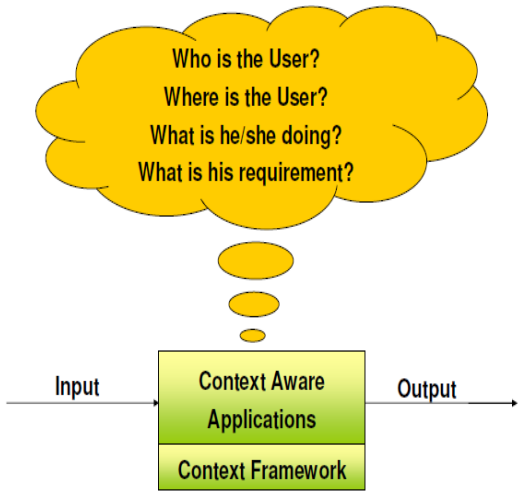

Fig.1: Context Aware Applications

Types of context-awareness:

- Active context awareness: An application automatically adapts to discover context, by changing the application's behaviour.
- Passive context awareness: An application presents the new or updated context to an interested user or makes it persistent for the user to retrieve it later.

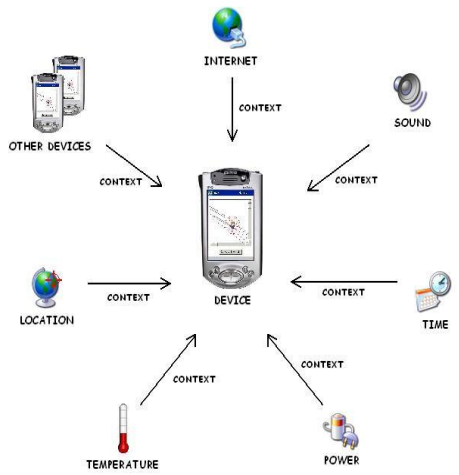

Fig.2: Context-aware computing

Types of context-aware applications:

- Automatically execute a service : e.g. Smart homes: turn off lights, adjust temperature, sending alerts and reminders
- Present the information and services to a user: e.g. Tour guide, Active Badges
- Tag the context to the information for later retrieval: e.g. Digital camera meta-data: time, location

II. General model of context-awareness

- Generation : Contextual information is obtained from UI or sensor
- Processing: Change raw data to meaningful information
- Usage: Use of contexts and possible reaction as output

III. Situation & Context

From sensor to context and activity:

- Raw contextual information
  - Location : location-tracking system (GPS)
  - Time : built-in clock of the computer
  - Light level : photodiode
  - Sound : microphone
- High-level contextual information
  - User's current activity: big challenge
  - Machine vision: camera technology, image processing
  - Artificial Intelligence

IV. Sensing context changes

Most of contextual information changes as time go on. Context changes can be sensed by

Periodic polling

Advertising

Selective polling

It is difficult to describe and detect a situation.

A car is going to have a serious accident

Two people are undecided what to buy

Someone is sleeping in a room

A family having dinner

But it is a prerequisite to recognize situations for building intelligent objects.

Example: Someone is sleeping in a room Sensors

Motion sensor overseeing the room (ON/OFF)

Weight sensor in each leg of the bed (0 - 100)

Light sensor (0 - 100)

Door sensor (OPEN/CLOSE)

Pressure mat in a rag on the floor (ON/OFF)

Microphone providing noise level (0 - 100)

Find a function that takes sensor values as input and that tells if someone in sleeping in the room or not.

Alternative approaches:

Top down

Situation -> Context -> Features -> Sensors

Bottom up

Sensors -> Features -> Context -> Situation

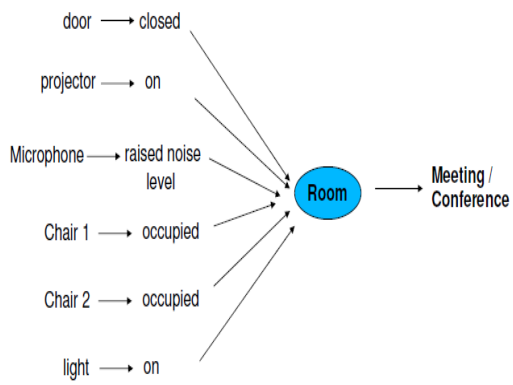

Fig.3: Higher Level Context

'A survey on context-aware system'

Architecture	Sensing	Context module	Context model & processing
CASS	Centralized middleware	Sensor nodes	Relational data model
CoBra	Agent based	Context acquisition module	Ontologies (OWL)
Context Management framework	Blackboard based	Resource server	Ontologies (RDF)
Context toolkit	Widget based	Context widgets	Attribute-value tuples
CORTEX	Sentient object model	Context component framework	Relational data model
Gaia	MVC (extended)	Context providers	4-ary predicates (DAML+OIL)
Hydogen	Three layered architecture	Adapter for various context types	Object-oriented
SOCAM	Distributed with centralized server	Context providers	Ontologies (OWL)

V. Challenges in building context-aware systems

Building and deploying context aware systems in open, dynamic environments raises a new set of research challenges.

- Sensor Data Acquisition
- Context Modelling
- Building Intelligence
- Execution of services according to context

VI. Sensor Data Acquisition

There are many challenges in acquiring data from sensors. Devices are heterogeneous, protocols, communications, interfaces etc.

Examples

- Devices may be with RS232 / USB interfaces
- Devices may have various protocols like Bluetooth, Zigbee, RFID etc.

Sensor data representation

- Implementing wrappers / plug in modules for heterogeneous devices
- Different sensor will give data in different format
- Convert them in to context information and infer appropriate context

VII. Context Modelling

A context model is needed to define and store context data. The context model reflects the designer's understanding towards context. It determines the way in which we organize the meaningless and out-of-order data of physical world into logical structural items of computing world it lays a foundation for the implementation of context aware functions. The basic elements of a model are objects that describe the physical or logical entities and relations between objects. Efficient context modelling is required in order to represent the entire set of context data (both dynamic such as location and static such as preferences) that need to be monitored, collected, stored and utilized in context aware system.

Various Modelling Approaches:

- Key-Value Model
- Markup Scheme Model
- Graphical Model
- Object Oriented Model
- Logic Based Model
- Ontology based Model

Context Modelling Approaches

- Key-Value-Pairs Models

  Most simple category of models

  Not very efficient for more sophisticated structuring purposes

  Only exact Matching

- Mark-up Scheme Models

  Scheme implements model

  Typical representatives: profiles

- Graphical Models

  Particularly useful for structuring, but usually not used on instance level

- Object Oriented Models

  Intention behind object orientation is (as always) encapsulation and reusability

- Logic Based Models

  Logic defines conditions on which a concluding expression or fact may be

derived from a set of other expressions or facts (reasoning)

→ Context is defined as facts, expressions and rules

High degree of formality

- Ontology Based Models

Ontology used as explicit specification of a shared conceptualization

→ Strong in the field of normalization and formality

Context is modelled as concept and facts

VIII. Conclusion

Context-aware computing offers many advantages, allowing systems to act more autonomously and take initiative, but informed by a better model of what their users need and want. Building and deploying context aware systems in open, dynamic environments raises a new set of research challenges.

- Sensor Data Acquisition
- Context Modelling
- Building Intelligence
- Execution of services according to context

Reference

1 Albrecht Schmidt. Ubiquitous Computing - Computing in Context

http://www.comp.lancs.ac.uk/~albrecht/phd/

http://albrecht-schmidt.blogspot.com/

2 Dey, A.K. and Abowd, G.D. (1999). Toward a better understanding of context and context-awareness. GVU Technical Report GIT-GVU-99- 22, College of Computing, Georgia Institute of Technology.

ftp://ftp.cc.gatech.edu/pub/gvu/tr/1999/99-22.pdf

3 'A survey on context-aware system' by Matthias Baldauf and Schahram Dustdar and Florian Rosenberg

4 "Challenges for Context Management Systems Imposed by Context Inference" by Korbinian Frank, Nikos Kalatzis, loanna Roussaki and Nicolas Liampotis

5 "Context modeling Survey" by Thomas Strang and Claudia Linnhoff-Popien

6 "Context Sharing Platform" by K.Nihei, Journal of Advanced Technology, vol 1, 2004

7 Lamming, M. and Flynn, M. (1994). Forget-me-not: Intimate Computing in Support of Human Memory, in proceedings of FRIEND 21: International Symposium on Next Generation Human Interfaces,Tokyo,1994 , pp. 125-128.

8 Context-Aware Applications Survey By Mari Korkea-aho, Department of Computer science, Helsinki University of Technology

9 The Aware Home: A Living Laboratory for Ubiquitous Computing Georgia Institute of Technology Atlanta, GA 30332-0280, USA