Proactive context-aware computing

PREETI BHARGAVA
Responsive, Reactive, Proactive

- Responsive – responds to a request
- Reactive – reacts to an event
- Proactive – Anticipates and performs an action
What do we mean by proactive context-aware computing?

- Delivering “the right information at the right place and the right time" i.e. relevant, personalized and timely information to users.

- Relevance - most important characteristic of a context-aware system and varies with different users. In today's world, where information overload can easily happen, it is necessary that the system retrieves and displays only that information which is relevant to the user in making effective decisions.

- Personalization - can be achieved by acquiring a user's context (location, preferences, etc.) through implicit or explicit means and filtering the relevant information to tailor it to the user's situation and preferences.

- Timeliness - Timely information delivery by a context-aware system can be achieved by adopting a pro-active approach for ‘pushing’ the personalized relevant information to the user at the time when the user needs it and can act upon it.
How can you build a proactive context-aware system?

We believe that this paradigm can be fulfilled by a system which acts as an efficient Personal Digital Assistant and:

- Recognizes and anticipates the user's current and future situations, activities or events as relevant information often depends on what the user is or will be doing.

- Determines the relevant information for each of the user's situations and further filters or personalizes it based on the user's context and preferences.

- Finally, `pushes' it to the user pro-actively before an explicit request is issued by him, in order for it to be useful and effective for the user.
Client agent applications

Send user’s context, behavior and situation

Predict future intent and behavior and take appropriate action

Provide Feedback for refinement

Learn, store and refine user behavioral and situation models

Rover II - Context-Aware Middleware

Extract context and behavior history

Store activity, event and context logs for every user

User Models

Logs

Sense user’s context and situation
Rover Architecture

- RDF-DB
- User Models
- Service Tier
- WWW
- HTTP
- Controller
- Context Interface
- Activity Engine
- Activity Store
- Planning Engine
- Rule-Based System
- Learning Engine
- Smartphone Client agents
- Web base Client agents
Smartphone client agent: SenseMe

- A system for automated multidimensional and multi-sensory temporal context and activity recognition in a robust, accurate, scalable, power efficient and non-invasive manner.
- Captures five dimensions of a user’s situation:
  - Environmental Context - Indoor, Outdoor, Indoor-Outdoor
  - Physical Activity - Stationary, Walking, Running, In-vehicle
  - Device Activity - Task the user is engaged in on the device such as phone call, browsing
  - Location - Set of locations determined by Wi-Fi (indoors) or reverse geocoding(outdoors) and their types (obtained from Google Places or Foursquare)
  - Social Context - Number of people around the user at any given instance
SenseMe architecture

SenseMe Service
- Environmental Context Recognition Service
- Physical Activity Recognition Service
- Localization Service
- Social Context Recognition Service
- Device Activity Recognition Service

Front End Application Layer
- SenseMe Vis

SenseMe Db

Environmental Context, Physical Activity, Device Activity, Environmental Context, Social Context
SenseMe Results

- High Accuracy for different dimension in the range of 90 – 95%
- Uses power conservation techniques like Suppression, Piggybacking and Adaptation to control the GPS duty cycle
- Uses techniques that are agnostic to orientation, body position, time, weather etc., so no calibration is required
- Scalable, robust and generic
- Device independent and universally applicable
- All computation and processing is carried out on the device without a backend server - minimum latency and data is secure and confidential on the device
- Non-invasive and can easily run in the background and collect and process user’s data without the need for any intervention.
Indoor Location: Locus

Access points data (RSSI, MAC address, SSID)

LocateMe

Room #,
Floor #,
Building Address,
Location Type
How Locus works

- The client Android application scans the environment for Wi-Fi access points
- The AP data (RSSI, MAC address, SSID) is then sent by the client to the Locus Location Server in XML format
- Locus server has a database with every AP's (x,y) coordinates and floor
- Locus uses heuristics derived from RSSI to determine the floor and interpolate location of the device on that floor
- Returns a room #, floor #, and a logical address to the client device which can then be displayed on a floor plan
- Floor Accuracy: Over 95% on an average
- Location accuracy: Room level accuracy (6.9 m)
Benefits

- First calibration-free system for floor as well as location determination in multi-story buildings
- Inexpensive solution suitable for indoor localization with minimum setup, deployment or maintenance expenses.
- By avoiding the dependence on radio maps, it is readily deployable and robust to environmental change (as opposed to fingerprinting)
- Relies on existing infrastructure and mobile device capabilities, and requires no proprietary hardware to be installed
- Scalable to buildings with any number of floors
- Low software and hardware complexity
- Can support multiple indoor location based context-aware applications for education, entertainment, retail and shopping, public safety, healthcare, etc.
Relevant Information: TellMe

- A novel system that performs Context and Situation-aware Relevant Information Discovery and Ranking

- It discovers and ranks services and the external information to be retrieved from them, based on the Semantic Relatedness of the content of the services with the user’s situation.

- Client side system retrieves information about future situations from several user generated information sources such as calendars, reminders, and alarms in order to generate the Situation Trajectory of a user (a sequence of the user’s current and anticipated future situations)

- Server side system has a Ranking algorithm iterates over the Situation Trajectory in order to discover and generate a ranked list of candidate services, as well as the most relevant category for each service
How you can contribute?

- Rover – help us build it!!
- SenseMe – add more dimensions of context and activities
- Locus – build iLBS that use it
Questions?