Background and Specification Progress Report

By

Joshua Simpson

Table of Contents

Background and Context

Contextual awareness:

Contextually aware services are becoming a more and more popular method to engage and inform a user, with constant advances in technology increasing the precision and nature of contextually aware services, and providing new and interesting tools to provide these services.

Context has been defined several times in several different papers depending on the needs of the researcher(s), but one of the most relevant to the project at hand would be Dey and Abowd, who define context as:

> *"... 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"*[1]

Building widely available contextually aware services has the main problem of requiring all client devices to possess the ability to measure their environment (and to the same scale) – this requires the use of an already well-established method of defining context.

Location-based services / route planning:

Location based services have made great progress over the last few years, bringing the ability to locate a cellular device down to a building using GPS technologies. Massive platforms have been built on using location to build a contextually aware service[2], so it makes sense that this field continues to advance at a rapid pace. Location based services work hand-in-hand with contextually aware services, providing the ability to give context, whilst also being able to take location using contextual markers.

A natural extension of these location services are route-planning services that allow users to find the most expeditious route to any location, from their current. Whilst the biggest problems that location technologies are constantly trying to solve are precision and accuracy, route planning adds more problems – namely efficiency, and correctness.

Crowdsourcing Data:

Defined as 'Obtaining (information or input into a particular task or project) by enlisting the services of a number of people, either paid or unpaid, typically via the Internet'[3], crowdsourcing is becoming a wildly popular method to collect large amounts of data with minimal effort on behalf of the client, with services such as CrowdFlower offering

[1] (Abowd, 1999)
[2] See Google, forgo Apple Maps
[3] http://www.oxforddictionaries.com/definition/english/crowdsource

incentives for users to give data. The biggest problems in this sector are encouraging user participation, and reliability of data.

Existing Solutions in these sectors

Google Maps:

Google already provide high level location-based services, allowing users to pinpoint their location incredibly accurately – this has been boosted by their use of public wireless points to increase the accuracy of triangulation using Wi-Fi as well as GPS. A major issue is that Google provide world-based maps, with local area being the lowest level, meaning that it cannot provide truly pinpointed contextual services throughout a building in a useful manner (usually it is just presented as a worldwide latitude / longitude).

Google also crowdsources a lot of the data that they use – mostly automatically, to provide auto completion services and the like; their Maps application can collect anonymous location data periodically to improve their own services[4][5].

Estimote:

A rapidly accelerating tech start up, Estimote provides ‘beacons’ using low-power Bluetooth in order to send an ID[6] to an Estimote enabled application – these can then be programmatically associated with locations, events, etc. and calculates its distance from the phone using RSSI (the received signal strength), meaning that it can be used very effectively to provide location-based services. This is currently one of the most prominent devices currently using its base technology (iBeacon). Currently it’s major drawback is the price, which limits Estimote beacons / stickers to small buildings – covering an area such as King’s would be incredibly costly, and any structural changes regarding the beacons would require significant effort to represent in an application developed to use Estimote. Another issue is that using RSSI to develop an idea of location can become very complex in a large building with lots of people in it (which would cause signal attenuation at different levels throughout the day).

Cisco Context-Aware Software

An established standard in networking, Cisco provide a complete package, based around monitoring ‘assets’ and providing back-end support to network hardware, along with some contextual services that allow location tracking of connected wireless users[7]. The software is designed with an open-ended API to allow the development of business

[4] (Google - Collection of anonymous location data, n.d.)
[5] (Google - Privacy Policy, n.d.)
[6] (Estimote - Intro To Beacons, 2014)
[7] (Cisco - Mobility Services - Context Aware Software, n.d.)

applications that integrate with the network, although this may still limit businesses in their ability to collect data, and provide useful functionality in return.

Relevant Papers:

Participation inequality: Encouraging more users to contribute (Nielsen, J., 2006)

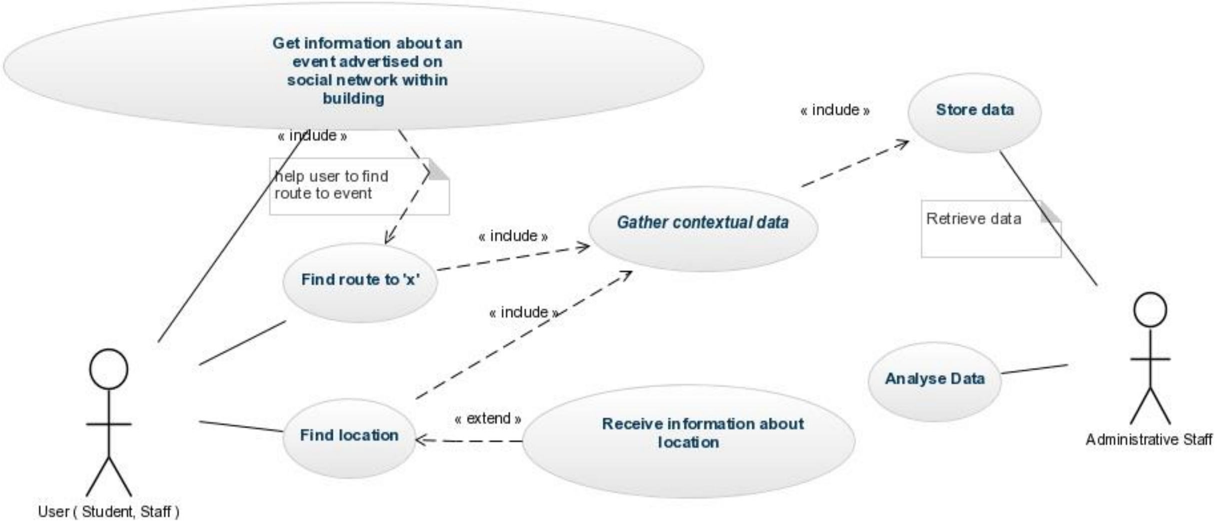
This article addresses a major problem with crowdsourcing data which Nielsen refers to as ‘participation inequality’, stating that ‘user participation often follows a 90-9-1 rule, with 90% of users being ‘lurkers’ or users that will observe but not contribute’[8]. Nielsen also covers how to overcome this participation inequality (after briefly summarizing with ‘you can’t’). One method in particular is of interest when considering the aims of this project and its associated application: ‘make participation a side effect’[9] – essentially the act of collecting data without the user having to do anything - this is less intrusive to the user’s day-to-day use of the application, but can also have obvious ethical ramifications if the user is unaware if they are submitting data.

A Survey on context-aware systems (2007)

This article provides some great insight into approaching the development of contextually aware applications – it lays forward architectural approaches, noting that modern software design principles emphasize encapsulation ‘in order to separate business logic and GUIs’[10]. It also details sensor types – the most interesting being ‘logical sensors’, which in practice would use multiple sources of information, ‘combining physical and virtual sensors with additional information from databases or various other sources in order to solve higher tasks’[11]. It even goes on to detail already existent location-aware systems, such as those used in tours.

[8] (Nielsen, 2006)
[9] *Ibid*
[10] (Baldauf, (2007))
[11] *Ibid*

Use Case Diagram:



Requirements Specification

Overview:

The end goal of this project is to create a mobile application that will provide location and route-planning services, using collected wireless data. It will use different aspects of the same wireless data to provide analytical information to a backend server, essentially crowd-sourcing useful data for King’s College London. The application should also be able to use social networking features to increase usability.

Functional Requirements

FR1.x: Get user's location

FR1.0 – Collect wireless data

FR1.01: Scan for wireless networks

FR1.02: Collect MAC addresses of available networks

FR1.03: Filter out duplicate networks (different SSIDs, with different MAC addresses, projecting from the same router)

FR1.1 – Use wireless data to find location

FR1.11: Compare available MAC wireless points with data stored in SQLLite database to ascertain user location

FR1.2 – Inform user of location

FR1.21: Display data retrieved from database to user

FR1.3 – Give user extra locational information

FR1.31: Display fact / information about user's current location

FR2.x: Route-finding

FR2.0 – Retrieve target location

FR2.01: Widget for user to input location

FR2.1 – Create route to target location

FR2.11: Load directed graph

FR2.12: Use path-finding algorithm (Dijkstra’s, experiment with others) to find optimal path to user's selected destination

FR3.x: Data collection

FR3.0 – Collect contextual data – room utilisation

FR3.01: Store current location information when user moves to new area

FR3.02: If user does not move to new area then store records containing timestamp and location

FR3.03: Once user has moved to a new area, send locational data and begin process again

FR3.1 – Collect contextual data – wireless access point functionality / preference

FR3.11: On connect to new wireless point, store point connected to

FR3.12: On connect to new wireless point, check if location is the same – if it is then send previous and current wireless point data

FR4.x: Social Integration

FR4.0 – Facebook Event Integration

FR4.01: Application should be able to connect to user's Facebook using Graph API, and collect event data, with the intention of using local events as targets in FR2.x

FR5.x: Data collection backend

FR5.0 – RESTful Backend

FR5.01: RESTful backend server that will take POST data from the mobile application and process the data to make it more useful to administrative staff

FR5.02: Backend needs to manage databases (DR3.0 & DR4.0), and store processed data into the relevant database

Data Requirements

DR1.0: Location / MAC address database

SQLite database containing five fields; LocationID, LocationName, MACOne, MACTwo, and MACThree, held locally on the mobile device to decrease access / query times

DR2.0: Directed Graph

Directed graph, containing vertices representing each location in DR1.0, and edges connecting locations that can be reached from one another, used for route-planning in FR2.x

DR3.0: Database to store room utilisation data

SQL database to store collected data, containing the location that was in use, and how long it was in use for in total – fields are EntryID, Location, TimeUsed, TimeProcessed

DR4.0: Database to store wireless access point functionality data

SQL database to store collected data on preferred wireless points – contains fields OriginalMAC, PreferredMAC, TimeStamp

Non-Functional Requirements

NF1.x: Performance

NF1.1: Application will not cause an 'ANR' (app not responding) error on any 'low-end' smartphone[12] (low end specifications are considered to be 1ghz dual core processor and 768mb RAM[13])

NF1.2: Application will find user's location in 95% of the King's College London Strand Campus

NF1.3: Application will be as cost-friendly as possible when considering battery life, using no more than 10% battery life whilst operating over 30 minutes

NF1.4: Application will minimise amount of data to be sent from the smartphone to reduce bandwidth usage

NF2.x: Security

NF2.1: Application will not breach the user's statutory rights, or the Data Protection Act, and as such, data will only be sent if the user consents.

NF2.2: Any data that is sent over a network will be anonymised, so as to not compromise the user's personal details.

NF2.3: Only applications using a hardcoded API key will be able to GET / POST data to the server.

[12] At time of writing

[13] Figures created by comparing phones shown at http://www.phonearena.com/news/Affordable-not-cheap-15-best-low-cost-smartphones_id58696

Bibliography

Abowd, A. K. (1999). *Towards a Better Understanding of Context and Context-Awareness.* Retrieved from smartech.gatch.edu: https://smartech.gatech.edu/bitstream/handle/1853/3389/99-22.pdf?sequence=1

Baldauf, M. D. ((2007)). A survey on context-aware systems. In *Int. J. Ad Hoc and Ubiquitous Computing Vol. 2* (pp. 263–277).

Cisco - Mobility Services - Context Aware Software. (n.d.). Retrieved from Cisco: http://www.cisco.com/c/en/us/products/collateral/wireless/mobility-services-engine/data_sheet_c78-470925.html

Estimote - Intro To Beacons. (2014). Retrieved from Estimote: http://estimote.com/api/getting-started/intro-to-beacons.html

Google - Collection of anonymous location data. (n.d.). Retrieved from Google: https://support.google.com/gmm/answer/2839958?hl=en

Google - Privacy Policy. (n.d.). Retrieved from Google: http://www.google.com/policies/privacy/#infocollect

Nielsen, J. (2006, October 9). *Participation Inequality.* Retrieved from Nielsen Norman Group: http://www.nngroup.com/articles/participation-inequality/

Schilit, B., Adams, N., & Want, R. (1994). Context-Aware Computing Applications. 1st International Workshop.