

Context

The context is all elements that are significant in a given situation and can influence the execution of mobile applications. There may be device context (current state of the device, e.g. RAM in use, battery level), user context, and environment context.

Since these contexts are quite different, we need a framework to define how context information is acquired and delivered.

Generic Information Sources

- Sensors, e.g. temperature, humidity
 - These may be internal, or e.g. ad-hoc external sensors using WLAN
- mobile device
- user
 - activities/profiles (e.g. at work, driving)
 - preferences
 - policies
 - * if there are several pieces of data at the same type, which should be chosen?
- time, location
- internet

When you have mobility, the context is always changing.

The level of abstraction of the data should be appropriate to the particular data.

The frequency with which the context is updated needs to be low enough to be useful.

Challenges

Data may change rapidly and may be incomplete.

Extracting context info by fusing data from several sources is complicated.

Context Framework

This is a generic model – usually only a subset of this functionality is used.

Applications talk to a context manager, which has components providing different services. Applications tell the context manager what type of context data they need, as well as how frequently they need updates, and what level of abstraction they need. The application is essentially subscribing to the context manager.

The context manager needs to create the different contexts based on the atomic data.

Context Manager Components

Resource Server

The resource server connects to any context data source and outputs context atoms. It has a pipeline structure.

Sensor measurement -> Preprocessing -> Feature extraction -> Quantisation and semantic labelling

Pre-processing stage builds measurement data arrays that store a certain amount of samples and calculates generic features for each time interval.

Feature extraction identifies which data is significant and extracts.

Context Recognition Service

The quantisation phase produces context atoms that can be used or further refined.

The resource server can use one of two methods for quantisation:

- crisp limits
 - e.g. sound intensity = silent/moderate/loud
- fuzzy set
 - silent: 0.6, moderate: 0.3, loud: 0.1
 - produces a weight for each category
 - the example is 60% silent, 30% moderate, and 10% loud

Context recognition uses either a set time-stamp or a time series of context atoms. It returns a single higher-level context.

###? Context Manager Work

A naïve Bayse classifier recognises higher-level contexts from lower-level context atoms.

The classifier [...]

[...]

Change Detection Service

Changes the context if there is a significant change in some context atoms.

Context Storage

The context history can be used to save computing resources by determining that the context follows a routine. We can fetch previously-determined contexts from storage and apply those.

Context Ontology

How do we represent the context?

A set of axioms (the ontology) can be used to express relationships between entities, e.g. a Father entity could have a hasDaughter relationship, which connects it to a Daughter entity. A context reasoner (inference engine) can parse the axioms to make inferences about what is true and false.

[...]

Resource Description Framework (RDF) can be used as the description syntax, which gives a common structure for representing information.