Modelling and Simulating Context Data in a Mobile Environment

Agathe Battestini and John A. Flanagan

Nokia Research Center
P.O. Box 407, FIN-00045, NOKIA GROUP, Finland
agathe.battestini@nokia.com, adrian.flanagan@nokia.com

Abstract. An important component of context awareness in mobile devices is sensing the user's environment. In context-aware research quite often there is little discussion on the quality of the context data which may be related to the lack and difficulty of obtaining real measured context data. Examples of real context data show the data can be noisy and cannot always be interpreted as might be expected. Simulation of context data is described based on a probability model. As well as providing a quick means of testing context aware applications it also contributes to the understanding of context data in a mobile environment.

1 Introduction

Sensing the user's environment using a variety of sensors is an important part of context-awareness in mobile devices [1]. Although the technology is available for sensing (GPS receivers, thermometers, pressure sensors, microphones, etc) the limitations of its application in a mobile environment applied to context awareness has not been analyzed in any type of formal manner. While sensors in general have problems associated with their sensing ability, including reliability of the hardware, calibration, drift, noise etc, in context aware applications further challenges are posed by the mobile nature of the sensors. As the sensors are mobile they should be small and require little power to operate. These two characteristics typically mean the quality and reliability of the sensors is decreased. Second, as illustrated later, due to the mobile nature of the sensors it is difficult to actually determine which aspect, if any, of the user's environment the sensor senses. For example a thermometer may sense a very different temperature depending on whether it is carried by the user in an outside or inside pocket.

As pointed out in [2], [3] many current context aware applications make unrealistic assumptions about the quality of the available context information. Typically these approaches, often based on ontologies [4], [5], [6], adopt a top-down strategy where there is an inherent assumption that the lower-level context information, typically coming from sensors, is error-free. The bottom-up approach has also been used where direct processing of the sensor signals has been used to extract features [7], [8], [9] typically based on some learning type algorithm. However none of these approaches address the nature of the signals being sensed and how it can effect the feature extraction and further context recognition task.

In section 2 some of the characteristics of sensor signals and their effect on the quality of the sensor measurements are discussed. There is a discussion on how in a mobile environment the sensors may not always record the user's environment. Since the ideas of modelling and simulating the context data are closely related, in section 3 there is a discussion on simulation followed by a description of the model used to simulate the context data. In section 4 one simulation scenario is described and a set of simulated features generated. A brief overview of the simulation tool is given in section 5, followed by a conclusion in section 6.

2 Sensor Signals

One of the contributing reasons to the increasing interest in context awareness is the availability of small, cheap, low power sensors that can be distributed in an environment or on the user. These sensors can be used to sense different environmental characteristics such as temperature, acceleration, humidity etc. Location in terms of GSM cell IDs and location area codes can also be considered as a signal, in essence, sensed by a mobile phone. Despite the fact that such sensor signals are considered as the basis of context awareness surprisingly little effort has been put into understanding the nature of these signals as they might appear in real situations. Due to the fact that these sensors are smaller, cheaper with low power consumption can also mean that the quality and stability of the sensors may not be as expected. Some properties of sensors and their ability to sense can be described as follows.

Accuracy Capacity of the sensor to give results close to the true value.

Resolution Minimal change of the input necessary to trigger a detectable change at the output.

Precision Capacity of the sensor to give the same results for the same input under the same conditions.

Sensitivity The slope of the calibration curve, which is a graphical display of the calibration record.

Response time Time between the occurrence of a constant change in the input and the time the sensor output has reached a percentage (i.e. 95%) of the value of the input.

Systematic errors Results from: interfering or modifying variables (i.e. temperature), drift (i.e. changes in mechanical stresses), measurement process, transmission process, human observers. Systematic errors can be corrected with compensation methods.

Random errors / noise True random errors follow a Gaussian distribution and come from environment noise, transmission noise, signal to noise ratio.

Drift Change in the signal over a long period of time.

Some of these characteristics such as systematic errors, and sensitivity can drift over time but can also be a function of other physical entities such as temperature. There are many approaches to minimizing these problems, such as recalibration of the sensors or using other auxiliary electronics to counteract these effects. Both approaches necessarily increase the cost of the sensors. It seems plausible that any sensor based context aware system, and the people developing them, should take into account or at least be aware of these effects in order to allow the system to function as expected over a period of time.

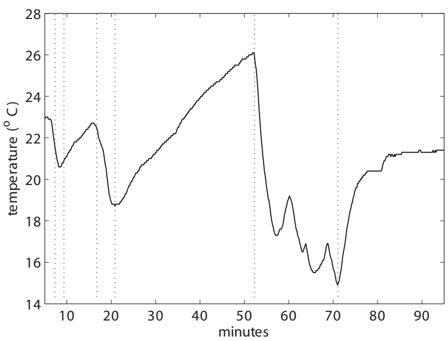
Given that such physical characteristics of sensors can be optimized so the sensors behave perfectly there is then the question of what the sensors actually measure in the user's environment. In the following we look at different types of sensor signals as recorded during the collection of data for the Nokia Context Database found at [10].

Consider first the temperature signal. The measurements for one recording session is shown in figure 1 (a). During this session the thermometer was in a sensor box placed in the user's outside pocket. The user was inside and then briefly walked outside from times $2-8$ minutes and then inside to take the metro $8-17$ minutes. From $17-21$ minutes the user was outside walking to the bus and from $21-52$ minutes the user was in a crowded bus. From the bus the user walked outside, briefly went inside to a shop and then continued home arriving there at approximately 72 minutes. Some time after 72 minutes the sensor box was removed from the pocket and placed on a table. The recording took place during the winter with a substantial difference between the inside and outside temperature. It is clear there is a certain time lag between a change in real temperature (i.e. between inside and outside) and the time it takes the thermometer to reach this real temperature. The reasons for this is due to the location of the thermometer in the user's outside pocket which could be considered semi-insulated. This is compared to the temperature variation after 72 minutes where it would seem that when the sensor box is left on the table and in contact with free air then the temperature stabilizes much quicker.

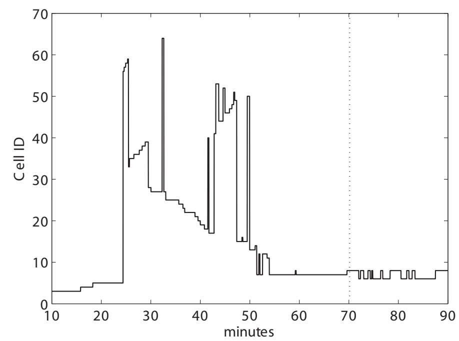
In the measurement of location, as determined by the Cell ID of the GSM network, figure 1 (b) shows a plot of the changes in the Cell ID for the same recording session. After 70 minutes when the physical location of the device is static it is clear however the Cell ID changes in a somewhat random manner between Cell IDs $7, 9$. This of course is related to the the conditions of the GSM network, GSM network load, and radio transmission environment.

In figure 1 (c) there is an illustration of how the user activity changes, as determined from the accelerometer, during the recording with 1 being low activity and 5 being high activity. The activity feature seems quite noisy and even during the bus trip when the user is stationary the activity level is obviously related to the movement of the bus. However, even when walking the activity level changes quite significantly.

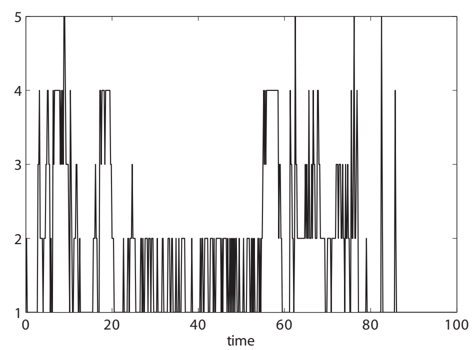
These examples of temperature, Cell ID based location and activity serve to illustrate that while everyday, human understandable concepts, may seem quite simple, interpreting the outputs of sensors measuring these quantities is not so straightforward. This has implications not only for modelling and simulating context data, but also has an impact on the types of methods and approaches that can be used in context interpretation and representation.



(a) Temperature



(b) Cell ID



(c) Activity

Fig. 1. Variation of temperature, Cell ID and activity during a single recording session.

3 Simulating Context Data

Our interest in this paper is to go some way towards understanding the nature and characteristics of sensed data in a mobile environment. One practical outcome and a means of verifying our understanding is by being able to generate data that is representative of what can be sensed on a mobile device. Our system is *subject + device* and the desired output of the simulation is a series of sensor data.

We have considered three different simulation models that could be applied for generating sensor data:

1. Represent each sensor with accurate electronic and mechanical models, represent the human or device (inter-)activities with mechanical or physical equations.
2. Model human and device (inter-) activities with mathematical or statistical functions and simulate the real output of the sensors (i.e. values: 0,122; 1,01; 3,15).
3. Model the statistical distribution of the data obtained after the measurement preprocessing and feature extraction steps, for different human-device (inter-) activities and simulate sensor data by generating random data following this distribution.

For the purpose of this paper, we use the second and third methods depending on the complexity of the context output to simulate and how relevant generating an accurate output would be. For example, determining user activity such as 'walking, running, stationary' using an accelerometer in a mobile device is difficult and error-prone even when the optimal conditions are ensured. Simulating series of accelerometer data would be difficult, and hardly representative of the real conditions, thus for such context information, we produce directly the data as if it came from the preprocessing or feature extraction stages.

It is assumed that context data in general can be modelled based on probability distributions. A starting point is to consider a mixture model of the form,

$$p_s(s) = \sum_{j=1}^{K} p(s|\theta_j)\pi_j \tag{1}$$

where π_j are the mixture probabilities and the $p(s|\theta_j)$ are the uni-modal component distributions with parameters θ_j. The distribution $p_s(s)$ is assumed to be a K-modal probability distribution and describes a cluster structure with K clusters. In terms of context the random variable represents a sample of the current user context, where for example s could be a vector containing information on the user location, time and user activity as follows, $s = (Location, Time, Activity)$. It is assumed that each cluster of the mixture model in equation (1) can be interpreted as a distinct user context. Each of the components of s can have their own associated probability distribution that best describes the distribution of the component. The mixture model of equation (1) describes the general probability distribution of the samples over all clusters or contexts, however when simulating a user's context based on this model the user is typically in one context which implies the samples should be essentially from a single cluster. In order to achieve this time variation of the user context the mixture probabilities are varied over time, where for cluster m then $\pi_m \approx 1$ for some period of time and for all other clusters v $\pi_v \approx 0$. The value of m is alternated over time based on some random or predetermined manner depending on the required scenario. Figure 2 shows an illustration of varying the mixture probabilities with time in order to generate a probability distribution that can produce sample context vectors for simulating real context data. In the next section it is shown how

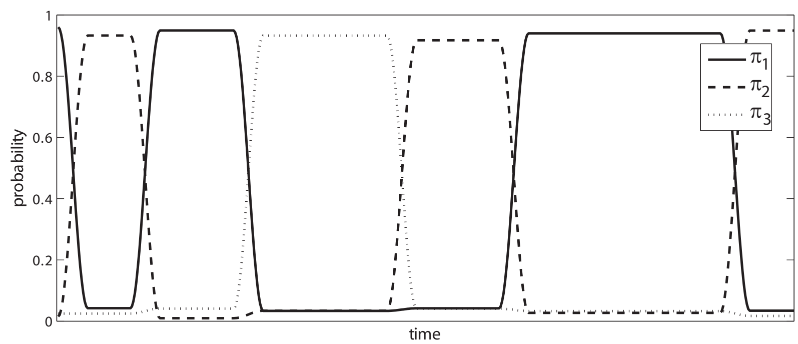


Fig. 2. Variation of mixture probabilities π_i over time for $K = 3$.

such a probability model can be applied to generating simulated context data. It is also seen that this type of model needs to be further expanded to take into account the cross correlations between different context features.

4 A Simulation Scenario

As an illustration in this section we simulate a *bus stop scenario* which describes a user walking from her workplace to the bus stop, waiting for the bus, then taking the bus, and interacting with mobile applications at various times. In this section, we have extracted from the full scenario two distinct contexts A and B that we define as:

A- The user walks to the bus stop following a predefined path.
B- The user stands or moves closely to the bus stop.

Context A and B are two clusters of the mixture model. We describe each context by assigning probabilities to each desired context feature according to the context model in equation (1). Probabilities account for sensor errors, uncertainty and variation and provide a way to better simulate reality. For example, it is very likely that the subject walking in the streets stops or walks at different speeds.

Table 1. Initial configuration for two context features, simulated independently from each other, and the probabilities associated in two different contexts A and B.

Feature		**A**	**B**
Activity	walking	0.80	0.25
	running	0.15	0.05
	stationary	0.05	0.20
Stability	stable	0.10	0.80
	unstable	0.90	0.20

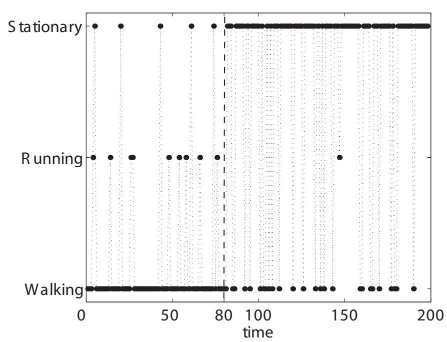
The resulting simulated data obtained after implementing the first version of the tool showed that simulating stability without taking into consideration activity was insufficient (Table 1). Thus we introduced cross-correlated probabilities between features, in order to define different sets of probabilities depending on the value of another feature.

In most scenarios, location is a context feature of importance; it sounds however unrealistic to generate random locations in all cases. We currently handle simulation of location in two different ways: one by defining a virtual path on which simulated coordinates are sampled (context A), the second by generating random (x,y) coordinates within a certain range (context B).

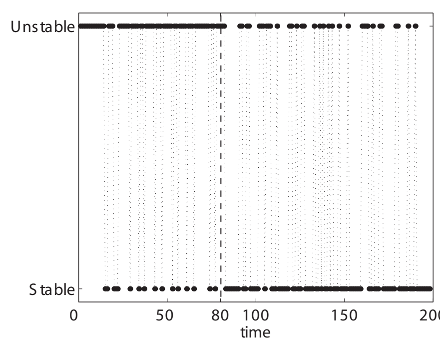
Table 2. Final configuration describing two different contexts A and B and the probabilities associated to the context features in each context.

Feature		**A**	**B**
Activity	walking	0.80	0.25
	running	0.15	0.05
	stationary	0.05	0.70
Stability (activity is walking)	stable	0.20	0.30
	unstable	0.80	0.70
Stability (Activity is running)	stable	0.00	0.00
	unstable	1.00	1.00
Stability (Activity is stationary)	stable	0.90	0.80
	unstable	0.10	0.20
GSM Cell ID	0 (36610)	0.95	0.05
	1 (42007)	0.05	0.90
	2 (317)	0.00	0.05
	4 (754)	0.00	0.00
Location	(x,y)	Samples from a virtual path: (0,0) (1,1) (2,0) (3,1) (3,2)..	Random within a disk centered on (4,3)
Timestamp	Unix timestamp		

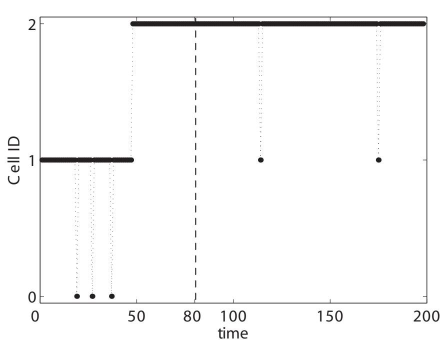
Table 2 shows the configuration for all variables simulated for context A and B, and excludes other types of context features such as: humidity, phone in hand, phone gesture, sound level, type of light source, temperature, which would have been simulated using the same model. The probabilities given are currently estimations, and can be modified to best fit the target scenario. Figure 3 shows the outputs of a simulator based on the probability model of equation (1) and the probabilities in table 2. At time 80 the context changes from A-B. Figure 3 (a) shows the activity which is dominantly walking in context A and at some times having also running or stationary activities. In context B the activity is dominantly stationary with some episodes of walking. In figure 3 (b) the stability is as might be expected and correlated to the activity in each context. Figure 3 (c) shows the Cell ID and how during context A while walking the Cell ID changes. Even though the activity is essentially stationary in context B there are some changes in the Cell ID. By simply comparing the activity and Cell ID signals from real measured signals in figure 1 to those simulated in figure 3 there is a certain similarity. Of course to achieve this similarity requires a good choice of probabilities.



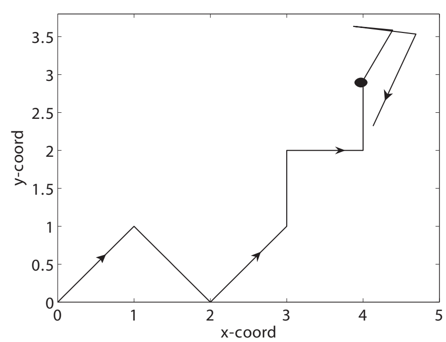
(a) Activity



(b) Stability



(c) Cell ID



(d) Route

Fig. 3. Output from context data simulator for consecutive contexts A-B. The change over from context A to context B occurs at time sample 80.

5 Simulation Tool Overview

Our simulation tool consists of a set of Python scripts that read a simulation configuration file, process the randomly generated context features, then process the deterministic or cross-correlated ones, and finally format and write the data into a file.

The simulation configuration file contains all information presented in table 2 in an XML format:

1. Declaration of the context features and the set of discrete values they can take.
2. The different user contexts or clusters are defined as sequences of context features with the associated probabilities.
3. The whole scenario or mixture model is defined as a sequence of user contexts. Additional information for each cluster is added such as time durations, or location parameters (i.e. (x,y) coordinates and distance in meters of the virtual path) and walking speed.

The simulation function operates as follows:

1. The function runs for each cluster of the mixture model.
2. Depending on the cluster's parameters, the simulation runs for a given time (context B), or as long as the distance has not been covered (context A).
3. For each context feature following a probability model, the function generates an array of random values, that are next converted to take values from the corresponding discrete range. For the feature location, the generated array is composed of (x,y) coordinates sampled from the virtual path (context A) or randomly picked (context B).
4. A final step is required to advance the timestamp, check location constraints - the next (x,y) coordinates is picked from the array so that the distance covered is coherent with the walking speed and the elapsed time - and select the next value for each feature from the arrays of previously generated data.

6 Conclusion

Sensing the user's environment is an important component of context awareness for mobile devices. In a mobile environment it is more difficult to control how and what sensors are measuring which can lead to problems of reliability, quality and interpretation of the sensor signals. We have shown that even in what might appear an elementary scenario sensor signals and their interpretation is not so straightforward.

We proposed a probability based model for contexts and used it to simulate context data from different sensors. The simulator allows different parameters to be set for different features and initial results appear to result in valid simulated context data. This approach allows for the modelling of context data in a mobile environment and contributes towards the understanding of this data.

In future work we would like to examine in a more detailed and statistical manner the real measured context data which would indicate appropriate values for the probabilities in the model of equation (1), as well as cross-correlations between features originating from the same source. There also remains the issue of formally describing and representing the inherently ambiguous characteristics of sensors and context data in representational formats such as ontologies.

Acknowledgements

This work has been performed in the framework of the IST project IST-2004-511607 MobiLife, which is partly funded by the European Union. The authors would like to acknowledge the contributions of their colleagues, although the views expressed are those of the authors and do not necessarily represent the project.

References

1. Gellersen, H.W., Schmidt, A., Beigl, M.: Multi-sensor context-awareness in mobile devices and smart artifacts. Mob. Netw. Appl. **7** (2002) 341–351
2. Henricksen, K., J.Indulska: Modelling and using imperfect context information. In: Pervasive Computing and Communications Workshops(PERCOMW). (2004) 33–37
3. Dey, A., Mankoff, J., Abowd, G., Carter, S.: Distributed mediation of ambiguous context in aware environments. In: UIST '02: Proceedings of the 15th annual ACM symposium on User interface software and technology, ACM Press (2002) 121–130
4. Staab, S.: Handbook on Ontologies. Springer (2004)
5. Korpipää, P., Mäntyjärvi, J.: An ontology for a mobile device sensor-based context awareness. In: Proc. Context03. LNAI no. 2680, Springer-Verlag (2003) 451–459
6. Chen, H., Finin, T., Joshi, A.: An ontology for context-aware pervasive computing environments. Knowledge Engineering Review **18** (2004) 197–207 Special Issue on Ontologies for Distributed Systems.
7. Mayrhofer, R.: An Architecture for Context Prediction. PhD thesis, Johannes Kepler Universität, Linz, Austria (2004)
8. Korpipää, P., Mäntyjärvi, J., Kela, J., Keränen, H., Malm, E.J.: Managing context information in mobile devices. Pervasive Computing, IEEE **2** (2003) 42–51
9. Schmidt, A., Aido, K.A., Takaluoma, A., Tuomela, U., van Laerhoven, K., van de Velde, W.: Advanced interaction in context. In: Proc. Intl Symp. Handheld and Ubiquitous Computing, LNCS 1707, Springer-Verlag (1999) 89–101
10. Mayrhofer, R.: Context database. http://www.soft.uni-linz.ac.at/Research/-Context_Database/ (2004)