On Uncertainty in Context-Aware Computing: Appealing to High-Level and Same-Level Context for Low-Level Context Verification[1]

Amir Padovitz, Seng Wai Loke, Arkady Zaslavsky
School of Computer Science and Software Engineering, Monash University, Caulfield East, VIC 3145, Australia

Abstract. There is an inherent chasm between the real-world and the world that can be perceived by computer systems, yielding uncertainty and ambiguity in system perceived context, with consequent effect on the performance of context-aware systems. While the problem is complex in depth and breadth, we explore an approach where context is characterized at different levels of abstraction, and where contextual information at high-levels of abstraction and sensed context at low-levels of abstraction can be used to validate and correct low-level sensed context such as location. We describe a randomly generated simulation of locations that might be sensed by a positioning technology, and how our approach can be used to validate and correct the sensed locations.

1 Introduction

One of the main challenges in pervasive and context-aware computing is the ability to handle uncertainties that emerge when systems try to become aware at runtime to desirable situations and are indecisive in reasoning about the true situation [1, 2, 10, 11].

We observe three factors that promote context uncertainty and highlight the need for context verification. The first is unsatisfactory combination of attribute types (either virtual or physical) [3] to infer a desired context, which results in low confidence in the inferred context – this may be the result of cost efficiency considerations. The second is an intrinsic ambiguity between two or more situations that impedes a straightforward reasoning about the correct context - this is often the case when two different situations are characterized by similar attribute values. If such ambiguous context states are comparable and context-states form a continuous function as values vary, it is possible to perform adaptation based on the fuzzy nature of the situation [6] rather than actually validating the true situation. However, in many cases two or more situations may be ambiguous and share a fuzzy region of attribute values but will not be comparable with respect to the specific application [7] or would enforce a different or opposite rule when they are inferred [5]. The third factor and focus of this paper is the often inherent inaccuracy and unreliability of many types of low-level sensors,

[1] We thank HP Labs and HP Australia for sponsorship of this work.

Padovitz A., Wai Loke S. and Zaslavsky A. (2004).
On Uncertainty in Context-Aware Computing: Appealing to High-Level and Same-Level Context for Low-Level Context Verification.
In *Proceedings of the 1st International Workshop on Ubiquitous Computing*, pages 62-72
DOI: 10.5220/0002664600620072

which may lead to contradicting or substantially different reasoning about context. When faced with contradicting sensorial data from two similar sensors, a context-aware system needs to resolve these discrepancies as well as high-level context ambiguities that result from the contradicting sensor readings.

Often, context verification is only a matter of optimizing and utilizing already existing context-reasoning techniques, which are perhaps not employed due to system constraints such as time costs or available resources. Verification of low-level context (e.g. ‘light in room’ or ‘high noise level’) is an exception, as the context is solely dependent on readings of a specific sensor. The accuracy of the sensor determines the quality of the context inferred.

We suggest a general high-level, logical approach that makes use of existing context reasoning and acquisition techniques that enables a context-aware system to resolve context ambiguities and optimize sensor-reading values. Our approach is the following: in order to verify a given sensor reading (i.e. low-level contextual information) such as location or light, we use other sensor readings and inferences upon such sensor readings.

2 Logical Verification

The central scheme in a logical verification of context is the ability to resort to other context information that would help us judge sensor attribute values. The verification process assumes the correctness of a specific possibility and determines its probability according to other contextual information; it then switches and assumes the correctness of other possibilities and assigns a probability to each as well. Finally, it selects the most probable alternative. Suppose for example that two identical light sensors in a room yield opposite readings; the first indicates the light is on and the other indicates the light is off. By resorting to other elements, such as the time of day or motion in the room or computing or other activities currently held in the room, the system can assign probabilities to both readings and select the most probable. Suppose that a person is detected to be located in two places at the same time by observing two similar location detector devices (e.g. his PDA location vs. his electronic badge location); the system then resorts to other contextual parameters, which indicate the more probable location value.

We first observe different levels of abstraction in context and in particular distinguish between context that can be obtained directly by observing low-level attribute values/sensor readings (e.g. temperature or location), and more abstract higher level context, which is inferred by a collection of low-level attributes values (e.g. ‘In a meeting’ or ‘Sleeping’). We argue that often more abstract contextual situations can assist in verifying sensor reading values or low-level contextual states. Following the logical verification scheme, we first assume correctness of a specific attribute value a_i^V of an attribute type a_i and then search for a more abstract contextual situation C_j that is made up of several attributes, including the attribute we are trying to verify (denoted by $C_j = (a_1, a_2, ..., a_i, ...a_n)$). We then observe how well the assumed attribute value corresponds to that contextual situation in the current system state (i.e. having current specific values for the other attributes), which yields $P(C_j \mid a_i^V)$ - the probability of currently having the contextual situation assuming a specific value for the ambiguous attribute. We repeat the process for each alternative value for the ambiguous attribute and compare the probabilities associated with each of the values.

3 Experimental Evaluation

The need to verify low level context (i.e. sensor readings) is not only required when two or more similar sensors yield different or contradicting results, but is useful when pervasive systems deal with sensors that are inherently inaccurate. Examples of such inaccuracies can be found in Global Positioning Systems (GPS), which may vary in accuracy between 0.0l meters and 15 meters [13, 4, 9] or indoor positioning mechanisms [12, 4, 9] whose accuracy depends on the number and proximity of wireless access points. Minimizing inferred location errors takes high importance when relatively short distances imply totally different context for context-aware systems, such as in the case of different contextual interpretations for different spaces in a building [11]. For example, the context of ‘Subject in a Meeting’ is completely different from the context of ‘Subject in Lunch’, even though some locations in the meeting room and the dining room are in close proximity, and are only separated by a thin wall.

The approach of verifying low-level context attributes by logically resorting to higher level contextual situations is also applicable in correcting (or what we term *filtering*) the sensor readings errors. By estimating a maximal deviation of an attribute value from its true value, we can assume different attribute’s sensed values and resort to other situations to estimate whether such situations combined with the adjusted attribute’s value are more probable.

We make use of this approach and present a system prototype that filters sensed location readings according to a logical scheme using high-level contextual situations. We also present a simulation, used for critically assessing the logical filtering approach.

The system prototype uses a simple inference mechanism which identifies a current situation by the manifestation of the current context-state within a specific situation [7, 8]. It also has knowledge on predefined location descriptions that are identified as possibly valid locations for sensed location readings (even with no attached situations).

The main scheme behind the logical location filtering is the assumption that the observed sensed attribute is inaccurate and therefore it is sensible to resort to other contextual parameters that can help minimize the incurred error of the observed value. As described in section 2, the system looks for helpful information by examining the probability of having other situations that are inferred using multiple attributes including the location attribute.

For example, consider a case in which no meaningful high-level context can be inferred from the combination of the currently sensed location reading and values of other contextual attributes (assuming here that we have a set of rules which map combinations of sensed attribute values, i.e., low-level context information, to high-level context) – whereas from a combination of the current location reading but slightly modified and values for the other contextual (non-location) attributes, we can infer a specific high-level context with high confidence. In such circumstances, it may be sensible to assume that the location reading was inaccurate and should be adjusted.. The magnitude of the adjustment from the original readings may also influence such decisions, as it can be assumed that errors follow a Normal distribution. Therefore, the greater the magnitude of the adjustment, the greater the error (or deviation from the true location) is assumed, and the less likely such an error would have taken place.

The main procedure for the logical verification used by the system prototype is presented in figure 2.

1. if sensor readings and context-state correspond to same context-space (current active context) then:
➔ Return sensor readings.

2. Adjust location parameters within acceptable error boundaries:
2.1 if location found in current active context and if passed probability distance test then:
➔ Return adjusted location, which has minimal distance to the original sensor readings.
2.2 if location found in other context spaces but not in current active context-space:
A ← location with minimal distance to any of the context spaces.

// reached here if no current active context found in acceptable error distance
3. if original sensor readings are not in a valid location, then force change in location, by
→ Return minimal distance to any valid area.
// reached here if original location is in a valid area
4 ➔ Return A .

Fig. 2. Logical verification procedure

The procedure distinguishes between three possible situations, namely when the original readings already lead to an inferred high-level context, when the readings do not lead to an inferred high-level context but are in a valid area (e.g., not indicating that the position of the person is inside a brick wall), and when the readings are outside any valid area description. In the last case, the system immediately adjusts the estimated location (as the current one is unquestionably wrong – e.g., the user can't be inside a brick wall, at least not typically!), even though such modifications may possibly mean that the original location reading was highly in error.

In the current implementation, the prototype handles two-dimensional information and modifies location values assuming a maximal error reading of five meters in both horizontal and vertical axes. The prototype assumes Normal distribution of the location reading error and for simplicity estimates a discrete pseudo-Normal distribution function for every meter unit of error.

3.1 Simulation and Analysis

We assess the characteristics of the logical filtering with a simulation of a context-aware system that makes use of location positioning system as part of its context inference. We make use of the logical filtering mechanism to judge the location readings and correct the inherent location sensing errors. The following experiment observes a simulated subject's pseudo-random (where randomness is computer generated) activity in terms of its location and high-level contextual situations the subject is locally participating in. It examines a simple floor plan consisting of a meeting room, dining room, the subject office and a corridor, and provides information to the logical filtering mechanism to identify the following contextual situations: 'Subject in office', 'Subject in lunch' and 'Subject in a meeting'. We make use of the following attributes for reasoning about context: 'location', 'office computing activity', 'office door status', 'time', 'subject scheduled meetings', 'motion detected in area *x*' and 'light sensed in area *x*'. The basic experiment's floor plan is illustrated in figure 3, and can be easily extended into more complex plans by defining more areas, valid locations and contextual situations and applying these to the same filtering mechanism. We also allow a degree of unpredictability by occasionally disproving the applicability of a specific current context state even though it can be inferred by the system. For example, if a subject is in the dining room, at lunchtime, having no particular scheduled meeting for that time, he may occasionally be only accompanying a friend than having lunch himself.

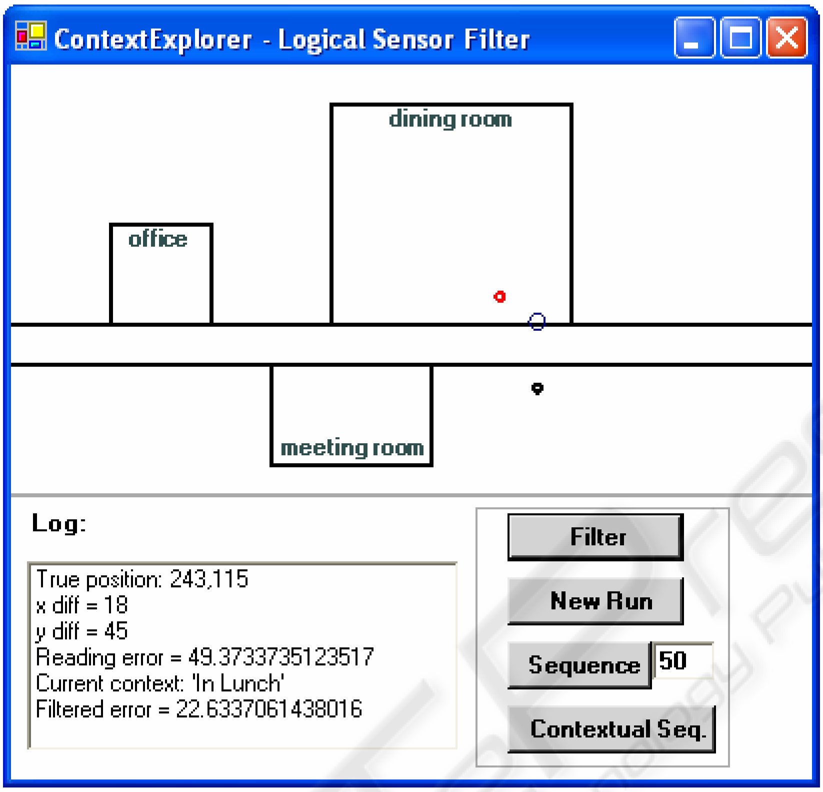

Fig. 3. Basic experiment floor plan

The three circles in the floor plan represent: the actual true subject's location – denoted by the red circle, the sensed location reading – denoted by the black circle and the new location estimation after logical filtering procedure – denoted by the thinner and wider blue circle.

We will now go through the actions of the logical filter for a typical scenario, which is illustrated in figure 4.

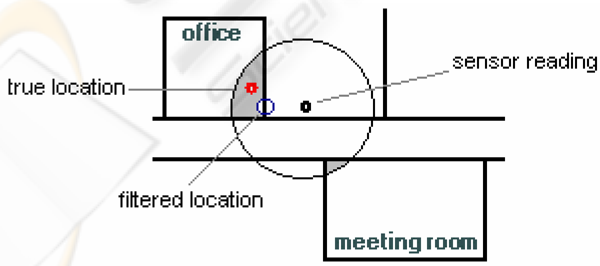

Fig. 4. Typical scenario illustration

The logical filtering procedure starts by receiving the sensor reading location, without knowing the actual true location. It then checks the probability of being in that specific location. For example, it tries to match this location with permissible positions of any of the predefined known situations. In the example of figure 4, the sensor reading location is actually outside any valid area in the floor plan. This location does not match any location definition for any known situation and is evidently wrong. Next, after assuming an error in the location reading, the procedure iterates on other possible positions for the true location, based on the maximal error distance it assumes (around 7 meters in this experiment). The large circle in figure 4, having the sensor reading in its center, denotes the area for the iteration process. At every iteration the procedure assumes the correctness of a new position within the large circle and checks if the current assumed position corresponds to some predefined contextual situation and if so, it checks whether this particular contextual situation is currently active. In our example, two areas correspond to known contextual situations, the shaded areas in the office and in the meeting room, corresponding to two possible high-level contextual situations the user could be in: ‘subject in office’ and ‘subject in a meeting’. The procedure checks which one of these situations is probable and if more than one is probable, which is the most probable. The ‘subject in a meeting’ situation does not seem to be active since no light and motion are sensed in the meeting room and no meeting is scheduled for the subject in the current time. The ‘office’ situation is more likely to be currently active, since the office door is not locked; light is sensed in the office as well as some computing activity. Together with having the subject location attribute corresponding to a position somewhere in the office, a good match with the ‘office’ situation is attained. The procedure assigns probabilities to possible areas while also considering the distance of the area from the sensor reading (assuming Normal distribution of the error implies less chance to have large distance errors). It then selects the most probable one if it exceeds a predetermined probability threshold, and in our case the shaded area in the office. Within this area it selects the position that is the shortest distance from the original sensor reading, again based on the Normal distribution assumption.

Figure 5 provides a collection of other typical experimental runs handled by the simulation.

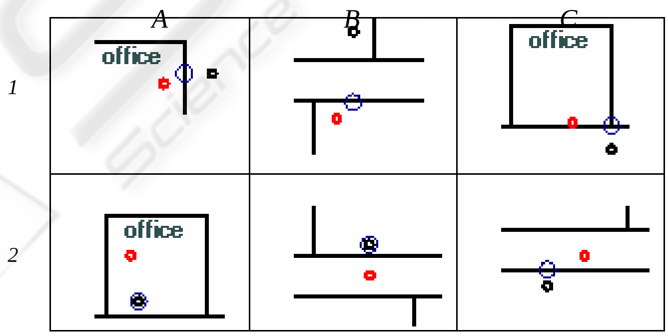

Fig. 5. Example logical filtering scenarios

We observe three typical occurrences in figure 5; the first 3 figures (*A1*, *B1* and *C1*) depict tangible filtering effects over the location sensor reading values. In these examples, the sensor readings are located outside the area of the true high-level context (e.g. in example *C1* the area of the high-level context is the office box and contains the subject's true location). The filtering procedure weighs the feasibility of each possible high-level contextual situation, each situation as inferred by a different location value, as the location values are changed within the acceptable range of values. As the application is not aware of the true subject's location (the red circle) it performs the minimal change possible in the estimated location so it would enforce the new contextual situation while deviating minimally from the sensor reading values.

Examples *A2* and *B2* depict situations where no gain is achieved with the logical filtering. In *A2*, although the sensed location is different from the true location, it is still within the currently inferred contextual situation; hence filtering does not provide additional useful information. In example *B2*, the subject's true location is in the corridor, which does not conform to any high-level context. The filtering mechanism cannot find any high-level contextual situation that would fit even when the location readings are adjusted (within its boundaries), and by default maintains the original sensor readings.

Example *C2* depicts a situation where the sensor reading values are outside the valid pre-defined areas. In such cases, even though the subject's true location does not conform to any known context, the procedure forces a change towards the closest valid area.

We performed experimental runs to examine the characteristics of logical filtering; the results are shown in figures 6. We observe a consistent trend of optimization of the location attribute, having lower error rates after applying logical filtering. Figure 7 shows data produced after 100 runs. Errors are compared to true location by Euclidean distance between the filtered location and true location, and between the true location and sensed location.

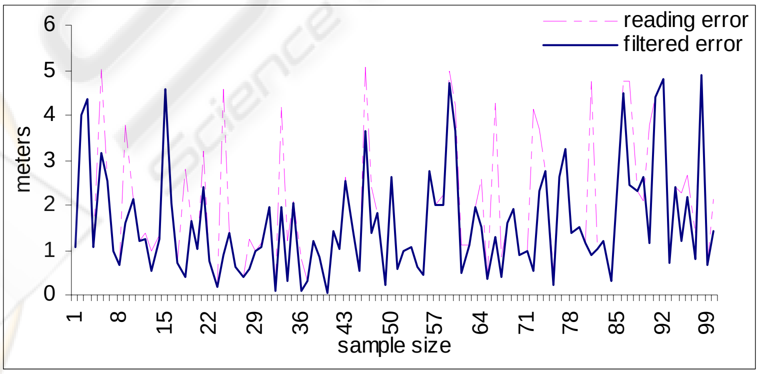

Fig. 6. Filtering errors vs. sensor reading errors

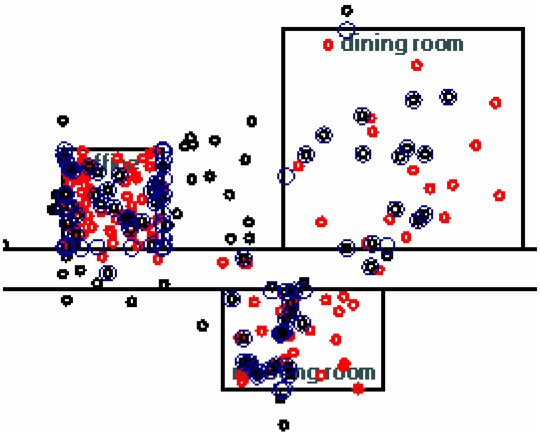

Fig. 7. Accumulated 100 runs

3 Conclusion and Critical Analysis

We have observed clear improvement in average location error after performing logical filtering for general positioning mechanisms. Although a relatively simple floor plan was used to demonstrate the impact of logical verification, the procedure can be extended to more complex and elaborate settings and can be used in a variety of scenarios concerning other types of low-level attributes. The degree of success of this approach is nevertheless dependent on the suitability of the system's contextual configuration, i.e. to the various predefined high-level contextual situations, the attributes they are inferred by and the accuracy of the process of reasoning about context in general.

The ability to reason about context incorrectly, given a false current sensor reading is in particular hazardous in a logical filtering process as we often assume new estimated locations and the filtering process may be tempted to assign importance to non-existent but inferred situations. To illustrate the problem, consider the following scenario. A subject is in close proximity to the dining room, say in the corridor, at lunchtime, with no scheduled meetings. The sensed location turned out to be in a non-valid area. The logical filtering process assumes different locations by modifying the location readings and verifying the new locations with the current non-location attribute values. It identifies that 'Subject in lunch' contextual situation will become possible if the sensed location was slightly adjusted. It favors this situation and changes the sensed location values, creating an even greater error. This scenario is illustrated in figure 8 where no probability considerations were assigned to the possible error distance, and is arguably a limitation of the basic logical filtering algorithm.

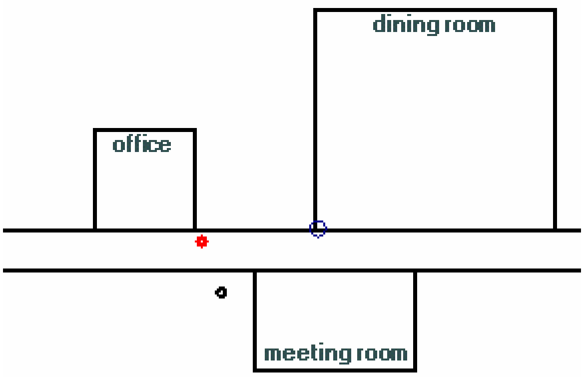

Fig. 8. Increase in error due to logical filtering

The likelihood of such scenarios occurring depends on the capability of having all attributes values except location denote a specific situation, while in reality, considering the true location reading, having a different inferred contextual situation. This scenario can possibly be avoided by a wise selection of attributes that eliminate such situations. It may also be possible to overcome this scenario by adding more complex rules to the basic logical filter such as tracing historical events for better inference or avoiding the proposed value changes if it incurs a great magnitude in distance correction.

Despite the mentioned limitation, a logical layer that makes use of the system's available context reasoning techniques for low-level context verification is evidently useful and can be used to decrease inherent sensor inaccuracies and resolve contextual ambiguities. As attribute information is obtained by the system in order to infer contextual situations, the contribution of the logical filter is in effect greater than only the actual gain from the decrease in the attribute inaccuracy. By applying the logical filter a significant contextual change occurs; the original sensed value may change even minimally but in such a way that it would be identified with another more probable contextual situation, consequently assisting the system in performing much better context inference. Generalizing from the approach, the use of semantics holds promise in resolving ambiguities and we believe we have merely touched the tip of the iceberg of what is possible.

References

1. Bardram J. E., Applications of Context-Aware Computing in Hospital Work – Examples and Design Principles, In Proceedings of ACM Symposium on Applied Computing (ACM SAC) 2004, ACM Press, 2004
2. Dey A. K., Jennifer Mankoff, Gregory D. Abowd and Scott Carter Proceedings of the 15th Annual Symposium on User Interface Software and Technology (UIST 2002), Paris, France, October 28-30, 2002. pp. 121-130.

3. Glassey R., Ferguson I., Modeling Location for Pervasive Environments, First UK-UbiNet Workshop, London, UK, 2003
4. Hightower J., Borriello G., Location systems for ubiquitous computing, IEEE Computer 34(8):57-66, August 2001
5. Huuskonen P., Pavel D., Tuomela U., Context-awareness: A new dimension for communication, http://www.nokia.com/nokia/0,,33758,00.html
6. Mäntyjärvi J., Seppänen T., Adapting Applications in Mobile Terminals Using Fuzzy Context Information, Mobile Human-Computer Interaction: 4th International Symposium, Mobile HCI 2002, Pisa, Italy, September 18-20, 2002
7. Padovitz A., Loke S. W., Zaslavsky A., Towards a Theory of Context Spaces, Workshop on Context Modeling and Reasoning (CoMoRea), at 2nd IEEE International Conference on Pervasive Computing and Communication (PerCom'04), Orlando, Florida, March 2004.
8. Padovitz A., Zaslavsky A Loke S. W., Burg B., Stability in Context-Aware Pervasive Systems: A State-Space Modeling Approach, Workshop on Ubiquitous Computing (IWUC 2004) at the 6th International Conference on Enterprise Information Systems (ICEIS 2004), Porto, Portugal, 2004, accepted for publication
9. Patterson C. A., Muntz R. R., Pancake C. M., Challenges in Location-Aware Computing, April-June 2003, issue of IEEE Pervasive Computing, available at: http://dsonline.computer.org/0306/d/bp2spo.htm
10. Satyanarayanan M., Pervasive Computing: Vision and Challenges, IEEE PCM August 2001, pp. 10-17.
11. Satyanarayan M Coping with Uncertainty,., IEEE CS Pervasive computing Journal, August 2001
12. Ekahau Positioning Engine, http://www.ekahau.com
13. GPS Information, available at: http://www.gpsworld.com, http://www.colorado.edu/geography/gcraft/notes/gps/gps_f.html