

An approach to Data Fusion for Context Awareness

Amir Padovitz¹, Seng W. Loke¹, Arkady Zaslavsky¹, Claudio Bartolini² and Bernard Burg²

¹ Centre for Distributed Systems and Software Engineering, 900 Dandenong Rd., Caulfield-East, Victoria, Australia

{amirp,swloke,arkady.zaslavsky}@csse.monash.edu.au

² HP Labs, Palo-Alto.

{claudio.bartolini}@hp.com

Abstract. We develop and propose an approach modeled with multi-attribute utility theory for sensor fusion in context-aware environments. Our approach is distinguished from existing general purpose fusion techniques by a number of factors including a general underlying context model it is built upon and a set of intuitions it covers. The technique is developed for context-aware applications and we argue that it provides various advantages for data fusion in context-aware scenarios. We experimentally evaluate the performance of our approach with actual use cases using real sensors.

1 Introduction

Various approaches attempting to partially reduce the incurred uncertainty associated with context and attempts to build effective models that describe context are still at an early stage. Context itself can reflect circumstances borrowed from a variety of domains, including social interactions and human perception, and reasoning about context often involves the task of identifying human users' situations. Subsequently, the diversity and complexity of the conditions involved in reasoning about context contribute to the inference challenge.

One key ingredient in inferring and interpreting context is fusing together sensor data. The integration of sensed information, preferably coupled with other (perhaps higher-level) reasoning methods provide a promising approach (although not complete) for reasoning about context.

In this paper we develop and propose a novel approach modeled with multi-attribute utility theory for fusing sensor data. Our approach is distinguished by its association with a general context model as the underlying description of context related knowledge. It attempts to incorporate various intuitions that should impact context inference, to produce a better fusion result. Our approach is specifically intended for use in context-aware applications, and exhibits many characteristics desirable in reasoning about context. The underlying general context model offers a new theoretical perspective on context that enables development of new reasoning techniques including the fusion technique discussed in this paper.

We organize this paper as follows. In Section 2 we introduce our theoretical modeling approach referred to as the Context Spaces model, and discuss different concepts, which are used during the development of the fusion process. In Section 3 we develop our fusion algorithm, as a practical extension of the Context Spaces model. We first discuss the intuitions we seek to cover in the fusion and then develop the technique itself. We experimentally evaluate our proposed approach in Section 4 in two stages. We use simulation experiments with which we analyze the performance of the fusion and the intuitions we seek to cover; and we provide results of our fusion technique in actual office-related use cases using combinations of real-life sensors. We conclude our paper in Section 5.

2 The Context Spaces Theoretical Model

Our objective is to develop an approach for inferring the occurrence of situations by a context-aware system for any type of situation it is familiar with. It should be able to provide a solution to a query such as: “Is situation X occurring?”

To enable this capability for general use (i.e. by different applications for different types of situations and available information) we develop a formal model that represents perceived context and situations of interests of a system. We then develop a technique for fusing the modeled information to achieve situation awareness.

The paradigm of context-aware computing can be regarded as an attempt to obtain information with limited sensing capabilities, but which nevertheless reflects circumstances useful to the application at hand. Designers of a context-aware application would seek to define perceivable context, which reflects interesting circumstances or real-life situations, as accurately as possible, governed by limitations such as technology and cost.

The fundamental nature of context (in context-aware computing) can therefore be regarded as a constrained view of a system over the world, which can either be immediately used (for triggering actions) or can represent real-life situations (in which case further computation may be required to relate the context to the situation, e.g. data fusion techniques).

The Context Spaces theory aims to model this fundamental nature of context and enable context and situation awareness for systems that are able to sense information with various degrees of imperfections at runtime. It is an attempt towards a general context model to aid thinking and describing context. The concepts use intuitions from geometrical spaces and state-space models - we hypothesize that geometrical metaphors such as states within spaces are useful to guide thinking about context, though the user will need to take the concepts and elaborate on them within the context of the application to be built.

In the process of developing our fusion technique we will make use of the following concepts.

2.1 Context Attribute

Definition 1. A context attribute (denoted by a_i) is defined as any type of data that is used in the process of inferring situations. A context attribute is often associated with sensors, virtual or physical, having the value of the sensor readings denote the context attribute value at a given time t (denoted by a_i^t).

2.2 Context State

Definition 2. A context state describes the application's current state in relation to chosen context, and is denoted by a vector S_i . It is a collection of context attributes' values that are used to represent a specific state of the system at time t . A context state is $S_i^t = (a_1^t, a_2^t, \dots, a_N^t)$ where S_i^t denotes a vector defined over a collection of N attribute-values, where each value a_i^t corresponds to an attribute a_i 's value at time t .

2.3 Situation space

Definition 3. A situation space represents a real-life situation. It is a collection of regions of attribute values corresponding to some predefined situation and denoted by a vector space $R_i = (a_1^R, a_2^R, \dots, a_N^R)$ (consisting of N acceptable regions for these attributes). An acceptable region a_i^R is defined as a set of elements V that satisfies a predicate P , i.e. $a_i^R = \{V | P(V)\}$.

For example, in numerical form the accepted region would often describe a domain of permitted real values for an attribute a_i . A region of acceptable values is defined as a set which satisfies some predicate; hence, it can contain any type of information, numerical or non-numerical.

We complete the basic model with a set of functions, which add more realism to the representation by relating interesting facts to the fundamental concepts. In this paper we will consider two of these functions, as follows.

2.4 Relevance function

Definition 4. Given a situation space i , a relevance function e_i associates weights w_1, \dots, w_N with regions of values a_1^R, \dots, a_N^R of i , respectively, where $\sum_{j=1}^N w_j = 1$. A weight $w_j \in (0,1]$ represents the relative importance of an attribute region a_j^R to other regions in the situation space's definition.

In many cases some types of information are more important than others for inferring a situation, e.g. high body temperature may be a strong indication of a general sickness of a person while other attributes may not be so important to infer that spe-

cific situation. For example, high respiratory rate may be caused by and therefore also indicative of other situations, such as a person doing physical exercise. To model this variation in the importance of context attributes for inferring a situation, we define the relevance function, which assigns weights to context attributes. The weights reflect how important each attribute is (relative to other attributes) in describing a situation.

2.5 Contribution function

Definition 5. A function \mathfrak{h}_j assigns a contribution level $c \in (0,1]$ for each element in a region of values j . The contribution level of an element in a region reflects how well is that element (an attribute value) associated with the modeled situation.

In the relevance function we model the relative importance between the regions of a situation space, whereas in the contribution function we model the individual contribution of elements within a specific region for inferring a situation.

This allows a differentiation between different values for the same context attribute. For example, it might be that some values are better reflecting the purpose of that specific region than others. E.g., for a context attribute of ‘body temperature’ in the definition of ‘subject is healthy’ the values between 36.5 and 36.7 would reflect a contribution of 1 (on a scale between 0 and 1) and values between 36.3 and 36.5 and between 36.7 and 36.8 would reflect a lesser degree of supportive contribution.

3 Data Fusion Algorithm for Context-Aware

The most basic intuition covered in the Context Space paradigm is that of representing context in terms of state and space. The existence of the context state within some situation space indicates, to a certain degree, the occurrence of the situation represented by that space. While this represents an initial reasoning approach, we have argued that additional information is necessary for more accurate and realistic modeling of the situation and therefore needs to be considered during inference.

We develop a data fusion technique, based on multi-attribute utility theory (MAUT) [6] that takes the information represented by the model (i.e. the condition of the context state and the definition of the situation space to be inferred) and computes a degree of support for the occurrence of that situation. The degree of support is then compared with support for other alternative situations or with a support threshold predefined by the system designers, to facilitate a decision regarding the occurrence of the situation.

The algorithm works by accumulating positive indications for the occurrence of a situation. These indications cover several intuitions that we believe are important to be considered in reasoning about context. They include the following:

(1) *Individual significance and contribution of context attributes -*

This information is modeled by the relevance and contribution functions in the model.

(2) Completeness of containment -

In context-aware computing application designers specifically choose a relatively small number of context attributes to represent a particular situation. Therefore it is significant that values of all chosen attributes correspond to the situation definition.

(3) Different characteristics of context attributes and their effect over the fusion -

We distinguish between two types of context attributes in regard to the definition of a situation space, which have different effects over the fusion outcome, as follows.

Intrinsic:

Definition 4. *An intrinsic context attribute is an attribute which negatively affects the inference of a situation if its value is not within the acceptable region of values of the same attribute type.*

If some context attribute (e.g. light level) is defined as intrinsic then having a value outside the corresponding region in the situation space would decrease the support for the occurrence of that situation.

Optional:

Definition 5. *An optional context attribute is an attribute that assists in inferring a situation (i.e. having values of such attributes within the accepted regions would increase the probability of the situation), but sensing values outside the accepted region would not weaken the support for having the situation.*

This distinction between different types of context attributes, therefore, exercises dissimilar impact over the final inference result. To clarify this distinction, examine the following intuitive examples. First consider a situation space defined for a situation of ‘subject is healthy’ with an intrinsic attribute of ‘Body Temperature’ and acceptable region of values between 36.6 and 37.2. Deg. Celsius. Any value outside this region, say 38 Deg. Celsius, would mean (with some degree of confidence) that the subject is unhealthy. Therefore the values of this type of attribute must always remain within the region of values specified for it in the situation space.

In contrast, when we use a context attribute ‘PDA location’ of a user to help us infer ‘user in a meeting’ situation, the accepted region for this attribute would be location information corresponding to the meeting room. Having indication of a PDA located in the meeting room will assist to infer the situation. However, if the PDA location is sensed to be elsewhere, the ‘user in a meeting’ situation may still be valid (e.g. the PDA was left at home or in the office). Therefore, this type of attribute would be referred to as optional, since it only assists in inferring a situation but does not refute a situation if its values are outside the situation space.

(4) Completeness of containment vs. individual contribution -

There is a trade-off between ensuring complete containment of all intrinsic attributes and their individual contributions. An important context attribute should produce greater impact on the final result of the fusion than one which is considered less important. In contrast, when several attributes are equally very significant for the evaluation of the real-life situation we may want to ensure that all of them are con-

tained, in particular when they are explicitly chosen at design time. We therefore need to account for this type of trade-off in the fusion process.

(5) Inaccuracies of sensors, yielding uncertain context attributes values -

Inherent inaccuracies associated with sensor readings should affect our confidence in the outcome of matching state (defined over inaccurate context attributes values) and space. The true state of an attribute which value is outside some region of values might actually be contained within that region, and vice-versa. Sensors are often inaccurate and it is important to naturally incorporate accuracy estimation in the fusion model.

3.1 A Utility Based Fusion Approach

We make use of MAUT as the basis for deriving a measure that reflects the accumulated support for the occurrence of a situation represented by the situation space. The process considers the intuitions previously described and results in a single numerical-valued measure ranging between 0 and 1.

MAUT provides a convenient way for combining together seemingly different contributions into a single measure, expressing the result in terms of utility [6]. In our case, utility can be defined as the contributed support of having a context attribute value within a corresponding region for inferring the situation, constrained by additional considerations. The more indications we have that the context state matches the definition of a situation space, greater utility is gained and vice versa.

MAUT is considered an evaluating scheme, which provides a general evaluation function $v(x)$ over an object x to denote the overall object's utility. The evaluation function is traditionally defined as a weighted accumulation of evaluating the objects' value dimensions [6, 3], which represents a combination of different contributions relevant to the object. Each dimension can also be individually evaluated using the general approach of weighted contribution by evaluating its own value dimensions.

In the following we combine the model's concepts and our intuitions to reflect a single utility measure. (If some intuition is irrelevant then it should not be included; the principle of a utility combination approach remains, however, the same.)

We start by defining a dimension d_1 , evaluating the fact that a context attribute value matches the situation space definition.

$$(1) \quad d_1 = \sum_{i=1}^n \hat{w}_i \cdot p(a_i^t \in a_i^R), \text{ where } p(a_i^t \in a_i^R) \text{ is the estimated probability of}$$

having a context attribute a_i 's value contained in its corresponding region of val-

ues a_i^R in the situation space, and \hat{w}_i (where $\sum_{i=1}^n \hat{w}_i = 1$) is a weight measure ex-

pressing the relative importance of the specific context attribute evaluation to the

overall utility. The calculations of \hat{w}_i is done using a process of weights redistribution.

In general, in terms of MAUT, $p(a_i^t \in a_i^R)$ reflects the evaluation of a particular attribute containment; the higher the probability of an attribute value being within the region, the greater contribution is evaluated for the attribute. The more confident we are in the containment of an attribute, the more this attribute can contribute to support the situation. \hat{w}_i reflects the individual weight or significance of the attribute in the overall evaluation process of d_1 . The need to determine weights with additional computation (\hat{w}_i rather than w_i) is the result of considering the characteristics of the context attributes, being either intrinsic or optional.

For example, an optional attribute may be an important ingredient in inferring a situation and therefore should be associated with a high weight value. However, if this attribute does not correspond well to the situation space, its absence should not affect the inference result (since it is defined as optional). This fact would affect the weight values of the remaining context attributes, which should still reflect the same relative importance between themselves.

The general steps for determining weights for equation (1) are the following.

Let $w_1, w_2, \dots, w_k, \dots, w_{k+m}, \dots, w_N$ denote initial weights associated with a given context state S_i^t for the specific fusion procedure, where weights are determined according to their relative importance in regard to a specific situation.

1. Repeat for each optional context attribute k :

1.1 If $p(a_k^t \in a_k^R) < C$ then $w_k = 0$, where C defines a threshold value

2. Repeat for each context attribute i (both optional and intrinsic):

$$\hat{w}_i = w_i / \sum_{j=1}^N w_j$$

During redistribution of weights, the system examines how well an optional attribute fits the definition of a situation space. If a sufficiently low confidence in the containment of that context attribute is gained (thereby reflecting the disassociation of the attribute with the space) it is ignored by assigning its weight to zero. Finally, after the adjustment of all optional attributes, the overall remaining attributes' relative weights are recalculated.

Next, we define a dimension d_2 , evaluating the contribution of the fact that the entire intrinsic attributes match the definition of the situation space. We compute d_2 as the joint probability of the containment of all intrinsic attributes in the situation space, where we assume attributes are independent, as follows.

$$(2) d_2 = \prod_{i=1}^m p(a_i^t \in a_i^R), \text{ where } a_i \text{ is an intrinsic context attribute}$$

We have so far defined two dimensions that express different aspects that are important in obtaining an accumulated support measure and contribute to an overall utility estimate (d_1 looks at individual attribute contribution and d_2 at having a complete containment of all attributes). We now proceed and combine these two dimensions in a general utility function, as follows.

$$(3) U = q_1 d_1 + q_2 d_2, \text{ where } q_1 + q_2 = 1$$

The overall utility U considers each dimension's relative weight in determining a combined result. The individual weight (i.e. q_i) of each of the two dimensions in the overall utility measure greatly depends on the situation at hand. More specifically, it depends on the distribution of the weights in equation (1).

An interpretation of d_1 and d_2 reveals a trade-off between the need to have all the attributes contained within a space (reflected by d_2), and the need to account for their individual contributions (d_1). If for example, all context attributes are equally significant (e.g., as an application developer determines), then it is sensible to put more emphasis on d_2 , i.e. having all the attributes contained within the space. So that if an attribute is outside the situation space, an overall low confidence would be achieved. If on the other hand, some context attributes are much less significant, then it would be more appropriate to give greater importance to d_1 , i.e. give more weight to the significant attributes containment, rather than insisting on having all attributes contained.

Consistent with this line of reasoning, we set the overall utility weights by analyzing the differences between the weights defined in the situation space definition. Greater differences increase q_1 and smaller differences increase q_2 . We suggest the following simple computation to determine the weights:

$$q_2 = \frac{\hat{w}_{\min}}{\hat{w}_{\max}}, \text{ where } \hat{w}_{\min} \neq 0, \text{ and } \hat{w}_{\min} \text{ and } \hat{w}_{\max} \text{ denote minimal and maximal}$$

weights after weights redistribution process, respectively. Alternatively, other approaches such as standard deviation analysis can be used.

3.2 Estimating containment

We have so far used the expression $p(a_i^t \in a_i^R)$ to denote an estimate for the containment of an attribute within its corresponding region. As long as no sensor inaccuracy is considered, a_i^t also represents the actual value or true state of the sensed attribute value. However, as sensors can in general be inaccurate, the true state might be significantly different than what is reflected by the sensor readings. We would therefore need to estimate the probability of the true state being within the region of values, i.e. $p(s_i^t \in a_i^R)$, where s_i^t denotes the true value. (If we are unable to per-

form such estimation, then $p(a_i^t \in a_i^R)$ would represent complete confidence in the actual observation represented by the sensor readings).

Knowledge about sensors and data accuracy can be obtained by various means. These include manufacturer specifications or confidence estimation as part of the provided service (e.g. in [9] the position engine provides a confidence measure associated with the inferred location). Alternatively, testing sensors for accuracy with sufficient sample sizes and knowledge of the true state can be used to derive estimates of readings errors. Similar approaches have been suggested in [7] for using ground truth knowledge on the current actual sensor performance and using historical knowledge on sensor performance as they age.

3.3 A General approach for Estimating Containment

We now develop a general approach for deriving confidence in containment of the true state of a context attribute sensed value in a region of values, given a known distribution of the reading errors or some estimation for it. For simplicity we assume that changes in the context attribute values do not significantly change the errors' distribution behavior (i.e. their mean and standard deviation). This estimation is intended for sensor readings that correspond to continuous region of values as well as discrete regions, which contain all possible elements between some minimum and maximum boundaries. (E.g. all enumerated values of some semantic non-numeric attribute between some maximum and minimum enumerated values).

Let $e_j = a_i^t - s_i^t$ denote an error of a sensor reading from the true state of the context attribute value it senses, where s_i^t represents the true state measured by a sensor for context attribute a_i and a_i^t denotes the sensor reading value. Note that the error is intentionally characterized as having signed value rather than absolute; (when a sensor is biased, reading errors may have significantly different densities for values higher or lower than the true state).

Let a_{\min}^R and a_{\max}^R denote minimum and maximum values of an accepted region of values a_i^R , which corresponds to a context attribute a_i .

Proposition. *The probability of the true state s_i^t being contained within the region*

a_i^R *can be computed by:*

$$p(s_i^t \in a_i^R) = p(e_j \text{ between } (a_i^t - a_{\min}^R) \text{ and } (a_i^t - a_{\max}^R)).$$

Proof: *For containment in a continuous region of values the context attribute true value must be greater or equal to a_{\min}^R and smaller or equal to a_{\max}^R , i.e.:*

$$(1) a_{\max}^R \geq s_i^t \geq a_{\min}^R. \text{ Given the definition } e_j = a_i^t - s_i^t, \text{ then } s_i^t = a_i^t - e_j.$$

Substitute s_i^t with $a_i^t - e_j$ to obtain: (2) $a_i^R \geq a_i^t - e_j \geq a_i^R$. Rewrite as:

$$(3) a_i^t - a_i^R \geq e_j \geq a_i^t - a_i^R$$

The practical implication of this is that given an estimation of the reading errors' distribution we can estimate the true state containment in the region by the observation (i.e. the inaccurate sensed value) alone. For example, if we know that the reading errors follows, say some *Normal* distribution estimation, than the containment of the state could be computed by: $p(e_j \leq a_i^t - a_i^R) - p(e_j < a_i^t - a_i^R)$.

4 Experimental evaluation

We have described a utility based approach, which considers various intuitions that we believe are important when fusing together sensors data for reasoning about context. We proceed with a set of experiments that examine the behavior of our measure and its sensitivity to various changes in the environment. We separate the experimental evaluation into two parts. In the first, we use a simulation of a variety of sensor types, inaccuracies and scenarios. We seek to show how our intuitions are captured in the outcome of the support measure. In the second part, we provide results of an experiment, which uses real-life sensors, where we try to distinguish between three types of situations in real-life office settings, using the fusion process.

4.1 Performance Analysis

We compute the support measure in a setting of 'user in a meeting' situation, using a variety of simulated sensor types, accuracy levels and varying knowledge over the sensors accuracies. A description of the sensors used here is provided in Table 1. Each sensor activity is simulated using random value generation according to natural changes in the true event it senses and its defined inaccuracy. We allow different knowledge and type of inaccuracy for different sensor types. For example, we produce *Normal* approximated discrete samples to characterize location errors, assign Boolean result with fixed known accuracy for light detection and assume no knowledge whatsoever about the inaccuracy of noise level detection.

We assign importance on a scale between 1 and 5 and associate a relative weight for each sensor, corresponding to its significance for inferring the situation. Here, we seek to obtain a general sense of relative importance, using expert knowledge, rather than an exact resolution of the weights, which may be harder to achieve. For example, in estimating if a specific user is in a meeting, the fact that the user is located in the meeting room is more significant than the fact that noise is detected in the room.

Finally, we characterize each context attribute in the situation space definition as being either optional or intrinsic. For example, we assume that a user always carries his RFID tag but may leave his PDA behind, therefore refer to user location inferred by the PDA location as optional. Similarly, a projector may or may not be active

during a meeting. Its activity would contribute to the inference of a meeting but not the opposite.

attribute name	importance (1-5)	optional	weight
User RFID Y Location	4	No	0.114286
User RFID X Location	4	No	0.114286
User PDA Y Location	3	Yes	0.085714
User PDA X Location	3	Yes	0.085714
MR Light Level 1	4	No	0.114286
MR Light Level 2	4	No	0.114286
MR Noise Level	2.5	No	0.071429
MR Motion Detected	2.5	No	0.071429
MR Projector Active	4	Yes	0.114286
MR Microphone Active	4	Yes	0.114286

Table 1 –Sensors definitions for simulation of ‘user in meeting’.

First, we examine the effect of optional attributes over the support measure. Figure 1 depicts the results of the measure for the ‘user in meeting’ situation for three scenarios. In the first scenario all sensors yield values that correspond to the definition of the situation space (i.e. all attributes values are contained in corresponding regions). This results in relatively high support for the specific situation (i.e. A) and varies in time according to inherent inaccuracies of sensors and changes in the experiment settings (such as user changes locations inside the meeting room). In the second scenario, we position the user’s PDA in his office. As this attribute is defined as optional the overall support remains nearly identical, yielding high support for the situation (i.e. B). Minor differences compared with the previous settings result from fewer number of attributes participating in the fusion (The PDA location is not considered).

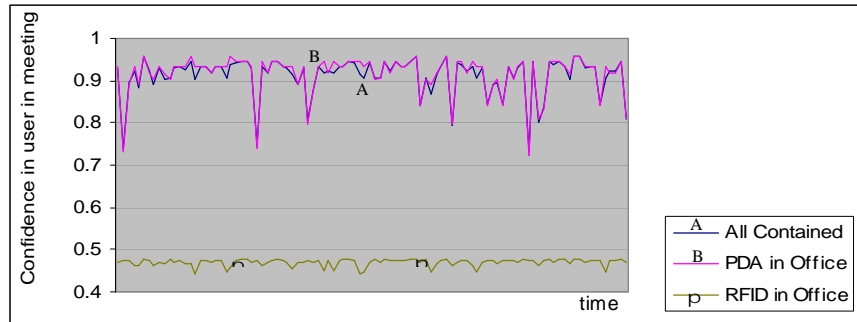


Fig. 1. Effects of optional attributes

In contrast, the effects of positioning the user’s RFID tag outside the meeting room are significant. The support measure immediately drops to reject the ‘user in meeting’ situation. This corresponds to a scenario when the user is actually in his office but forgot his PDA for example, in the meeting room.

Next we examine the general behavior of the measure when we significantly change the readings of various sensors. Figure 2 depicts different support levels for the situation in three different scenarios. In the first scenario all sensors yield values, which are contained in their corresponding regions of values. In the second scenario, values are only partially contained in the regions, due to their inherent inaccuracy and experiment settings.

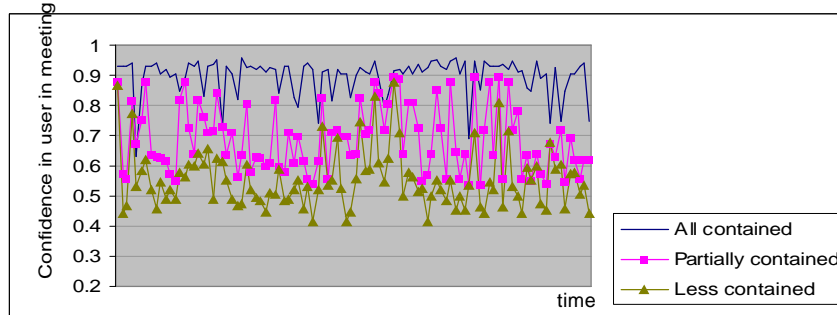


Fig. 2. Effects of sensors values

For example, simulating the user being located close to the boundaries of the room together with high inaccuracy of the positioning mechanism. The last scenario, depicts situations where some of the events the sensors sense are truly outside the region of values. For example, when the user is in the corridor close to the meeting room and do not participate in the meeting. As expected, the results show degradation in support as sensed values reflect notably different situations.

Finally, we observe in figure 3 the effect of associated inaccuracies of sensors. The higher the inaccuracy of a sensor, a lower degree of support is gained for the situation and vice versa.

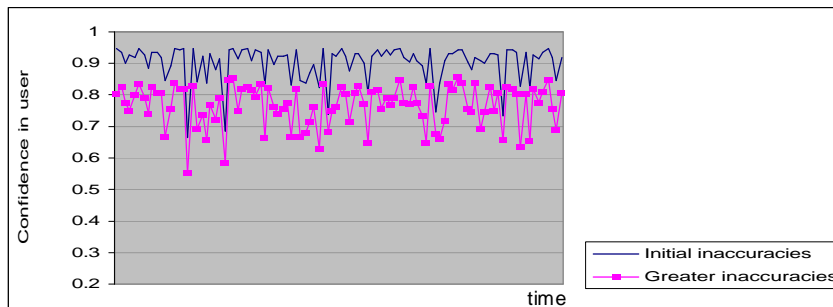


Fig. 3. Effects of sensors inaccuracies

The experiment's results reaffirm the expected behaviour of the fusion and the intuitions it attempts to capture.

4.2 Performance in a real-life use case

We follow with a performance analysis of a real-life use case fusing together a blend of different real sensors and data sources. In the following experiment we were interested in identifying and distinguishing between three types of user activities, taking place in our smart meeting room, namely, (1) a user giving a presentation, (2) a user attending another's presentation and (3) a user attending a meeting.

To reason about these activities we have selected four basic context attributes physically positioned in different locations, specifying the user location, the meeting room light level, the user's notebook keyboard and mouse activity and presence of active presentation processes on the user notebook. We defined the corresponding region of values of the context attributes for each situation space with different appropriate values, including optional and intrinsic attributes and associated inaccuracies of readings. For user location we used Ekahau Positioning Engine [9] that tracks the user's personal devices such as her PDA and notebook. The positioning service computes spatial positions by analyzing wireless signal strengths and comparing them to previous calibration. It provides an associated confidence of its inferred location with a measure between 0 and 1. We used Berkeley Motes [10] for sensing and communicating light levels in the meeting room. For retrieving information about the user's presentation activity we have implemented a service that hooks to the notebook operating system and provides information on latest keyboard and mouse activity. We also provide a service that identifies active presentation processes in the notebook.

In the meeting room we use a portable light-weight presentation projector that is connected to the presenting user's notebook. It is a common practice and the assumption in this experiment that each presenter uses his personal notebook.

During experimentation we have revealed significant inaccuracy in location estimation due to less than optimal wireless network settings that affect signal strength and number of optimal access points. The Positioning Engine Server itself is located in a remote location and communicates with an agent that performs the fusion via wireless 802.11b infrastructure.

The functional architecture of the reasoning process is as follows. An agent working on behalf of the user and running on the user notebook is equipped with communication protocols for exchanging information with remote sensing services as well as with local independent processes that provide information on notebook activity. The agent performs fusion using the inputs of the incoming information and assigns a support measure to each of the candidate situation spaces, which represent the real-life activities. Before applying the fusion, information is pre-processed either by the agent or remotely, depending on the type of data. For example, light levels are sampled by the motes sensors a number of times and then averaged. The Motes Interface Service, which handles this information, matches the averaged result against predefined light levels and communicates back a predefined value. In contrast, interpretation of keyboard and mouse activity is preformed directly by the agent. Here, the amount of time lapsed since the last captured activity influences the confidence that the user is currently using his notebook. Various similar approaches of preprocessing

raw sensor readings (also known as cooking the data or using cues) are common in fusion for context recognition and have been shown to significantly improve inference results (e.g. [2, 1]).

During actual experimentation we have switched between the activities and observed the results of the measures for the different situations. Figure 4 provides support measures for the three activities. In this experimental run we have started with the user presenting first for 15 minutes then attending a colleague's presentation for 15 minutes and finally participating in a discussion or general meeting on the topics presented (again for 15 minutes). Interpretation of the results reveals matching support levels with the actual activity taking place. At the time the user is presenting, the support for this particular activity averages around 0.9 whereas support for other situations is significantly lower. A change in the situation towards the user only attending another's presentation results in a drop of the 'User Presenting' situation to support levels below 0.4, and a rise in the support for the 'User attending a presentation' situation to levels around 0.9.

Similarly, when a discussion (equivalent to a meeting) over the presentations involving our user is starting right after the second presentation, the support for 'User in a meeting' situation rises to 0.9 and support for the former situation drops significantly.

The successful results of this experiment can be partly contributed to observing assumed behaviors associated with the defined activities, such as having the lights switched off during presentations and on during the following general discussion, or controlling the presentation shown from the user's notebook

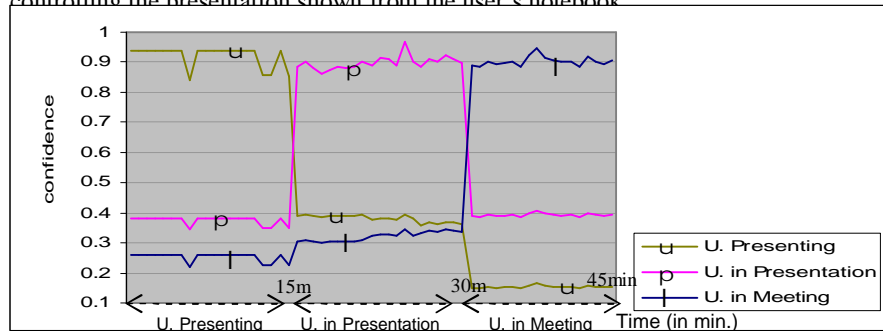


Fig. 4. Experimental run

In order to relax these assumptions more context attributes need to be introduced, such as those discussed and demonstrated in section 4.1. Here, the choice and design of sensor networks clearly affects the quality of inference [8, 4, 5]. In this experiment we have shown that our approach is applicable for fusion in real-life scenarios using real sensors and data.

5. Conclusion

We have presented an approach for fusing sensor readings that considers intuitions that we believe are pertinent in reasoning about context. Our approach can be considered within the general field of sensor fusion modeled with MAUT for situation classification. We have also offered a unique theoretical perspective about context that we use to develop a general context model and general purpose inference procedures.

Having an inference technique, which is an extension of a general context model, is an appealing feature in itself, as it can be applied seamlessly over different domains and context-aware scenarios. Given information captured by the general context model, the fusion can be immediately applied to infer the occurrence of a situation of interest.

Acknowledgments

We thank HP for sponsorship of this work

References

- [1] Gellersen H.W., Schmidt A., Beigl M., Multi-Sensor Context-Awareness in Mobile Devices and Smart Artifacts, in *Mobile Networks and Applications (MONET)*, Oct 2002.
- [2] Golding A. Lesh R., N., Indoor navigation using a diverse set of cheap, wearable sensors, In *Proceedings of the third International Symposium on Wearable Computers*, 1999.
- [3] Schafer R., Rules for Using Multi-Attribute Utility Theory for Estimating a User's Interests, workshop on Adaptivity and User Modelling in Interactive Systems (ABIS), University of Dortmund, Germany, October 2001.
- [4] Schmidt A., Van Laerhoven K., How to build smart appliances. *IEEE Personal Communications, Special Issue on Pervasive Computing*, 8(4):66-71, August 2001.
- [5] Van Laerhoven K., Schmidt A., Gellersen H.W. Multi-Sensor Context Aware Clothing, 6th Intl. Symposium on Wearable Computers, October 07 - 10, 2002, Seattle, Washington.
- [6] Winterfeld D., Edwards W. *Decision Analysis and Behavioural Research*, Cambridge, England, Cambridge University Press 1986.
- [7] Wu H., Siegel M., Steifelhagen R., Yang J., Sensor Fusion Using Dempster-Shafer Theory, *IEEE Instrumentation and Measurement Technology Conference*, Anchorage, AK, USA, May 21-23, 2002, May, 2002.
- [8] Yoshimi B., On Sensor Frameworks for Pervasive Systems, *Workshop on Software Engineering for Wearable and Pervasive Computing (SEWPC)*, ICSE 2000, Limerick, Ireland.
- [9] Ekahau Positioning Engine, <http://www.ekahau.com>
- [10] Information on Motes available at: www.tinyos.net/, www.xbow.com/