# Strathprints Institutional Repository

Thomson, G. and Nixon, P. and Terzis, S. (2004) *Towards ad-hoc situation determination.* In: First International Workshop on Advanced Context Modelling, Reasoning And Management, 2004-01-01, Nottingham, England.

# Towards Ad-hoc Situation Determination

**Graham Thomson, Paddy Nixon and Sotirios Terzis**
Global and Pervasive Computing Group
Department of Computer and Information Sciences
University of Strathclyde
Glasgow, UK
*firstname.lastname*@cis.strath.ac.uk

**ABSTRACT**

Toolkits such as PlaceLab [1] have been successful in making location information freely available for use in experimental ubiquitous computing applications. As users' expectations of ubiquitous computing applications grow, we envisage a need for tools that can deliver a much richer set of contextual information. The high-level situation of the current environment is a key contextual element, and this position paper focuses on a method to provide this information for an ad-hoc group of people and devices. The contributions of this paper are i) a demonstration of how information retrieval (IR) techniques can be applied to situation determination in context-aware systems, ii) a proposal of a novel approach to situation determination that combines these adapted IR techniques with a process of cooperative interaction, and iii) a report of preliminary results. The approach offers a high level of utility and accuracy, with a greater level of automation than other contemporary approaches.

**INTRODUCTION**

The original vision of ubiquitous computing proposed by Mark Weiser describes an environment where the computing machinery contained within it silently and automatically adapts to its inhabitants' behaviours, "invisibly enhancing the world" [2].

In order for such computers to adapt, they must be able to sense and analyse the environment in which they exist. That is, they must be context-aware [3].

Location information is an essential element of context, and toolkits such as PlaceLab [1] have been successful in making location information freely available for use in experimental ubiquitous computing applications.

As users' expectations of ubiquitous computing applications grow, we envisage a need for tools that can deliver a much richer set of contextual information. The high-level situation of the current environment is a key contextual element. It is a natural pivot to which users and application programmers can associate behaviours.

The approaches to situation determination offered by the state-of-art context-aware infrastructures [4, 5] experience the following drawbacks:

- An expert of the particular environment is required to specify the correlation of the available contextual information elements with the situations that occur.
- Reasoning is performed by large logic programs [5] or Bayesian networks [4], which must be manually constructed and maintained.
- As the number of available contextual information elements and situations that occur increases, it becomes increasingly difficult for an expert to decipher and specify correlations.
- The situation specifications will suffer from the subjective bias of the expert who programmed them.
- It is a single common 'oracle' that performs the situation determination, and therefore cannot incorporate knowledge of the environment's inhabitants that is not public.
- No support for ad-hoc situation determination. Recognition is limited to the fixed number of cases programmed by the expert for the local environment.

This paper presents a novel approach to situation determination that attempts to address these issues. The approach is automatic as far as can be made possible. There is no need for an environment expert, the only manual configuration that is required is that users click a button to capture a 'snapshot' of the situations they wish their devices to recognise. Specific contextual information captured in a snapshot is abstracted, so that ad-hoc situations, as well as the specific situations captured, can be recognised. By combining the snapshots of different users, situations are recognised by their 'true nature', that is, the characteristic features that are common to almost all snapshots. All inhabitants of the environment collaboratively perform the determination process. An individual can incorporate knowledge of the environment that is only known to him / her, and so can identify situations more specifically. Each time the situation is determined in an environment, each individual learns how to identify the situation more accurately. The determination process continuously improves, and also automatically adapts to situations as they evolve over time.

## SCENARIO

Imagine an office manager, Jane, who carries with her a new smartphone. She wishes to configure the smartphone such that the device will alert her to incoming calls and messages by the most appropriate means depending on her current situation. During her coffee break, Jane selects a 'capture situation' option on her smartphone, which then takes a snapshot of the contextual information detectable by her smartphone in the current environment, just as a photographic camera would capture the visible image. She then marks this snapshot as 'Afternoon coffee break', and instructs the smartphone that both calls and messages should be announced audibly when she is in that situation. She then uses her smartphone to connect to the company's main context server, and searches for the contextual information captured during a design review meeting she had attended yesterday. She captures a snapshot from during this time and marks it as 'Formal meeting', and instructs her smartphone that all calls and messages must be silenced during a formal meeting, and announced on-screen when the meeting has ended. Noticing the time, Jane makes her way across town where she must meet with clients from a different company to negotiate a contract. On entering the other company's meeting room, her smartphone does not recognise her situation, so sends a request for the current situation to the other devices in the room. The context server for the room replies, informing Jane's smartphone that the current situation is a formal meeting, and the smartphone then applies Jane's preferred configuration automatically.

This example illustrates the three main stages of our proposed situation determination approach. Initially, Jane captures and stores the current situation. Therefore, it is necessary to have a concrete representation for the situation snapshots. On entering the other company's meeting room, her smartphone begins a recognition phase. Failing to identify the situation by itself, the smartphone enters into the final stage of cooperative interaction. In the following sections, we shall look at each of these stages in turn.

## REPRESENTING THE SITUATION

The notion of contextual information is central to this work. Several definitions for context are provided in the literature, though commonly these are difficult to apply operationally, and assume an intuitive understanding of the notion of situation [6].

Coutaz and Rey offer an operational definition of context that bundles the key concepts drawn from the literature, which is drawn from an explicitly defined notion of situation [6]. Their definition provides a succinct mathematical framework on which we base our work:

> Given a set of users $U$, a task $T$, and two instants of observation, $t_0$ and $t$, where $t_0$ is the temporal reference for observations, the Context at $t$ that is related to $U$ for performing $T$, is the composition of the Situations observed between $t_0$ and $t$ that relate to $U$ for performing $T$.
>
> $$context^{U,T}(t) = COMPOSITION($$
>
> $$situation^{U,T}(t_0), ..., situation^{U,T}(t))$$
>
> where: $situation^{U,T}(t)$ is the Situation at $t$ that relates to $U$ for performing $T$. The Situation is a set of the values observed at $t$ of the peripheral state variables that relate to $U$ for performing $T$, as well as their relations. Peripheral state variables denote the entities that are not central to $U$ at $t$ for performing $T$, but that may have an impact on $T$, now and/or in the future.

We define an ontology that specifies the relations and the classes of the observed values. A relation links a single instance of a class to an instance of the same or another class. By making the contextual information that is used conform to an ontology, it enables snapshots captured in different ubiquitous environments to be exchanged easily and interpreted correctly, as we can define translations between the terms of the different ontologies used in the environments. Also, as we shall see later, by exploiting the subclass relations in the ontology we can gain greater reasoning power when matching situations. Both these aspects are essential to the nomadic nature of our approach.

Ubiquitous computing environments are open environments. That is, any number and variety of people, devices, and software may appear within them. The contextual information produced by such environments is therefore also open, as the instances of a relation are drawn from a potentially infinite set. Reasoning about contextual information is made difficult by this fact, as many traditional data mining and machine learning techniques make strict demands on the structure and constraints of the data. For example, the well-known C4.5 classification algorithm requires that each attribute considered for classification, as well as each attribute's range of possible values, be stated prior to the start of the classification process. Such approaches are clearly unsuitable for contextual information. Even if a particular ubiquitous computing environment defined a tight structure and constraints upon the contextual information it used, there is no guarantee that another environment would do the same. A more free form approach is required.

The task of text classification in the field of information retrieval aims to automatically assign documents to a given set of categories. The text in documents may exhibit no regular structure. The set of known terms used in the documents may grow with each new document that is classified. It is easy to see similarities with the task of text classification to that of situation determination. The following section demonstrates how common document representation techniques can be adapted to the representation of a situation snapshot. In doing so, it allows us to apply the large body of research from the field of information retrieval on text classification to our situation determination approach.

### Situation as a document

It is common in information retrieval to treat a document as a bag of terms. Similarly, we treat a situation snapshot as a bag of relations. The relations captured from the environment links two instances of a class. For example, Jane 'owns' her smartphone. When reasoning about snapshots, we wish to be able to do so at an abstract level. We have greater utility reasoning about a person who owns a smartphone and who is sitting in a coffee lounge, rather than specifically about Jane who owns her particular smartphone and is sitting in the coffee lounge of her company building. To accommodate this, an expansion algorithm processes each of the relations in the bag. For each instance on the left and right hand side of the relation, the algorithm traverses the inheritance hierarchy from the class of the instance to the class that is linked by the relation. New relations are created and added to the bag that connect the instance and each class of the left hand side to the instance and each class of the right hand side. In its expanded form, this collection of relations explicitly declares the layers of abstraction that are lost when considering just the instance-based relation.

In order to reason about snapshots, we must transform them into a representation suitable for machine learning algorithms. To achieve this, we use the vector space model, which is commonly used by IR systems to represent documents. The concepts of the vector space model are sketched here briefly. A fuller treatment of the subject can be found in [7].

The vector space model considers a document to be a vector in a multi-dimensional Euclidean space. Each axis corresponds to a term. The coordinate of each axis of the vector is determined by the following function:

$$d_t = TF(d,t)IDF(t)$$

where $d_t$ is the coordinate of document d in axis t, $TF(d,f)$ is the term frequency of term $t$ in document $d$, and $IDF(t)$ is the inverse document frequency of term $t$.

Term frequency is calculated as follows:

$$TF(d,t) = \begin{cases} 0 & \text{if } n(d,t) = 0 \\ 1 + \log(1 + \log(n(d,t))) & \text{otherwise} \end{cases}$$

where $n(d,t)$ is simply the number of times term $t$ appears in document $d$. The logarithms of this value are taken to normalise for document length.

Not all axes in the vector space are equally important as some terms shall appear many times in a document regardless of its content. Applying the inverse document frequency seeks to scale down the coordinates of these terms. It is calculated as follows:

$$IDF(t) = \log \frac{1 + |D|}{|D_t|}$$

where D is the set of document, and $D_t$ the set of documents that contain term $t$.

Any combination of instances or classes in a relation are considered a unique term. For example, if the labels in double quotes denote an instance, those in single quotes to be a relation, and those without quotes to be a class, "Jane" 'works with' "Robert", "Robert" 'works with' "Jane", "Robert" 'works with' Person, and Person 'works with' Person, are all unique terms.

The number of terms in a collection of documents is likely to follow a $zipf$ distribution, that is, where a few terms occur very often while many others occur rarely. The $IDF$ scaling is used so that rare terms, which may characterise a document, are not swamped by more common terms. Through experimentation, we found that the number of relations in the situations we considered also follow a $zipf$ distribution. For example any social situation will be swamped with friendship relations. This makes $IDF$ scaling also appropriate for our situation determination approach.

To incorporate the time of capture of a situation snapshot, an extra axis is added to the vector. The time axis is not subject to $TFIDF$ scaling, but scaled from 0 (midnight), to 1 (just before midnight).

## DETERMINING THE SITUATION

In this section we will consider the case where an individual device is attempting to determine the situation from the contextual information it can gather from its environment.

Determining the situation is a classification problem. The aim is to assign the current situation to one of the marked-up situation categories, for example, to 'Coffee break' or 'Formal meeting'. Support Vector Machines (SVM) are currently the most accurate classifiers for text [8], and it is this approach that we use to classify the current situation.

A high level overview of SVMs is given here, please refer to Chen et al. for more details [9].

Consider the case where we wish to determine if the current situation snapshot represents a coffee break. We can construct a binary SVM classifier that has the ability to determine if the snapshot is in the category labelled 'coffee break', or not in that category. The SVM is trained on a set of examples, each marked as a positive or negative example for the 'coffee break' situation. The SVM splits the vector space into two partitions, one for each class. The partition in which the current situation snapshot is positioned represents the category to which the snapshot belongs.

To illustrate how the partition is calculated, consider a training set of $n$ situations consisting of positive and negative examples, that are linearly separable by a hyperplane. The SVM seeks to find a hyperplane such that the distance from any training example is maximised. The optimal separator is orthogonal to the shortest line connecting the convex hulls of the two categories, and intersects it halfway. An example is shown in Fig. 1.

More complex SVMs can be constructed to handle cases that are not linearly separable. Other varieties of SVM include non-linear and multiple-class models. The LIBSVM soft-

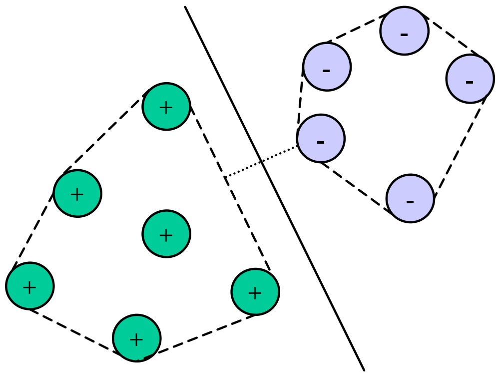


**Figure 1: A hyperplane partitioning the situation space in a binary SVM classifier. The circles represent positive and negative examples of a category. The dashed lines represent the convex hulls. The solid line is the optimal hyperplane.**

ware supports all these types of SVM, and it is this package that we use to classify situations [10].

### COOPERATIVE INTERACTION

So far we have focused on how an individual device can perform situation determination. In a ubiquitous computing environment, there are likely to be many devices, both personal and common. By determining the situation individually, we are ignoring the utility offered by including the other devices.

In a given environment, each user and device may have different access control policies applied to him or her, and as such, the available contextual information about the current environment will differ in each case. Therefore, each participant will be able to reason about the current situation to a lesser or greater degree. By cooperatively interacting with one another, devices that can confidently determine the current situation correctly can assist the other devices that are failing to do so.

In our approach, we partition contextual information into three privacy categories. Both the individual relations and the full snapshot itself may be assigned to a particular category. The three privacy categories are as follows:

**Public** The information is available to every other participant in the environment.

**Privileged** The information is only available to certain people and/or devices, as specified by an access control policy.

**Private** The information is known only to the owner.

The view that a single participant will have of the context of the current environment will be the union of the set of private relations known only to it, each of the sets of privileged relations it has access to, and the set of public relations.

Privacy constraints must also be enforced when snapshots are shared, which happens when devices collaborate to identify the situation. For example, when a public or privileged snapshot is shared, it must first be stripped of any private relations, as well as the privileged relations to which the receiver has not been granted access.

The process of determining the situation within an environment is a cooperative effort between each participant within the environment. Clearly, the participants who are cooperating to determine the situation, must all be in the same situation. We make the simplifying assumption that participants are considered to be in the same situation if they are in the same room. In other cases, participants could be regarded as being in the same situation when they occupy the same logical zone such as a seating area in a train, or simply by having been in each other's company for a certain length of time.

When the process of interaction begins, each participant proposes what it believes the situation to be, based on the set of relations that is visible to it. It is this step that was described in the previous section. If all participants agree on the same situation, then the situation has been successfully determined.

In the case where the participants do not agree, further action must be taken, and this begins the cooperative interaction process.

The first step in this process is for all participants to share relevant snapshots. Which snapshots that are relevant will be those that the SVM has currently marked as support vectors. The support vectors will be those that are closest to, but not over, the boundary for the category of the snapshot. All participants then re-evaluate the situation. If there is still disagreement, the cooperative process enters a correction phase.

The contextual information that is used to determine the situation cannot be assumed to be perfect. As demonstrated by Henricksen et al., by identifying the source of contextual information, we can reason about its accuracy [11]. Context that is supplied by a user is likely to be correct, though it may be stale in circumstances where the user has neglected to keep the information up-to-date. In cases where there is conflicting user-supplied information, we could take the most recently updated copy to be correct. Context that is detected by sensors may have several quality attributes associated with a reading such as freshness, accuracy, and confidence. When sensor readings conflict, depending on the quality attributes associated with them, we could take the readings that are freshest, have the highest confidence, or take the average of readings that are reported within a certain accuracy. In cases where context is derived from other context, we may favour the results of a derivation algorithm that uses more accurate inputs or is known to out-perform the alternative. Marking up relations with such metadata allows the correction stage to exploit the presence of multiple copies of the same contextual information, which may have been gathered from several different sources, to increase the reliability of the information used to determine the situation.

If all previous stages fail to unanimously identify the situation, the situation must be chosen based on a heuristic measure. Past experience has shown that a certain situation may be most strongly characterised by a non-obvious indicator. For example, determining whether a particular person was busy or not was most strongly indicated by a single sensor reading - whether or not the person's office door was open or closed. As such strongest non-obvious indicators could be held as private relations of a participant, it is likely that a heuristic based on the highest individual confidence will give the best performance.

Cooperative interaction concludes with a learning stage. The product of cooperative interaction is a new pairing of a snapshot to a situation label. Each of the participants can take this pairing as a new training example. This creates a process of continuous automated learning. It may even be desirable to discard older training cases in favour of these new cases, so that any 'drift' in a situation can be managed without manual intervention.

## PRELIMINARY RESULTS

In this section, we demonstrate a proof-of-concept implementation of our situation determination approach. We wanted to discover if the situations that occur in the meeting room of our research lab could be accurately determined by our approach.

Before constructing the necessary infrastructure to collect live data, we wished to experiment with simulated situations to evaluate the potential value of collecting different contextual information elements, and gauge the approximate situation determination accuracy that we could hope to achieve.

The relations of a situation snapshot are produced from a probabilistic generative model. For a given situation, each event that can occur is assigned a probability of it occurring independently in that situation. When a new snapshot is created, the likelihood of it containing a relation corresponds to the probability distribution for that situation. For each situation, the probability distribution was constructed by hand, based on expert knowledge.

The ontology that was defined for this experiment was based upon the relations that could realistically be measured in our meeting room. These include the identity of the devices in the room, and as we assume that the presence of a person's PDA to be synonymous with the presence of the person, we also include the identities of the people in room. Process table information is probed to determine which applications are running on a device. Static relations between people, devices, and groups are drawn from address book entries, personal profile information, and administrative records. These are friendship, supervision, group membership, and device ownership relations.

The following list states the different situations that we aimed to identify, accompanied by a short description of their characteristics.

**Formal group meeting** The formal group meeting is held once a week at the same time. The attendees are likely to be from the research lab group, and roughly ten in number. The minutes of the meeting are always noted, and occasionally presentations are given.

**European project meeting** European project meetings are held frequently, and arranged at a time that suits each of its four group members. The meetings are mostly for discussion, though occasionally drafts of papers are worked on collaboratively.

**Informal meeting** An informal meeting may take place at any time and can involve any member of the department. The number of people present will commonly be small, though larger groups are possible.

**Coffee break** Coffee breaks tends to happen at mid-morning and late afternoon. Music is often played in the room during a coffee break.

**Private study** The meeting room is occasionally used for private study by individual PhD students from the research lab.

**Movie night** The meeting room is sometimes transformed into a small movie theatre. A laptop is linked to the projector and used to play a DVD. People from the whole department may attend.

The probability distributions of each situation model were constructed to reflect these characteristics.

We examined the special case of an environment in which all contextual information and snapshots were public. This is equivalent to the case of having a single set of relations to reason with, and a single participant performing the determination. In doing so we can estimate the maximum accuracy that can be achieved by this approach.

Text classification tasks often draw on a large set of examples during the training phase. If our approach also required a large set of training examples, it would not be suitable for situation determination. Ubiquitous computing environments are dynamic. New situations will continually appear, while current situations will cease to recur. It is therefore necessary to recognise new situations as quickly as possible, that is, the system must achieve an acceptable determination accuracy using as few snapshots as possible. Figure 2 shows the results of determining the situation based on a varying number of available snapshots. An average accuracy of 72% is reached after only 5 snapshots have been accrued for each situation, increasing to 86.2% when each situation has 50 snapshots. Investigating the misclassified situations revealed that almost all misclassifications occurred between the three meeting situations. Note that continuous automated learning is an inbuilt feature of our approach. Each time the situation is determined, a new training example is created. A large repository of snapshots will build up quickly over time.

partitions has on the overall determination accuracy.

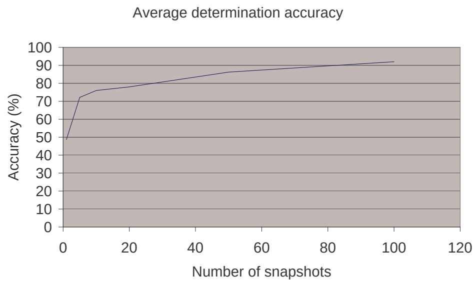


**Figure 2: This chart shows the average classification accuracy of a trained SVM attempting to determine a single situation given a fixed number of training example snapshots for each situation. A multi-class $\nu$-SVM classifier with a linear kernel function is used. The results are averaged over the result of forty runs for each situation.**

## SUMMARY AND FUTURE WORK

In this paper, we proposed a novel technique for ad-hoc situation determination. Initially, we exemplified the level of manual configuration required by our approach, which was simply to click a button at a key moment to mark a situation, and detailed the technical expertise that is required for configuration by other contemporary approaches. We illustrated how information retrieval techniques can be applied to both the representation and reasoning of contextual information, allowing us to effectively manage the open nature of context in ubiquitous computing environments. By treating a situation snapshot as a document of terms, it was shown that the vector space model could be applied to the representation of contextual information, positioning a situation snapshot in situation space. We highlighted that by drawing relations from an ontology and exploiting inheritance relations, we can determine ad-hoc, as well as specific, situations. We then explained how the situation of a snapshot could be automatically identified using powerful Support Vector Machine techniques. Our cooperative interaction process was described, which illustrated the potential of the presence of many ubiquitous computing devices to enable the correction of conflicting contextual information, increase the determination accuracy of devices for which contextual information or situation snapshots are scarce, and facilitate continual automated learning, such that the overall determination accuracy improves automatically over time. The paper concluded with a demonstration of our approach based on simulated data. The results were encouraging, showing that a reasonable accuracy can be achieved using a relatively small set of situation snapshots.

We are currently developing the necessary software to allow us to assess the value that each contextual information element adds to the determination process. The aim is that such feature selection will also be performed automatically. We are also experimenting with the cooperative interaction techniques and analysing the effect that introducing privacy partitions has on the overall determination accuracy.

A related project within our lab will provide the software infrastructure that will enable the collection of live snapshot data. We are also working on the development of SVM clients for Bluetooth-enabled devices, which will provide a prototype system allowing us to experiment with real-time ad-hoc situation determination.

## REFERENCES


1. B. Schilit, A. LaMarca, G. Borriello, W. Griswold, D. McDonald, E. Lazowska, A. Balachandran, J. Hong, and V. Iverson. Challenge: Ubiquitous location-aware computing and the place lab initiative. In *Proceedings of The First ACM International Workshop on Wireless Mobile Applications and Services on WLAN (WMASH 2003), San Diego, CA*, September 2003.
2. M. Weiser. The computer for the 21st century. *Scientific American*, pages 94–104, September 1991.
3. A. K. Dey. Understanding and using context. *Personal and Ubiquitous Computing*, 5(1):4–7, 2001.
4. A. Ranganathan, J. Al-Muhtadi, and R. H. Campbell. Reasoning about uncertain contexts in pervasive computing environments. *IEEE Pervasive Computing*, 3(2):62–70, 2004.
5. H. Chen, T. Finin, and A. Joshi. A context broker for building smart meeting rooms. In *Proceedings of the Knowledge Representation and Ontology for Autonomous Systems Symposium, 2004 AAAI Spring Symposium*. AAAI, March 2004.
6. J. Coutaz and G. Rey. Foundations for a theory of contextors. In *Computer Aided Design of User Interfaces, Springer Verlag*, 2002.
7. R. Baeza-Yates and B. Ribeiro-Neto. *Modern Information Retrieval*. Addison-Wesley Longman Publishing Co., Inc., 1999.
8. Y. Yang and X. Liu. A re-examination of text categorization methods. In *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 42–49. ACM Press, 1999.
9. P. H. Chen, C. J. Lin, and B. Schlkopf. A tutorial on nu-support vector machines. *Applied Stochastic Models in Business and Industry*, 2004.
10. Libsvm - a library for support vector machines. Available online at: http://www.csie.ntu.edu.tw/ cjlin/libsvm/.
11. K. Henricksen and J. Indulska. Modelling and using imperfect context information. In *Second IEEE Annual Conference on Pervasive Computing and Communications Workshops*, 2004.