CONTEXT AWARENESS IN A MOBILE DEVICE: ONTOLOGIES VERSUS UNSUPERVISED/SUPERVISED LEARNING

John A. Flanagan

Nokia Research Centre
P.O.Box 407, FIN-00045 NOKIA GROUP, FINLAND
adrian.flanagan@nokia.com

ABSTRACT

Context awareness is presented as a means to facilitate the interaction between a user and a mobile device. Facilitating this interaction requires a personalization of the context awareness. The ontology approach to context awareness is top down,where human understandable concepts are defined. However, defining these concepts in terms of low-level information is complex and is a major barrier to personalization. On the other hand unsupervised learning can be used to generate higher level user contexts from low-level information in a simple manner. It is argued that an unsupervised approach to context awareness is workable compared to any ontology based context awareness.

1. INTRODUCTION

One of the goals of mobile or ubiquitous computing is to enable devices to sense changes in their environment and to automatically adapt to these changes based on user needs and preferences. We refer to this ability as *context awareness*. Consider a simple example of context awareness and its possible use in mobile devices. A context aware device, ideally, should sense the user's context and react in an appropriate manner that facilitates the user interaction with the device. This raises the problem of *personalization*. The question of personalization is not restricted to defining user needs and it is argued in what follows that personalization in the definition of user contexts is also required.

The most commonly cited definition of context quoted in the literature related to mobile computation is by Dey [1],

Definition 1 (Context) *Any information that can be used to characterize the situation of an entity. An entity is a person, place, or object that is considered relevant to the interaction between a user and an application, including the user and applications themselves.*

Given this entity based definition of context it is not surprising the most dominant approach to the problem of context recognition in the mobile environment is based on ontologies and inference or reasoning based on ontologies [2], [3], [4]. In what follows it is argued that the characterization of real-world entities is a complex information processing task on real world information sources, such as sensor signals. This complexity can severely restrict the ability of users to personalize the context awareness functionality.

Another approach to the context awareness problem has been taken in [5],[6] where unsupervised learning, or more specifically the Symbol String Clustering Map (SCM) [7], for context recognition has been used. In this approach, environment information such as sensor signals, are measured and basic features are extracted and quantized. The fusion of the quantized feature states of several information sources at any time results in a symbol string. The SCM clusters the symbol string data in an unsupervised manner and the resulting clusters are interpreted as contexts.

In Sec.2 a general overview of ontologies and learning is given in order to distinguish what are the essential conceptual differences between the two approaches. This is followed in Sec. 3 by a development of the reason why mobile devices can and should be context aware and a more specific discussion of the application of both ontologies and learning to context awareness and their relative advantages and disadvantages. A quasi-supervised learning method for labelling the contexts arising from unsupervised learning is presented in Sec. 4. A short conclusion is given in Sec. 5.

2. LEARNING AND ONTOLOGIES

2.1. Ontologies and Inference

The first use of ontologies in computer science goes back to Artificial Intelligence (AI) in the 1960's [8]. The following is a dictionary definition of an ontology with respect to its use in AI[1].

Definition 2 (Ontology) *An explicit formal specification of how to represent the objects, concepts and other entities that are assumed to exist in some area of interest and the relationships that hold among them*

For AI systems, what "exists" is that which can be represented. When the knowledge about a domain is represented in a declarative language, the set of objects that can be represented is called the universe of discourse. We

[1] From www.dictionary.com

can describe an ontology by defining a set of representational terms. Definitions associate the names of entities in the universe of discourse (e.g. classes, relations, functions or other objects) with human-readable text describing what the names mean, and formal axioms that constrain the interpretation and well-formed use of these terms. Formally, an ontology is the statement of a logical theory.

From this definition the first obvious fact is that something exists only if it can be represented and that representable knowledge defines the ”universe” that can be considered. This type of representable and closed universe is both practical and necessary. Practical in that the different entities and their interpretation in the universe can be labelled in a human understandable form and in a form that can be manipulated by software in a computer. The resulting ontology is governed by a set of logical statements about the represented knowledge. The closed universe is necessary to ensure that the logical statements are consistent[2].

Reasoning or inference on an ontology using a descriptive logic (DL) [9] can be explained as reasoning in two parts referred to as the T-Box and A-Box. Actors from the real world or the domain of interest are mapped into items or instances in the A-Box. The formal definitions of concepts or classes in the T-Box are used to map items from the A-Box into these concepts. Hence inference on real world items is possible and generalized by considering real world items as instances of classes in the ontology, on which the inference is defined. Figure 1 shows an illustration of the T-, A-Box and real world representation.

Context awareness can then be defined using this approach as inference on the ontology. For example, consider the ”work meeting” context. Using DL a meeting in the ontology can be defined as a certain group of people who are work colleagues, in a given location which correspond to the T-Box concepts or classes. Consider real world items such as ”Mary”, ”Bob” and ”Tom” who are colleagues and a meeting room ”Room B4”. These real world items are considered to be the relevant actors and are mapped into A-Box instances of the T-Box concepts. Using inference on the concepts in the T-Box the context in this situation can be inferred to be a ”work meeting” the defined concept in the T-Box ontology.

2.2. Unsupervised Learning and the SCM

The SCM algorithm processes symbol strings in a time sequential and unsupervised manner. Each symbol in a symbol string represents the state of an information source at that instant in time. The symbol string representing the fusion of the states of the information sources at that time instant. The functioning of the SCM is independent of the origin or characteristics of the information source. After training with a set of input data the result is a set of representative symbol strings which define clusters in the input data. Note there is no supervision here and the outcome of the training is entirely dependent on the input data used in the training. This means there is a certain unpredictability of the outcome if there is no a-priori information on the input data. This unpredictability is in the sense that it is not possible to predict which node will represent which cluster in the data. On the other hand for the same data set the resulting clusters should be consistently the same.

[2] It is unclear whether this statement contradicts Gödel’s incompleteness theorem

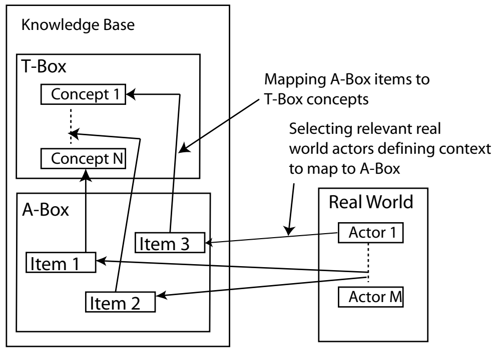

Figure 1. T-Box, A-Box and real world entities for context awareness using ontologies.

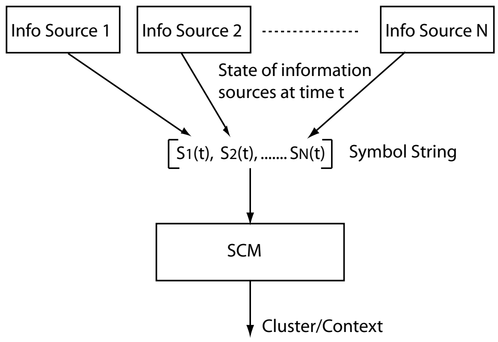

Figure 2. Illustration of the process of generating and clustering symbol strings with the SCM.

The structure of the SCM in its simplest form (K-SCM, [10]) consists of a set of nodes. Each node has an associated representative symbol string which in turn has an associated weight vector. For a given input symbol string the best matching node is determined and referred to as the winner. This node in fact represents the cluster to which the input string is identified as belonging to. When used in context awareness the winning node for a given input symbol string is also interpreted as representing the context. Figure 2 shows an illustration of the states of a set of information sources fused into a symbol string at time t and processed by the SCM to produce a cluster or context label.

3. CONTEXT AWARENESS THE SCM AND ONTOLOGIES

As mobile device and network technology evolve there is a real possibility of information overload for the user. Furthermore the technological capabilities of the mobile terminal is set to increase considerably in order to incorporate all these advances in sources and applications. A mobile device has quite a limited UI, both input (i.e. keypad) and output (i.e. small screen). Hence the problem is further exaggerated by the limited ability of the user to interact with the device to manage all this information, services and technology. Context awareness in some sense can act as an information filter, providing the user with a select set of information or services appropriate for the user in a given context enabling the user to manage the device in a more effective manner.

3.1. Context Awareness and Ontologies

The use of ontologies for context awareness has some obvious advantages, not least the ability to communicate context information. A second benefit of ontologies in context awareness is the ability to name different concepts in machine readable fashion which could be even useful to the user allowing for the use of everyday words and concepts when interacting with the technology.

On the other hand ontologies by their very definition are based on well defined concepts and logic. In many cases people generally understand what is meant by different concepts, for example everyday ones such as ”at home” or ”at work”. In a context aware application based on ontologies it would seem necessary for each user to define ”at home” and ”at work”. In technical terms it means defining an inference rule on the ontology. Assuming there is basic information available such as time and location etc. each user has to define what for them is ”at home” and ”at work”, not always straightforward.

While defining concepts in ontologies requires input from the user, maintaining the ontologies also requires user intervention. Consider the example used in section 2.1 with ”Tom”, ”Bob” and ”Mary” as ”people” who are also ”work colleagues”. Now the question must be asked as to who defines ”Tom”, ”Bob” and ”Mary” as ”work colleagues”. If ”Tom”, ”Bob” and ”Mary” are senior management in the company would the person who also works for the company and cleans the offices be considered their ”work colleague”. Maybe there is a new colleague ”Anne”. How can the information needed to define the concepts be fed into the application? Using the mobile terminal, or should it be done on a PC using another application and by who ?

In conclusion it would seem that when using ontologies for context awareness, even if it is possible to have named concepts in an ontology that people understand and can be handled in a logical and consistent manner, mapping and maintaining the mapping between the concepts and real world items in order to personalize the technology is not so straightforward. This requires exposing the user to the technology either directly or indirectly and can involve quite an effort and may not even be a tractable problem.

This mapping between real world items and ontologies is further compounded by the fact that ontologies do not handle, *uncertainty*. Ontologies rely on logic which assumes all information required to make a logical decision is available and produces either true, false or undeterminable statements. Uncertainty on the other hand produces similar statements but with degrees of truth or falseness. When applied to context awareness, the case where uncertainty is not present is very much an exception.

The size and scale of an ontology capable of handling context awareness is another problem. Ontologies are expandable and new concepts can be added, for example by a user. Each time a new concept is added to an ontology a consistency check should be made, as well as updating all interpretations of concepts in the ontology to take into consideration the new concept.

3.2. Context Awareness and the SCM

The SCM algorithm [7] is an unsupervised, symbol string, clustering algorithm where each of the learned clusters is interpreted as a recognized context. The symbol string is used to describe the state of a set of context information sources that combined together describe the user’s context at a given instant of time. The result is learned contexts personal to the user with no user intervention or understanding required. Unsupervised learning also makes the implicit assumption that the data being handled is noisy and can tolerate any missing pieces of information.

One difference between the learning and ontology approach is once the ontology is defined then it can be available immediately whereas in the learning approach each context has to be experienced at least once before being recognized again. It can also happen that a learned context can be quite general and over time it is split into more specific less general contexts based on the user’s behavior. So once a context is learned it can be later modified as well.

The main perceived drawback of context awareness by unsupervised learning in the SCM is the absence of labelling of the learned contexts in a standard manner. In the next section the problem of labelling contexts is addressed based on a quasi-supervised learning approach with the user an unwitting supervisor.

4. QUASI-SUPERVISED LABELLING OF CONTEXTS

One of the major differences between context awareness through ontologies and unsupervised learning as already discussed is the ability to label the contexts. In what follows a supervised learning algorithm is described that can be used to label user contexts with user needs in those contexts.

In [11] a learning approach is used to associate the requirements of a user in a given context learned by the SCM. In this case if a user interacts with a mobile device, for example, starts an application or accesses a WEB page the context is learned through the SCM and a second learning mechanism (i.e. not the SCM) associates the application, WEB page, or communication type with that learned context. For each context there is then a list of applications, WEB pages and communication types associated with that context which label it. Of course the la-

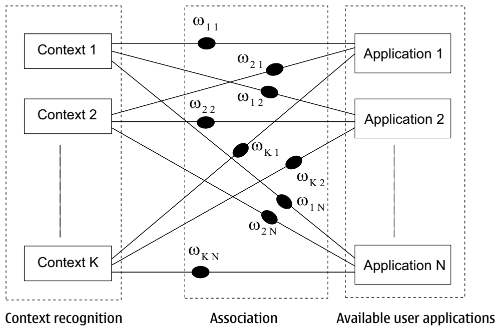

Figure 3. Defining the associations between recognized contexts and application use by the user.

bel is personal to the user and comes without any explicit labelling by the user. Figure 3 shows a low-level implementation of associating different applications with recognized contexts, using the association weights $\omega_{ij} \in [0,1]$. In fact as a result of the learning process the weights ω_{ij} represent a probability that the user uses an application in a given context. It is assumed there are a set of K recognizable contexts. At any given time the recognition algorithm recognizes the user is in context j, if the user is in context j for a continuous period of time $[0,T]$ and the user uses applications $a_1, a_2, \ldots, a_l$ in context j then there is a weight $\omega_{ja_i} \in [0,1]$ associated with each of these applications which at the time T are updated as,

$$\omega_{ja_i} = \omega_{ja_i} + \alpha\,(1.0 - \omega_{ja_i}) \;\; i = 1,\ldots,l \quad (1)$$

where $0 < \alpha \ll 1$ is an adaptation parameter, assumed constant here. Also assume there are a set of weights $\omega_{jb_1}, \omega_{jb_2}, \ldots, \omega_{jb_m}$ for applications $b_1, b_2, \ldots, b_m$ which are not used by the user in the period $[0,T]$, then these weights are updated as,

$$\omega_{jb_i} = \omega_{jb_i} + \alpha\,(0.0 - \omega_{jb_i}) \;\; i = 1,\ldots,m. \quad (2)$$

With this adaptation it is very simple to show that $\omega_{jk} \rightarrow p_{jk}$, where p_{jk} is the probability that the user uses application k when in the recognized context j.

5. CONCLUSION

Context awareness is possible and may even be necessary for mobile devices, however, if it is to work it should be possible to personalize it to the user. The effort to perform this personalization should be significantly less than the benefits to the user from the context awareness. Context awareness based on ontologies is a dominant approach, however ontologies are based on high level definitions of concepts and knowledge about their interaction. While different concepts may be understandable to people, defining these same contexts for each user is a more complex problem. On the other hand context awareness through unsupervised learning processes the information available in the user's world and builds up a representation of the user's context. Labelling of the user's contexts then becomes a problem. However, it is argued that user contexts do not need to be labelled in the same manner as in the ontology approach, but what is more significant is knowing the user requirements in a given context. The learned user requirements act as labels for the contexts. This provides a very simple means of avoiding user interaction with any form of context awareness functionality while at the same time allowing the user to benefit from its implementation.

6. REFERENCES

[1] A.K. Dey, "Understanding and using context," *Personal and Ubiquitous Computing*, vol. 5, no. 1, pp. 4–7, 2001.

[2] Harry Chen, Tim Finin, and Anupam Joshi, "An ontology for context-aware pervasive computing environments," *Knowledge Engineering Review*, vol. 18, no. 3, pp. 197–207, May 2004, Special Issue on Ontologies for Distributed Systems.

[3] Xiaohang Wang, Da Qing Zhang, Tao Gu, and Hung Keng Pung, "Ontology-based context modeling and reasoning using OWL," in *Workshop on Context Modeling and Reasoning at IEEE International Conference on Pervasive Computing and Communication (PerCom'04)*, Orlando, Florida, US, March 2004.

[4] D. Khushraj, O. Lassila, and T. Finin, "sTuples: semantic tuple spaces," in *Mobile and Ubiquitous Systems: Networking and Services*, 2004, pp. 268–277.

[5] J. A. Flanagan, J. Himberg, and J. Mäntyjärvi, "Unsupervised clustering of symbol strings and context recognition," in *IEEE Intl' Conf. on Data Mining (ICDM02)*, Maebashi City, Japan, 2002, pp. 171–178.

[6] J. Himberg, J. A. Flanagan, and J. Mäntyjärvi, "Towards context awareness using symbol clustering map," in *WSOM 03 (Intelligent systems and innovational computing)*, Kitakyushu, Japan, 2003, pp. 249–254.

[7] John A. Flanagan, "Unsupervised clustering of symbol strings," in *Intl' Joint Conference on Neural Networks, IJCNN'03*, Portland Oregon, USA, 2003, pp. 3250–3255.

[8] S. Russell and P. Norvig, *Artificial Intelligence, A Modern Approach*, Prentice-Hall, New Jersey, 1995.

[9] S. Staab, *Handbook on Ontologies*, Springer, 2004.

[10] J.A. Flanagan, "Unsupervised clustering of context data and learning user requirements for a mobile device," Accepted to Context-05, 2005.

[11] J.A. Flanagan, J. Himberg, and J. Mäntyjärvi, "Context and facilitating user interaction in mobile devices," in *Artificial Intelligence in Mobile System (AIMS 2003)*, Seattle, USA, 2003, pp. 66–73, (in conjunction with Ubicomp 2003).