

Reaching Activities by Places in the Context-Aware Environments Using Software Sensors

Kamran Taj Pathan¹, Stephan Reiff-Marganiec¹, Asad Ali Shaikh², Nizamuddin Channa¹

¹Department of Computer Science
University of Leicester
Leicester, LE1 7RH, United Kingdom
ktp2, srm13, nc180 {@le.ac.uk}

²Institute of Information & Communication Technology
University of Sindh, Jamshoro
Sindh, 76080, Pakistan
dr.asadshaikh@yahoo.com

ABSTRACT

Context-aware systems sense the user's situation and provide an interaction that adapts and behave accordingly. To reach a level of user's satisfaction the systems need to be aware of activity context (i.e. what user is doing?). To sense the activity context the popular methods are hardware sensors for example tracking devices etc. or recordings of the behaviour which occurred in the past. However, as we use web services in our day to day life to get our tasks done, for example navigation, calendar, email, profile etc., stimulates the research to sense the user's information using software sensors which can also be most relevant and circumstantial. The challenge is not only to sense the most relevant data from exchanges of messages between a user and service providers using software sensors but also to provide this data a suitable model that helps to process and store that data efficiently in both generic and more specific form. To achieve this an architecture is also needed to gather context information from software sensors. We have proposed a model that generalizes the concept according to the literature and which covers the generality in a sense which can further be lowered down to the specific domain. This paper focuses on our work regarding a novel generic semantic model for a service-based context-aware computing with emphasis on places' information that helps systems to detect the activity of the user. It provides general taxonomy and further extends to properties and relations specific to our context-aware scenario. We have also provided an architecture that utilizes software sensors to capture user's context, in which this model is embedded. In the final an application to a case study rounds the paper off.

Keywords: *Context-aware systems, Software Sensors, Context Modelling, Place Ontology*

1. INTRODUCTION

Pervasive computing is a discipline where users interact with environments full of devices. It provides applications to aid users in conducting their daily tasks wherever they go. Context-aware systems are one of the research areas in ubiquitous computing concerned with applications and services that adapt automatically to the environments in which they operate. "Context is any information that can be used to characterize the situation of entities (i.e. whether a person, place or object) that are considered relevant to the interaction between a user and an application, including the user and application themselves" [2]. Typical examples of the use of context-aware systems can range from large systems such as home care systems for people with initial-stage cognitive deficiencies or the elderly to small mobile systems that might alert a user of something in his interest being close by. Typically these systems make use of hardware sensors to identify changes in the environment. To develop the

context-aware systems is time consuming and a complex task for developers [9][11] with one core issue being the representation of context. If a generic model of context can be found, then it is beneficial for systems or services which are interacting with each other and the end users, who ultimately tries to accomplish her daily tasks. The use of standard representation languages will make sharing, reuse and extension of context models and context data easy. In addition, a more formal context model provides the ability to infer new facts from gathered context information. Context modelling has been an active research area for a few years. The first models had very close links to location specific applications such as Xerox ParcTab [23] or Active Badges [26]: The context models would allow to store the location of a person and possibly who else was in the same location. Later work focuses on attempting to create more generic models, using XML (for example the Context Specification Language [15], while others make use of Entity-Relationship models [27] and UML diagrams [14]. A different view is to model context is proposed as Ontologies, which have been first proposed

in [20]. These models allow for combining and normalizing the knowledge from different areas, thus being better positioned for reuse. The CoBrA system [5] provides concepts that are based on Ontologies to characterize entities (i.e. persons or places) within their context. Most of the existing work focuses on describing people (the who) and their locations (the where); it also considers time (the when). However, it usually does not consider the activity (that is the what) or the motivation (the why). In this paper we present a generic model for context. The model represents context that provides concepts, instances and relations between them. This provides lower Ontologies to extend for their required domains. The model is intended to allow us to store data at a higher level of abstraction than that gained from context sensors, in some sense one could refer to this as situation. Additionally, much of the required data in our work is not derived from hardware sensors (although they would be complimentary), but from what we call software sensors. We present a framework that at its core depends on the context model, but shows how the same interacts with the acquisition of context data and its use. Section 2 presents a service-based architecture for the system. Section 3 outlines the design consideration of context-aware systems. In Section 4 we describe our semantic based context model which further extends to a lower ontology by taking care of place ontology for activity-aware meeting scenario. Section 5 looks at the related work and Section 6 concludes and provides future work.

2. AN ARCHITECTURE FOR THE SCENARIO

While there is a strong need to develop generic context models, it is also paramount to understand the context in which they are intended to be used. This becomes even more crucial when considering new input sources and different levels of abstraction of the context data. In general we should have an architecture that helps to make the development of context-aware applications easier [4][7]. We present a service-based architecture, which presents the developer with a context platform that can be used inside other applications. Figure 1 depicts the infrastructure of the Activity Context in this context information is acquired from services, these include web services and applications. With the help of those data, software can infer about the user's current activity. Context Providers are the services which provide the user information such as profile, calendar, timetable etc. Profile Service provides the entity information which can help to collect and aggregate information according to user's own record. Online calendar service shows the scheduled activity of the user. Email service can sometimes be used to infer scheduled or deduced activities of the sender with

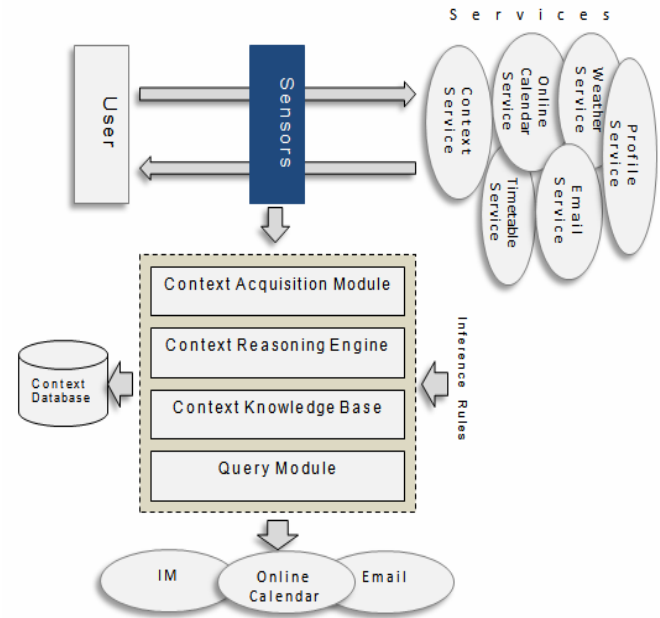


Figure 1: Activity context architecture for context-aware systems using software sensors

the help of looking at sender, receiver and subject of the message only. Timetable service shows the scheduled activity of the user. If by any means he has not updated his calendar we can get benefited from this service to look at his activity and Weather service can strengthen the deduced activity should the rules are applied on the collected context. In this architecture we are showing that whenever any user is using services for their day to day tasks, actually sending the data in the form of SOAP messages which are received and sent from the services provider which carry a useful data if interpreted right. The sensors here are capturing that data to further carried out by mapping module and Reasoning Engine to store triples in the knowledge base. Suppose a student want to know about his teacher's context whose class is about to begin in a particular time. The query module takes care of the queries being sent to the system. In the next section we will discuss about the design considerations of the context model.

3. DESIGN CONSIDERATIONS

The most researched area in context-aware applications is Location. Context-aware applications make use of Location Services to exploit the information of user's location. But Context is not the location only; rather it is much more than that. In our work we have used literature to exploit four semantic types which reflect the

so called W4H concept: who, where, when and what, which are according to context entity, place, time and activity respectively and we have left the device (how) bit as a subclass of object, which is a sub class of entity.

- who and how (Entity)
- where (Place)
- when (Time) and
- what (Activity)

The *who* component of context-aware system deals with entity which performs some action and can be a person, or an object (machine or another service etc.). The social connection between actors must also be considered e.g. a student behaviour in the class room.

The *where* component describes not only physical location of a user in a pervasive environment but also the resources surrounding him that can limit the activities. The place also holds some information (e.g. a room with 20 chairs and 10 computers along with a projector) limits the activities one can perform in that particular place with such resources hence affect the overall situation of the user. Location is not static but dynamic and can also be the different locations. If we know where the interaction takes place we can design the system architecture. Dynamic location of a person or a device is a key issue in the design.

The *when* addresses the time of that interaction in systems or services. The simple approach is registering everything or relevant events indexed according to the time they occurred hence an important one to the event notification services.

The *what* describes an activity performed in a pervasive environment. It is very hard to obtain the information regarding the activity of the entity. The systems exploited the hardware sensors to solve the problem but our model can exploit software sensors to solve the same problem.

Context-aware computing needs information to be exchanged between no. of entities, those might be user or services, and the context model should support that interoperability which is semantic as well. Taking an example of the scenario user's activity can easily be understood from his profile, calendar, timetable and email services. According to the definition given by Dey and Abowd [6] context has a great variety.

Out of that variety we are more concerned about activity context here, which can also affect the other factors of context. Context information is interrelated as per the above definition for example in our scenario if User has updated his calendar for a trip but because of

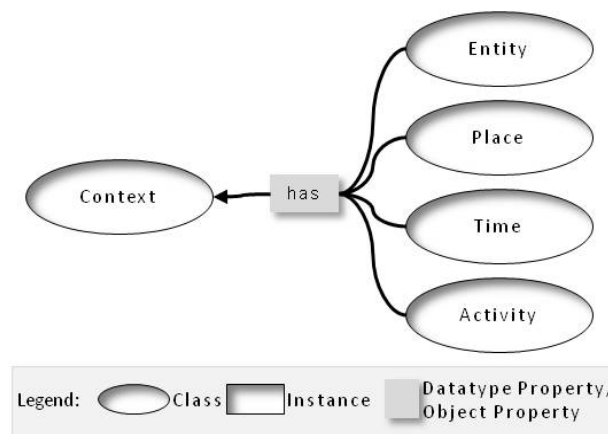


Figure 2: Upper Ontology of the Context Model

weather condition he is unable to fly then we will have to consider this factor also. In this discussion we have come to know that ontology based modelling is the approach which uses proper knowledge management, avoids inconsistency, and applies reasoning rules. The beauty of this approach is that in the future context sources become reusable and extendable. For software sensors to work without any conflict these features play a vital role.

4. CONTEXT MODEL

Once the data regarding the user's context is acquired then the need is a well designed model which can not only able to process the data up to the requirements, but also can provide acceptability at the larger audience in the context-aware systems. We portray here existing context models used to represent, store and exchange context data. Strang and Linhoff-Popien [24] provide a survey of models with respect to software sensors in which they also introduce a classification of models based on the used data structures. Key-value pairs were used by Schilit et al. [22] to model the context. The basis for Markup Scheme Models are hierarchical data structure which consists markup tags with content and attributes. They are usually based upon XML type languages such as RDF/S and have the advantage of easy tool access, but lack of formality and expressiveness. Unified Modelling Language (UML) is a language to model the context using UML diagrams. Bauer [3] and Henriksen et. al. [12][13] modelled in air traffic management and a context model which is graphics oriented. Like any object oriented approach, the object oriented models offer key features i.e. reusability and encapsulation. Like object oriented paradigms object level details are encapsulated (hidden) to other components. A first logic based context modelling approach has been published by McCarthy et al. [17], who

termed context as a help in artificial intelligence. Ontology is a mechanism to specify concepts and their interrelations [8]. Context models based on Ontologies have been first proposed by Otzturk and Aamodt [19].

They have first proposed the concept by analyzing psychological studies in combination with contextual information. They normalized and combined the knowledge from various domains. They proposed an ontological model because of they know formalization well. The knowledge is evaluated using a reasoner. Another context modelling approach based on Ontologies is the CoBrA system [5]. These systems provide runtime support by using broker-centric agent architecture. e.g. intelligent meeting rooms.

Early models mainly modelled for specific domain or even if it is in general area then it is lacking in the sensors. Generic models attract the interest and various applications can take advantage from these and can share knowledge across different platforms. And on top of it model-oriented approach supports extendibility and reusability. Context reasoning is usually based on semantic web technologies.

Context has also been modeled by Gu et. al. [10], who has modelled context based on an ontology-oriented approach but this lacks the upper ontology by not making it more general according to software sensors, which may affect context reasoning in the domain specific Ontologies. In this section we have extended their work and present our ontology-based model for context that mentions these limitations which occur when we deal with software sensors kind of systems. We have emphasized more on activity context because to describe a rich source of information for advanced adaptation in collaboration like inContext [25], activity context plays a major role. We have chosen Web Ontology Language (OWL) because it fulfils all the requirements we need on systems like expressiveness compared to other non-ontology based languages, it has the capability to be distributed across different platforms, scalability, compatibility, accessibility, openness and extensibility, automated reasoning and of all a W3C [1] recommendation. Also being a Web language it

is an obvious choice to be used in connection with services.

These Ontologies are divided into lower Ontologies and upper Ontologies. Upper here means more general ontology that models general context knowledge about the real world along with keeping an eye on software sensors in context-aware computing environments. The low-level Ontologies model the concepts their properties and restrictions in the sub-domain. With the change of an environment the domain-specific ontology like the one we are using for our scenario can exploit the upper ontology as per their needs. For example, when a user is sending email then it switches to an email service. To make the upper ontology more generic, we have classified the context into four main types, the Entity, Place, Time and Activity as in Figure 2. Upper Ontology of the Context Model provides an entrance for declaring the ontology. .

4.1 Entity Ontology

The main ontology from other Ontologies drive to is Entity ontology. When we talk about user's context then it is the focal point to model the profile of the entities in a context-aware systems environment e.g. Actor and Object. We have modelled this ontology by keeping an eye on software sensors in which services are hierarchically represented as Object<-Computer<-Services. Service selection has to select services to help a user complete a specific activity. This selection is influenced by the service context and the user context. User context is usually gathered by use of hardware sensors and possibly by mining of past behaviour records. However, as we more and more use communicating software, especially services, they can provide a lot of information on the user activities. We have placed computer or non-computer category in Object because we are dealing here with software sensors and due to that if some kind of computerized devices or non-computerized material are available at the work space then by applying rules on the aggregated data e.g. Person's profile and the nature of the objects, activity can be inferred that what user would be doing at that particular place.

<http://www.cisjournal.org>

We have categorized here entity as different roles performed by a person, organizational profile, different groups, projects and person's profile. In the object category further to computerized and non computerized objects that reflect the very nature of software sensors which covers everything from services data to inventory listing that can possibly affect the circumstances. For example, a building X is used as a class only, the room 3 of that building is the lab for Postgraduate students only. The Object describes computerized and non-computerized devices, which include meta-data that model the features of different devices and resources regarding different platforms like: Services, Applications, Devices, Network nearby thing available, and Agents. These platforms model the data about the devices e.g. what devices are available, what is the capabilities of I/O devices. So that inference mechanism can take place that if the user requirements are to conduct a class of 50 students and to demonstrate for that particular time the colour monitor and overhead projector are needed; whether that particular room having these descriptions fulfils the user demand and so the activity of user is conducting a class in that room.

4.2 Place Ontology

This is very important ontology if we are dealing with software sensors; we believe that availability of the user at different places affects the activities they perform. All those consideration are according to resources available at that place. For example if a user is in his own room does not states that he cannot teach, if all the resources are available in his room that can convey his message to the wider audiences can deduce that he can teach. Places may have the names, category (e.g. restaurant), location (i.e. address, coordinates), and resources (computers, tables, chairs and TV etc.) etc that defines the meta-data about that Location therefore the upper class of location as in Figure 3. Going further down location ontology describes the geographic coordinates (e.g. altitude, longitude, latitude) of the real world entities.

The user's activity become more effective by the place he is in. This does not only deal with the coordinates but also usual information like street, city, building, room etc. Locations are further divided into indoor and outdoor. Outdoor location can best located through the use of GPS systems but while talking about indoor locations then along with the other methods tracking devices (i.e. hardware sensors) are the best approach so far. But this research tries to prove that those costly and time consuming methods may be lessened to go through the other options by looking at not only the location but also the activity of the user.

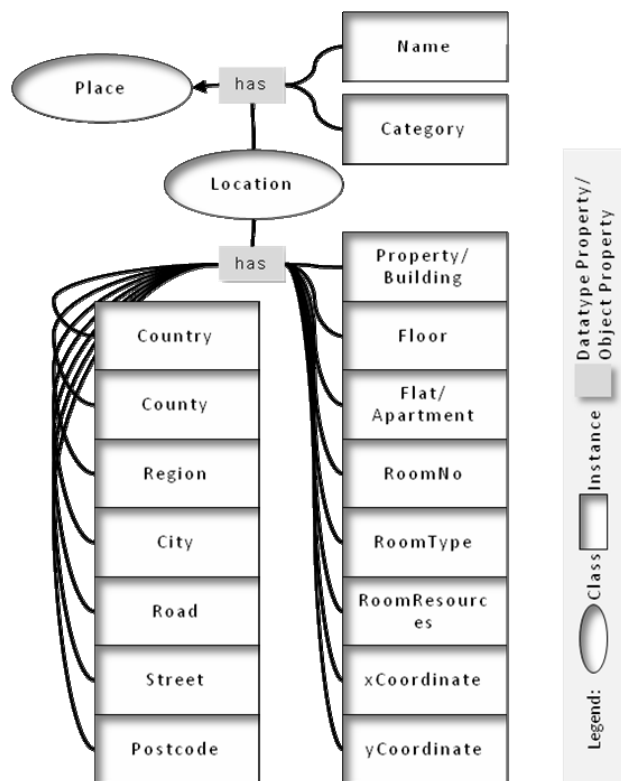


Figure 3: Place Ontology

To know the exact location of a user, systems get help from hardware sensors, but as we are dealing here with software sensors, we can infer the activity of the person by applying rules on his current profile, places of interest, frequent visits etc.

4.3 Time Ontology

For every event to occur there is a starting time and ending time no matter it takes this to happen even in the fraction of time in the real world. This may include time interval of activities (e.g. conducting a surgery class for 01 hour(s), 23 minute(s) and 17 second(s)) which is nearest to exact or might include dates (e.g. Monday, May 03, 2011).

Interval of events and the related information is covered by Time ontology. A temporal instant is any one of the points on the universal timeline, enclosure of distinct and convex time instants makes a temporal interval [18]. While modelling some properties of event intervals and temporal relations between intervals and dealing with services we believe that time duration of activities are as important as the reaching time at that particular station. Looking at the available Ontologies of Time we have

modelled Time ontology according to our requirements to represent time in the occurrence of events in the context of the user.

4.4 Activity Ontology

The Activity ontology describes ‘What the user is doing?’ of the model, the actions that a user do which describes situation of the user. Activities are of two types: Scheduled and Deduced. The former one represents the activities which are planned according to date, time and location (e.g. project meeting), whereas the later represents the activities that occurs in an informal manner and can be inferred by combining scheduled along with the complete context and applying rules on the available data.

Table 1: An example of some of the properties of the lower Ontologies for activity-aware meeting scenario

Property	Domain	Range
isSupervisorOf	Researcher	PostGraduateStudent
isDirectorOf	Administrator	MastersProgram
startTime	Event	Time
endTime	Event	Time
isTeaching	Teacher	Class
isA	Faculty	Researcher
isA	Faculty	Administrator
isA	Faculty	Teacher

isAttending	Student	Class
hasPlace	Faculty	Place
hasLocation	Place	Location
hasCategory	Place	Category
hasResources	Place	Resources

4.5 A lower ontology for activity-aware scenario

To demonstrate our scenario in conjunction with the model, lower Ontologies are presented for the activity-aware scenario in the university domain. As we have already discussed that we are concentrating to make use of software sensors hence modelled our Ontologies in such a way that can exploit software sensors. To give an idea the nodes of the concept shown in Figure 4 represent the subclass relation between concept of each Ontology; we have also presented some of the properties and relations between them in Table 1.

These taxonomies represent the extended model for the case study and are extended according to their properties only. This model classifies two disjoint activities a) ScheduledActivity and b) DeducedActivity. Considering the activity-aware university domain, Scheduled Activities are the activities which are updated either by user his self by maintaining his calendar or by organization which publishes timetable. On the other hand Deduced Activities in the scenario include: activities which are inferred by looking at the available data and applying inference mechanism onto it.

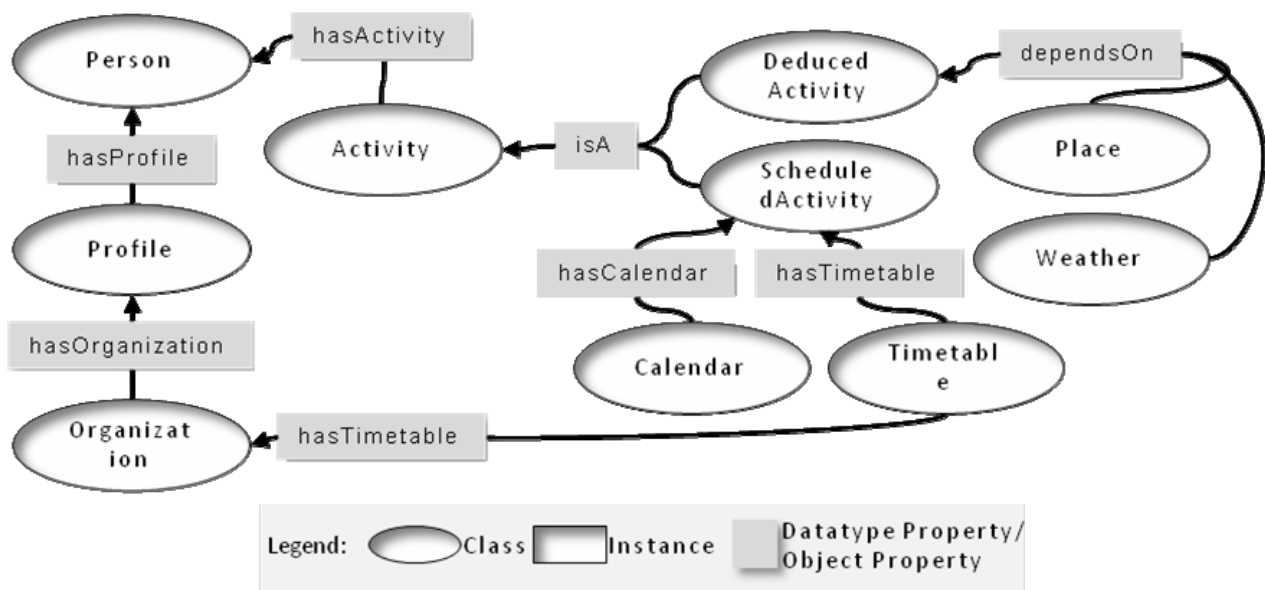


Figure 4: Lower Ontology for Context-aware Scenario

By considering the activity-aware meeting scenario in a university domain some of the common scheduled activities are *ConductingClass*, *LabSession* and *SurgeryClass* which are *subClassOf Teaching*; *UGResearch*, *PGResearch* and *ResearchProject* which are *subClassOf Research*.

The same can also apply to deduced activities through inference mechanism. Activities performed in a class room as *subClassOf ConductingClass*, activities performed in a surgery as *subClassOf SurgeryClass* depends upon the role of the person who is either a teacher or a student. Activities performed in the Indoor Location by the persons in campus e.g. *ClassRoom*, *LabRoom*, *LectureRoom*, *StudyRoom* etc. The places can describe what resources are placed in that location which can affect the activity of the person. e.g. *Room02* of such building can only be used as a Computer Lab not as a group discussion or class. For example in our scenario, scheduled activity context Faculty's current activity can be inferred from Faculty's calendar or Organization's timetable, although deduced activity context can be inferred from personal information, organization, timetable, place, weather, object, time and location.

Taking that, if a Postgraduate Students's place is a University's specific building at a specific date and specific time infers that his activity is "*Attending a Postgraduate Class*" hence a deduced activity. Later one presents the scheduled activity like If according to Faculty's Organization Timetable and the same has also updated his Calendar to teach some classes then his activity is *Teaching*.

The next section talks about the Semantic Rules which help to infer the knowledge out of the existing facts.

4.6 Reasoning Rules

Applying rules on existing facts to derive a new fact is called inference mechanism. Rules help us to reach at the results. To infer an activity context, scheduled activities can be inferred by defined activities however, deduced activities can be inferred from individually or combining two or more than two context types. For example scheduled activity context i.e. teacher's current activity can be inferred from Faculty's work calendar or Organization's timetable, although deduced activity context can be inferred either from personal information, personal calendar, organization timetable, place, object and time individually or combining these all otherwise.

Suppose, we have instances in Ontologies, onto which we are then going to apply reasoning rules into the existing facts to derive new facts. Some of the existing

facts lie in our Context Ontology which can be used as a generic data model as far as upper Ontologies are concerned and can be used as specific data model for the scenario we are discussing in this paper. However, it fulfills the general requirements of the context itself but considering knowledge has no boundaries and any of the term can be added afterwards). These facts (derived) then will be an input to update the Context Database.

Some of the rules are mentioned here:

1. $Place(f, p) \wedge Time(t) \Rightarrow teachingClass(f, c)$

Where f = Faculty, p = Place Name/Name of a Building, t = Time Zone and c = Postgraduate Class

And

2. $Timetable(f, o) \wedge Weather(l) \Rightarrow teachinClass(af, c)$

Where f = Faculty, o = Organization, l = Location, af = Alternative Faculty and c = Postgraduate Class

Taking an example of Rule 2, it states that according to Organization Calendar the Faculty is liable to teach the postgraduate class and the exception will only be accepted if the alternative item is of the same or more importance than the agreed item. Here for example duties of teaching a postgraduate class can only be transferred to an alternative faculty if the weather is worst and the actual faculty will not be able to turn up and conduct that class.

First one is the example of scheduled activities and the last one is the example to inferred or deduced activities.

5. RELATED WORK

In the field of pervasive computing one of the challenges is to make applications context-aware. A key issue for this is sensing activity so that a system adapts its behavior according to the situation. A high-level context model might be the answer to this challenge. This section presents various context models from the literature. Models differ according to the domain information, concepts, and modeling techniques. Initially, models were built for specific domain [23] [26]. As these models were built for specific applications which make them difficult to share and reuse, and mainly use hardware sensors to acquire location information. However, our model is based on semantic web technologies that can be used in various domains and further can be extended, reused and shared. The model we propose represents the context in its entirety

<http://www.cisjournal.org>

by exploiting Entity, Place, Time and Activity. Cooltown model [16] lacks in the expressiveness and formality for context reasoning. Our model which is based on ontology supports context reasoning. As an evolution context modeling includes XML-based representations of context acquired from hardware sensors [6]. Despite of the information and XML syntax, these models lacks in the formality and expressiveness. Also, the use of Ontologies provide us with all the features along with the standards, by allowing the exchange of information between various systems. A sensor fusion system represents Entity-Relationship models of the context [27]. Combining E-R and UML techniques, it models context as physical, history and social relationship [14]. Models were proposed which represents five context dimensions in a similar way proposed by Neto and Pimentel [18] but the difference is that the later is based on Ontologies. Both approaches provide context modeling including temporal information but the former do not allow inference mechanism and the later deals mainly with physical sensors. Artificial intelligence based methods have also been used to model the context. Ranganathan et al. [21] model context as First Order Predicates. To validate the predicates the ontology is used. In comparison to this our model provides inference over the context and uses RDF triple model between context sources and the user. The COBRA ontology provides a set of standard Ontologies for ubiquitous computing applications [4]. Neto and Pimentel [18] also use OWL to model context and the context dimensions are also same. In comparison to that we propose supporting Ontologies in a software sensors environment to ease system designers to model according to their own requirements. The CONON ontology is also an OWL-based ontology [9]. It addresses context classification and dependency. We have also covered temporal information like [18] and meta-data for modeling profiles of entity.

6. CONCLUSION & FUTURE WORK

In this paper, we have proposed that using Places information in a generic semantic model for software sensors in a service based context-aware systems, the activities can be sensed. This model is built by means of: determining the W4H design space for context; place, entity, activity and time, and taking concepts from existing vocabularies to avoid the process of inference over context. We have been developing the service based architecture that senses the context of the user using software sensors and acquire the exchange of messages between user and services and based on it maps and adds semantics to context data acquired from software sensors on our semantic model. In future, we will build a prototype of activity-aware context monitoring for service based context-aware systems.

ACKNOWLEDGMENTS

The authors^{1,4} thank University of Sindh Jamshoro and HEC for scholarship.

REFERENCES

1. Dean, M., & Schreiber, G. (eds): OWL Web Ontology Language Reference. W3C (2004). Available at <http://www.w3.org/TR/2004/REC-owl-ref-20040210/>, last accessed 14 Oct 2011.
2. Abowd, G., Dey, A., Brown, P. et al.: Towards a Better Understanding of Context and Context-Awareness. (1999) 304-307
3. Bauer, J., Kutsche, R. D., Ehrmanntraut, R.: Identification and Modeling of Contexts for Different Information Scenarios in Air Traffic. Technische Universität Berlin, März, (2003)
4. Chen, H., Finin, T., Joshi, A. et al.: Intelligent Agents Meet the Semantic Web in Smart Spaces. IEEE Internet Comput., (2004) 69-79
5. Chen, H., Finin, T., Joshi, A. et al.: Using OWL in a pervasive computing broker. Defense Technical Information Center (2005)
6. Dey, A. K.: Understanding and using Context. Personal and ubiquitous computing, **5** (2001) 4-7
7. Goularte, R., dos Santos Moreira, E., da Graça, C. P.: Structuring Interactive TV Documents. (2003) 51
8. Gruber, T. R.: A Translation Approach to Portable Ontology Specifications. KNOWLEDGE ACQUISITION, **5** (1993) 199-199
9. Gu, T., & Pung, H. K.: Toward an OSGi-Based Infrastructure for Context-Aware Applications. IEEE Pervasive Computing, (2004) 66-74
10. Gu, T., Wang, X. H., Pung, H. K. et al.: An Ontology-Based Context Model in Intelligent Environments. **2004** (2004)
11. Helal, S.: Programming Pervasive Spaces. IEEE Pervasive Computing, (2005) 84-87
12. Henricksen, K., Indulska, J., Rakotonirainy, A.: Generating Context Management Infrastructure from Context Models. (2003) 1-6
13. Henricksen, K., Indulska, J., Rakotonirainy, A.: Modeling Context Information in Pervasive Computing Systems. LECTURE NOTES IN COMPUTER SCIENCE, (2002) 167-180



<http://www.cisjournal.org>

14. Henricksen, K., Indulska, J., Rakotonirainy, A.: Modeling Context Information in Pervasive Computing Systems. Lecture notes in computer science, (2002) 167-180
15. Hong, J. I., & Landay, J. A.: An Infrastructure Approach to Context-Aware Computing. Hum. - Comput. Interact., **16** (2001) 287-303
16. Kindberg, T., Barton, J., Morgan, J. et al.: People, Places, Things: Web Presence for the Real World. Mobile Networks and Applications, **7** (2002) 365-376
17. McCarthy, J.: Notes on Formalizing Context. **13** (1993) 555-555
18. Neto, R. F. B., & Pimentel, M. G. C.: Toward a Domain-Independent Semantic Model for Context-Aware Computing. (2005) 61
19. Öztürk, P., & Aamodt, A.: Towards a Model of Context for Case-Based Diagnostic Problem Solving. 198-208
20. Öztürk, P., & Aamodt, A.: Towards a Model of Context for Case-Based Diagnostic Problem Solving. (1997) 198-208
21. Ranganathan, A., & Campbell, R. H.: An Infrastructure for Context-Awareness Based on First Order Logic. Personal and Ubiquitous Computing, **7** (2003) 353-364
22. Schilit, B., Adams, N., Want, R.: Context-Aware Computing Applications. (1994) 85-90
23. Schilit, B. N., Adams, N., Gold, R. et al.: The PARCTAB Mobile Computing System. (1993) 2
24. Strang, T., & Linnhoff-Popien, C.: A Context Modeling Survey. (2004)
25. Truong, H. L., Dustdar, S., Baggio, D. et al.: InContext: A Pervasive and Collaborative Working Environment for Emerging Team Forms. (2008)
26. Want, R., Hopper, A., Falcao, V. et al.: The Active Badge Location System. ACM Transactions on Information Systems (TOIS), **10** (1992) 102
27. Wu, H., Siegel, M., Ablay, S.: Sensor Fusion for Context-Understanding. **1** (2002) 13-18