# Construction of a Contextually-Aware Pervasive Computing Environment

Ian Millard, David DeRoure, Nigel Shadbolt

School of Electronics and Computer Science
University of Southampton,
Southampton, SO17 1BJ, UK
{icm02r,dder,nrs}@ecs.soton.ac.uk

**Abstract.** This paper outlines work in progress related to the construction of a system which is able to deliver information in a contextually sensitive manner within a pervasive computing environment, through the use of semantic and knowledge technologies. Our approach involves modelling of task and domain as well as location and device. We discuss ideas and steps already taken in the development of prototype components, and outline our future work in this area.

## 1 Introduction

During the course of the last decade, our lives have become increasingly dependent on the availability and exchange of information. In today's world, knowledge can be equated to people, money, flexibility, power, and competitive advantage. Knowledge is more relevant to sustained business than capital, labour or land, yet in most corporations it remains the most neglected asset. For large businesses and organisations, being able to access the right information at the right time is key to success.

Another notable trend in recent times is the increasing level and availability of hi-tech devices within the workplace, including a wide range of portable and/or personal computing and communications devices, laptops, desktop machines, plasma screens, and projection displays. Often these devices are powerful, offer significant storage capacity, and feature wired or wireless networking technologies; yet a lack of interoperability prevents the seamless transfer of information. Instead, we are presented with the chores of having to remember where information is stored, and manually transferring data between devices – a far cry from the integrated, all-knowing solutions promised to us in visions of the future.

The contextually aware environment aims to harness the abundance of technological resources to aid workers in these situations, by presenting the right information to the right users, at the right time and in the right place. In order to achieve this, a system must have a thorough understanding of its environment, the people and devices that exist within it, their interests and capabilities, and the tasks and activities that are being undertaken. Such a system must recognise that people often work on a number of different projects, collaborate with other colleagues, and have a unique set of skills and concerns. That is to say, the system must primarily be able to identify where, and under what context each person is working.

Experience drawn from the Advanced Knowledge Technologies (AKT) interdisciplinary research collaboration gives us confidence in taking a semantically-orientated approach to these modelling tasks, using the robustness of formal knowledge representations, the descriptiveness of ontologies, and the power of inference which they offer. Given such a rich representation of a particular domain, a wealth of knowledge or agent-based services may be envisaged, utilising the data acquired to provide useful and proactive assistance to the users of that environment.

The following section gives an overview of the major requirements for such a contextually aware system, and details some of the modelling and data-acquisition issues relating to them.

## 2 Requirements

We have already indicated that the major elements in a contextually aware environment involve modelling the location of users within that environment, their activities and interests, and the devices and resources which are available.

### 2.1 Location

Sensing the location of people and devices within a building is non-trivial. In large open outdoor spaces, technologies such as GPS and RF signal strength analysis usually allow location to be determined accurately, typically with a resolution of a few metres. However, inside a building or in built up areas, these techniques become unreliable, if not dysfunctional, hence other methodologies are required.

Several research projects have been undertaken in the area of localised high resolution location tracking, with prototype systems including triangulation of position utilising infra-red [16] ultrasonic [2][9] or radio signals [10][11], location sensitive flooring [6], and body mounted accelerometers [5]. However, deploying any of these location sensing technologies throughout a reasonable sized office or academic research environment would require a significant investment in proprietary infrastructure.

Conversely, there are many off-the-shelf systems that can be used to sense the presence of a tag, fob or card in close proximity to a specific receiver unit. These include RFID, magnetic swipe cards, smart cards and iButtons, and are often used for access control or the identification of products in industrial applications. However, such technologies typically have an active range of only a few hundred millimetres, rendering them only suitable for detecting the presence of a user at a particular door, desk, or workstation for example.

For applications and services to operate efficiently within a pervasive environment, some form of middleware is required to monitor all of these potential sources of location information, and present that data in a coherent fashion through an integrated interface. To achieve this, we have chosen to represent the location information a common model mediated by an ontology, as this is easily extendible, permits some levels of reasoning over the location data, and enables interoperability with other information models. The model [7] is not designed primarily from a spatial perspective, as it is thought that exact positioning is not critical for most pervasive applications. Rather, it is more important to represent the type and purpose of different areas, and their relative proximity. The ontology uses a hierarchical notion of locations and sub-spaces built from the principle concept of an abstract space, and properties are defined to describe the relationships between these spaces.

In addition to this hierarchical notion of inter-connected symbolic spaces, the ontology also supports coordinate or point locations, supporting both absolute systems such as GPS, and schemes using relative positions from a given known location. This arrangement permits different location sensing schemes to operate at different resolutions, without requiring knowledge of the location system in which they are situated.

By describing the physical properties of an environment in RDF using the ontology, and storing them in a repository, we will achieve a concrete model of the space in which users carry out their activities. Subsequently, the various sensors within the environment should contribute their snippets of information to this repository, forming a stateful representation of the environment. The population of the base model with location information from a variety of sensors offers a powerful means of data fusion, and combined with potential inferences offered through the ontology offers a richer picture than it is possible to construct from the sensor data alone. In addition, the combined model formed in the repository is an ideal platform for a query interface, or direct manipulation by other applications.

### 2.2 Activity and User modelling

In any contextually aware environment, there is an obvious need to collect and represent a wealth of information regarding the skills, interests, personal preferences, privacy requirements and the relationships between people in that environment. However, representing a generic notion of context would be a hugely complex and difficult task. In addition, there is some doubt to the usefulness of such a generic representation, should one be created, as it is likely that it would not afford the level of detail required by many applications. Instead, we propose a domain specific task-oriented approach, as most activities carried out by people are task-oriented.

For example, most working days can be conceptualised as a sequence of different tasks, such as a project meeting, document review, teleconference, patient consultation, or student supervision. Each of these tasks may have a variety of important properties, ranging from the date/time/location, to meaningful relationships between people and/or resources that feature in or are required for a particular task, which would be impossible to represent in a generic context ontology.

It is the authors' view that contextually aware systems can be more usefully constructed in a fashion that is tailored to a specific environment, such as that of an academic research department, hospital ward, or lawyers practice, through the application of domain specific ontologies and rules which may be applied to common infrastructure components.

To represent the activities of an individual, the concept of a schedule is perceived, containing a reference to each of the tasks or events which that person will be undertaking. For a given domain, it is likely that a high level classification may be made of the typical tasks carried out. These tasks, or events, may be identified as complying with one or more templates, implying that common properties, features and/or resources are required, in a similar fashion to implementing an interface in an object oriented programming language. These high level templates may provide the basis for a hierarchy of more specialised instances of those tasks, defined specifically for the domain in which they are applicable. They may additionally include data such as the location, persons attending, topics of interest, importance and 'interruptability' relating to a particular instance of an event or class of events.

For any given system deployment, it is likely that an additional array of information, concepts and relationships will be essential for representing domain or background knowledge. This data can only properly be represented in detail by domain specific ontologies, utilised within the more general notions of location and task. As an example, we have looked to develop a prototype within the academic domain, focusing on the activities of a senior faculty member. Often having a variety of different research interests, busy schedules which may include significant travelling, and a number of different teaching, research, and supervisory duties, the life of the academic can be hectic and diverse. Throughout a typical day, an academic may switch context on a frequent basis as they attend different meetings and work on different papers or projects, and each of these tasks are likely to address differing research interests, and require a different set of resources to be at hand. In our prototype work these properties are ontologically represented by the AKT Reference Ontology [1] and extensions thereof.

However, given these modelling capabilities, acquiring contextual information relating to people within an environment remains a non-trivial task. Many indicators offer snippets of information, and knowledge of the environment may permit other inferences to be made. For example, let us consider the determination of which activity in a schedule the user is currently undertaking. Clearly the time of day is a strong indicator of when we can anticipate an activity to have started. However schedules often slip, and we can achieve a more thorough assertion by additionally considering knowledge within the system relating to the location of individuals due to participate in an event, as we know, for example, that a meeting between two people cannot have commenced if they are not co-located (either physically or by some virtual means). Other indicators may be inferred from monitoring computing devices, such as ascertaining an idle state and/or availability of the user through observing screensavers or instant messaging clients.

### 2.3 Device representation

To facilitate the delivery of information in situ to a person's location, a contextually aware system must also be able to comprehend what computing resources are available, their capabilities, and location. To achieve this, a device ontology [4] has been created which permits a particular set of features relating to a computer interface to be described. These focus mainly on the input/output and user-interaction capabilities of the device, rather than the typical system hierarchy approach of describing processor, disk, and memory specifications, although the ontology could easily be extended (or others incorporated) if these details were thought necessary at a later date.

Properties are defined to express the visual output resources in some detail, including accepted content types and the dimensions, resolution, type of the display, available screen layouts, and current status. Given these properties, it should be possible for services wishing to present information to locate suitable display resources, taking into account the intended recipient(s) and the format/sensitivity of the data. In addition, provision is made for representing relationships between devices and their users, and devices may be permitted to update information in the repository relating to whether the current user is actively engaging with the device, what applications are running, and in the case of portable display devices, their location, if known.

## 3 Prototype

The remainder of this paper describes a prototype system, currently under development, which has the aim of delivering short messages or memos to users within an academic research environment in a contextually-sensitive manner. Messages are injected into the system by email, and picked up by a daemon process monitoring a designated inbox. After analysis, it is determined whether it is suitable in the current context for that message to be delivered to the recipient(s), and is subsequently processed or queued until it becomes suitable.

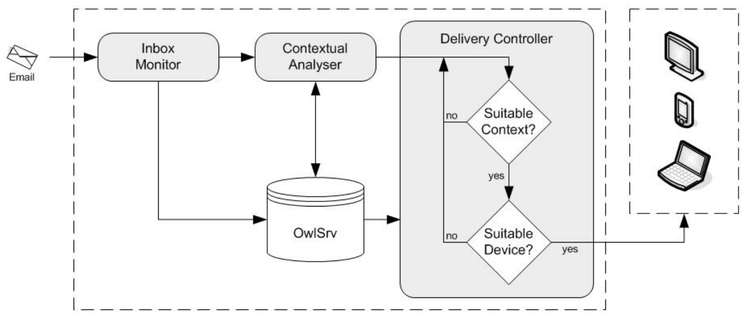


**Fig. 1.** Conceptual architecture for the prototype system

### 3.1 Repository

We have developed a combined RDF repository and OWL inference engine, 'OwlSrv', which is capable of executing custom inference rules and handling near real-time updates. This plays a central role by providing core services of data storage, query and inferential capabilities required to form the ‘intelligence hub’ of the contextually aware environment. OwlSrv consists of a Jena-based [8] dynamic repository and inference engine, operating on a number of ontologies together with the set of domain specific custom rules. Presenting a similar interface and function as 3Store [13], OwlSrv is not as efficient or as massively scalable, yet it provides the crucial additional features of OWL support, inference and dynamic updates which 3Store does not.

Testing has shown that the experimental OWL reasoner, shipped as part of the Jena toolkit, is rather inefficient when running over the complex ontologies. The main cause is thought to be the overheads introduced in satisfying the subsumption clauses inherent in OWL Full, and as these features are not essential those rules concerned with the propagation of domain/range inference, restriction, equivalence and disjointness have been removed.

Persistent data in the OwlSrv repository is stored through Jena in a MySQL database, and can be browsed through an interface hosted through the integral web server. RDQL queries are accepted through the HTTP post method, and can return results as chunks of RDF or as variable-value records. Data in the model can be updated by asserting or retracting RDF models, again through the HTTP interface, and additionally triple assertion and retraction may be performed through an interface to the Elvin distributed messaging system [3], permitting easy integration with a range of existing location sensing components and services.

### 3.2 Orchestration

The main point of data and service orchestration is that of the inference engine within OwlSrv. This has been extended from the generic Jena reasoner through the creation of a number of domain-specific predicates, which are used within the inference rules. Some of these predicates perform specific analyses on the model which are too complex to express in the rule language, and act as boolean variables in other inference rules, while others utilise the Elvin interface to alert external processes of particular events or to request the specific analysis on a given resource. It is these rules, fired as and when data in the repository matches their requirements, which action the various tasks required for the analysis and delivery of messages.

For example, when a new message is received by OwlSrv, a custom predicate is used to determine whether the necessary analysis between the sender and recipient has been performed. If not, another predicate in the inference ruleset requests an external ‘contextual analyser’ to prepare information concerning persons related to the message.

### 3.3 Analysis

In the prototype system, the contextual analyser is realised through a process which, on request, executes a number of RDQL queries read from a configuration file in order to identify specific relationships within domain repositories relating to a sender-recipient pair. The queries are arranged and weighted to calculate a number of metrics relevant to the domain, with each successful query result contributing a decimal increment to one or more of these metrics. The weighting of each query result for a given metric is also specified in the configuration file, hence realising the different levels of relevance a single query may have on different metrics. The result of any individual RDQL query is a bag of variable/value bindings, to which we have permitted the application of simple set operations. Set union can be used to (somewhat crudely) merge results from multiple repositories, and set subtraction enhances the expressiveness of our metric calculations by chaining together RDQL queries in the form of 'find people who have relationships prescribed by *query x*, but which do not match properties given by *query y'* (or even *query z,* etc).

Having performed the analysis to generate metrics classifying important relationships between the sender and recipient(s) of a message, and considering other information within the repository, the 'delivery controller' then makes an assessment by applying further domain-specific rules as to whether it is appropriate in the current context to deliver that message. These rules may consider factors such as the type of activity currently being undertaken, properties expressed directly in the message, e.g. 'personal' or 'urgent', or a manual indication from users indicating their availability or willingness to be disturbed, in conjunction with the domain metrics just calculated.

For example, in the academic domain, one factor we may look to is the relative superiority of the sender and recipient, such that when the recipient is engaged in activities deemed to be of a certain importance, only urgent interruptions from their superiors or line managers are permitted. In the case of where the recipient is collocated with other persons, for example in a meeting, we should perhaps consider the relationship between the sender and meeting participants as a whole. Again, a thorough understanding of the environment into which the contextually aware system is to be deployed is essential for the creation of these rule sets.

### 3.4 Interfaces

One demonstrator in the contextually aware environment is a client program, prototyped as an HTML based interface, which is designed to assist academics by presenting their schedule and information relevant to their current task. These resources are automatically determined based on custom inference rules running over data held in OwlSrv, and may include events of interest, papers, required documents, and links to previous occurrences of events. Intended to be run on personal laptop or desktop computers, this application also provides a strong indication of when users switch task context, returning the currently selected task to the repository.

For the display of messages within the environment, the repository will be queried to find displays which are suitably co-located with the recipient, that can handle the required content types, and which satisfy any requirements for maintaining the confidentiality of personal or private messages. The Signage Project [15] at Southampton has a core interest in the deployment of display devices throughout the environment, and in cooperation with this activity a number of wall-mounted interfaces have been developed. A daemon running on these displays, in addition to users' laptops and desktop PCs, will await instructions to deliver pop-up notifications as required. Devices that permit user interaction may be used to acknowledge the receipt of the message, whereas passive displays may show the information for a limited period of time, or for as long as the user is located in the proximity.

## 5 Future Work

The final component of the prototype system to be implemented is the decision logic involved with determining whether a particular message is suitable for delivery in the given context, and the discovery of suitable devices. This logic could possibly be implemented in the main OwlSrv inference engine through further custom extensions, or by and external service accessed on demand in a similar fashion to the contextual analyser. Currently the latter option is preferred, as this offers a more flexible solution, in better service-oriented design, but more importantly will not introduce overheads of potentially complex rules that need not be considered in the norm.

Several of the components in the prototype system use custom data formats, which does not tend towards interoperability or reuse. Emerging standards such as SPARQL [12] and SWRL [14] should be implemented in any form of production system, in addition to a more formal representation of the interaction between services, such as that offered by OWL-S.

## 6 Conclusions

In this paper we have discussed the novel application of semantic technologies to pervasive computing scenarios in order to enable the development of contextually aware environments. Our notion of context pays particular attention to modelling the user's task, and we believe this to be an essential view, to be coupled with a systems perspective on context. Our ideas and work to date on the construction of a prototype application have been presented.

## Acknowledgments

Part of this work has been supported under the Advanced Knowledge Technologies (AKT) Interdisciplinary Research Collaboration (IRC) in conjunction with the Equator IRC, which are sponsored by the UK Engineering and Physical Sciences Research Council under grant numbers GR/N15764/01 and GR/N15986/01 respectively.

## References


1 AKT Reference Ontology – http://www.aktors.org/ontology/
2 Andy Ward, Alan Jones, Andy Hopper. *A New Location Technique for the Active Office*. IEEE Personal Communications 1997, 4(5):42-47.
3 Bill Segall, David Arnold, Julian Boot, Michael Henderson and Ted Phelps. *Content Based Routing with Elvin4*, in Proceedings of AUUG2K, Canberra, Australia, June 2000.
4 Device ontology – http://signage.ecs.soton.ac.uk/ontologies/device
5 Elena Vildjiounaite, Esko-Juhani Malm, Jouni Kaartinen, Petteri Alahuhta. *Location Estimation Indoors by Means of Small Computing Power Devices, Accelerometers, Magnetic Sensors, and Map Knowledge*. In Proceedings of Pervasive Computing 2002, 211-224.
6 L. McElligott, M. Dillon, K. Leydon, B. Richardson, M. Fernstrom, J.A. Paradiso. *'ForSe fIElds' - Force Sensors for Interactive Environments*. In Proceedings of UbiComp 2002, 168-175.
7 Location ontology – http://signage.ecs.soton.ac.uk/ontologies/location
8 McBride, B., *Jena: Implementing the RDF Model and Syntax specification*. in 2nd International Semantic Web Workshop (2001).
9 N. B. Priyantha, A. Chakraborty, and H. Balakrishnan. *The Cricket location support system*. In proceedings of the Sixth International Conference on Mobile Computing and Networking (ACM MobiCom) August 2000, 32-43.
10 P. Bahl and V. N. Padmanabhan. *RADAR: an in-building RF-based user location and tracking system*. In Proceedings of IEEE Conference on Computer Communications (INFOCOM) 2000, 2:775-84.
11 Paul Castro, Patrick Chiu, Ted Kremenek, and Richard Muntz. *A probabilistic room location service for wireless networked environments*. In Proceedings of UbiComp 2001, 18-34.
12 SPARQL – *Query Language for RDF* – http://www.w3.org/TR/rdf-sparql-query/
13 Stephen Harris and Nicholas Gibbins. *3store: Efficient Bulk RDF Storage*. In Proceedings 1st International Workshop on Practical and Scalable Semantic Web Systems, Sanibel Island, Florida, USA, 2003.
14 SWRL – *A Semantic Web Rule Language combining OWL and RuleML* – http://www.daml.org/rules/proposal/
15 The Signage Project, Intelligence Agents Multimedia group, University of Southampton – http://signage.ecs.soton.ac.uk/
16 Want, R., Hopper, A., Falcão, V., Gibbons, J., *The Active Badge Location System*. ACM Transactions on Information Systems, Vol. 10, No. 1, January 1992, 91-102.