Contextually Aware Information Delivery in Pervasive Computing Environments

Ian Millard, David De Roure, and 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

The pervasive computing vision leads us to believe that future working environments may well feature a wide range of interconnected portable and/or personal devices, in conjunction with static displays populating the surroundings. While this infiltration of technology into everyday life is aimed at improving access to information, communication, and the ease of work, there is a danger that users may become inundated with data or distracted from the task at hand to such an extent that productivity begins to suffer.

The authors propose that by enabling a system to understand and reason about the activities of the occupants of such an environment, then that environment can be significantly more supportive of those working within it. Given the correct knowledge relating to both the general environment and the current situation, a contextually aware system may provide access to resources required for undertaking a given task, and offer interesting or related information, while at the same time removing unwanted or inappropriate data or distractions until such a time that they are more suitable.

In our earlier paper [11], we outlined the notion of a contextually aware environment which aims to present the right information to the right users, at the right time and in the right place. In order to achieve this, the system must clearly have a sufficient 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. That is to say, the system must be able to identify where, and under what context each person is engaged with their current task.

T. Strang and C. Linnhoff-Popien (Eds.): LoCA 2005, LNCS 3479, pp. 189 – 197, 2005.

2 Contextual Modelling

As we have discussed, central to any contextually aware system is the need to represent and collect a wealth of information such as current activities, skills, interests, personal preferences, privacy requirements and the relationships between people. However, the authors believe that attempting to create a generic notion of context would be a hugely complex and difficult task, and would probably be of little use as it is likely that it would not afford the levels of detail required by an implemented system. Instead, we put forward the case that the creation of generic systems or frameworks, which can be specialised for use within a given domain, would be more appropriate in the majority of cases.

2.1 Location Modelling

One area identified as a key component for the representation and understanding of activities within a general working environment is that of location. We must have good knowledge of where people and devices are, though not necessarily in a spatial representation as it is more important to comprehend semantic information regarding locations, such as the type and/or purpose of different areas or spaces, their relative proximity and positioning, facilities offered, or activities usually undertaken in that area. We have created a location ontology [8] which permits these kinds of relationships to be described, and can be used as a basis for combining sensed location data and inferring additional facts about the environment, as discussed in [11].

2.2 Task Modelling

Another important area is representing what people are doing at any particular time. We propose a task-oriented model, as 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.

In addition, many tasks are repetitive or recurring, such as project meetings, preparing accounts, or performing routine maintenance. Each instance of these types of event is likely to be very similar, exhibiting many of the same properties. In recognition of this, each instance of a task or event may be identified as complying with one or more hierarchical 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. 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/or 'interruptability' relating to a particular instance of an event or class of events.

2.3 Domain Modelling

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. In our prototype work, which focuses on the academic domain, these properties are ontologically represented by the AKT Reference Ontology [1] and extensions thereof.

2.4 Device Modelling

Finally, to facilitate the display of information in situ to a person's location, a contextually aware system must also be able to comprehend what display 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,

Existing work in this area, such as Composite Compatibilities/Preference Profiles [3], has generally looked at techniques for adapting content for a particular display based around the inclusion, exclusion, or modification of various components within a document, given a minimalist representation of available screen resources. While such capabilities may be useful for repurposing content across many different types of device, the content must be generated with such intentions in mind. In addition, representation of display resources is not sufficient to satisfy the requirements discussed above.

3 Acquisition of Contextual Data

In addition to having the capability of modelling the events and activities within a pervasive environment, we must be able to obtain data in near real-time to populate these models for a system to have any chance of success.

Sensing the location of people and devices within a building is non-trivial. Coordinate position technologies range from GPS, offering relatively low accuracy over a very large (outdoor) area, to small scale localised systems offering much higher resolution. 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 at a fixed-point location.

The different forms of physical location technology offer presence detection with a variety of different accuracies, reliabilities, and ranges in which those observations can be made. A location system is likely therefore to have to take into account data from a number of different information sources, potentially including both spatial (coordinate) and symbolic point information. Combining these two fundamentally different types of system is non-trivial and is an important research topic. [7]

For applications and services to operate efficiently within a pervasive environment, some form of middleware is required to monitor these potential sources of location information, and present that data through an integrated interface. To achieve this, it is proposed that the location ontology is used in conjunction with a real-time OWL inference engine, permitting location assertions to be instantiated and retracted as and when appropriate sensor data is received. The capabilities offered by the ontology and OWL engine permit data from various sources to be easily combined, and further facts inferred.

Determining contextual information relating to people within an environment is also 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 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.

4 Prototype System

A prototype system is being developed, with the aim of being capable of delivering messages to users within an academic environment in a contextually-sensitive manner.

As described in [11], we have already constructed some components which will be useful in achieving our goals. 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, providing all of the data storage, query and inferential capabilities required. OwlSrv consists of a Jena-based [10] dynamic repository and inference engine, operating on a number of OWL ontologies together with the set of custom rules, and it presents a similar interface and function as the 3Store [6] system with which we have worked previously.

Our earlier demonstrator application consists of a client program designed to assist academics by presenting their schedule and information relevant to their current task, automatically determined based on custom inference rules running over data held in OwlSrv. It offers a strong indication of when users switch task context.

Our prototype system focuses only on delivering messages to users, as opposed to additionally presenting documents and resources, as issues surrounding real-time event generation are more interesting, and our previous application covers the latter areas reasonably well.

The conceptual architecture of our prototype is shown below in Fig. 1 below.

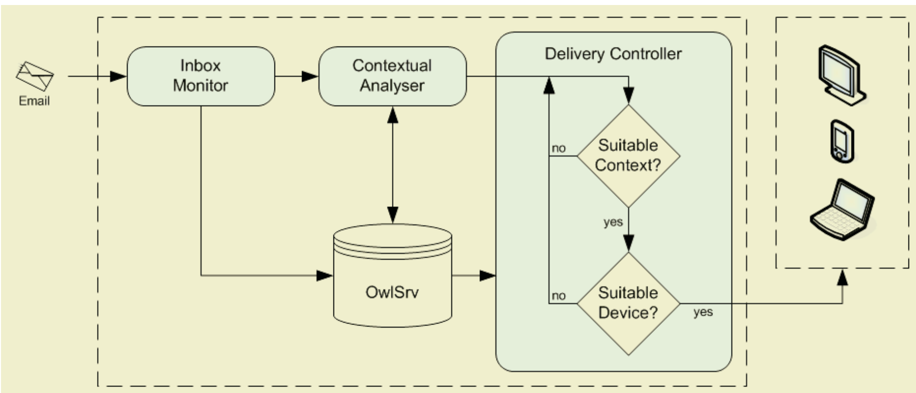

Fig. 1. Conceptual Architecture of the Contextually Aware Message Delivery System

4.1 Message Injection

To enable messages to be easily fired into the system, email has been chosen as the source message format. A daemon process monitors a specified IMAP inbox for the user, asserting RDF representations of each message received in the OwlSrv repository. Upon arrival of a new message, a 'contextual analyser' is invoked to prepare information concerning persons related to the message.

4.2 Analysis of Message

Many pieces of information within the data repository can be used to build up the contextual picture for a given domain. To assist in this task the contextual analyser has been created, which permits domain specific queries to be executed, with each result contributing to a metric or weighting for a number of different concepts relating two entities. For example, this tool can be used to give a value indicating whether two individuals are of a personal or professional acquaintance, or to give a notion of superiority based on line management or academic status. Given appropriate understanding of the domain in which a contextual message delivery system is deployed, suitable queries should be definable to identify the important factors which make up the contextual picture of that particular domain.

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).

In contrast the majority, if not all, of these queries could be performed by the OWL reasoner, and perhaps with some minor extensions it could also be able to perform the weighting summation operations. However, this approach is less favourable than the on-demand queries, as performing the analysis arbitrarily in the reasoner for all possible person–person pairings will cause a vast overhead, which is likely to be incomputable or lead to very unsatisfactory performance. Performance reasons also lead us to store the result of the analyses once calculated, for further reuse if required again at a later stage.

Other systems which analyse large data sets in order to extract underlying relationships, such as that of Community of Practise analysis [2], often use a search or 'growing' algorithm which starts from a given node and follows links between nodes propagating weightings as they do so. While these methods can be used to provide good indication of relationships of an individual within in a community, it does not guarantee us the analysis of the relationships between a pair of individuals. This kind of analysis may play a part as a background low-priority task, quietly performing analysis in case it is one day required, although it should be noted that by relaxing the constraint of either the sender or recipient in our query/metric system, a very similar function could be performed.

4.3 Delivery of Message

Having performed the analysis to generate metrics classifying important relationships between the sender and recipient(s) of a message, and 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 co-located 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.

The location and execution process of these rules is yet to be determined. They are likely to reside as part of the custom rule set loaded into the OWL reasoner, acting when data relating to a message is asserted, and information relating to its sender and recipient(s) is available. The rules needed to model the required delivery behaviour may be numerous and complex, though the data over which they are applicable is likely to be small, and each rule will seldom be fired. However, experimentation may indicate that an external 'on-demand' processing cycle acting in a similar fashion to the contextual analyser may be required on grounds of performance.

If a message is deemed suitable or relevant for delivery given the current contextual representation, the next stage of analysis for the delivery controller is to determine if that message can actually be delivered to the recipient in an appropriate manner. Using the descriptions of devices within our pervasive environment, as described earlier, suitable device(s) may be notified to display the message. This selection must consider which devices are currently being used by, or are in proximity to the recipient, the type and format of the message, its urgency, and whether or not the message is of a personal nature. Similar implementation issues arise here as seen with the determination of whether a message is suitable for delivery.

Messages received but deemed to be inappropriate for delivery in the current context, or those which are unable to be delivered suitably, shall remain queued indefinitely until such a time that their delivery is both appropriate and achievable.

5 Future Work

The delivery controller is the next component to be implemented, and a number of architectural options present themselves. It may be possible to integrate the required functionality tightly into OwlSrv, utilising custom hooks into the inference engine to perform more complex tasks as rules are fired. However, building monolithic applications often leads to poorly adaptable or extendable systems, hence encapsulating the required functionality in an external service or application seems a more logical solution.

Coordination of data and instructions between the display controller, OwlSrv and display devices may be achieved through a number of mechanisms, including publish/subscribe message space models such as Elvin [13] or EQUIP (a platform developed for the Equator project), through the existing HTTP query and update interfaces to OwlSrv, or through Web Service models.

Several of the components of the prototype system use custom data formats, which do not tend towards interoperability. However, ongoing work of particular interest in the field of standardisation includes the SPARQL query language [12] for accessing

RDF repositories, and the Semantic Web Rules Language (SWRL) [14]. The use of both of these standards, once released, could greatly enhance the potential for the reuse of the inherent domain knowledge and behaviours built into a specific contextually aware system.

As a step towards a fully distributed implementation of our system, we adopt a service-oriented view. Descriptions will then be needed not only for the devices but also for publishing and discovering the services. For this we can turn to Semantic Web Services approaches such as the OWL Web Ontology Language for Services (OWL-S) [9], which is an OWL-based Web service ontology designed to facilitate fuller automation of Web service tasks, such as Web service discovery, execution, composition and interoperation.

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 [5]. Our ideas and work to date on the construction of a prototype application have been presented.

Acknowledgements

This work is 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. Alani, H., Dasmahapatra, S., O'Hara, K. and Shadbolt, N. *Identifying Communities of Practice through Ontology Network Analysis*. IEEE Intelligent Systems, 18 (2). 18-25. (2003)
3. Composite Capabilities/Preference Profiles – http://www.w3.org/Mobile/CCPP/
4. Device Ontology – http://signage.ecs.soton.ac.uk/ontologies/device
5. Fallis, S., Millard, I. and De Roure, D., *Challenges in Context*, W3C "Mobile Web Initiative" Workshop, Barcelona, Spain, November 18-19 (2004).
6. Harris, S. and Gibbins, N., *3store: Efficient Bulk RDF Storage*. in 1st International Workshop on Practical and Scalable Semantic Web Systems, Sanibel Island, Florida, USA (2003).
7. Leonhardt, U. *Supporting Location-Awareness in Open Distributed Systems*, Imperial College of Science, Technology and Medicine, University of London (1998).
8. Location Ontology – http://signage.ecs.soton.ac.uk/ontologies/location

9. Martin, D., Paolucci, M., McIlraith, S., Burstein, M., McDermott, D., McGuinness, D., Parsia, B., Payne, T., Sabou, M., Solanki, M., Srinivasan, N., and Sycara, K., "*Bringing Semantics to Web Services: The OWL-S Approach*," presented at First International Workshop on Semantic Web Services and Web Process Composition (SWSWPC 2004), San Diego, California, USA. (2004).
10. McBride, B., *Jena: Implementing the RDF Model and Syntax specification*. in 2nd International Semantic Web Workshop (2001).
11. Millard, I., De Roure, D. and Shadbolt, N. *The Use of Ontologies in Contextually Aware Environments*. In Proceedings of First International Workshop on Advanced Context Modelling, Reasoning and Management, pages pp. 42-47, Nottingham, UK. (2004)
12. SPARQL Query Language for RDF – http://www.w3.org/TR/rdf-sparql-query/
13. Sutton, P., Arkins, R. and Segall, B., *Supporting Disconnectedness - Transparent Information Delivery for Mobile and Invisible Computing*, CCGrid 2001 IEEE International Symposium on Cluster Computing and the Grid, 15-18 May, Brisbane, Australia.(2001).
14. SWRL: A Semantic Web Rule Language combining OWL and RuleML http://www.daml.org/rules/proposal/