Context-Awareness for the Mobile Environment*

Andry Rakotonirainy[††], Seng Wai Loke[†] and Geraldine Fitzpatrick [††]

[†]CRC for Distributed Systems Technology, Monash University

School of Computer Science and Software Engineering

Monash University, Caulfield VIC 3145, Australia

`swloke@dstc.monash.edu.au`

[††]CRC for Distributed Systems Technology, University of Queensland

Level 7, General Purpose, The University of Queensland, QLD 4072, Australia

`andry@dstc.edu.au, g.fitzpatrick@dstc.edu.au`

January 31, 2000

Abstract

The proliferation of mobile devices in our daily-life implies a complete re-organisation of the IT architecture. Mobile objects (people and software or hardware or combinations thereof) face dynamic and drastic changes of context. Context awareness helps an application respond appropriately to, and take advantage of, these context changes. This paper discusses the concepts of context and context-awareness for the mobile computing environment, and sketches our context-awareness model which is based on an event notification service. The model enables each individual entity of a mobile environment to be aware not only of its own context but also the context of other entities. We discuss why contextual information is needed, what contextual information we need, and how to get them.

1 Why is Context-Awareness Useful in the Mobile Environment?

The answer in a word is "adaptation". An entity needs contextual information to choose between alternative strategies in order to reach its objectives (or to do useful work). Context gives hints about what is or what is not achievable. If the initial objectives are not reachable then they are changed to suit the current context, i.e. the objectives change or actions are taken according to the context

*The work reported in this paper has been funded in part by the Co-operative Research Centre Program through the Department of Industry, Science & Tourism of the Commonwealth Government of Australia.

to maintain an objective. The entire mobile environment is then seen as a set of entities associated with a set of adaptability rules. Activation of adaptability rules depends on the current context of entities.

Knowing its own context is important in a mobile environment because mobile objects tends to wake up in an unknown and hostile environment and has to "survive".

Not only knowing its own context is important but it is equally important to know the context of other entities in the environment.

The environment of the entity includes

1. its own context (e.g., bandwidth, cpu, screen definition, etc)
2. the context (as observed by this entity) of entities this entity is interacting with.

Each entity has the freedom to reveal fully or partially its context.

Awareness of own context is important for an application to have sensible Quality of Service requirements. In a mobile environment, if a QoS goal cannot be reached then this application might decide to lower its QoS requirement. For example, instead of requiring high definition pictures from a remote server, the user might lower his/her requirements by requesting a textual description of the picture due to the the usage price of the network. This simple example shows that the application corresponding to an image viewer is aware of the context manifested by the user preference and network usage rate provided by an ISP. Indeed, exchange of contextual information might be part of QoS negotiations.

2 What is Context?

There is not a unique definition of context. However, we give a broad definition of context from the mobile computation perspective.

Dey & Abowd (1999) [1] defines context as

> "any information that can be used to characterise the situation of an entity, where an entity can be a person, place, physical or computational object."

Normally, many characteristics can be used (and in real-life, perhaps infinitely many possible characteristics). However, to an application (and practically so) only some characteristics are relevant, i.e. the context is formed by a set of relevant information/characteristics.

In a mobile environment, there is not only the issue of relevance but also obtainability of information. Some contextual information might not be obtainable due to resource changes on moving from one environment to another (e.g., one environment has some sensors whereas another does not). Also, different entities in a distributed environment can have different capabilities and so an entity in one environment can obtain some kinds of information whereas the same (or different) entity in a different environment cannot obtain such information. Therefore, we extend the definition of context as follows which looks at context from the point of view of individual entities:

> "Context (from an entity's viewpoint) is information that can be used to characterise the situation of an entity and can be obtained by the entity, where an entity can be a person, place, physical or computational object."

When an entity asks itself, "what is my context?", information it cannot obtain will not be part of the answer at all. Similar, when an entity asks itself, "what is the other entity's context?", information it cannot obtain about the other entity will not be part of the answer at all.

The following two subsections introduce the concepts of abstract context and concrete context which emphasise that some information about the context might not be knowable to an entity (or cannot be found out by the entity).

The following definitions allow us to have two different meanings to context-awareness:

- aware of abstract context means "aware of what you can know about your context"
- aware of concrete context means "aware of what you do know about your context"

2.1 Abstract Context

An abstract context consists of the *description* of different observable dimensions (or attributes) characterising a situation. The description is separated from any measurement or values. A situation of an entity has a scope, it corresponds to a set of actions (something which happens) that has/can impact on the behaviour of the entity. The behaviour of an entity can be influenced by internal or external actions. Situation refers to the external actions. These dimensions (due to the external actions) must be observable by the entity, i.e. the entity must have the capability/means to observe them. Abstract context can change (say due to mobility) as the set of observable attributes characterising a situation of an entity changes (as the capabilities or "power of observation" of the entity changes). So, being aware of the abstract context is being aware of what can be known/observed about a situation.

From an implementation viewpoint, it might consists of metric descriptions or type definitions (e.g., logical domains like location, or time). Ranges might be specified for values of these attributes as part of the abstract context.

What are examples of such dimensions/attributes? A context provides a variety of information and answers questions such as where I'm, what did I do, what am I doing, where I'm going, what can I do, what am I becoming, what are the others are doing or have done.

Therefore the context should describe information related to:

- the user, e.g. for capabilities and preferences exchange, ad-hoc preferences
- location (logical, domain)
- devices, hardware profiles
- network: bandwidth, disconnection rate, price
- computational objects, availability, reliability, security ...

Access to the above information allow an object to choose the best options between alternative adaptation actions.

2.2 Concrete Context

A concrete context is an instantiation of an abstract context. Each attribute of the abstract context is assigned a value. Information in the current concrete context is restricted by the current abstract context (you can't have a value for a non-existing attribute and the value might be constrained by a range).

The values of the attributes belonging to a concrete context can change. Its value could become unknown (for some time), or unknowable (attribute not even in the abstract context). The absence of information about an attribute is still considered as useful information - "knowable but not known (yet)."

3 How can Awareness of the Contexts of Distributed and Mobile Entities be Achieved?

We briefly mention our model for context-awareness in distributed mobile systems. The design of our context-awareness model is driven by the following requirements:

- A clear separation between context definition, context measurement, context interpretation (e.g recognition techniques for analysis), context management and context dissemination.
- Asynchronous and scalable context information access from any distributed mobile entity. Asynchrony is suitable for network disconnections or latencies which characterise the mobile environment.
- Support for standard context description to allow platform independence.
- Support for context aggregation (e.g Room 12 maps to building C).
- Support for dynamic updates of context contents (evolutionary).
- Support for storing context history.
- Support for context filtering. As sensors such as camera, microphones, GPS receiver will increase data about context we need a way to filter the amount of input information.

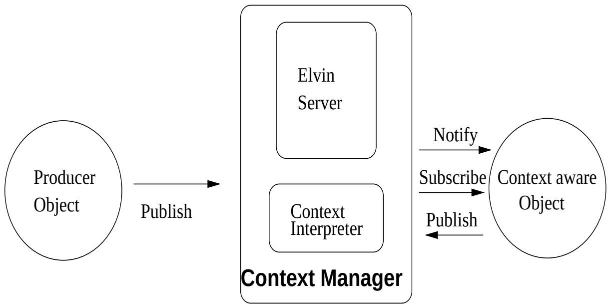

Figure 1: Context Manager

We assume each entity has both an abstract context and a concrete context. Also, we assume that a context manager is located in each site. A context manager can be accessed locally/remotely provided that they have permission to do so. The architecture of a Context Manager is shown in Fig 1. It is based on the Elvin Event Notification Service [2] which is a content-based publish and subscribe system with multi-party interaction. A producer makes visible what it want others to be aware of and the consumers decide to be aware (or notified) of whatever they are interested in. The publish/subscribe model provides the asynchrony. The context interpreter contains an interpreter that analyses different the notification, publication and subscription. For example a subscription to a given event might imply the unsubscription to others due to some defined policy.

A context is described in a standard notation (e.g., RDF,XML as used in the W3C CC/PP working group[1]) which enables attributes and values to be represented, and used as a input for the subscription. An example of a subscription is:

```
exists(usenet)and
(subject matches(``.*[Cc]ontext*)
or from_email matches(``andry@dstc.*''))
 and crossposts < 5
```

The above subscription browse usenet and retrieve posts that have the subject matching with the string “Context” or “context” and any post from “andry@dstc”.

Context aggregation can be achieved by correlating multiple notifications, i.e. combining several event notifications to form one notification. For example, when every device in the system has enough memory (as we know from combining the contexts of these devices), that could be the context for some application.

[1] See http://w3c.bilkent.edu.tr/TR/NOTE-CCPP/

Updates of context contents is required when an entity A has knowledge of another entity's (say B's) context but B's context changes, thereby making A's view of B's context obsolete. Updates can be propagated by A subscribing to changes in B's context. Changes can occur either to an entity's concrete context (only values change) or to the entity's abstract context (attributes change according to "power of knowing") which in turn affects the concrete context instantiated from this abstract context. A context history can be archived by channelling such notifications to a database.

4 Conclusion and Future Work

Key attributes of adaptability in mobile computation is context-awarness of the environment.Context-awareness allows an entity to adapt its behaviour to the circumstances in which it finds itself in. We have presented motivations for contexts in mobile environments, and an architecture for its implementation. Future work involves defining the interfaces that allows context-awareness and fleshing out the abstract context descriptions.

We have outlined our motivations for context in mobile environments and presented an architecture for its implementation. Future work at this infrastructural level involves prototyping and fleshing out the abstract context descriptions.

But this is only part of the story for putting mobility to work for people engaged in their day-to-day activities. To complement the infrastructural view of context, we will also be exploring the social, organisational and work contexts that underpin people's motivations in using mobile environments. These contexts are complex, dynamic and evolving, often in unpredictable ways. They include both computational objects and devices but also physical objects, and, more interestingly, 'soft' objects, i.e., people and their intelligence and creativity in achieving what they want to do.

In exploring these richer notions of context, some of the questions we will be asking include: how well do our abstract context descriptions map to the user's needs and perspectives? What are the implications for how we implement and manage context 'at work' for people?

References

[1] A. Dey et al. The conference assistant: Combining context -awareness with wearable computing. In *Proceedings of the 3rd International Symposiumon Wearable Computers*, 1999. The Conference Assistant: Combining Context -Awareness with Wearable Computing. San Francisco, CA: pp. To Appear.

[2] B. Segall and D. Arnold. Elvin has Left the Building: a Publish/subscribe Notification Service with Quenching . In *Proceedings of AUUG97*, Brisbane, Australia, September 1997. Available at <http://www.dstc.edu.au/Elvin/doc/papers/auug97/AUUG97.html>.