

The Scooby Event-Based Pervasive Computing Infrastructure

Jon Robinson & Ian Wakeman
School of Informatics
Network Lab
University of Sussex
{ ianw@cogs.susx.ac.uk, jrr20@cogs.susx.ac.uk }

Introduction

In this abstract we describe the lower levels of the Natural Language Composition project. We are developing a pervasive computing infrastructure, which we have called Scooby¹, through which we will be investigating a number of issues in the development of pervasive computing. We describe these issues and the current state of the Scooby environment.

Research Issues

Services and features described by some form of logic, where actions occur when some predicate based on events on the system become true. The key research question we will be addressing is what this logic should look like. This is currently under investigation.

When the logic has been decided, a number of other issues have to be addressed. For instance, what form should the programming language take? The language will primarily be machine generated and processed. However, any service composed will have to be map onto a natural language description, and may have to be modified by people. Therefore, we have to ensure that the syntax of the language is human readable. Naturally, in the design of any programming language, other issues are raised, such as the operational semantics, and the potential efficiency of any implementation.

Service composition will take place through the human manipulation of objects described in an ontological layer above Scooby. We hope to leverage off work within the semantic web community, building upon the Web Services Description Language [1]. In particular, we will be hoping to provide ways to help software engineers automatically generate service descriptions which enable their services to be easily integrated into the ontological layer.

Scooby and event based pervasive computing

Scooby, our initial system, is an infrastructure based around the Elvin content based router [2]. Sensors, applications and people generate events from various input channels (email, SMS) into the Elvin messaging system. Applications and people add subscriptions to the router, and when their subscription patterns match events, the

¹ Scooby emerged from our initial name, the “Softsys in Cogs UBIquitous thingie”.

event is forwarded to them through whatever channel they wish. We have designed a prototype composition language and implementation for users to control the propagation of events and to designate actions when particular event combinations become true.

Elvin is a content based router, in which events described by free-form keys and data are delivered to the router. The router then matches the events against subscriptions. For each subscription match, the event is delivered to the subscriber. The choice of Elvin was in many ways a pragmatic one – it is being widely used within the Equator project [3], and is well-supported locally. Elvin would appear to be well-suited to the development of event-based pervasive computing, but there are a number of issues yet to be fully explored, such as the scalability, the security and extensibility of the content based router. These may have an impact upon the implementation of the Scooby composition language.

We have provided a syntax for events in the Scooby system, detailing source service or sensor, expiry time, notification type for destination discrimination and general data. These form the basis of events which are propagated throughout the environment as the primary form of communication between services. The event description is also utilised in the propagation of discovery events for the purposes of service identification, but also includes additional information regarding the location of a service.

We have provided a number of services such as service discovery and location, a blackboard service to allow agent based personalisation and gateways to SMS and email. We have built services such as “we’re going to the pub” notification system, where users can configure the destination of the message based on their current whereabouts. Pub notification events can either be generated directly from the user’s desktop, or can be received via the SMS or email gateways. A desktop application can be programmed to route the event through SMS or email based on when certain events were true. This is programmed through the Scooby composition language.

Conclusion

The primary goal of this investigation is to identify the issues involved in producing a pervasive computing infrastructure which supports service composition. We have discussed the current incarnation of the initial system, with a scenario which demonstrates a use of the system, and ability to specify services using the Scooby composition language. Issues still remain on resolving the integration of services into the broader context of web services, along with scalability and extensibility challenges which can result from larger scale systems.

[1] Web Services Description Language, <http://www.w3.org/TR/wsdl>

[2] *Elvin*, <http://elvin.dstc.edu.au>

[3] Equator Project, <http://www.equator.ac.uk/>