Rapid User-Centred Evaluation for Context-Aware Systems

Eleanor O'Neill[1], David Lewis[1], Kris McGlinn[1], Simon Dobson[2]

[1] Department of Computer Science, Trinity College Dublin IE
[2] Systems Research Group, School of Computer Science and Informatics, UCD Dublin IE
Eleanor.ONeill@cs.tcd.ie

Abstract. This paper describes a platform for the user-centred design and evaluation of adaptive, context-aware services in the wireless, mobile and pervasive computing markets. It focuses on evaluating the user interactions with context-aware adaptive systems while synchronising the control of the environmental context that drives adaptivity and the user's perception of that environment. The platform uses a 3D virtual reality simulation to present the environment to the user and to drive the generation of simulated environmental context. The platform thereby delivers repeatable, instrumented, context-dependent evaluations of adaptive services over a range of contexts. It aims to reduce development costs and facilitate the development of more effective, user-empowering services.

Keywords: User-centred design and evaluation, usability, context-aware services, adaptive services, 3D virtual environment.

1. Introduction

More so than other distributed systems, pervasive computing services and other context-aware mobile services must dynamically adapt to the needs of the user and to the current physical, social and task context in which those needs are formed. For instance a weather forecast service may localize its content based on the user's current location. Alternatively, a news notification service on a user's PDA may adapt the volume of its alerting tone based on the level of ambient noise detected or mute itself if the user's calendar indicates she is in a meeting.

The effectiveness of the exhibited adaptive behaviour is highly dependent on the subjective experience which is influenced by their perception of and interaction with the environmental and social context of the task they are currently attempting. For example, too little adaptivity does not offer any significant benefits; too much means that users cannot predict how the system will behave in a given situation. Developing an effective context-aware adaptive service therefore requires extensive user-centred design and testing as the proposed adaptive functionality for the service evolves.

To produce successful services, developers must be able to exercise services prior to deployment, and to incrementally add (or remove) adaptive behaviours in response to feedback from users experiencing that behaviour in a range of context spaces.

Developing and testing context-aware services is extremely challenging since there are few effective ways to carry out on-going user evaluation with a controlled, repeatable profile of context-change. It is the large number and type heterogeneity of the independent variables associated with such real-world context, e.g. physical, social and task contexts, that makes testing expensive and thus problematic to integrate into the overall engineering process.

At present mobile service developers address these issues through unit testing and final integration testing. In the case of large systems, integration testing typically occurs immediately prior to deployment without realistic user assessment and ruling out possibility for major change. However progress in rapid prototyping methods and tools has been identified as central to overcoming the barriers to widespread development and deployment of ubiquitous computing applications, according to Davies et al [23].

Here we present a platform for the user-centred evaluation of context-aware services which provides a 3D simulated pervasive computing environment. The simulated environment is sufficiently realistic to accurately convey changing physical and social context to the user through the virtual representation of the environment. In conjunction, the adaptive system under test also receives a simulated electronically sensed view of the environment based on the configuration of embedded simulated sensors in the virtual environment. The service can thus create its own view of the physical and social setting of the user.

The problem of controlling synchronized user and system views of context is addressed through the simulated virtual reality environment An adaptive service model will express the relationship between a service's core behaviour and variation in that behaviour in response to context changes. Ultimately this model will drive the rapid prototyping of the service itself, together with the configuration of the usability evaluation instruments.

In this paper we describe the initial implementation of our platform and report on its usability from the point of view of the experimenter. In section 2 we discuss the state of the art in the evaluation of context-aware adaptive services and in the related use of 3D simulations. We then discuss the current implementation of the platform in section 3. In section 4 we report on our experiences in configuring a complex simulation for evaluating a composite set of adaptive services. In section 5 we describe how we aim to extend the platform to support usability testing for context-aware services, using ontology-based semantics.

2. Relation to State of the Art

Currently, user acceptance testing of mobile services involves expensive field trials where the usage context and the user's experiences may be hard to instrument. Usage tests of pervasive computing services that integrate with situated sensors have been largely lab-based and thus are a poor representation of the variety of real-world context users will encounter in the course of their every day lives. For larger location-aware or pervasive computing applications the cost of user-testing a full service deployment quickly becomes prohibitive [2], especially where the interaction between context variation and the behaviour of the service is still being explored, thereby making effective experimental design problematic.

Groups such as the Future Computing Environments Group at Georgia Institute of Technology working on the "Aware Home" [19] and Tatsuya Yamazaki of National Institute of Information and Communications Technology, Japan working on the "Ubiquitous Home" [18] have completed real-life test home environments for accurate simulation of the home environment. Both groups aim to perfectly emulate a real domestic environment and intend to have test-subjects spend significant periods of time in these simulated home environments carrying out domestic activities. However, such live usage testbeds are expensive and difficult to reconfigure to emulate a wide range of different contexts.

Kerttula and Tokkonen [16] have identified "the total user experience" as an area of concern and aim achieve it through early product and system simulations. This idea moves away from testing in isolation and moves towards a simulation where services are tested in parallel and valued over longer periods of time. This approach uses accurate simulation/prototyping of services focussing on features such as the user interface, audio properties and product behaviour, but not including the user's surrounding physical environment.

Similar to our platform, Huebscher and McCann [17] aim to allow initial testing of context-aware applications without requiring a physical deployment. However Huebscher and McCann are working to simulate sensor data e.g. temperature, humidity or location, from a description of context or a simulation model of contexts. This in turn will be used to test the context-logic of a context-aware application.

In the past, virtual reality simulation of pervasive computing environments has been used in a small number of research efforts, specifically QuakeSim [5] and HP Lab's UbiWise [6]. These have demonstrated that 3D virtual reality computer game engines potentially provide a cost effective platform for simulating pervasive computing environments with sufficient realism to accurately test human interaction with pervasive computing software systems.

More recently Shirehjini and Klar have been developing 3DSim[20], a 3D tool for rapidly prototyping Ambient Intelligence building blocks e.g. situation-recognition, goal-based interaction. 3DSim aids the development of human-ambient-interaction systems such as PDA based control systems, adaptive user interfaces, multimedia output coordination or goal-based interaction systems. During a simulation, sensor data is derived from a 2D GUI and gesture elements which are the result of an avatar can pointing at devices.

The team at GIST U-VR Lab, S. Korea have been working on creating a unified context model and a method for the integration of contexts for unified context-aware applications. To loosen the coupling between services and context, they have developed a unified context that represents user-centric contextual information in terms of 5W1H (Who, What, Where, When, How and Why) [21]. To demonstrate user-centric integration of contexts for a unified context-aware application model (the ubi-UCAM), they created a simple 3D simulated environment [22]. By using the simulator they were able to test the effectiveness of the Context Integrator when there were multiple users working with the service simultaneously. The simulated environment allowed them to assess the capabilities of their Context Integrator before bringing it into a real world situation (ubiHome).

Overall, our platform is distinguished from existing ubiquitous computing simulation approaches in that we focus on providing a flexible and easy to configure platform for the tester/experimenter with the target of integrating seamlessly into a wider rapid prototyping process.

3. An Evaluation Platform for Context-Aware Services

The overall goal of the platform is to provide for the rapid user-centred evaluation of adaptive context-aware services by effectively and efficiently testing and evaluating usability and thus increasing productivity in the development of these services. The platform must support the rapid prototyping of adaptive service behaviour through ease of use in the design, execution and instrumentation of user acceptance tests.

Elements, collectively referred to as contextors, must be provided to sense context both from the physical world (sensors) and gather data from personal and other information (data mining). Individual adaptive behaviours for services, must be provided, the most visible being the user interface [3] but also including adaptive information storage and retrieval and operation of actuators in the pervasive computing environment. A service's behaviour must be verified as remaining within a well-defined behavioural envelope across its exhibited adaptivity [4].

The net effect of the tool is to increase the effectiveness of services by incrementally maximizing user acceptance and thus reducing the risk involved in full scale field trials or deployment.

3.1 Interactive Context Simulator

The interactive context simulator has been implemented to allow a researcher rapidly configure and run an experiment for a prototype of their software, using simulated context generated at runtime. The context generator features a multi-user 3D simulation component, a proxy gateway which interfaces services to the simulated 3D world, the under-lying network infrastructure and a real time execution environment. With many users interacting with the service under conditions set by the tester, this provides the service developer with a sophisticated method for experimenting with collaborative, context-aware systems.

The 3D simulation component of the platform is provided by the Half-Life 2 (HL2) game engine [7] which has been modified to enable extraction of information from the environment in XML encoded messages. The game engine has been further tailored towards a pervasive computing environment through the addition of pervasive computing sensor models. Creating a new simulated pervasive computing environment uses existing HL2 modelling tools to place sensors in the virtual world so that at run-time user activity and movement in the virtual world activates the sensors in accordance with experimental objectives. On activation, a sensor model responds by generating an XML encoded message containing information related to the event e.g. username or location data.

Simulated sensors have been modelled to be visible or invisible. We use visible simulated sensors to represent physical devices e.g. pressure mats or wireless access points. Invisible simulated sensors are used to model the field of view or signal range of these devices where required. Supplementary simulators have also been interfaced to the platform to support this approach by providing realistic simulation of RF signal propagation and location information through triangulation [15].

The sensors are programmed to be event-driven, polling or a combination of the two. For instance, a pressure mat responds to the *event* of a user stepping on it, where as a Bluetooth master *polls* to detect new slaves. Using a game engine allows flexibility in the type and quantity of sensors featured by the test environment. For the most part, this is not yet realisable in the real-world where the expense and logistics are prohibitive.

Fig. 1. Multiplayer Virtual Environment.

Interfacing the system-under-test (SUT) to the simulator is done via a Java application or Proxy. The platform can host and manage the connections between multiple services and multiple test environments simultaneously. This allows multiple services to access a single environment, or vice versa, a single service to access multiple environments. Services are not obliged to subscribe to all simulated environments and only receive information about relevant experiments.

Prior to connection, the experiment designer will have created or adapted an already existing 3D map. Hammer 4, a map editor provided as part of the Half-Life 2 SDK (HL2 SDK), is the tool that developers currently use to do this. The flexibility of the HL2 SDK means a wide variety of environments and sensor types can be modelled. These sensors can then be deployed into a map, in the positions, densities and numbers that are required for a particular experiment.

Although developing a large map takes some effort, considerable productivity can be achieved by using a blank version of an existing environment to outline an experiment. The effort to populate blank maps with sensors is minimal by comparison to developing a map of a new environment from scratch. The experimental design and set-up process makes use of reusable resources in keeping with the iterative and incremental approach required by rapid development, testing and experimentation. Among these reusable resources are the map files that define the experimental environment, the sensors and the experiment definition XML profiles for a service.

A new experiment commences when a service contacts the simulator with an experiment configuration file. This configuration file contains an experiment ID, a map name, a game-server address and data subscription information. The service is registered and the simulator creates a new database [8] collection using sensor information parsed from the map file. The simulator invokes a new game-server on the remote host and subsequently establishes a connection with the simulation for experimental data transfer.

At run-time, messages flow between the virtual environment and the adaptive service. Data leaving the simulator becomes the contextual information on which services base their decisions and thus respond to the user's needs. In response, services send asynchronous instructions to alter the state of the environment through device or entity actuation, e.g. opening a door or switching on a light. Only a single connection to game-server hosting the experiment is required since underlying game infrastructure ensures game-clients are also updated in a time that is imperceptible to the player/developer. Ultimately, the sensors will send their information to the services under test *via* a contextual services layer.

4. Experiences in Configuring Experiments on the Platform

Here we report on our experience to date with the Interactive Context Simulator in setting up experiments with a collaborative context-aware service that had been developed as a research prototype by colleagues in our department.

4.1 Modelling the Physical Environment

We have had experience both in successfully importing existing 3D maps and in building maps from scratch. Prior to the experiments described here, a three story office building model was constructed which was an accurate representation of a portion of the Computer Science Department at Trinity College Dublin. The model features 104 rooms comprised of offices, computer labs and lecture rooms. In total these rooms are furnished with 520 desks, 352 chairs and 257 replica desktop computers. An undergraduate intern, untrained in the Hammer map editor, completed this map in 22 working days. The resulting model is a substantial resource, supporting experiments where users can roam on a scale that dwarfs that of indoor lab-based emulations used elsewhere.

We also opted for an accurate population of office furniture over more sparsely furnished rooms, since we wanted to replicate as closely as possible the user's experience of the real spaces so that we could conduct comparative experiment in the real world building at a future date. In the process we gained experience in how to produce such maps more efficiently in future, for instance in the use of overlapping polygons and transparent textures.

4.2 Experimenting with Context-Aware services

We have uncovered some of the merits and difficulties in using this platform for the evaluation of pervasive computing services under development, by observing colleagues using it to configure usability experiments on context aware systems they have developed. In this context, the system under test is an Instant Messaging (IM) application that can display the location of other users as part of their presence information, but only when permitted through a sophisticated policy-based access control mechanism.

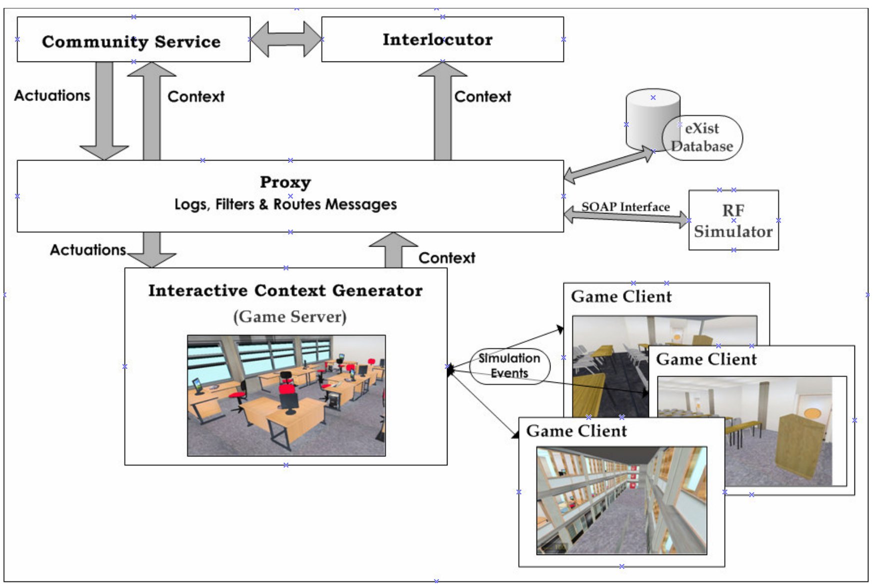

Fig. 2. Experimenting with Context Aware Services.

This Location-aware Instant Messaging system extends an existing standard IM server infrastructure, based on JABBER, with a decentralised communications infrastructure employing content-based routing. This adds an additional level of flexibility, allowing applications to subscribe to events of interest based on the event's content rather than overall message type [23]. The IM application allows online users only to view the locations of fellow online members under access control based on policies employing concepts of Community and Trust.

The simulation platform is beneficial as it supplies context in a more dynamic and unanticipated manner since its events are user-invoked and user-controlled. The IM service was tested on an environment where six users were interacting with the environment and service. To enhance location-aware testing, the simulation platform has interfaced to an RF simulator to provide simulated signal propagation values and location information through triangulation of signals [15].

4.3 Experimenting with Policy-based Access Control

Community Based Policy Management [24] models an organisation's structure as a set of communities. Networks of collaborating users can self-organise because the framework supports dynamic definition of sub-community structure and operational rules. This provides for autonomous sub-communities and allows migration of decision making responsibility to the most appropriate communities in the organisation.

A policy based access control system for the dissemination of location information has been incorporated into the Instant Messaging application above. The policy-based system is characterised by allowing communities of users to agree on policies for the formation of buddy lists, or rosters, in the application. The system also allows policies to be created for determining who is able to monitor the physical location of other users and, in addition, which users can access particular locations (or rooms) in a building. This latter facility was enabled by linking a decision to lock or unlock a door to a request to the access control decision function in this system under test - thus also linking actuation in the simulated environment to the system. The researcher in question wished to determine the ease with which users could collaboratively configure complex access control policies and be satisfied with the operation of those policies as they collectively engaged with the pervasive computing environment.

To evaluate this in situ would only be possible over a long period of use in a location aware environment however the platform provided quick and easy means to designing an experiment to accelerate the usually ponderous interaction of users with policy authoring systems under controlled conditions. This was achieved through experiments where users engage in team games, e.g. catch the flag, in the simulated environment, but using the policy management interface between games/experiments to investigate different rules for the games by changing access control policies for viewing the location of other via the IM application and for enabling room access to the user's avatars. Though the situation is somewhat artificial, the simulator provides a low cost but never-the-less engaging environment for the user in which they can be stimulated into interacting with the adaptive aspects of an application.

4.4 Platform Usability

Based on their experiences, researchers are very receptive to the experimental approach proposed by this tool. The findings from the researchers' use of the platform outline the researchers' opinions on the simulator's usefulness and usability.

Setting up the first version of an experiment, in a new iteration sequence requires the most overhead in terms of time and effort. However, subsequent adjustment to the files is minimal or even non-existent when an experiment is well-defined and fine-tuned. During interviews, researchers reported installation of the simulator, and its accompanying tools, taking on average 30-45 minutes. Further, to initially populate a basic map with sensors and make the XML file associations an additional 60-90 minutes were required. This step partially depends on how powerful the user's computer is. On the grounds that researchers were not at the time familiar with the toolset, i.e. the Hammer map editor, it is expected that these times will improve.

Using the platform, the developer of the location-aware instant messaging service was able to test and debug the system from his desk during the design and development cycle. In particular, it greatly eased the testing of the interaction between the IM application and various configurations of simulated location sensors which the developer confirmed would otherwise have been virtually impossible due primarily to budget constraints. However, even with a substantial equipment budget, the developer would still have faced logistical hurdles regarding deployment of sensors in campus buildings within the college which would be insurmountable at the proof-of-concept stage of development.

A further benefit for the developer meant that it was not necessary to enlist a group of volunteer test users to use a real-deployment of the service. Instead the developer was helped by lab partners to manipulate the location of virtual users in the virtual environment. Since the virtual environment is based on a game engine and very intuitive, the learning curve for new users is minimal. As a result, to run a multi-user experiment required little organisational effort and reduced planning and scheduling of testers and timeslots. These experiments could be run regularly and at short-notice, which was helpful for the debugging process.

5. Extending the Platform

Ultimately, our aim is to combine user-centred design and evaluation. This requires extending the platform as has been implemented to date. Our next steps are to integrate into the platform: support for more sophisticated context processing; more realistic simulation of sensors and adaptive user instrumentation. These extensions are discussed in more detail below.

5.1 Context Services Layer

As is clear from above, one may not treat sensor information as fact but only as evidence of fact, to do otherwise would expose applications directly to all the noise in the environment [9]. This implies that sensor information must be combined with information from other sensors, users models etc, in order to arrive at a stable model of the environment.

A number of approaches to such sensor fusion have been reported in the literature, [10], [11], and [12]. It is not clear that any single approach has yet demonstrated superiority, and within the platform we are experimenting with four complementary approaches; Bayesian networks, fuzzy logic, Dempster-Shaffer evidence theory, and machine learning. Early explorations suggest that each is a plausible candidate for performing high-level context fusion. It is worth noting that, while machine learning may prove extremely useful for adjusting the prior probabilities of sensor events to match observed conditions, these probabilities are functions of context themselves, as mentioned above, and the only criteria for deciding on the accuracy of a sensor observation is the fused result of other sensor observations; a result which itself may not have a strong confidence level.

Our solution to supporting sensor fusion is to run adaptive service tests on top of a contextual services layer based on Construct [13]. Construct provides a highly scalable, distributed platform for collecting and managing contextual information represented using the World Wide Web Consortium Resource Description Framework (RDF). Contextual fusion is supported at the model level, with applications either querying the model or being driven by a truth-maintenance framework. The use of RDF abstracts the details of the sensors underlying a particular installation; sensors write information to the knowledge base under the appropriate ontology, which may be accessed by applications without being aware of the detailed sensor population. This improves the robustness of applications to individual sensor failures and simplifies the addition of new sensor capabilities.

Overall the services under test can be isolated from the fact that their inputs are coming from a virtual environment, they are simply given access to a contextual model to which they can react. The behaviour of the service is less well-abstracted at present, in that some behaviour is targeted directly at devices which must be simulated within the virtual world. Since the models of sensors and their fusion are expressed within the Construct framework, it is straightforward to change the parameters of the simulation to, for example, cause sensors to fail or exhibit more inaccuracy. We conjecture that this will assist in the development of applications that are more robust to sensor noise, sensor failure and uncertain information in general.

The design consists of a closely integrated set of tools and accompanying methodologies, Fig.3. Experimental design involves the tester building a simulation model of the physical environments encountered by the human test subject and populating it with simulated sensors that would provide user driven excitation of the region of context space being investigated. The accurate propagation of realistic context information to the adaptive service under test is mediated by the contextual services layer. The adaptive service model is used to help identify and generate the simulated environment through the identification of experimental goals, test cases and the required control of contextual services. It is also used in generating the user and service evaluation instruments for a test, i.e. which user and service behaviour parameters to monitor and log, along with the structure of user evaluation questioning.

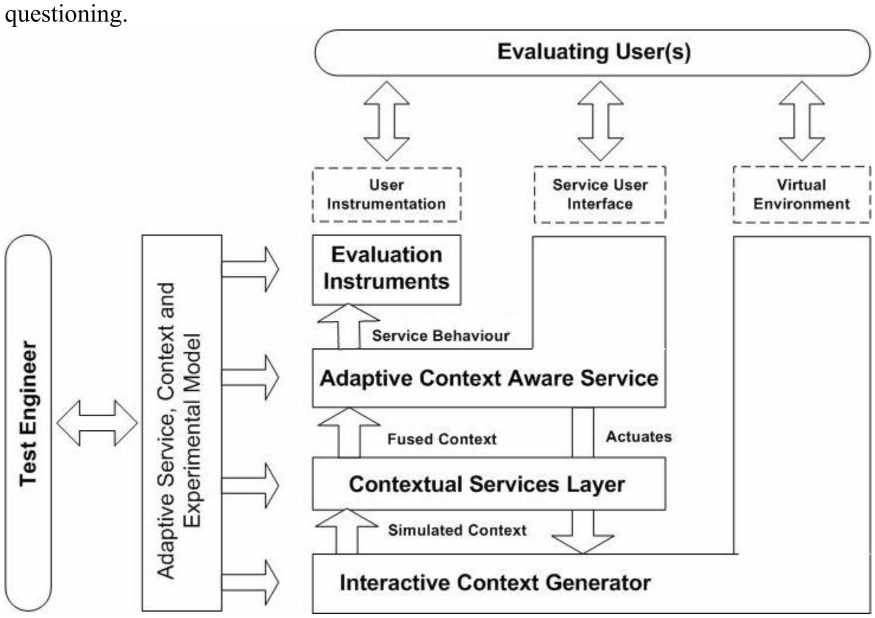

Fig. 3. Overview of Platform Architecture.

5.2 Sensor Simulation

Sensors in the platform are implemented as objects in the virtual world, as mentioned above, which expose certain facets of that world to the services under test. The simple approach currently taken in the simulator defines the range and capabilities of a particular sensor, which would then be used by the virtual world to generate sensor events. We recognise, however, that such an approach is naïve for two reasons.

Firstly, all sensors include significant noise components. These manifest themselves as inaccuracies (*i.e.* a location sensor reporting the presence of the wrong person) and imprecision (*i.e.* a location being an area rather than a point). Evaluating a service against such sensors might lead to services that function well in the presence of correct and precise information but which fail when exposed to real-world inaccuracies and imprecision.

Secondly, sensors are themselves sensitive to the context in which they are installed. Two examples illustrate this:

- A wireless communications system will encounter signal reflections and attenuations in buildings with substantial amounts of metal in the walls. These reduce the effective range and bandwidth of the communications channel.

- RFID tag readers frequently fail under heavy load, such as when several people move past a reader in quick succession. This manifests itself as the reader only observing a fraction of the tags that actually come into range.

These two constraints mean that the characteristics of a sensor must take account of both the static and dynamic context into which it is deployed. This is essentially the same problem as encountered in modelling the larger adaptive service, and means that we must provide 'user models' of sensors together with context-aware behaviour, which are then used to generate the information for the services under test.

5.3 User Instrumentation

Integrated user evaluation instruments will be used to conduct tests by allowing the user to roam within and interact with the virtual physical environment. This system will provide task instructions to user test subjects based on context change or adaptive service notification received from the Simulation Infrastructure. It will present the user with evaluation instruments, such as questions or prompts for free style comment.

Responses will be logged aligned to the activities the user has performed in the simulator, the context changes and the adaptive behaviour to which the user has just been exposed. In this way users may interact with the simulation in a non-linear way, exploring the environment and deviated from tasks as the might to in the real world, but the instrumentation of their experience will be recorded in context to assure accurate analysis of the results.

6. Conclusions and Summary of Further Work

To date this Platform has largely been used as a simulator of pervasive computing environments and context generator for experimenting with context aware services. We aim to build on this by exploring the capture of a fuller semantic model for the adaptive behaviour exhibited by a service, beginning with the integration of the contextual service layer with the Interactive Context Simulator. The model will map the context space within which an adaptive service operates to the behaviour space that the service can exhibit. This model will be used both to provide abstractions for rapid, script-based prototyping of new adaptive behaviour, which will thus be integrated, with the ontology-driven experimental design and configuration.

This will lead to improvement in the productivity of the testing and evaluation phase of the rapid development cycle for context-aware adaptive services. This improvement will be achieved through the model-driven configuration of test cases, shorter test development life-cycles, more targeted and relevant user evaluation, a low-cost test infrastructure and the facility for on-line user testing, thereby resulting in a lower overall cost for the test portion of the development cycle.

In line with the range of behaviour exhibited by adaptive services, the aim in evaluation will not be to verify correctness of system behaviour, but to confirm that it operates within a well defined behavioural envelope given for specific regions of the possible context space that corresponds to the overall testing goals. The adaptive service model will be used to identify these experimental goals as well as to identify test cases, and context and control variables. It will also assist in the generation of the simulated environment and user and service evaluation framework through a set of closely integrated tools and methodologies.

Additionally, future development work will fully integrate a set of user evaluation instruments into the simulator to allow online, runtime questioning of users. Usability Engineering offers a number of techniques for evaluating ease of use and user acceptance. Though tool support for the former in mobile services is maturing, in focusing on the latter we address an area where effectiveness remains elusive as evidenced by high-profile failures in mobile data services.

Recent developments allow adaptation of hypermedia documents to be based on the selection of narrative, content and user meta-data [18]. This will be applied to adaptively assemble different evaluation instruments based on questioning style (e.g. factual, opinion, attitude, open/closed ended or Likert style), instrument topic (e.g. adaptive behaviour experienced) and user data (e.g. demography, previous responses).

Extensions to the simulation element of the platform are planned to include integration with a wireless indoor signal propagation simulator. This will allow us to factor in communication performance degradation for both context and service delivery, as well as location sensing inaccuracies for WLAN signal strength based location system. We have performed some initial integration tests with wireless simulators capable of modelling RF reflection and absorption characteristics of different structural materials as well as of the human occupants themselves as they move in the space [15].

Acknowledgements

This work has been partially supported by the Irish Higher Education Authority under the M-Zones programme and by Enterprise Ireland under the PUDECAS [TD 2005 217-A/B] project.

References

1. Czarkowski, M. and Kay, J., (2003). Challenges of Scrutable Adaptivity. In Proceedings of AIED Conference, 11th International Conference on Artificial Intelligence in Education, IOS Press 404 - 407.
2. Yamazaki, T, (2005). Ubiquitous Home: Real-Life Testbed for Home Context-Aware Service. Proceedings of IEEE 1st International Conference on Testbeds and Research Infrastructures for the DEvelopment of NeTworks and COMmunities (Tridentcom 2005).
3. Calvary G., Coutaz, J., Thevenin, D., (2004). A unifying reference framework for the development of plastic user interfaces. In Proceedings of EHCI/DSVIS'04.
4. Dobson, S., Nixon, P. (2004). More principled design of pervasive computing systems. In Proceedings of EHCI/DSVIS'04.
5. Bylund, M., Espinoza, F., (2002). Testing and Demonstrating Context-Aware Services with Quake III Arena. Communications of the ACM, Vol. 45, No 1, pp46-48.

6. Barton, J., Vijayaraghavan. V., (2002). UbiWise, A Ubiquitous Wireless Infrastructure Simulation Environment. HP technical report.
7. Half-Life 2, Valve Corporation (2004). http://www.half-life2.com/
8. eXist, Open Source Native XML Database. http://exist.sourceforge.net/
9. Dobson, S., (2005). Hybridising events and knowledge in an infrastructure for context-adaptive systems. Proceedings of the IJCAI'05 Workshop on Artificial Intelligence and Autonomic Communications.
10. Gonzalez, A. and Ahlers, T., (1999). Context-based representation of intelligent behaviour in training simulations. Transactions of the Society for Computer Simulations International 15(4). March 1999.
11. Henrickson, K. et al, (2002). Modelling context information in pervasive computing systems. Proceedings of Pervasive'02. LNCS 2414. Springer Verlag.
12. Ranganathan, A., Al-Muhtadi, J. and Campbell, R., (2004). Reasoning about uncertain contexts in pervasive computing environments. IEEE Pervasive Computing 3(2). April-June 2004.
13. Stevenson, G., Nixon, P. and Dobson, S., (2005). Towards a reliable wide-are infrastructure for the context-based self-management of communication. Proceedings of the 2nd IFIP Workshop on Autonomic Communications. LNCS. Springer Verlag. 2005. To appear.
14. Dagger, D., Wade, V., Conlan, O., (2004). Developing Active Learning Experiences for Adaptive Personalised eLearning, Third International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems, AH2004, LNCS3137 p55-64
15. O'Neill, E. et al, (2005). A Testbed for Evaluating Human Interaction with Ubiquitous Computing Environments. Proceedings of IEEE 1st International Conference on Testbeds and Research Infrastructures for the DEvelopment of NeTworks and COMmunities (Tridentcom 2005).
16. Kerttula, M. and Tokkonen, T., (2001). The Total User Experience – How to make it Positive in Future Wireless Systems and Services. WWRF Annual Workshop in Paris 2001.
17. Huebscher, M. and McCann, J., (2004). Simulation Model for Self-Adaptive Applications in Pervasive Computing. 2nd International Workshop on Self-Adaptive and Autonomic Computing Systems (SAACS '04).
18. Yamazaki, T., (2005). Ubiquitous home: real-life testbed for home context-aware service. Proceedings of IEEE 1st International Conference on Testbeds and Research Infrastructures for the DEvelopment of NeTworks and COMmunities (Tridentcom 2005).
19. Abowd, G., (2000). Living laboratories: The future computing environments group at the Georgia institute of technology. ACM Conference on Human Factors in Computing Systems, 2000.
20. Nazari Shirehjini, Ali A., Klar, F., (2005). 3DSim: Rapid Prototyping Ambient Intelligence. Joint sOc-EUSAI conference.
21. Seiie Jang, Eun-Jung Ko, and Woontack Woo (2005) Unified User-Centric Context: Who, Where, When, What, How and Why. UbiPCMM: Personalized Context Modelling and Management for Ubicomp Applications.
22. Yoosoo Oh, Woontack Woo (2005). User-centric Integration of Contexts for a Unified Context-aware Application Model. Joint sOc-EUSAI conference.
23. Kenny, A., Lewis, D., O'Sullivan, D., (2006). Interlocutor: Decentralised Infrastructure for Adaptive Interaction. To appear in proceedings of 3rd International Workshop on Managing Ubiquitous Communications and Services (MUCS), 2006.
24. Feeney, K., Lewis, D., Wade V., (2004). Community Based Policy Management for Smart Spaces. Proceedings of 5th IEEE International Workshop on Policies and Distributed Systems and Networks, IEEE, 2004.