The EQUIP Platform:
Bringing Together Physical and Virtual Worlds

Chris Greenhalgh, Shahram Izadi, Tom Rodden, Steve Benford

University of Nottingham
School of Computer Science and IT
Nottingham, NG8 1BB, UK
`{cmg, sxi, tar, sdb}@cs.nott.ac.uk`

Abstract. This paper presents EQUIP, a platform to support the convergence of real and virtual worlds by allowing a diverse set of heterogeneous devices to share information with each other and with software systems such as multi-user virtual worlds. The paper presents an overview of the platform, describes some of its key innovations, and presents some initial examples of the platform in use to develop new arrangements of interactive devices.

1 Introduction

Ubiquitous computing is often characterized in terms of environments that place digital information in physical spaces [14]. The supporting infrastructure often exploits the contextual information provided by the space to manage the interaction between devices, services and users [9]. Similarly, the development of virtual environments and the construction of collaborative virtual environments (CVEs) have used virtual spaces to manage interaction. As part of the Equator project we are exploring the advantages to be gained by allowing a shared virtual environment to be overlaid on top of a shared physical space. A number of key advantages motivate our desire to combine the physical and virtual to support ubiquitous computing:

- The ability to exploit the coextensive virtual world as a ‘behind the scenes’ resource for coordinating and managing devices and interaction in the physical space.
- The opportunity to develop applications that span the physical and digital realms, for example that required for collaboration between field operatives and control-room personnel.
- The chance to support new kinds of interactive experience, combining elements from virtual worlds (e.g. rich media content, high interactivity) with varied modes of access over extended geographical areas and periods of time (e.g. across a city, over a period of days or weeks).

This paper describes the EQUIP platform developed to support the merging of physical and virtual environments. Physical and virtual worlds each have distinct characteristics and place slightly different requirements on the supporting infrastructure. Virtual worlds are characterised by:

- Significant amounts of structured state information (describing the virtual world and the users within it);
- Rapid updates on some data (e.g. user movement and interaction);
- Demanding performance requirements (e.g. for realtime rendering and interaction);
- Tight integration of multiple media (e.g. embedded realtime audio and video streams).

On the other hand, devices placed in the physical world are characterised by:

- Extreme diversity of devices, including diverse computational, storage, input and display capabilities and variations in networking (e.g. bandwidth, reliability, disconnection);
- Coordinated use of collections of devices, both by the same person (at the same or different times) and by different people in the same physical environment.

As well as addressing this range of requirements, EQUIP must also facilitate interfacing to a range of existing and independent software-based systems and services to allow digital devices to access as wide a range of resources as possible.

2 The EQUIP Philosophy and Architecture

The majority of supporting services and protocols for ubiquitous computing have focused on representing and accessing services and the provision of communication mechanisms between devices. This has included the extension of protocols used within World Wide Web [15] and a number of general device discovery mechanisms [13], [11].

Each of these different platforms often embody their own philosophy of approach and place fairly significant demands on devices that wish to link with a supporting infrastructure. In contrast we wish to focus on a low cost lightweight infrastructure that allows a number of devices to share information. Although existing platforms focus on communication between devices some of these protocols already incorporate certain facilities for sharing. For example, Jini incorporates JavaSpaces [10]. A similar model of shared state has been exploited within the GUIDE system [3]. Similar shared state based models exist in collaborative virtual environments such as Dive[12], Massive[6] and Spline[1]. In the development of EQUIP we have focused on allowing a number of devices to communicate across the boundary between real and virtual, thus permitting the convergence of these worlds. EQUIP provides an active distributed real-time data sharing platform that supports the sharing of arbitrary data among heterogeneous distributed applications. This embraces applications written in both C++ and Java and allows applications to make selected information available to other applications through a standard set of simple interfaces. Once internal application data has been made available to the platform updates and alterations to the shared state are propagated to all interested applications using EQUIP. This general arrangement is shown in figure 1.

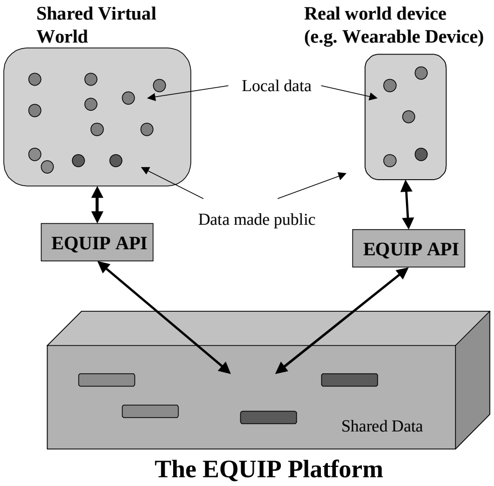

Figure. 1. The general view of the Platform for Shared Interaction

The EQUIP platform extends work on tuple spaces (e.g. Tspaces [7], Limbo [4]) and shared virtual environments to provide an active shared state infrastructure that is equally accessible from large virtual environments and small handheld devices. As well as application state, the platform also allows service information to be shared, enabling a number of flexible context driven resource discovery policies to coexist.

2.1 The EQUIP Shared Data Service

Data sharing in EQUIP is organised into ‘data spaces’ which are named using an URL-like scheme. There can be any number of data spaces and each data space can contain an arbitrary number of data items. Any number of processes can connect to a data space and add to and access the data items in it. Data items are passed between processes using EQUIP’s own serialisation protocol together with dynamic code loading in both C++ and Java.

Consequently all communication is performed indirectly, via the data space, rather than through direct inter-application communication. This same kind of indirect communication is seen in other pattern-driven distribution systems, such as tuple space and generic event distribution systems such as Elvin [5]. This indirection facilitates the construction of loosely coupled distributed systems that are needed to bridge between the physical and digital.

Building on our work with shared virtual environments each data space supports ownership-oriented publishing and updating primitives. In particular, a process can:

- *Add an item to the dataspace* (this is comparable to a tuplespace ‘out’ operation);
- *Update an item in the dataspace*, replacing its current value with a new value; and
- *Delete an item from the dataspace* (this is comparable to a tuplespace ‘in’ operation applied to the unique name of a tuple or data item).

Obtaining information from a data space relies on the use of *patterns*: consumers and users of data use patterns to express their information interests, and the data space routes relevant information to them. Using this approach it is relatively easy to create generic components and to reuse them in a range of applications. Components can be combined by using common data types and by being configured to use matching data spaces and data item naming schemes. Any process connected to a data space can:

- *Add a pattern to the dataspace*. Each pattern is itself a kind of data item (which can be exploited when distributing patterns across the system). These include arrays of prototypical event objects and data items that it may match.
- *Remove a pattern from the dataspace*. This mechanism is not normally required for traditional tuplespace-style patterns, which remove themselves once they are satisfied, however it is required for some of the other pattern types that are supported by EQUIP (see below).

In EQUIP a pattern normally matches an item if: (a) it is of the same or a sub-class of the item; and (b) its fields match by value or are null (which is effectively a wildcard for reference and string fields).

3 Key Techniques

A number of specific techniques give EQUIP its particular strengths in supporting the integration of physical and virtual worlds and allow it to combine the high performance demands of virtual worlds with the communication demands of physical devices. Specific feature include

- ***Support for multiple languages*** with parallel support for C++ and Java. This allows performance-critical components to be implemented in C++ (e.g. 3D renderers) while other components can be implemented in Java (e.g. for rapid development or cross-platform GUIs).
- ***Support for multiple independent data spaces***. This facilitates scalability and load distribution (each data space may run on a different machine); data partitioning and efficiency (pattern matching is scoped within a single data space); and availability and continuity of services (separate data spaces can be restarted independently).
- ***Flexible discovery using specialised data spaces***. Service discovery and activation information is kept distinct from application data spaces in order to maintain stability. These discovery data spaces use a standard set of data types to

indicate available services and to request activation of services for particular applications. This allows service discovery to exploit the full range of pattern matching facilities provided by EQUIP.

- ***Persistent active pattern matching***. Traditional transient pattern matching operations (comparable to tuple-space read and copy-collect) are complemented by a persistent pattern matching operation, the ‘item monitor’. When added to a data space this acts initially like a copy-collect, returning all currently known matching data items. However, it also continues to exist within the system, returning further matches as they arrive, until it is explicitly removed from the data space. Because an item monitor represents an ongoing interest in a certain kind of data it is used to drive data replication and event distribution within the system: an event or data item is only forwarded to a client if a pattern explicitly indicates that they are interested in it. This is similar to Elvin’s use of event filtering, but is also tightly integrated with support for data items.

- ***Context sensitive pattern-matching***. The use of serialisation and dynamic code loading to pass data items between processes means that applications can change pattern matching operations within EQUIP dynamically to support context-sensitive data discovery by using attributes that describe context and location in a specialised pattern matching operation.

- ***An explicit ‘update’ operation***. Allows a data space to be more efficient than the in/out combination required in a normal tuple space, and allows the implicit coherence between successive values of the same data item to be exploited. For example, the system performs intelligent suppression of unreliable updates that are superceded while in transit.

- ***Asynchronous communication between data spaces***. This supports better throughput than a synchronous system, and has also been engineered to recover from temporary disconnection of clients (e.g. in a wireless network). In future versions this will be further developed, and enhanced with additional facilities such as prioritisation of communication on slow or interrupted connections.

The EQUIP platform has been used to support a number of applications that bring together a range of different devices and overlay the real world with a virtual environment. For example, as part of a project called SHAPE we have explored the development of a museum application where the real world is populated by virtual artefacts to be unearthed by visitors using a range of devices to aid their digital archaeology.

4 EQUIP and the SHAPE Museum Demonstrator

In the SHAPE museum demonstrator EQUIP performs an integration role, connecting the physical world – in the form of various input and output devices embedded or mobile within it – to a corresponding virtual world, realised in the MASSIVE-3 collaborative virtual environment (CVE) [2]. This virtual world can be experienced by those in the real world through interfaces on a handheld device (currently a Compaq iPAQ) through an audio landscape (presented via a Laptop PC) and through a special purpose display/projection device called the periscope. Each of these devices present different experiences of the shared environment.

The MASSIVE-3 CVE system uses its own data distribution facilities to construct and maintain a shared 3D audio-graphical virtual world. Two standard EQUIP components provide a general-purpose bridge between EQUIP and MASSIVE-3. These are:

- A ‘Management server’, that can create and render a 3D graphical view of a MASSIVE-3 virtual world. It also conveys the user’s mouse and keyboard input to EQUIP, and can relay information about avatars and objects in the virtual world to EQUIP.
- An ‘Entity Generator’, that can create and animate 3D virtual objects in MASSIVE-3 worlds according to data in an EQUIP dataspace.

Figure 2 shows how these components are used in the SHAPE museum demonstrator in order to:

- give the outdoor wireless user an avatar (virtual body) within the MASSIVE-3 virtual world (from a GPS client to the Entity Generator); and
- give the outdoor wireless user information about objects in the virtual world such as their proximity (from the management server to a simple 2D client running on the PDA).

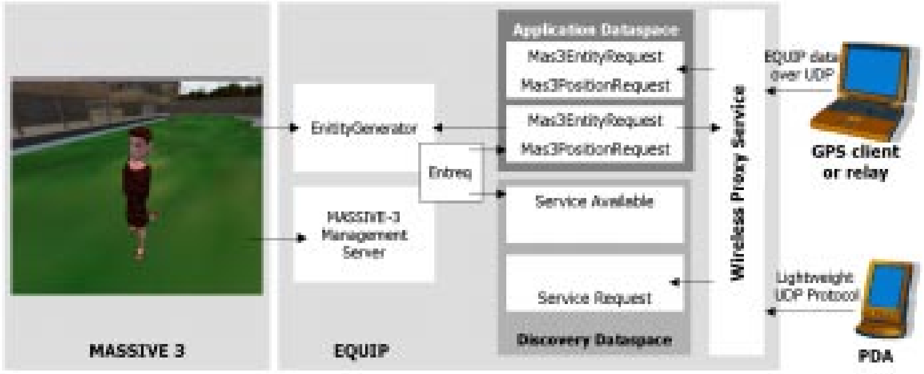

Figure. 2. Supporting the wireless user in the SHAPE museum demonstrator

Access for intermittently connected wireless devices is provided through a proxy service. This arrangement allows clients to send and receive EQUIP data items using the connectionless UDP transport protocol. If the proxy receives an EQUIP item that does not appear within the application dataspace it will add the item on behalf of the client. Further updates can also be issued through the proxy, as and when relevant data items arrive from the client. When experimenting with this arrangement we found that low memory devices could not receive and unmarshal large numbers of events at a reasonable rate. As a result, an additional lightweight ASCI protocol was developed to communicate between the proxy and low resource devices.

5 Conclusions and Future Work

In this paper we have presented EQUIP, a platform to allow information sharing and integration between the physical world – represented by heterogeneous computing and interaction devices – and virtual worlds. The currently developed system allows users to explore the real world and virtual world together and for their interactions to be simultaneously available from within both these worlds.

We are currently looking to exploit the semantics embodied within the real world to best support and promote contextual awareness. This work builds upon our previous exploration of spatial models within virtual environments [6] and will allow us to exploit the spatial arrangement of users and devices in both the real and virtual world in order to drive decisions surrounding resource discovery and communication.

For further informaton on EQUIP, including downloads, see **http://www.equator.ac.uk/equip/**.

6 Acknowledgements

EQUATOR is funded the UK's Engineering and Physical Sciences Research Council. We would like to express particular thanks to our collaborators in the SHAPE workshop in which this demonstration was developed, in particular John Bowers, Paul Chandler, Luigina Ciolfi, Martin Flintham, Mike Fraser, Tony Hall, Sten Olof Hellström, Holger Schnädelbach, Ian Taylor.

References

1. Barrus, J.W., Waters, R.C., Anderson, D.B..: Locales: Supporting Lange Multiuser Virtual Environments, in IEEE Computer Graphics and Applications (1996) 50-57
2. Benford, S., Bowers, J., Chandler, P., Ciolfi, L., Flintham, M., Fraser, M., Greenhalgh, C., Hall, T., Hellström, S.O., Izadi, S., Rodden, T., Schnädelbach, H., Taylor, I.: Unearthing virtual history: using diverse interfaces to reveal hidden virtual worlds (Submitted to UBICOMP 2001)
3. Cheverst, K., Davies, N., Mitchell, K., Friday, A.: The Role of Connectivity in Supporting Context-Sensitive Applications, Lecture Notes in Computer Science, No. 1707, Springer-Verlag, Heidelberg, Germany (1999) 193-207
4. Davies, N. , Wade, S.P., Friday, A., Blair, G.S.: Limbo: A Tuple Space Based Platform for Adaptive Mobile Applications, Proc. Int. Conference on Open Distributed Processing/Distributed Platforms (ICODP/ICDP '97), Toronto, Canada (1997) 291-302

5. Fitzpatrick, G., Kaplan, S. Mansfield, T., Arnold D., Segall B.: Supporting public availability and accessibility with Elvin: Experiences and reflections, Computer Supported Cooperative Work: the Journal of Collaborative Computing (To Appear)
6. Greenhalgh, C., Purbrick J., Snowdon, D.: Inside MASSIVE-3: Flexible Support for Data Consistency and World Structuring, Proc. of ACM Conference on Collaborative Virtual Environments (CVE2000), San Francisco, USA (2000) 119-127
7. IBM Corp.: TSpaces Technology, http://www.almaden.ibm.com/cs/TSpaces/
8. Krebs, A., Dorohonceanu, B., Marsic, I.: Collaboration using Heterogeneous Devices - from 3D Workstations to PDAs, Proc. of the IASTED International Conference on Internet and Multimedia Systems and Applications (IMSA'2000), Las Vegas, USA, (2000) 309-313
9. Streitz, N.A., Geißler, J., Holmer, T., Konomi, S., Müller-Tomfelde, C., Reischl, W., Rexroth, P., Seitz, P., Steinmetz, R.: i-LAND: An interactive Landscape for Creativity and Innovation, Proc. ACM Conference on Human Factors in Computing Systems (CHI '99) , Pennsylvania, USA, (1999) 120-127
10. Sun Microsystems Inc.: JavaSpaces Technology, http://java.sun.com/products/javaspaces
11. Sun Microsystems Inc.: Jini connection technology, http://www.sun.com/jini/
12. Swedish Institute of Computer Science (SICS): DIVE - A Toolkit for Distributed VR Applications, http://www.sics.se/dce/dive
13. Universal Plug and Play Forum: Universal Plug and Play (UPnP), http://www.upnp.org/
14. Want, R., Schilit, B.N., Adams, N.I., Gold, R., Petersen, K., Goldberg, D., Ellis, J.R., Weiser, M.: An overview of the ParcTab ubiquitous computing experiment. IEEE Personal Communications Magazine (1995) 28-43
15. World Wide Web Consortium.: eXtensible Markup Language, http://www.w3.org/XML/