

Enterprise-Enabled Ubiquitous Workspaces

Terence Blackburn¹

Rudi Vernik^{1,2}

Damien Bright¹

¹School of Computer and Information Science
University of South Australia
Mawson Lakes Blvd
Mawson Lakes South Australia 5095

²Defence Science and Technology
P O Box 1500
Edinburgh South Australia 5111

terence@cs.unisa.edu.au
rudi.vernik@dsto.defence.gov.au
damien.bright@unisa.edu.au

Abstract

This paper reports on a case study that was conducted to investigate infrastructure requirements for enterprise-enabled, ubiquitous workspaces of the future. We first set the context for future work environments by reviewing current research activities in the areas of intelligent interactive rooms and enterprise computing. The work reported in this paper involved integrating MIT's Metaglu multi-agent, ubiquitous computing infrastructure with an enterprise bus and workflow capability derived from DSTC's Open Distributed Services Infrastructure (ODSI), a framework for future enterprise computing systems.

Keywords: future workspaces, intelligent environments, ubiquitous computing, software infrastructures.

1 Introduction

This paper examines ubiquitous computing infrastructure from the perspective of next generation work environments. One of the driving factors behind the current trend to pervasiveness is the need to evolve from the use of workstations as the primary human computing interface to environmental or workspace interfaces. In these future environments, keyboards and mice, the main human-computer interfaces for the last couple of decades, will be replaced by more human-centric approaches. This will be facilitated by the move from being "cabled and connected" to environments where "wireless and wandering" will be the norm. These trends will allow humans to be decoupled from computers. The new interfaces will be more intuitive, context aware and adaptive. These future workspaces will use multiple display devices and support interactions through various pointing instruments, the use of gestures, speech, and haptic interfaces. Computing applications will move from being hosted on workstations to being hosted within physical environments such as rooms and even buildings.

This paper focuses on the challenge of providing the underlying architectures and software infrastructure to accommodate these types of future environments and associated workspace applications. As with individual workstations, which require an operating system to coordinate and run applications, ubiquitous computing applications will require an underlying operating environment. Research is well underway in various laboratories, e.g. [1, 2], to more fully understand what this underlying infrastructure should be. In this paper, we begin to explore how the infrastructure for these interactive environments might be extended and integrated into broader enterprise-level solutions. Some of the characteristics that need to be considered in this respect include the use of workflow for orchestrating and coordinating work activities, distribution, information management and integration with various enterprise services such as ontology servers, data repositories, and web services.

Section 2 sets the context for the paper by discussing the desired characteristics of future workspaces and by providing examples of the types of activities that will need to be supported in these types of environments. We also examine the infrastructure requirements and the likely components that will form the foundations for these new work environments.

Section 3 discusses four separate approaches that begin to address the infrastructure requirements for future workspaces. These include commercial initiatives such as Jini as well as research work that has been undertaken to provide the ubiquitous computing infrastructure for intelligent and interactive rooms. We also discuss research being undertaken to provide for future enterprise computing infrastructure.

Section 4 describes a case study where the Metaglu [1] multi-agent ubiquitous computing infrastructure, which forms the basis of the MIT Intelligent room, is integrated with the ODSI enterprise infrastructure from DSTC [9]. ODSI implements the concept of an enterprise bus that allows the integration and coordination of a range of

services across an enterprise. Of particular interest is ODSI's ability to use enterprise knowledge (in the form of organizational structures, context, and processes) to orchestrate and coordinate work activities and allow flexible access to, and use of, various enterprise applications and services. We use a simple meeting scenario as the driver for the case study to show the value of using an enterprise-enabled, ubiquitous computing workspace. We also report on the results of integrating an agent-based ubiquitous computing infrastructure with a publish-subscribe, enterprise bus implementation.

Section five concludes by summarizing the key results to date and discusses future work.

2 Future Work Environments

Ubiquitous and pervasive computing initiatives are providing the components for augmenting physical workspaces with devices that will allow people to interact more effectively with each other and with technology [12]. Future workspaces will be augmented with various interactive display devices, personal information appliances, speech, and natural language interfaces (see Figure 1). Approaches such as augmented reality, virtual presence and conversational agents will help transform the way in which people use information to support individual and collaborative workspace activities.

Several research projects are currently underway to look at how these devices and technologies can be integrated and managed as part of a future workspace. For example, MIT's Intelligent Room integrates the use of speech, gestures and natural language interfaces to help users interact with workspace applications [14]. Context plays an important part in enabling the room to be tailored to the needs of workspace participants. Stanford's Interactive Room [2] looks at the management of information in multidisplay environments and allows user interaction between personal handheld devices and large interactive displays. Each of these initiatives has as its core an infrastructure component that addresses important requirements such as display management, interaction management and context awareness.

Related work has been done to look at how these future workspaces might support collaborative work activities. For example, Stanford's Interactive Room has been used to support architectural design activities while MIT's Intelligent Room supports applications such as the Meeting Manager [13]. However, a review of the literature suggests that relatively little work has been done in looking at how these types of workspaces might be deployed as part of an overall enterprise environment. For example, we are interested in future command and intelligence environments where several such distributed workspaces might be used to collectively plan and execute military missions. In such situations, the work within individual workspaces needs to be orchestrated and synchronised with other workspaces so that overall enterprise goals can be realised.

The types of activities that need to be supported in individual workspaces include those associated with analysis, planning, briefing and execution. Each activity generates and requires access to information hence easy access to enterprise-wide data sources is critical. Workspaces also need to adapt readily to the type of activity being undertaken and the people involved; they need to be enterprise aware [16]. Moreover, activities need to be orchestrated within individual workspaces to ensure that goals are met within a required timeframe.

As discussed above, research into intelligent and interactive rooms identifies a number of desirable characteristics that need to be supported through ubiquitous computing infrastructure. These include display management, interaction management, mobility, and context awareness. In taking an enterprise perspective on the use of these future workspaces, there are several additional characteristics that we believe need to be addressed. These include distribution, process orchestration and synchronisation, enterprise awareness, information management, and security. As discussed in Section 3, these aspects are being considered in various research projects that look at future enterprise computing architectures and infrastructure. Our challenge is to investigate infrastructure for future work environments that exhibit a combination of desirable characteristics from both ubiquitous computing and enterprise computing domains.

3 Overview of Candidate Architectures and Infrastructures

Several candidate architectures and software infrastructures have been identified as providing at least some of the desirable characteristics for the type of enterprise-enabled workspaces outlined above. In this section we draw from the four examples to highlight how these characteristics have been accommodated and implemented.

3.1 Metaglu

Metaglu Version 0.5 [1] is a multi-agent based system that provides computational glue for large groups of collaborative software agents and device controllers, such as those used in the Intelligent Room at MIT Cambridge, MA [11]. It provides communication and discovery services and enables users to interact, subject to access control, with software and data from any space. It also arbitrates among applications competing for resources [15].

Metaglu is implemented in Java, replacing the Remote Method Invocation (RMI) mechanism with one that allows dynamic reconfiguration and reconnection so that agents can invisibly resume previously established, but broken connections. It has support for self-dissemination through a process called glue spreading where new agents can be created throughout the infrastructure as required.

Metaglug provides the support for managing systems of interactive, distributed computations i.e. those in which different components run asynchronously on a heterogeneous collection of networked computers. It provides high-level support for writing groups of interacting software agents and automates the connection and distribution of these agents according to their computational requirements.

Metaglug separates software that acts on behalf of users from software that controls spaces. Societies of adaptive agents can be attached to any entity in an intelligent room and can move with the entity eg. camera and microphone agents connected to someone who walks across the room would move from one set of device controllers to another located near the person's new location.

3.2 Intelligent Room Operating System (iROS)

The Interactive Workspaces project [2, 3] at Stanford University is developing a high level architecture for organising multi-person, multi-modal interactions in an integrated space that combines multiple computer systems. It has wall-sized interactive murals, a "collaborative table" with a bottom-projected display and support for large-scale collaborative applications. The infrastructure also supports wireless networking and allows the integration of handheld devices like PDAs and palm top computers.

The infrastructure supporting the intelligent room is based on iROS, the Intelligent Room Operating System. It uses a tuple space environment and incorporates the Tspaces server from IBM [17]. Instead of a traditional event queue, the architecture utilises a central event heap abstraction, called the Event Heap [22], as the main communication mechanism among the software components. The event queue metaphor, which works well for a single user sitting in front of a single computer using a GUI, breaks down in an interactive workspace with multiple users all using common hardware and software applications. The Event Heap is a mechanism by which multiple users, machines and applications can all simultaneously interact as consumers and generators of system events. Because the events are tuples in a tuple-space [18], analogous to a shared blackboard, publishing and subscribing to the tuplespace is sufficient to participate in the room infrastructure while allowing components to remain very loosely coupled.

A key requirement for the interactive workspace is a software infrastructure that allows users with mobile computing devices, including laptops and PDA's, to seamlessly enter and leave the interactive workspace. While in the workspace, they can use their devices to interact with the current application and/or control various hardware devices in the room.

The Event Heap is designed to be flexible, robust to the failure of individual nodes and lightweight. However, issues like lack of security limit its practical applications in an enterprise network. Events auto-age and are garbage

collected, so events can be posted without knowing whether or not they will be consumed.

3.3 ODSI

The Open Distributed Services Infrastructure (ODSI) is a framework that supports the collaboration of enterprise resources to meet the challenge of an adapting enterprise [9]. ODSI concentrates on collaborative services rather than data-driven applications and hence maintains an enterprise perspective rather than a software focus.

It provides a service integration environment designed to combine enterprise-modeling techniques, which incorporate enterprise information such as policy, role and component descriptions, with lightweight workflow and dynamic service integration technology. It provides lifecycle management for distributed software services through a peer infrastructure and supports their collaboration to perform complex business processes [19].

ODSI utilises the Breeze workflow engine for resource coordination and process orchestration and an adaptive enterprise bus based on Elvin Version 3 [8, 20], a content-based routing system. Elvin is a lightweight notification service. Producers detect events, changes in the state of an object, and notify Elvin, which in turn passes the notification to any clients that have registered an interest with a subscription expression. The difference between Elvin and most messaging systems is that Elvin's actions are motivated by the message content rather than an address or topic name. The advantage of this difficult to implement approach is that it decouples producers and consumers and thus promotes system evolution, integration and client homogeneity [10].

3.4 Jini

Jini network technology [4, 5] from Sun Microsystems is an open architecture that enables developers to create a federation of highly adaptive, network-centric entities, devices or services. Jini technology, which is written in Java, has been designed to build networks that are scalable, evolvable and flexible. These types of networks are those that are typically required in dynamic, distributed computing environments.

The overall goal is to turn the network into a flexible, easily administered tool with which resources can be found by any device or user. Resources can be implemented as hardware devices, software programs or a combination of the two. One focus of the system is to add a dynamic element to the network that better reflects the nature of a workgroup by enabling the ability to add and delete services flexibly without stopping and starting the system.

4 Case Study

Each of the four infrastructures has a specific focus and each uses a different technological approach to provide support for a particular type of ubiquitous computing

environment. As a result, no individual framework addresses the broad range of enterprise needs for the various domains that may one day deploy the technologies. This project has adopted the enterprise perspective and has focused on integrating two of the architectures in an effort to extend the range of resources available to clients in future work environments and to promote a higher level of collaboration and workplace adaptivity.

The aim of the case study was to experiment with the application and integration of two infrastructures: MIT's Metaglué and DSTC's ODSI. Metaglué was chosen for its strength in integrating and managing hardware and software devices in a workspace environment. ODSI was selected for providing an enterprise focus primarily, in this case, through the Breeze workflow engine and the Elvin-based enterprise bus.

4.1 LiveSpace Environment

A future workspace environment called LiveSpace has been configured with off the shelf technology to support current and future research activities. It has been set up with computers that host the Linux operating system for running the Breeze workflow engine, a Microsoft Windows 2000 environment for supporting the web displays and a Solaris UNIX environment, which hosts the Elvin [8] messaging server. The workspace includes a 1.5 x 1.2 meter touch-sensitive Smart Board operated through a front mounted projector. IBM's viaVoice speech recognition software processes the voice input and provides the speech-to-text functionality. R-JAX, a program from Stryon, processes event generated messages and passes them to MS Windows applications. These applications can be run collaboratively on the smart board displays or on personal notepad computers, which can communicate via an 802.11 wireless LAN. X10 device controllers are used to control various LiveSpace devices such as the projectors, lights, and coffee machine.

4.2 Scenario

An augmented dedicated meeting space was selected as being a relevant scenario for these initial investigations. This scenario was considered as being representative of the types of workspace activities that we might be interested in for future command and intelligence environments. The aim was to extend the meeting paradigm to incorporate context awareness, adaptability, collaboration and workflow process orchestration.

4.3 Set up

Figure 1 shows the set up used for the case study. The resulting composite workspace infrastructure is best explained in terms of the simple meeting scenario that was used for the case study.

- The chair starts the meeting with a command spoken into the microphone.
- viaVoice converts the words into text, which is passed in turn to a Metaglué speech agent, the Metaglué Notifier (the agent responsible for sending event notifications to subscribers), the "Elvin client" agent and the Elvin messaging server, which sends it on to an ODSI peer subscriber.
- The Breeze workflow engine, the ODSI peer, processes the message and starts the appropriate workflow.
- The first workflow turns on the coffee machine, powers on the SmartBoard projector, loads a browser window and displays the opening page for the meeting.
- Elvin sends these instructions back to the peer/agent, which relays them to the Metaglué Notifier, which in turn passes them to the appropriate agents.
- The first message is sent to a device controller that turns on the coffee maker and the second message is processed by a device controller that activates the SmartBoard.
- A wrapper agent for the web browser consumes the third message and passes it on to the R-JAX program, which starts the browser on the Smart Board and sends a URL to the browser, the address of which was in the fourth message.

This is the first workflow action for the meeting. The second and subsequent sub-workflows also involve voice-activated commands and include calling for further pages to be displayed on the Smart Board. The participants can also interact with the Smart Board and any notes or diagrams etc can be saved and stored as a file.

4.4 Results

Preliminary results demonstrated that it was possible to aggregate and blend the strengths and benefits of the individual software infrastructures to extend the interactive workspace with enterprise-enabled characteristics. ODSI has a strong process orchestration focus and was able to leverage the adaptive, device management features of Metaglué while Metaglué has strong context awareness and collaborative features that were enhanced by integrating the ODSI enterprise workflow scenarios. One of the problems encountered when integrating infrastructures of this type is creating the bridge that facilitates bi-directional messaging or event notification across different architectural paradigms. In this instance there were three environments to span: ODSI a publish-subscribe based infrastructure, the Metaglué multi agent infrastructure and Windows. A number of products have addressed the issue of bridging from a Java environment (e.g. Metaglué) to Microsoft Windows for application integration and manipulation. J-Integra, a product from Intrinsyc Software [7], allows COM components to be accessed from the Java environment. This solution exposes the methods available in the target application and converts the target data types

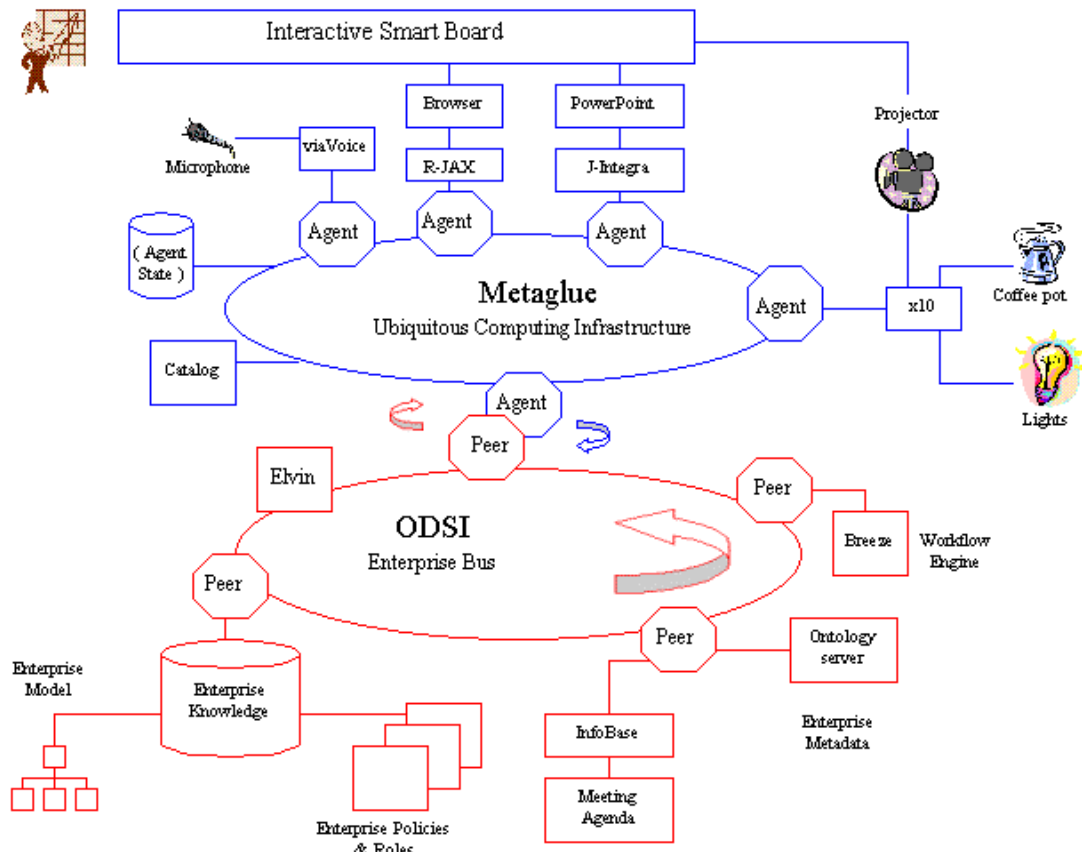


Figure 1 Case study set up showing the integration of the ODSI and Metaglué infrastructures

into Java data types. However there were initial problems with resources not being disposed of correctly and a second solution was chosen and implemented; R-JAX [6], from Stryon. This product did not seem to be as flexible as J-Integra but it was very easy to install and use. It provided an intuitive interface to the Microsoft application methods.

A second bridge was required to connect the Elvin messaging server in ODSI to the Metaglué architecture. Fortunately, Elvin clients can be written in a number of languages including Java. The resulting hybrid architecture was not technically difficult to create and implement. One of the reasons for this is the platform independent nature of the Java programming language. When a Virtual Machine is running in any environment, method calls from an object on one machine can be made pervasively to an object in a second VM on another computer, straddling any firewall friendly networks. In this way, a Metaglué agent was able to communicate concurrently with the Elvin architecture and the Metaglué agent system. It would have been more robust to implement the Elvin client as two separate agents. One would subscribe to the Metaglué Notifier and publish to the Elvin server and the second agent would subscribe to the Elvin server and publish to the Metaglué Notifier. In this

example scenario, the Elvin client/agent fulfilled both of the roles of producer and consumer to both architectures.

One of the useful extended features of Metaglué is the `relies_on()` method. This allows more than one agent to attach to a particular agent controlling a device. The `relies_on()` method is used instead of creating a new instance of the agent object. In this way, if one agent changes the device's state, all agents are notified. This would not happen if each agent was associated with a separate instance of the device controller agent.

More than one agent can interact with a device. For example, if two or more agents, representing entities like applications or people, wish to send separate URLs to the browser, each can communicate individually. In this way collaboration is fostered by group interaction with the application.

A shortcoming that will affect some domains is the lack of security. Elvin is a content based messaging service and any client can subscribe to any service in a federation of Elvin servers. The situation is similar with Metaglué and this can allow unwanted access to any enterprise resources exposed within the architecture.

Another shortcoming in this type of lightweight environment is the lack of message persistence. It may be important in some situations for notification to be

received regarding the success or failure of an event. Both ODSI and Metagluue have a built in mechanism to partially address the issue but their implementations are quite different.

In addition, Metagluue has the problem of no redundancy for the Catalog, the central agent registration mechanism. If the Catalog crashes, all agent state is lost. However, if a connection between agents is severed for some reason, individual agents may try and reconnect to the agent or device that they were connected to. They may try for a certain period of time or a certain number of times before restarting the service with another agent.

The end result is a change to a more productive methodology for implementing various units of work. This is manifested in faster access to relevant information, facilitation of group interaction involved in the decision-making process and more efficient dissemination of context appropriate information to the enterprise. It also provided a set of smarter interfaces to the enterprise.

5 Conclusions and Future Work

This paper reports on a case study that was conducted to investigate infrastructure requirements for enterprise-enabled, ubiquitous workspaces of the future. We have experimented with the integration of two dissimilar but complementary infrastructures for the purpose of creating interactive workspaces that interface with and draw on the services available through enterprise computing infrastructure. Preliminary results suggest that there is merit in pursuing this avenue of research. The resulting integrated framework highlighted several potential weaknesses. For example, more work is required to integrate the respective failure mechanisms to create a more manageable and predictable architecture. Issues of security need to be investigated for these types of environments.

Work will continue on further defining and validating requirements for architectures and software infrastructure for future work environments of the type outlined in this paper. The LiveSpace environment will be enhanced to support a greater range of environmental interfaces such as interactive display devices, augmented reality interfaces, and dialog interfaces. Experiments will be conducted with Stanford's iROS to investigate the use of tuples spaces and event heaps for enhanced display and interaction management in multidisplay, mulit-user environments.

Further case studies will be undertaken to address enterprise-wide aspects related to the deployment and use of interactive workspaces. Of specific interest will be issues related to the orchestration and synchronization of workspace activities, information management and augmentation resulting from the inclusion of additional enterprise services such as

simulation. Case studies will be conducted in the domains of future military command and intelligence environments, software production enterprises and medical domains.

6 References

- [1] PHILLIPS, B. (1999) Metagluue: A Programming Language for Multi-Agent Systems. M.Eng. Thesis. MIT, Cambridge, MA.
<http://www.ai.mit.edu/projects/iroom/publications/brenton-thesis.pdf>
- [2] Web site for Stanford Interactive Workspaces project,
<http://www.graphics.stanford.edu/projects/iwork/> visited June, 2002
- [3] WINOGRAD, (1999) T. Towards a Human-Centered Interaction Architecture. *Working paper for Stanford project on Interactive Workspaces*
<http://graphics.stanford.edu/projects/iwork/papers/humcent/index.html>
- [4] Web site home page for Jini.
<http://www.sun.com/software/jini/> visited June 2002
- [5] The Jini Page, JGuru Web site.
<http://www.jguru.com/faq/Jini> visited June, 2002
- [6] Web site home page for Stryon.
<http://www.stryon.com/> visited June, 2002
- [7] Web site home page for Intrinsic.
<http://www.intrinsic.com/> visited June, 2002
- [8] Web site home page for Elvin.
<http://elvin.dstc.edu.au/> last accessed June 28, 2002
- [9] BOND, A. (2001) ODSI: Enterprise Service Coordination.
www.dstc.edu.au/Research/Projects/Integrator/documents/DOA-abstract.htm visited June, 2002
- [10] FITZPATRICK, G., MANSFIELD, T., KAPLAN, S., ARNOLD, D., PHELPS, T. and SEGALL, B. (1999) Augmenting the Workaday World with Elvin. In *Proceedings of ECSCW'99*, Copenhagen, 431-451, Kluwer Academic Publishers.
- [11] BROOKS, R., with contributions from COEN, M., DANG, D., DEBONET, J., KRAMER, J., LOZANO-PEREZ, T., MELLOR, J., POOK, P., STAUFFER, C., STEIN, L., TORRANCE, M., and WESSLER, M. (1997) The Intelligent Room Project. In *Proceedings of the Second International Cognitive Technology Conference (CT'97)*, Aizu, Japan
- [12] HANSSENS, N., KULKARNI, A., RATTAPOOM TUCHINDA, R. and HORTON, T. (2002) *Building Agent-Based Intelligent Workspaces*. Proceedings of

The International Workshop on Agents for Business Automation. Las Vegas, NV, 2002.

[13] Oh, A., TUCHINDA, R. and Wu, L. (2001) *MeetingManager: A Collaborative Tool in the Intelligent Room*. Student Oxygen Workshop, Cambridge, MA, 2001

[14] Oh, A., Fox, H., Van Kleek, M., Adler, A., GAJOS, K., Morency, L. and DARRELL, T. (2002). *Evaluating Look-to-Talk: A Gaze-Aware Interface in a Collaborative Environment*. Proceedings of CHI 2002. Minneapolis, MN, 2002.

[15] GAJOS.K. (2001) *Rascal - a Resource Manager For Multi Agent Systems In Smart Spaces*. Proceedings of CEEMAS'01, Kraków, Poland, 2001.

[16] HAMMOND, T., GAJOS, K., Davis, R. and SHROBE H. (2002) *An Agent-Based System For Capturing And Indexing Software Design Meetings*. Proceedings of the International Workshop On Agents in Design. Cambridge, MA, August 2002.

[17] Home page for IBM's alphaWorks Tspaces server, 2001,
<http://www.alphaworks.ibm.com/tech/tspaces>

[18] (Multiple papers and authors), 1996, *Linda Group Technical Reports*, <http://www.cs.yale.edu/Linda/tech-reports.html>

[19] BOND, A. (2001) *Service Composition for Enterprise Programming* Working Conference on Complex and Dynamic Systems Architecture 2001

[20] B. Segall, D. Arnold, J. Boot, M. Henderson and T. Phelps, 2000, *Content Based Routing with Elvin4*, CRC for Enterprise Distributed Systems Technology (DSTC), The University of Queensland, St Lucia, 4072, Australia
<http://elvin.dstc.edu.au/doc/papers/auug2k/auug2k.pdf>

[21] JOHANSON, B., FOX, A., HANRAHAN, P. and WINOGRAD T. (2000) *Integrating Information Appliances into an Interactive Workspace* IEEE Computer Graphics and Applications, May/June 2000

[22] JOHANSON, B., FOX, A., HANRAHAN, P. and WINOGRAD T. (2000) *The Event Heap: An Enabling Infrastructure for Interactive Workspaces* CS Tech Report CS 2000-02