Facilitating Ubiquitous interaction using Intelligent Agents

Abey Campbell, Rem Collier, Mauro Dragone, Levent Görgü, Thomas Holz,
Michael J. O'Grady, G.M.P. O'Hare, Antonella Sassu, John Stafford

Abstract Facilitating intuitive interaction is a prerequisite for the ubiquitous computing paradigm in all its manifestations. How to achieve such interaction in practice remains an open question. Such interfaces must be perceived as being intuitive across a variety of contexts, including those of the hosting devices. Indeed, the heterogeneity of the device population raises significant challenges. While individual devices and the interaction modalities supported by, each satisfy the requirements of individual domains, integrating diverse devices such that the user experiences is perceived as consistent and intuitive is problematic. This chapter discusses and illustrates how intelligent agents may be harnessed for integrating a range of diverse interface and interaction modalities such that the ubiquitous user interface concept may be validated.

1 Introduction

Interaction defines the computing experience in many instances. A perceived deficiency in interface design or interaction modality will have a negative impact on how software is perceived; any deficiency in these aspects cannot be compensated for through creativity and ingenuity. In such situations, end-users will indeed as it were, judge the book by its cover, and, in situations where end-users can exercise

Abey Campbell, Mauro Dragone, Levent Görgü, Thomas Holz, Michael J. O'Grady, G.M.P. O'Hare, Antonella Sassu
CLARITY:Centre for Sensor Web Technology, University College Dublin, Belfield, Dublin 4, Ireland, e-mail: mauro.dragone@ucd.ie

Rem Collier
School of Computer Science & Informatics, University College Dublin, Belfield, Dublin 4, Ireland, e-mail: rem.collier@ucd.ie

John Stafford
Nova UCD, Belfield, Dublin 4, Ireland, e-mail: john.stafford@nova.ucd.ie

choice, such users will refuse to use, and will certainly not pay, software that is considered unusable. This emphasis on usability transcends all end-user computing paradigms, including ubiquitous computing.

Historically, ubiquitous computing has envisaged interfaces that are seamlessly embedded in a wide variety of artefacts dispersed throughout the environment. Such interfaces should be intuitive, through it has never been documented how an interface might measure up to this characteristic, at least objectively. Ambient Intelligence (AmI) advocates Intelligent User Interfaces as a means of managing the diversity inherent in AmI, and implicitly ubiquitous, computing systems ; however, before the merits of any approach can even begin to be analysed, the issue of managing interface diversity needs to be addressed at a more basic level. Interface designers have an unparallel range of interface technologies and interaction modalities to chose from. But the issue of meaningful integration to realise practical ubiquitous interfaces remains to be overcome.

In this chapter, the potential of intelligent agents as a software architecture for integrating and managing the diverse range of technologies essential to ubiquitous interfaces is explored. Prior to considering the agent approach, it is instructive to reflect on how a ubiquitous interface might be harnessed in a real life scenario.

1.1 Ubiquitous Interfaces: A Futuristic Scenario

To aid in the discussion about benefits and abilities that agency can bring to Human-Computer Interaction (HCI), a futuristic scenario about the day in the life of a Computer Science student named Greg will be given first. Throughout this chapter, the scenario will not only aid in illuminating the possibilities of agents mediating Human-Computer Interaction but will also serve as a reference point for already implemented proof of concept projects.

Greg starts his day like most others when his alarm clock wakes him at 8am, having been set by his personal agent the night before based on his class schedule for the day. Greg recently bought a Brain-Computer Interface (BCI) that he is still learning to use effectively. His agent knows of Greg's goal to master the interface and reminds him to complete a training session. The training consists of an Augmented Reality (AR) game in which Greg controls his agent using the interface. Greg asks his agent to find a friend that is up for playing a game with him. His agent informs him that his friend Michael, who lives in America, is still up and looking for a game as well and both their agents agree to setup a game. Greg has an AR Head-Mounted Display (HMD) that allows him to see the models and game world in front of him placed in the real world. Michael, who does not have a HMD, simply uses a normal web camera to see the same game world on his monitor. So sitting at his breakfast table, Greg places a paper marker in the centre of the table to place the virtual game field. Wearing his AR HMD, Greg can see his agent appear as a 3D model in the centre of the table, and in front of it is placed Michael's. The game aims to challenge the user by giving one user the ability to get their virtual agent to throw axes at the opponent. Using the BCI, the other player has to dodge the attack. If the player is relaxed, his agent will move itself to the left, if the player concentrates, it will move to the right — a simple game that helps the user train to use the interface. Greg notices that the model embodied by his agent has a frown, reminding him that he lost this game the last few times. However, after playing a few rounds, Greg finally beats Michael for the first time and gets ready to head to college. At this point, the agent migrates from his home network to his phone, knowing he is about to leave based on the current time and the fact that he has requested to shut down his home computer.

Still feeling a little hungry, Greg eats a chocolate bar before he leaves. As he is currently on a diet, he informs his agent of his additional food intake through his mobile phone. His agent is in charge of his diet and suggests a mobile game to help him burn those additional calories. On his mobile phone, a 2D version of his agent appear and outlines a game in which Greg has to find a hidden treasure chest using a virtual treasure map of the real world around him. The agent has set the final destination as Greg's college. Greg is also wearing one of his new shirts which contains a heart monitor. As the game starts, Greg is shown an overhead map pointing him to the end of the road. His agent appears on the overhead map, representing a competing computer controlled character. From the map, Greg can see his competitor has already taken the lead. Greg runs ahead looking occasionally at his mobile phone. He reaches the first destination point and is told that in order to get to the next location, he has to take a left at the palm tree. Greg knows that there is no palm tree in the physical world, so he picks up his phone and takes a picture of the junction ahead of him. The photo taken on his camera is augmented with an image of the virtual world. Through this magic mirror he sees a palm tree placed at the corner of the road. Knowing his new direction he quickly heads down the road. As Greg's agent senses that his heartbeat has not increased sufficiently for a good workout, it sets the next target on top of a hill just outside the college. After this location, Greg finally arrives in college, sweating a little as he had to outrun his virtual opponent. The final destination is placed just outside his Department, and Greg takes a picture of the front door believing this to be the location of the virtual bounty. The image returned contains only simple text, however, telling him the treasure is "hidden in the place that others would just waste." Looking around, he notices a waste bin and takes another picture. The image appears with a virtual treasure chest replacing the waste bin, and his agent congratulates him.

Entering the Department, Greg is greeted by Peter, his lab partner for the ubiquitous robotics course. Peter asks if Greg has brought the test robot today, as they have been challenged to a friendly competition. Greg forgot the robot and Peter points out that they could use the backup robot. It's an older model with sensors that are not as accurate as the test robot Greg had been given previously. Greg had been coding the test robot at home using his agent. Luckily for him, the agent keeps a memory of all its different actuators and perceptors for each body it has migrated too. The agent migrates to the new robot, and Greg gets an AR HMD in order to visualise the agent's mental state within the robot. Peter has written new code for his agent and it wirelessly communicates with Greg's agent on the robot to transfer the new abilities to Greg's agent. In the AR world that Greg sees through his HMD, he can see that his agent has adapted to its new body and that his agent's 3D model has been placed on top of the robot. Seeing the virtual model's eyes allows Greg to know where the agent is looking. This AR view also shows the agent's beliefs of the locations of its opponent and ball as geometric objects. As the sensors are not as good as the original test robot, Greg notices they are a little larger than the real world objects he previously tested it on. Greg is in charge of writing the machine vision code which, combined with Peter's code for the sonar on the robot, allows the agent to identify and quickly move to get the ball. Using its ability to sense the world around it, the agent competes against a rival robot to play a game of soccer. Greg's agent, with the help of Peter's additional code, wins and Greg transfers his agent back to his phone, including the new code written by Peter for the next time they compete. After a successful day in college, Greg returns home playing his AR game searching for a dragon's treasure chest. In the morning, Greg starts playing a game again against Michael but this time he notices that the 3D Model of his Agent is smiling.

This futuristic scenario demonstrates the ideas how Ubiquitous interaction with multiple devices could be facilitated using intelligent Agents . Each of these interactions create a corresponding list of properties that intelligent agents would need to possess if they are to create a Ubiquitous environment. The following section will discuss these properties of pervasiveness, personalisation, context awareness, scalability and privacy

2 Human-Computer Interaction in Ubiquitous Environments

The futuristic scenario described in the previous section is one example of a ubiquitous computing vision. The ubiquitous computing vision, as articulated by Weiser [48], [49], [50] envisages a world saturated with embedded computational artefacts networked with each other. Trends such as inexpensive internet access and the diffusion of wireless computing devices is making this vision a practical reality, supporting the delivery of services anytime and anywhere [2] [27]. This networking element is essential to the creation of these ever-present services that can only be delivered through the cooperation, as enabling social interactions among multiple users [53], [32].

From the Greg's perspective in the futuristic scenario, the lack of standard WIMP (Window, Icon, Menu, Pointing devices) facilities represents the most obvious difference with conventional computing. Ubiquitous computing advocates the disappearance of the technology in the environment. This phenomenon resembles the psychological effect that when people are familiar with something, they cease to be aware of it. Weiser explains this concept in terms of centre and periphery of attention. The calmness property of ubiquitous systems is related to their capacity to move from the centre of attention to the periphery without disturbing the user. How this is achieved is open to question; however, an examination of the literature tends to suggest that the issues of pervasiveness and personalisation, context-awareness, scalability and privacy are prerequisites to ubiquitous interaction.

Pervasiveness and Personalisation The central tenet behind AmI and ubiquitous environments requires to support seamless system-human interaction and multiple services while also accounting for user's mobility, heterogeneous interfaces, as well as power outage and network disruptions. At the same time, the fact that we are all individuals influences our expectations and design requirements for those tools and systems with which we interact. However, reconciling pervasiveness and ubiquity, with concepts such as personalisation across multiple interaction medias still remains a largely unexplored area [30]. *Greg's day embraces different environments and applications. It also involves different social partners both at home and at work, where he has to operate through different computational devices and interface media. In all these activities, Greg is assisted by services that are personalised to his needs and personal characteristics, such as diet requirements, fitness level and health problems. The same services also respond to Greg's personal goals and commitments, e.g. in terms of school schedule and projects. Crucially, our example shows how throughout his day this interaction may be eased by availing of a consistent persona (the agent) to represent and interface with the underlying services.*

Context Awareness For Abowd and Munatt [2] the requirement for context awareness is also a source of many practical difficulties, including how to define the context (e.g. its boundaries), how to encode it, how to gather and fuse information from multiple sources (which often gives noisy and uncertain information), and how to manage its evolution in the presence of new elements.

In order to gain advantage from all its knowledge on Greg's characteristics and goals, and provide assistance in an unobtrusive manner, the system described in our introduction gathers and manages a wealth of information concerning Greg's activities. Monitoring the user's location is essential for inferring other useful information associated to the location, such as the slope of a certain path or the presence of other people in the vicinity of the user.

Scalability System scalability is another important issue, which refers to the expansion of the system from a prototypical world to the real world (with its contingencies) [49, 2]. Scalability is not a direct issue of human-computer interaction. It is rather a technical problem. The change of the environment is unforeseeable and the environment's contingencies are by definition dynamic. The system can have difficult to understand the new context and to adapt to the new circumstances. Moreover, the increase of data and elements of the context can soon lead to an unmanageable quantity of total information to process. This causes the whole system to slow-down, and ultimately impacts on the user's perception of the technology, in terms of reliance and trust.

In our scenario, Greg plays a treasure game. What happens if the street where the treasure is located is blocked for some work or the game server becomes too slow, or the GPS information unusable?

Privacy Finally, serious considerations should be given to privacy issues within ubiquitous systems. As personal and contextual data is collected and dissemi-

nated across the many participants in the environment, the user is increasingly exposed to potential problems caused by data leakage and eavesdropping [44], which may compromise the confidentiality of sensitive and personal information [46]. In order to be used safely and provide effective natural interaction, ubiquitous systems should also defend the user against unwanted interactions [44], and against attempts to compromise data integrity [46]. Traditional solutions to these issues are based on three steps, namely: user's identification (user declares his identity), authentication (the system verify the authenticity of this claims) and authorisation (user is granted specific access rights) [46]. However, these issues have only recently being addressed in ubiquitous environments [29], whereby they also contrast with the need for natural forms of interaction that do not require active intervention by the user.
In a ubiquitous system, only necessary data should be communicated to the current device or other service providers. The user's personal information is spread throughout the system thus a bug or intentional attack could reveal it to everyone. What does the system know of Greg? The agent knows: the current location; the relationship; the school; the food habits; the healthy and so on.

One initiative that seeks to address this issue of ubiquitous interaction is that of Ambient Intelligence (AmI) [1]. AmI was conceived in response to the observation that many embedded artefacts all competing for the user's attention would result in environments that were uninhabitable. Agent-based intelligent user interfaces are seen as key enabling technology for AmI, and a solution to the interaction issue.

Harnessing intelligent agents for manipulating user interfaces in an effort to improve the user experience is a concept that has been discussed extensively in the literature for well over a decade. Agents can adapt the interface in response to a variety of prevailing contexts. Likewise, agents have been harnessed to animate a range of digital assistants, for example, in form of virtual characters interacting with the use. It has been advocated that the effectiveness of these agent-based intelligent interfaces in general stem from their deliberate exploitation of the fact that end-users consistently assume, if only on a subconscious level, an intentional stance toward computer systems [19].

Much of this research pre-dates many of the key developments in mobile computing technologies in the last decade, and focuses on meeting the needs of standard workstation users. Though understandable, this reduces the applicability of interface agents in ubiquitous computing scenarios. However, development in agent systems, in particular the availaibility of agent platforms for light and embedded devices such as PDAs and cellular phones have radically altered the landscape.

Embedded Agents [38][23] are lightweight agents that can operate on computationally limited devices. While obviously constrained by the platform on which they are hosted, nevertheless, such agents encapsulate many of the characteristics that one would traditionally associate with interfaces. It is their potential as a platform for integrating the diverse suite of artefacts necessary for realizing practical ubiquitous user interfaces that is of most interest. For ubiquitous interfaces, this is a seminal development as a homogenous architecture can now be deployed to encompass a range of artefacts.

One of the foremost aids to the design of agent-based ubiquitous/pervasive systems comes from the developments in agent-oriented software engineering (AOSE). AOSE promotes the designing and the developing of applications in terms of multiple autonomous software agents for their characteristic ability of delivering context sensitive, adaptive and scalable solutions. For this purpose, an increasing number of these systems are employing sophisticated agent control architectures and they have been successfully deployed in many mobile computing domains including tourism like Gulliver's Genie [37], mobile commerce like Easishop [26] and in outdoor exergaming (a mixture of gaming and exercise) environment [22].

Often, the effectiveness of these systems depends on the ability of the agents to take into account the state of the physical environment where their components are deployed, as well as the state of their human users. What we see in action is agents assistants that are embodied in the same environment of the user, for instance, as Virtual Avatars placed visualised through Augmented Reality (AR) interfaces, as in the Nexus [36] [39] system discussed later in this chapter. Nexus is not the only project attempting to bring embodied agents to a Augmented/Mixed Reality environment. The conversational agent Welbo [3], for example, is an interface agent that guides, helps, and serves the users in an "Mixed Reality Living Room" where the users can visually simulate the location of virtual furniture. Pedagogical Embodied Conversational Agents (PECA) [12] are similar virtual agents that apply proven pedagogical techniques to interact with human learners in various learning scenarios, including outdoor student tours of historical buildings. In addition to position-tracking devices, these systems may also employ other sensors, such as light and temperature sensors for gathering information about the physical environment in which they are visualized.

Furthermore, their ability to migrate between devices in the network and adapt to different circumstances, e.g. different computational power or different interface capabilities, makes this type of embedded interface agents the natural candidates for maintaining the connection with the user through his daily activities [5]. Noticeably, this also provides a simple basis upon which to tackle the privacy issue, as agents operate within the clear confines of agent platforms and usually avails of interpreted agent languages (similar to the Java sand-box), as illustrated in the next section of this chapter. Furthermore, it also helps to exploit the same social and psychological insights on human social behaviour that are informing the fields of social agent-human interaction. This paradigms are the most recent evolution of intelligent interface agents for the creation of agents that are capable of acting as more equal social partners. This includes developing and maintaining long-term social relationships based upon believability, credibility and trust with humans, as demonstrated within the robotics arena [17] [18].

In order to use intelligent interface agents, a framework for the development of Agent based system is required. For the three concept projects discussed in this chapter, Agent Factory provides the Intelligent Agents that each project utilisess.

3 Agent Factory

Agent Factory (AF) is an open-source framework for the development of agent based systems [11] (see Figure 1). Since 2001, AF has been structured over a number of layers, with the lower run-time environment layer providing FIPA-standards based support for agent interoperability much in the same way as is done in other agent platforms, such as JADE [8].

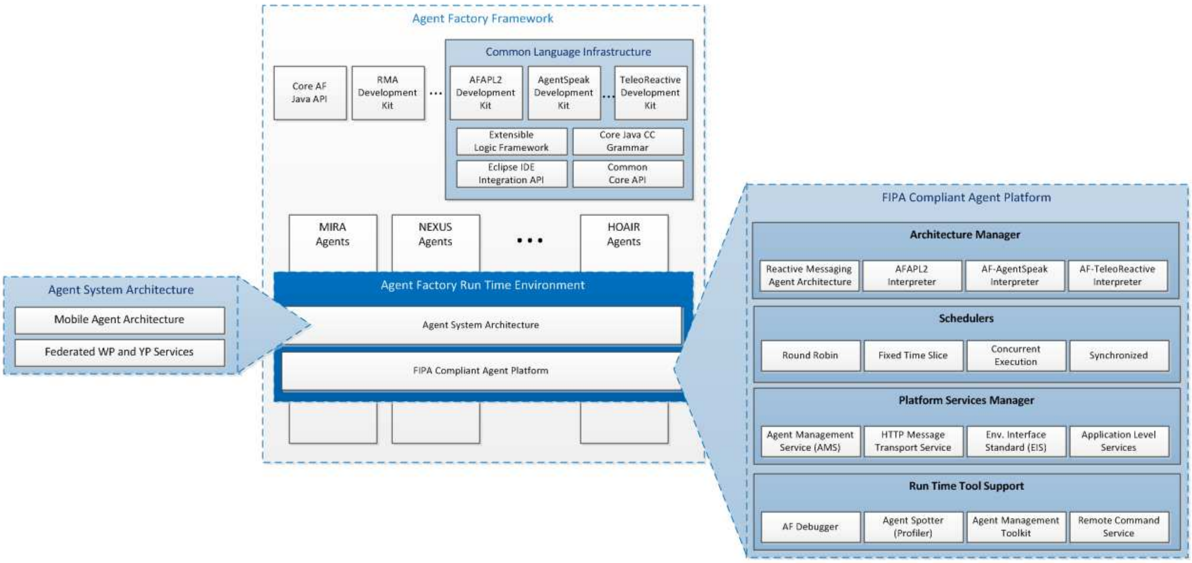

Agent Factory Framework
Common Language Infrastructure
Core AF Java API
RMA Development Kit
AFAPL2 Development Kit
AgentSpeak Development Kit
TeleoReactive Development Kit
Extensible Logic Framework
Core Java CC Grammar
Eclipse IDE Integration API
Common Core API
MIRA Agents
NEXUS Agents
HOAIR Agents
Agent Factory Run Time Environment
Agent System Architecture
FIPA Compliant Agent Platform
Agent System Architecture
Mobile Agent Architecture
Federated WP and YP Services
FIPA Compliant Agent Platform
Architecture Manager
Reactive Messaging Agent Architecture
AFAPL2 Interpreter
AF-AgentSpeak Interpreter
AF-TeleoReactive Interpreter
Schedulers
Round Robin
Fixed Time Slice
Concurrent Execution
Synchronized
Platform Services Manager
Agent Management Service (AMS)
HTTP Message Transport Service
Env. Interface Standard (EIS)
Application Level Services
Run Time Tool Support
AF Debugger
Agent Spotter (Profiler)
Agent Management Toolkit
Remote Command Service

Fig. 1 Agent Factory architecture diagram

The AF family tools includes Agent Factory Micro Edition (AFME) [34] , a minimised footprint agent platform developed to enable the creation of intentional agents for mobile devices. It is an open source project and is available for download from SourceForge. It targets the Constrained Limited Device Configuration (CLDC)/Mobile Information Device Profile (MIDP) subset of the Java Micro Edition (JME) specification

Both AF and AFME use an Agent-Oriented Programming language adhering to the popular Belief, Desire, Intention (BDI) agent model [42]. Such a model represents agents as mental entities whose internal state consists of beliefs, goals and commitments. Beliefs represent the agent's current state of its environment, while commitments represent the outcome of an underlying reasoning process through which the agent selects what activities it should perform. In addition to atomic actions, each agents can avail of a library of pre-compiled plans, i.e. procedural knowledge of the steps necessary to achieve given goals in given circumstances.

The goal-based nature of these agents helps the design of context-aware systems, and it also constitutes an important source of modularity, as designers can concentrate on writing plans for a subset of essential situations and can construct plans for more specific situations as they debug the system. Plans are invoked either in response to particular situations or based on their purpose, by availing of meta-reasoning heuristics used to select the best possible plan out of a number of suitable options.

Execution of an Agent Factory Agent Programming Language (AFAPL) [11] program involves the update of the agent's mental state by repeatedly applying (driven by the AF Platform Scheduler Service) an internal reasoning process that combines:

1. **Update of the agents beliefs via perception of the environment through a set of auxiliary Java components, known as perceptors.**
2. **The adoption of new commitments, through the evaluation (via logic resolution) of a set of PROLOG-like commitment rules, which map belief states onto commitments that should be adopted should that state arise.**
3. **The realisation of commitments by performing actions, which are implemented through a set of auxiliary Java components, known as actuators.**

Both actuators and perceptors in AFAPL are placeholders defining the connection between the constructs of the agent language and primitive action/perception modules. As such, they act as a standard interface layer allowing the connection between the agents' reasoning apparatus and the particular applicative environment where the agents operate.

More recently, these interface capabilities has been updated with the creation of AF adapter services to enable agents' interoperability with standard computational environments, as implemented in a number of proposal in the AOSE arena. For instance, the EIS (Environment Interface Standard) adapter [7] enables to connect to arbitrary environments, while the CArtAgo (Common ARTifact infrastructure for AGents Open environments) [43] adapter enables agents to work with artifacts - resources and tools that may be utilised in a standardised manner by agents in the satisfaction of their objectives. Finally, our SoSAA (Socially Situated Agent Architecture) adapter [15] enables AF to supervise component-based framework, such as Java Beans and OSGi. All these interface capabilities are essential to facilitate the interoperability of our agent solutions with heterogeneous legacy and third party software and technologies, as illustrated in the following sections.

In the remainder of this chapter, the three technology offerings of NeXuS, FreeGaming and MIRA will be described together with illustrated Concept Demonstrations. Each project utilises the Agent Factory Framework and together demonstrate how the futuristic scenario envisioned in the introduction of this chapter can be achieved.

4 NeXuS

NeXuS aims to create a distributed Augmented Reality (AR) framework that capitalises upon the properties of intentional agents in order to create a single world in which people can interact in a Ubiquitous manner with purely virtual entities in their own space. In order to reach this goal it is necessary for the embodied agents in the world to exhibit realistic behaviour. The goal of NeXuS is such that a real world user interacting in an AR environment must be able to suspend their disbelief towards a virtual avatar, this goal becomes unattainable if an avatar does not properly perceive the user's presence. It is the goal of achieving behavioural realism that motivates NeXuS to embed intentional agents within virtual avatars. To achieve a goal of behavioural realism for our Avatars, each embodied agent must given virtual senses so they are able to perceive the real world around them.

4.1 Architecture

The NeXuS architecture (see Figure 2) facilitates the design and delivery of experiments and experiences that fuse physical and virtual spaces into a coherent singular AR environment employing agent based technologies. A stratified architecture allows for flexibility in modifying pre-installed components or indeed adding new components to enhance the system.

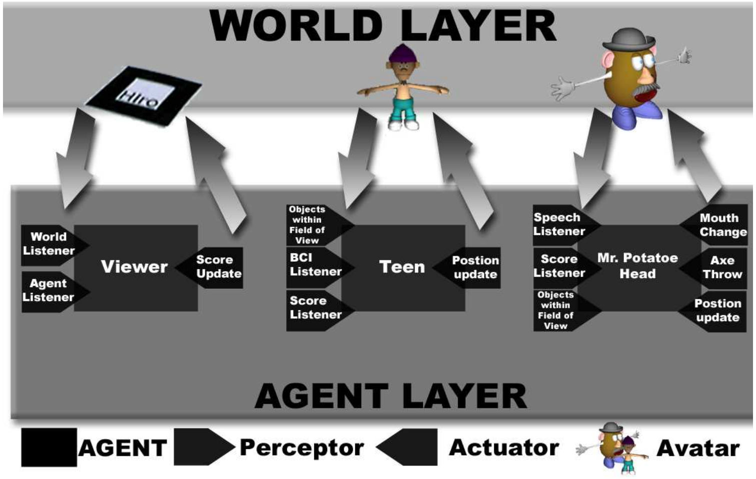


Fig. 2 The NeXuS architecture showing the interaction between different agents actuators and perceptors with the world layer.

Agent Layer The intelligent software agents layer exists at the centre of the NeXuS architecture. The agent's perceptors constantly check the state of the world for any relevant changes. If any such changes do occur the perceptor accordingly generates alterations to the belief state of the agent. The role of an actuator is to enact appropriate modifications upon the agent's environmental model whenever the belief state of the agent corresponds to a predefined inference rule. Perceptors and actuators may be used to give an agent virtual senses of touch, sight and hearing.

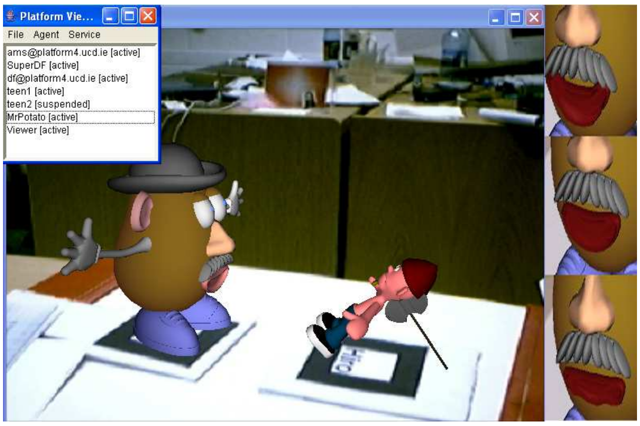
World Layer The upper most layer of the system architecture is the world layer itself. This layer is subdivided into two domains, the physical and the virtual. The physical domain corresponds to the hardware input/output (I/O) system components including the microphone, the display and the camera. The virtual domain encapsulates the software used to process the I/O data.

Virtual Component The world that is presented to the user is created with Sun Microsystems Java3D API. The use of Java 3D [40]increases the simplicity of integration with the java-based Agent Factory agents. The embodied agents' avatars that inhabit the 3D world exist as either 3D Studio Max [4]or Milkshape 3D [10] models. Each of these models may also have a set of additional components that can be interchanged to better express the internalised beliefs of the agent that underpins their behaviour. This set of components is specified in an XML[47] file that is parsed into the system at runtime. The ARToolKit [25] is used by NeXuS to support both video input and the tracking of events in the physical world. The open-source jARToolKit [20] java binding was used to enable the integration of the C based ARToolKit with Java3D. NeXuS facilitates the use of speech recognition with integrated support for Carnegie Mellon University's open-source Sphinx-4 [28].

Physical Component The Physical World Layer contains the real world (I/O) components of the system with which the user directly interfaces. This layer represents the non-virtual elements of the AR environment in which the agents are embodied. The principle input technologies within this domain are the camera and the microphone. Visual output can be achieved on monitor, head mounted display(HMD) and projector. Using the HMD in conjunction with a camera placed on top of it allows for 'Video-See-Through' Augmented Reality experience. Two Brain-Computer Interface devices, the OCZ Neural Impulse Actuator[35] and Emotiv headset[16], are on sale and have been integrated with the NeXuS project. Both utilise Electroencephalography (EEG)'s that can read if a user is agitated or relaxed.

4.2 Concept Demonstration

The 'Axe Game' concept demonstration shows how BDI agents may be used to mediate interaction between the physical world and the virtual avatars. In figure 3, Greg's Avatar is described playing the axe game in the introduction. This figure



ams@platform4.ucd.ie [active]
SuperDF [active]
df@platform4.ucd.ie [active]
teen1 [active]
teen2 [suspended]
MrPotato [active]
Viewer [active]

Fig. 3 The left side of the figure is the axe colliding with the Teen Avatar. On the right, a close up of the Mouth Transitions for the Mr.Potato Head avatar

also shows the fiducial marker use to register the 3D object within the real world. The agents can exist on different computers and thus can be organised to play the game anywhere in the world. These agents have the ability to pro-actively respond to dynamic alterations of their belief states triggered by changes in both the physical and virtual domains of the augmented reality environment.

The axe game is a simple 'shoot and dodge' scenario visually played out between a Mr. Potato Head agent and a teen agent. Both agents require human input in order to fully engage in the scenario. For Mr. Potato Head this takes the form of vocal commands issued through a microphone while the teen obtains input by using a brain computer interface. As the user relaxes the Teen avatar will move to left and if they concentrate it will move to the right. Each agent is aware of the current state of the game through both its own belief state and a *Score* module that is constantly maintained and updated by a *Score* actuator and a *score* perceptor.

The Mr. Potato Head agent has the ability to visually express its mental state by changing its 3D avatar components. This corresponds with the interchangeable facial components of the 'real life' Mr. Potato Head toy and is symbolic of the overall modular nature of the NeXuS framework. The facial expressions displayed by the Mr. Potato Head model reflect the current state of the axe game. Depending on whether it is winning, losing or in a tied position, the avatar will respectively appear to be happy, sad or neutral (see figure 3). Supplementary to this functionality, Mr. Potato Head's eyes move left or right to a position that indicates where the teen is located in its field of vision ..

Once Mr. Potato Head's speech perceptor detects a fire command from the first player, its belief states are altered and a series of communications between itself and the viewer agent are triggered off, resulting in the creation of a virtual axe. A confirmed hit is visually represented by an appropriate change in Mr. Potato Head's facial expression and an animation of the teen falling back from the 'force' of the impact as shown in Figure 3.

This simple concept application demonstrates the ability of the agent paradigm to facilitating users to experience interaction with the system using multiple modalities. It is this ability to seamless interact with these different modalities that can be considered Ubiquitous interface.

5 FreeGaming

The exergaming concept searches for ways to combine exercising and gaming [9] by disguising the tiresome side of working out with the uplifting side of playing to make the exercise process more attractive. The players in indoor type exergames are generally represented in the game environment by a virtual representative (avatar). In such settings, it is relatively easy to capture the player's motion, for example through cameras and other sensors, and integrate player's exercise with the game. The feedback can be delivered through haptic interfaces or screen facilities and objects/virtual characters surrounding the user's avatar can be easily modified and driven.

The aim of Freegaming is to extend this exergaming concept to the outdoors. Its concept offers generating a play anywhere, anytime and collaborative exergaming environment.

Mobile FreeGaming allows outdoor and ubiquitous exergaming. While indoor exergaming creates natural limits on what exercises can be performed, adding *mobility* to the exergames can open up the opportunity for encouraging many type of different exercises.

Augmented Freegaming gives the ability to the developers to utilize *augmented reality* when developing, to break away from the traditional avatar paradigm. This allows the player to feel an immersive experience where it is their own physical body which interacts with the game instead of mediated experience through a virtual avatar.

Collaborative Freegaming architecture is designed to allow both single and multiplayer gaming. It offers a suitable system to combine the socialisation feature of *mobile collaboration games* with the positive effects of exergames on physical health.

Adaptive Freegaming targets allowing outdoor exergaming. The objective of the Freegaming architecture is to *adapt* to this constantly changing outdoor environment by using intelligent agents technology.

Freegaming platform is using the intelligent agent paradigm and implemented by using Agent Factory Mobile Edition (AFME) [33]. It has been coded using AFME's (Belief, Desire, Intention) BDI-style agents which allow Freegaming Mobile Application (FMA) to use any sensor data to adapt itself both to players' body information and also information coming about environmental context.

5.1 Architecture

Freegaming platform includes two main components. The first one is Freegaming Server and the other one is Freegaming Mobile Application (FMA) which runs on the mobile devices as a client application. Freegaming Server consists of two separate servers which are Freegaming Game Server (FGS) and the Machine Vision Server (MVS).

Freegaming Game Server includes the Freegaming Database (FDB) which maintains the game status and holds all the necessary information for each players during the game. FGS also consists the Freegaming Servlets (FSERV) which allows the FMA to access and modify the game databases.

Machine Vision Server is responsible from creating an Augmented Reality environment [31]. Freegaming ambition to promote fitness requires that it creates incentives for players to complete their goals within the game. To achieve this the Freegaming architecture allows the developers to augment on planar surfaces in the photos taken by the mobile phone camera. The players are able to use their mobile as a window into a world where the virtual objects are coexisting with their physical counterparts. To generate the augmented photos, the Speeded Up Robust Features (SURF) [6] algorithm is utilised, features can be tracked from a template image onto an image taken by the phone. As the phone processor is not powerful enough to achieve this goal within a usable time frame, the MVS on Freegaming Server is utilised to achieve this goal.

Freegaming Mobile Application also has two main parts, first one is the J2ME[41] application and the other one is Game Agent (GA). J2ME application includes the necessary screens for the game like, map view and is mainly responsible from the graphical user interface (GUI). The GA makes decisions over the game status and scenario which directly effects the J2ME application. A Perceptor in the GA continuously reads the Global Positioning System (GPS) data during the runtime and updates its belief set concerning the state of the player and the game itself. It uses this belief set to make logical commitments for the game play; for example, identifying those zones into which the player should move. In this case, the J2ME application would generate an audio alert and show the new objective to the player.

During the game, locations of all players are obtained from built-in GPS receivers, GA uses these data to continuously update the game scenario. FMA communicates with FGS via FSERV and accesses to or updates the FDB. Also, the images captured during the game are sent to the FGS which delivers these images to MVS. MVS uses template image databases for necessary SURF calculations to analyse the newly captured image and decide on where to augment the virtual objects. The results from MVS is then reflected to the FMA via FSERV. J2ME application uses these data to augment the virtual objects on the planar surfaces over the photo taken by the player. All communications between the Freegaming components are done over web connections. The communication between the FMA and the FGS is done over

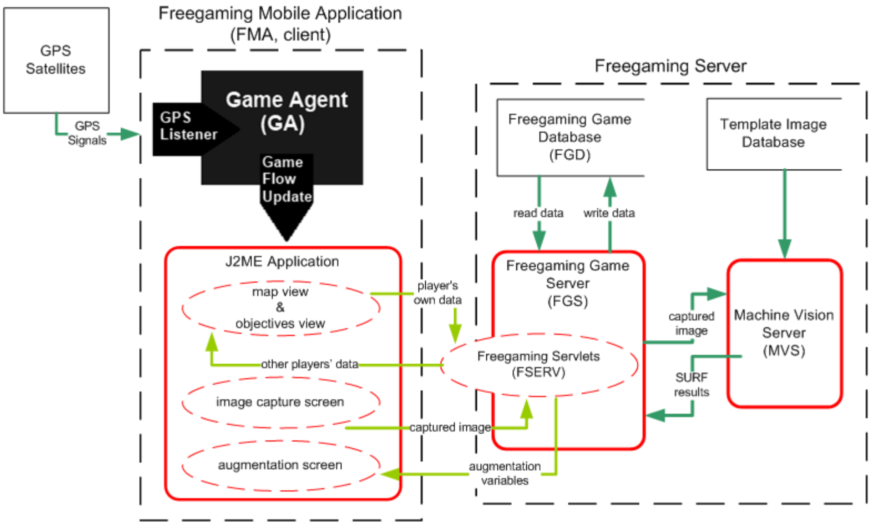


Fig. 4 Freegaming Architecture Dataflow Diagram

either wireless internet connection or 3G connection where it is possible. Dataflow diagram between the components of Freegaming Platform can be seen in Figure 4.

5.2 *Concept Demonstration*

Freegaming platform is designed to allow game designers to generate location sensitive exergames. Because of this nature of the system, game developers need to define special regions in a pre-defined gaming area. All special regions will have their related GPS coordinates defined in the GA which will control the game flow during runtime. At the same time, game designer needs to generate a template images database for each special region. As mentioned before, these template images will be used in the MVS for SURF calculations (Figure 5).

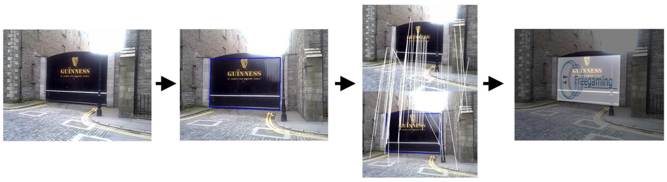


Fig. 5 Augmented Reality, Freegaming Machine Vision Server and SURF calculations.

To implement the Freegaming Platform's capabilities a prototype game is generated which is not using the multiplayer functionality and can only be played as a single player game. The screenshots in Figure 6 display this game as it is implemented on a Nokia N97. There are two special regions in this game and four objectives. At the beginning, the player enters his/her age. When the game begins, players can see the map view which shows the player icon overlaid on the map of gaming area. For the first objective, the player needs to go to the starting area for the game. When the player reaches to this point, GA detects this achievement automatically by reading the GPS signal and changes the player's state. Also the player gets prompted by J2ME application with a sound and a visual alert about a new objective. This new objective is to take a picture of the scene, especially the walls or windows. When the player captures a proper picture of the scene, MVS detects the windows inside this photo and augments virtual arrows on them that shows the direction of the next target location. The map view also gets updated by the GA (Figure 6). When the second target zone is reached, the player is alerted again with a new image capturing goal. This time, game asks for the gate in the target area. When the player takes the picture, GA decides on which message to augment on the gate based on the personalisation information given at the beginning of the game. It augments the direction for the nearest pub if the player is over 18 or the storehouse if the player is under 18 years old (Figure 6). After that, the game informs the player that the game is finished.

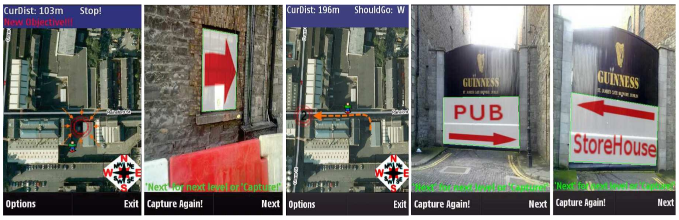


Fig. 6 Sample screen shots from Map View and Augmentation screens on mobile phone screen. Freegaming Game Agent augments different visual object based on personalization data.

6 Mixed Reality Agent (MiRA)

The Mixed Reality Agent (MiRA) project merges physical robots with virtual avatars in a bid to overcome the limitations of a purely real or virtual embodiment. A Mixed Reality Agent combines the tangible physical presence of a robot agent with the rich expressive capabilities and personalisation features of a virtual persona that are complex and expensive to realise with purely hardware-based solutions.

Figure 7 shows a typical MiRA system, akin to the one referenced in the introduction. In the figure, Greg's avatar takes the form of a boy and can be seen sitting on the robot, illustrating that it has migrated to and is now in control of that particular robot. Being rendered via Augmented Reality, it is able to spatially reference real world objects via deictic gestures, like the ball it is supposed to fetch, and can use its body language and facial expressions to provide feedback, for example, smiling when it sees or has fetched the ball.

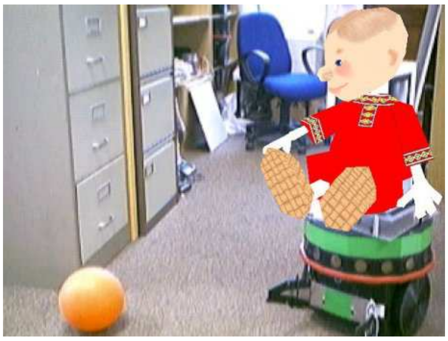


Fig. 7 A typical MiRA scenario showing a boy avatar in charge of a simple two-wheeled robot.

In contrast to virtual characters visualised on a screen attached to a robot, such as GRACE [45] and VALERIE [21], the Mixed Reality characters are visible from all angles and are not subjected to diminishing visibility at greater distances.

Other recent efforts that seek to combine virtual characters and physical robots for social human-robot interaction include the Jeeves project [51], which similarly combines a robot (the iRobot Roomba vacuum cleaner) with a cartoon-like character in order to investigate the use of cartoon art (i.e. simplified and exaggerated facial expressions) for intuitive social interaction with humans. Such an approach is intended to offer insights into the robot's state while, at the same time, avoiding to overly increase people's expectations as more realistic and human-like representations could.

Due to their mixed reality interface, both these systems are advantageous in applications with a high robot-to-user ratio, as a single interface can augment the interaction capabilities of multiple simple robots (e.g. without screens, heads or arms) and, possibly, even portray different characters to different users simultaneously [24]

6.1 Architecture

In essence, a MiRA is the result of a collaboration between a robot and a user node (see Figure 8), which communicate over an ad-hoc wireless network link in order to exhibit cohesion and behavioural consistency to the observer. Tracking is achieved by simply placing a cubic marker on top of the robot and tracking its position from

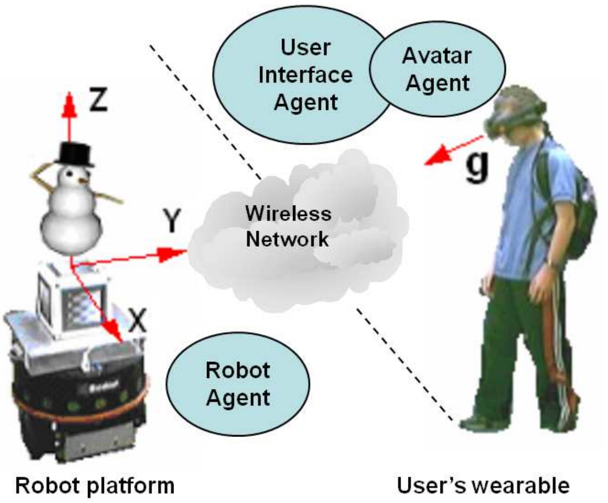


Fig. 8 SoSAA nodes for MiRAs.

the camera associated with the user's HMD. The tracking information is then used to align the image of the virtual character with that of the real robot.

Figure 8 shows how a MiRA system is implemented as the collaboration between different nodes deployed, rispectively: those in control of the robotic platforms, those managing the user interfaces (running on a wearable computer connected to the user's head mounted display (HMD), and those in control of the virtual characters. Such an organisation enables the MiRA to adapt to different users and different robots, as there is no predefined coupling between the robot and the appearance or behaviour of its associated virtual character. Instead, thanks to their communication with the robot agent, both user interface agent and avatar agent can take context sensitive decisions in order to deliver a personalised and adaptive human-robot interaction (HRI) interface.

In order to be a believable component of the mixed reality agent, the behaviour of the virtual character needs to exhibit a degree of awareness of its surroundings, comparable to a robot being physically embodied through an array of physical sensors and actuators. In MiRA, the instrument for such situatedness is the update input stream, which notifies the MR interface about the nature and relative position of the obstacles and other objects of interest perceived by the robot, and also about the intentions of the robot. This allows direct control of deictic and anticipatory animations of the associated virtual character, which can visually represent the robot's intentions (similar to Jeeves's cartoon-like expressions of the robot's state [52]).

On the other hand, since the MR overlay of virtual images onto the user's field of vision requires exact knowledge about the position and gaze of the user, this information can also be relayed to the robot. In doing so, the wearable interface helps the robot to know the position (and the identity) of the user, while the user can use his gaze direction to spatially reference objects and way-points in order to influence the robot's behaviour. The communication between wearable interface and robot is therefore essential in reinforcing the embodiment of each part of a MiRA and augmenting the system's HRI capabilities by merging these parts into a single agent.

Its agent-based implementation is essential to the open, ubiquitous nature of the MiRA system. Specifically, the different agents in the system use a User Datagram Protocol(UDP) based peer-discovery service to create connections whenever the user gets in the proximity with a MiRA-equipped robot. As soon as a robot agents starts to collaborate with the user's wearable node, the two can exchange information, e.g. about each other's identity, and also agree on migrating some functionalities from the user node to the robot node before the robot enters the visual field of the user's HMD. After this first connection, and until robot and human depart from each other, the two nodes will then collaborate to deliver a MiRA composed of the real robot and a virtual character visualised through the user's HMD.

6.2 Concept Implementation

In order to drive the implementation of the new MiRA system, we implemented a simple application scenario to demonstrate the joint exploitation of gaze tracking and positional awareness[13], and the expressive capabilities of MiRAs [14] by asking users to observe and collaborate with a soccer playing robot. Specifically, the user can ask the robot to find and fetch an orange coloured ball, and also direct some of the robot's movements to facilitate the successful and speedy execution of the task.

Figure 9 approximately illustrates the organisation of part of the wearable SoSAA node in the MiRA system. Within the node, component agents supervise two functional areas, namely the user interface and the control of the virtual character (avatar) associated with the robot. The user interface components control the display of text and other 2D graphic overlays in the user's HMD, and process user input. Through them, the user can be informed of details of the task and the state of the robot. These components also process user utterances availing of the IBM ViaVoice^{TM}speech recogniser and the Java Speech API (http://java.sun.com/products/java-media/speech/). Finally, the behavioural animations of the virtual character are implemented via OpenVRML (http://www.openvrml.org), an open source C++ library enabling a programmatic access to the VRML model of the avatar.

7 Conclusions

This chapter has proposed intelligent agents as a software architecture for integrating and managing the diverse range of technologies essential to ubiquitous interfaces. The three technology offerings give us concept demonstrations to illustrate

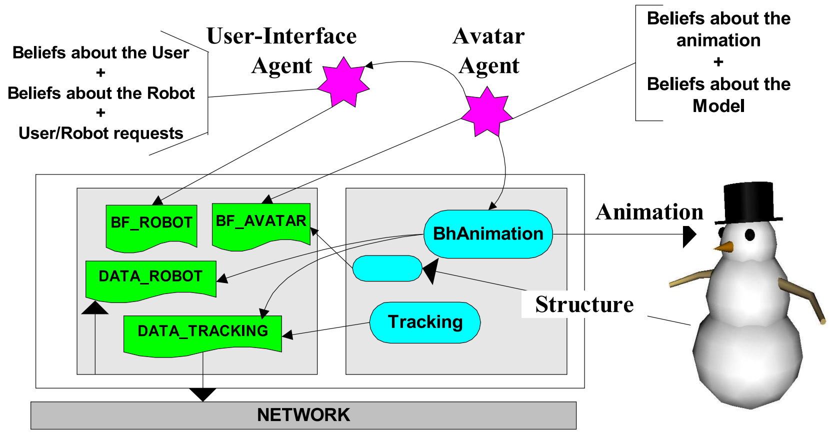


Fig. 9 Organisation of the user's MR node.

how our futurist scenario's ubiquitous interfaces could be achieved. Freegaming demonstrates how the fast advances in the mobile device industry made realistic and ubiquitous AR environments possible. NeXuS shows how a ubiquitous interfaces can be facilitated in an Augmented Reality environment. MIRA merges physical robots with virtual avatars to achieve a ubiquitous interface. To achieve the goals of Pervasiveness and Personalisation, NeXuS demonstrates how a consistent persona of Greg's Avatar could be emotionally represent achievements past and present. FreeGamming shows us how personalisation could be used to help create Augmented Reality Exergaming where the Greg's Agent was aware of his diet requirements, fitness level and health problems. The pervasiveness of the agent operating through different computational device is exemplified in the MIRA concept demonstration where an agent takes on both Virtual and Physical forms, thus showing how Greg and Peter could interact with their robot in the futurist scenario. Context Awareness is crucial to any ubiquitous interfaces. FreeGaming concept demonstration shows how this can be achieved in an unobtrusive manner monitoring a users location and changing the game accordingly. NeXuS allowing different interfaces to control Avatars based on what devices the user has currently available. Greg and Michael could play the same game without using the same interface. Any ubiquitous environment require Scalability , the concept demonstrations are idealised scenarios as is the futuristic scenario. Dealing with unexpect problems such as a road been blocked in Freegaming or the network connection between players using NeXuS failing would need additional intelligent agents to help mediate. In each concept demonstrations, any personal information is kept secret by the Agents themselves and not shared out unless require to complete a task.

The concept applications demonstrate how embedded agents equipped with BDI (Belief, Desire, Intention) control systems and a range of interface mechanisms can be used to support a wide range of collaborative applications scenarios with multiple agents and multiple users, where agents posses a high degree of autonomy. This approach provides the reasoning apparatus for creating believable characters that are responsive to modifications and stimuli in their environment, but are also proactive and goal-oriented. The approach also addresses the challenge of how to deal with the heterogeneity of the device population in a ubiquitous environment. Facilitating intuitive interaction is a prerequisite for the ubiquitous computing paradigm , using Intelligent Agents this prerequisite can be achieved.

Acknowledgements This work is supported by Science Foundation Ireland (SFI) under grant 07/CE/I1147 and supported by the National Digital Research Centre(NDRC) Research Grant entitled FreeGaming Project number SP/010.

References

1. E. Aarts, P. Res, and N. Eindhoven. Ambient intelligence: A multimedia perspective. *IEEE multimedia*, 11(1):1219, 2004.
2. G. D Abowd and E. D Mynatt. Charting past, present, and future research in ubiquitous computing. *ACM Transactions on Computer-Human Interaction (TOCHI)*, 7(1):58, 2000.
3. Mahoro Anabuki, Hiroyuki Kakuta, Hiroyuki Yamamoto, and Hideyuki Tamura. Welbo: An embodied conversational agent living in mixed reality space. In *Proceedings of the Conference on Human Factors in Computing Systems - CHI 00*, pages 10–11, The Hague, The Netherlands, April 2000.
4. autodesk.
5. Istvan Barakonyi, Thomas Psik, and Dieter Schmalstieg. Agents that talk and hit back: Animated agents in augmented reality. In *ISMAR '04: Proceedings of the 3rd IEEE/ACM International Symposium on Mixed and Augmented Reality*, pages 141–150, Washington, DC, USA, 2004. IEEE Computer Society.
6. Herbert Bay, Andreas Ess, Tinne Tuytelaars, and Luc Van Gool. Surf: Speeded up robust features. *Computer Vision and Image Understanding (CVIU)*, 110(3):346–359, 2008.
7. Tristan M. Behrens, Jürgen Dix, and Koen V. Hindriks. Towards an environment interface standard for agent-oriented programming. Technical Report IfI-09-09, Clausthal University of Technology, 2009.
8. Fabio Bellifemine, Agostino Poggi, and Giovanni Rimassa. JADE - a FIPA-compliant agent framework. In *Proceedings of the Practical Applications of Intelligent Agents*, 1999.
9. Ian Bogost. The rhetoric of exergaming. In *Proceedings of the Digital Arts and Cultures(DAC)*, USA, 2005.
10. chUmbaLum sOft. Milkshape 3d. http://www.swissquake.ch/chumbalum-soft/ms3d/.
11. Rem W. Collier, Gregory M. P. O'Hare, T. D. Lowen, and Colm Rooney. Beyond prototyping in the factory of agents. In *CEEMAS*, pages 383–393, 2003.
12. Jayfus Tucker Doswell. It's virtually pedagogical: Pedagogical agents in mixed reality learning environments. In *Proceedings of the Thirtysecond International Conference on Computer Graphics and Interactive Techniques - SIGGRAPH 2005 - Educators Program*, page 25, Los Angeles, California, 2005.
13. Mauro Dragone, Thomas Holz, and Gregory M. P. O'Hare. Mixing robotic realities. In *Proceedings of the 2006 International Conference on Intelligent User Interfaces - IUI 2006*, Sydney, New South Wales, Australia, January 2006.
14. Mauro Dragone, Thomas Holz, and Gregory M.P. O'Hare. Using mixed reality agents as social interfaces for robots. In *RO-MAN '07: Proceedings of the 16th IEEE International Workshop on Robot and Human Interactive Communication*, Jeju Island, Korea, August 2007. IEEE Press.

15. Mauro Dragone, David Lillis, Rem W Collier, and G M P O'Hare. Practical Development of Hybrid Intelligent Agent Systems with SoSAA. In *Proceedings of the 20th Irish Conference on Artificial Intelligence and Cognitive Science*, Dublin, Ireland, 2009.
16. emotiv.
17. B. J. Fogg. Persuasive technologies - introduction. *Commun. ACM*, 42(5):26–29, 1999.
18. Terrence W. Fong, Illah Nourbakhsh, and Kerstin Dautenhahn. A survey of socially interactive robots. *Robotics and Autonomous Systems*, 2003.
19. Batya Friedman. "it's the computer's fault": reasoning about computers as moral agents. In *CHI 95 Conference Companion*, pages 226–227, 1995.
20. C. Geiger, C. Reimann, J. Stöcklein, and V. Paelke. Jartoolkit - a java binding for artoolkit. In *Augmented Reality Toolkit, The First IEEE International Workshop*, page 5, 2002.
21. Rachel Gockley, Allison Bruce, Jodi Forlizzi, Marek Michalowski, Anne Mundell, Stephanie Rosenthal, Brennan Sellner, Reid Simmons, Kevin Snipes, Alan C. Schultz, and Jue Wang. Designing robots for long-term social interaction. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems - IROS 2005*, August 2005.
22. L. Görgü, G.M.P. OHare, and M.J. O'Grady. Towards mobile collaborative exergaming. *International Conference on Advances in Human-oriented and Personalized Mechanisms, Technologies, and Services (CENTRIC2009)*, 2009.
23. H. Hagras, V. Callaghan, M. Colley, G. Clarke, A. Pounds-Cornish, and H. Duman. Creating an ambient-intelligence environment using embedded agents. *Intelligent Systems, IEEE*, 19(6):12 – 20, nov. 2004.
24. Thomas Holz, Mauro Dragone, Gregory M. P. O'Hare, Alan Martin, and Brian R. Duffy. Mixed reality agents as museum guides. In *ABSHL '06: Agent-Based Systems for Human Learning, AAMAS 2006 Workshop*, New York, NY, USA, 2006. ACM Press.
25. Hirokazu Kato and Mark Billinghurst. Marker tracking and hmd calibration for a video-based augmented reality conferencing system. In *Proceedings The 2nd International Workshop on Augmented Reality*, 1999.
26. Stephen Keegan, Gregory M. P. O'Hare, and Michael J. O'Grady. Easishop: Ambient intelligence assists everyday shopping. *Information Sciences*, 178(3):588–611, 2008.
27. J. H Kim, Y. D Kim, and K. H Lee. The third generation of robotics: Ubiquitous robot. In *Proc of the 2nd Int Conf on Autonomous Robots and Agents*, 2004.
28. Paul Lamere, Philip Kwok, William Walker, Evandro Gouva, Rita Singh, Bhiksha Raj, and Peter Wolf. Design of the cmu sphinx-4 decoder. In *8th European Conference on Speech Communication and Technology (EUROSPEECH 2003)*, September 2003.
29. T. Ma, S.D. Kim, J. Wang, and Y. Zhao. Privacy Preserving in Ubiquitous Computing: Challenges & Issues. In *IEEE International Conference on e-Business Engineering*, pages 297–301. IEEE, 2008.
30. Alan Martin, Gregory M. P. O'Hare, Bianca Schön, John F. Bradley, and Brian R. Duffy. Intentional embodied agents. In *Proceedings of the Eighteenth International Conference on Computer Animation and Social Agents - CASA 2005*, Hong Kong, October 2005.
31. Paul Milgram, Haruo Takemura, Akira Utsumi, and Fumio Kishino. Augmented reality: A class of displays on the reality-virtuality continuum. In *Proceedings of SPIE*, volume 2351, pages 282–292, 1995.
32. M. Morris, J. Lundell, and E. Dishman. Catalyzing social interaction with ubiquitous computing: a needs assessment of elders coping with cognitive decline. In *CHI'04 extended abstracts on Human factors in computing systems*, page 1154. ACM, 2004.
33. C. Muldoon, G.M.P. O'Hare, R. Collier, and M.J. O'Grady. Towards pervasive intelligence: Reflections on the evolution of the agent factory framework. *In Multi-Agent Programming: Languages, Platforms and Applications.*, pages 187–212, 2009.
34. Conor Muldoon, Gregory M. P. O'Hare, and John F. Bradley. Towards reflective mobile agents for resource-constrained mobile devices. In *AAMAS*, page 141, 2007.
35. OCZ.
36. Gregory M. P. O'Hare, Brian R. Duffy, and Abraham G. Campbell. NEXUS: Mixed reality experiments with embodied intentional agents. In *Proceedings of Computer Animation and Social Agents - CASA 2004*, Geneva, Switzerland, July 2004.

37. Gregory M. P. O'Hare and Michael J. O'Grady. Gulliver's genie: a multi-agent system for ubiquitous and intelligent content delivery. *Computer Communications*, 26(11):1177–1187, 2003.
38. Gregory M. P. O'Hare, Michael J. O'Grady, Conor Muldoon, and John F. Bradley. Embedded Agents: A Paradigm for Mobile Services. *International Journal of Web and Grid Services (IJWGS)*, 2(4):379–405, December 2006.
39. Gregory M.P. O'Hare, Abraham G. Campbell, John W. Stafford, and Robert Aiken. NeXuS: Behavioural realism in mixed reality scenarios through virtual sensing. In *Proceedings of the Eighteenth International Conference on Computer Animation and Social Agents - CASA 2005*, Hong Kong, October 2005.
40. Oracle.
41. Oracle.
42. Anand S. Rao and Michael P. Georgeff. Bdi agents: From theory to practice. In *Proceedings of the 1st International Conference on Multi-Agent Systems (ICMAS-95)*, pages 312–319, San Francisco, California, USA, June 1995.
43. Alessandro Ricci, Mirko Viroli, and Andrea Omicini. Cartago: a framework for prototyping artifact-based environments in mas. In *Proceedings of the 3rd international conference on Environments for multi-agent systems III*, E4MAS'06, pages 67–86, Berlin, Heidelberg, 2007. Springer-Verlag.
44. T.S. Saponas, J. Lester, C. Hartung, S. Agarwal, and T. Kohno. Devices that tell on you: Privacy trends in consumer ubiquitous computing. In *Proceedings of 16th USENIX Security Symposium on USENIX Security Symposium*, pages 1–16. USENIX Association, 2007.
45. Reid Simmons, Dani Goldberg, Adam Goode, Michael Monetmerlo, Nicholas Roy, Brennan Sellner, Chris Urmson, Alan Schultz, Myriam Abramson, William Adams, Amin Atrash, Magda Bugajska, Michael Coblenz, Matt MacMahon, Dennis Perzanowski, Ian Horswill, Robert Zubek, David Kortenkamp, Bryn Wolfe, Tod Milman, and Bruce Maxwell. Grace: An autonomous robot for the AAAI robot challenge. *AI Magazine*, 24:51–72, 2003.
46. F. Stajano and R. Anderson. The resurrecting duckling: security issues for ubiquitous computing. *Computer*, 35(4):22–26, 2002.
47. W3C.
48. M. Weiser. Some computer science issues in ubiquitous computing. *Communications of the ACM*, 36(7):75–84, 1993.
49. M. Weiser. The computer for the 21st century. *Scientific American*, 272(3):78–89, 1995.
50. M. Weiser and J.S. Brown. The coming age of calm technology [1]. *Xerox PARC. Retrieved July*, 8:2007, 1996.
51. J. E. Young and E. Sharlin. Sharing spaces with robots: An integrated environment for human-robot interaction. In *Proceedings of the First International Symposium on Intelligent Environments - ISIE 06*, Cambridge, England, April 2006.
52. J. E. Young, Min Xin, and Ehud Sharlin. Robot expressionism through cartooning. In *Proceedings of the 2007 ACM/IEEE International Conference on Human-Robot Interaction*, Arlington, Virginia, USA, March 2007.
53. G. Zhang, Q. Jin, and M. Lin. A framework of social interaction support for ubiquitous learning. In *19th International Conference on Advanced Information Networking and Applications, 2005. AINA 2005*, pages 639–643, 2005.