Computers **2013**, *3*, 132-151; doi:10.3390/computers2030132

OPEN ACCESS

computers

ISSN 2073-431X

www.mdpi.com/journal/computers

Article

Towards Realising FollowMe User Profiles for Macro-Intelligent Environments

Luke Whittington [1,*], James Dooley [1], Martin Henson [1] and Abdullah Al-Malaise Al-Ghamdi [2]

[1] Department of Computer Science and Electronic Engineering, University of Essex, Colchester, CO3 3SQ, UK; E-Mails: jpdool@essex.ac.uk (J.D.); hensm@essex.ac.uk (M.H.)

[2] Faculty of Computing and Information Technology, King Abdulaziz University, Jeddah 22254, KSA; E-Mail: aalmalaise@kau.edu.sa

* Author to whom correspondence should be addressed; E-Mail: lwhittl@essex.ac.uk; Tel: (+44)7800 633787.

Received: 30 November 2012; in revised form: 12 March 2013 / Accepted: 30 April 2013 / Published: 30 July 2013

Abstract: In this paper, we introduce the concept of a Large-Scale Intelligent Environment (LSIE) and provide an introduction to the use of bigraphs as a formal method for description and modelling. We then propose our *MacroIE* model as a solution to the LSIE problem and describe how that model may be implemented to achieve a continuity-of-experience to end users as they travel from place-to-place (a technology we call *FollowMe*). Our initial experiments with these implementations are presented, providing some valuable insights and promise for future refinement towards real-world deployment.

Keywords: ambient intelligent environments; middleware; formal methods; mobile computing; server architecture; bigraphs

---

1. Introduction

Intelligent Environments are a wonderful example of ubiquitous computing [1,2], but ask two different research groups what an Intelligent Environment (IE) is, and you may get two fundamentally different answers. In our vision, an IE is a "common" space (such as a home, classroom or office) that contains a plethora of embedded computer devices that are interconnected and work together to enrich user experiences. These devices are generally controlled by a group of intelligent software agents that sense,

reason and act to achieve certain goals on behalf of the user. Thus, an IE itself exhibits an ambient intelligence (AmI) quality that we as occupants perceive through environment adaptation.

An IE is able to recognise human occupants, reason with context and adapt itself to meet occupant needs by learning from their behaviour [3]. The University of Essex has a purpose built IE called the iSpace, which is a fully functioning apartment (complete with bedrooms, kitchen, bathrooms, *etc.*), that has been augmented by a plethora of sensors and imbued with AmI. The iSpace contains false walls and false ceilings, allowing devices to be embedded directly into the fabric of the apartment. By using a distributed architecture for device deployment and interconnection, the iSpace acts as a template for the creation of new spaces. This architecture accounts for technology heterogeneity by using gateways that act as proxies for individual technologies; each gateway presents virtual devices to the IE network and translates action invocations into native actions (using some suitable middleware, such as Universal Plug and Play (UPnP) [4,5]). In this way, the technology deployment within an IE can be heterogeneous, but the network is homogeneous, and so, intelligent software agents can communicate with any device that is connected. This virtualisation approach also permits the creation of abstract devices; to give a practical example: each individual light in an environment can be represented on the network to expose control functionality (on/off/set-level). This allows remote control of the lighting, as well as the brightness level of each individual lighting unit. Having control over individual units, while desirable in some circumstances, is not always convenient. Ideally, we would also have control over groups of lights (e.g., living room lights, kitchen lights, and so on). Our architecture allows for these abstract groupings to also be natively represented.

Each IE has a set of users, each user owning their own user profile and each user profile containing a unique set of preferences and applications. Applications has a specific meaning in this context; users can create their own applications by composing resources that are available within an IE on an *ad hoc* basis. It follows that an event occurring (e.g., a DVD player powering on) would trigger a rule in an intelligent agent, causing the agent to dim the living room lights and close the curtains. The user could call this their "movie application", and it would be unique to that particular user.

The majority of our prior research has been conducted at this apartment scale and has led us to engage in a project called ScaleUp that is investigating the theme of increased IE deployment size towards Large-Scale Intelligent Environments (LSIE). ScaleUp is a collaboration between the University of Essex (UK) and King Abdulaziz University (Saudi Arabia), with the aim of addressing the scalability issues associated with realising a real-world LSIE deployment. Unfortunately, it is not as simple as taking the existing principles and applying those to a larger physical space. As the space grows in physical size, it becomes exponentially more expensive in terms of implementation time, money and resource management. It is a natural progression for these environments to scale up, so it is vital that these scalability issues are resolved (the outline of this project is given in greater detail in a previous paper [6]).

We begin this paper by providing a brief overview of work that relates to LSIE realisation (Section 2). This is followed by an introduction to the use of bigraphical notation in describing IE systems (Section 3)—a methodology we are currently experimenting with and hope to make popular across the field. The focus of this paper proposes the use of IE composition to realise an LSIE and address the main scalability issues; we call this a MacroIE (Section 4). To illustrate this, we describe how the user profile that is associated with a person can follow them from IE to IE (an extension of our previous FollowMe

work). The work is made concrete through three implementation approaches (Section 5) and associated experimental results (Section 6).

2. Related Work

Some of the earliest purpose built IE examples are the "Intelligent room", built in Bristol, UK [7], and the iSpace, built at The University of Essex [8]. Since then, the field has made massive strides of progress, with other spaces being built; examples include workplace environments: the "smart lab" at the University of Deusto [9]; and home place environments: the Phillips "HomeLab"[10], which is a fully functional apartment similar to the iSpace. The Cisco "Internet House", while larger than an apartment, was built to show an environment with an always-on internet connection and appliances that could be controlled via the internet (it also is an example of how different research groups interpret an IE differently). The model that these other environments propose would allow them to fit into the model this paper proposes, creating potential opportunities for collaboration between research groups, which would be the first step towards the unification of research in this area.

Much of the existing literature within the field reports on work that has had a focus on the internal aspects of an IE, but there are indications that the field as a whole is starting to consider IE deployment on a larger scale [11–13].

The increasingly popular movement of the Internet of Things (IoT) [14] is converging towards an IE ideology [15]. IoT started life with the vision of creating an environment full of objects that were uniquely identified by pointers to a centralised database, in order to change a user's experience of that environment for the better [16,17]. This vision has evolved and now has been stated as *"Things having identities and virtual personalities operating in smart spaces using intelligent interfaces to connect and communicate within social, environmental and user contexts"* [17]. Although AmI was not part of the original IoT vision, the convergence of ideas has seen its incorporation [18,19]. The IoT suffers from the similar problems of IEs in that there is very apparent fragmentation in the research taking place; many different bodies actually supplying different, yet overlapping definitions of the IoT [11,20–22]. It will be interesting to follow the trajectory of this research to see if it starts to overlap with the field of IEs further.

Publications are starting to appear that note that there is a desire to start scaling up existing implementations of pervasive computing [23,24]. The fact that the majority of these publications only briefly touch upon the topic of inter-IE communication and scaling provides further emphasis on the originality and novelty of the proposed research area. Just as publications are starting to appear on scaling up these environments, other research groups have stated a need for a beneficial formal framework [25–27] through which to model and describe the various works. These papers tend to focus on implementations of proprietary methods for specific, existing implementations. It is apparent that the community needs a more abstract solution that would serve the entire community as a whole, rather than niche cases. Habib published a paper [28] on bringing together geographically separated IEs; however, this paper uses a virtual world in an attempt to bring together these geographically separated environments (similar to the concepts outlined in [29,30]). Our work is focussed more towards bringing together these environments in the physical world; though it would be feasible to say that these ideas could be implemented in the virtual world, as well. There has been some movement into creating larger

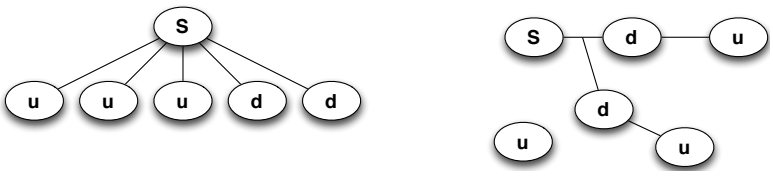
scale environments, such as [31], but this particular instance is aimed at creating large data sets from a series of individual environments.

3. Bigraphical Notation for IE Description

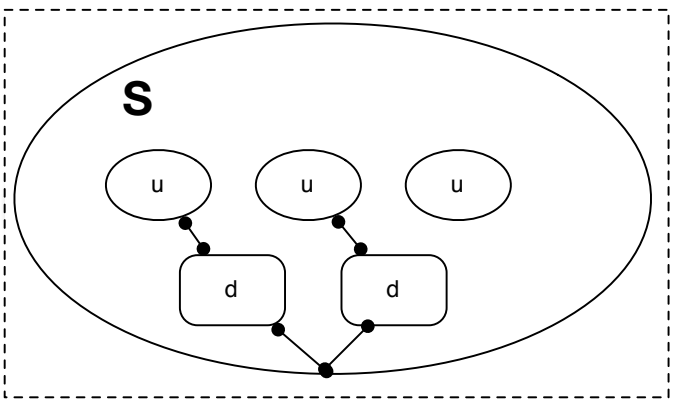
There is an outstanding and recognised need within the IE field for common formalisms that enable IE design to be described and modelled. This is synonymous with the need for Unified Markup Language (UML) to describe and model software systems. The need is reflected by the diversity of description frameworks used across the field, leaving comparison of different models difficult. To address this problem, we have been examining the use of bigraphical representations to describe IE problems and design; this section provides a brief overview of bigraphs as a primer for later sections.

Bigraphs offer a way to diagrammatically describe and represent a system, so that it can be easily understood by visual inspection, whilst simultaneously encoding structure and entity relationships [32]. This is all backed by mathematical principles that underpin the model and can be reasoned with robustly if the additional detail is required. As the name suggests, a bigraph consist of two graphs (Figure 1a): a place graph and a link graph. These two graphs share nodes. The place graph is restricted to being a tree (no cycles) and is contained within a forest, whereas a link graph tends to be a hyper-graph (a link can connect more than two objects) [33]. By representing both components of the bigraph in one picture (Figure 1b), we can get an impression of object locations and connections simultaneously [34].

Figure 1. This figure shows the underlying bigraphs (**a**) containing the link graph and place graph, respectively, and, then, the result of representing both the graphs in one picture (**b**). The information nodes repsent differ based on context; in this particular instance S represents a space, U represents a user and D represents a device.



(**a**)



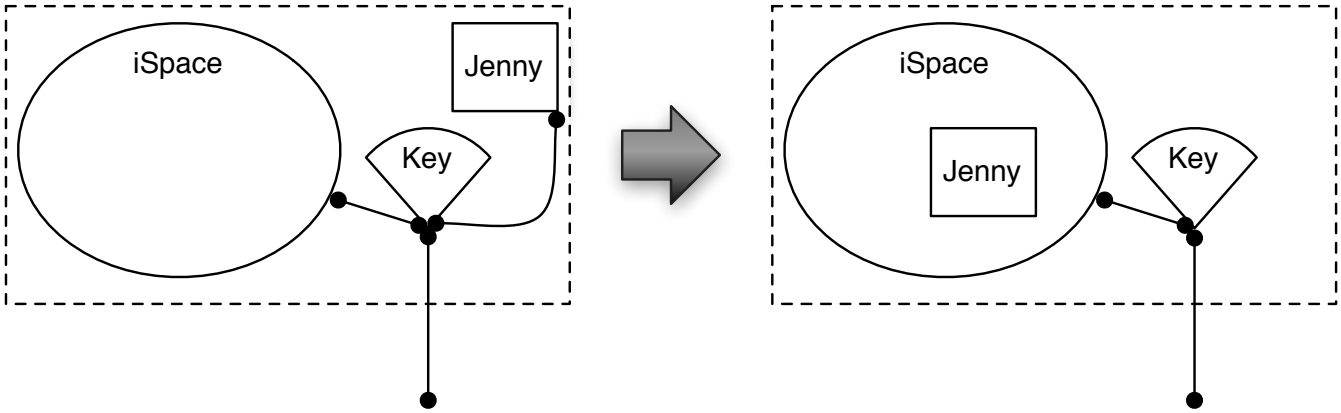
(**b**)

Bigraphical Reactive Systems introduced in [35] are a "*general graphical semantical framework for reactive systems unifying the feature of dynamic communication links introduced with the π-calculus* [36] *and the feature of mobile nested spatial structures introduced with the Mobile Ambients calculus* [37]". Simply put, Bigraphical Reactive Systems are comprised of a set of bigraphs and a set of reaction rules, which are used to reconfigure the bigraphs.

Bigraphs offer a simple understanding of newly introduced concepts and show possibilities for specifying behaviours at more than one level of abstraction. In Figure 1, you can see that the nodes in the graphs represent physical entities (people, spaces, iPads, *etc*.), but they also include the notion of connections or communication. Figure 1b shows users connected to a device (which could be a user typing on their iPad) and these devices connected to the space (which could show the wireless connection to the space's local area network). However, bigraphs can represent more than just the physical layer; the nodes themselves can represent conceptual ideas, too, such as passwords, agents, bits of software, *etc*. The idea of the nodes is to be quite general. This generality allows us to gain different perspectives on all aspects of a system, whether it be how a user gains entry to a space or how the agents within a space communicate with one another.

While bigraphs are a high-level, visual formal model, it is possible to break them down into their algebraic form. To give an example of their use, Figure 2 shows a lecturer, Jenny, transitioning from outside an IE to inside, using a persistent key (which, in this case, is a keypad on the wall next to the door).

Figure 2. Jenny uses the keypad to gain entry into the iSpace. The dotted line represents a region.



By simple inspection, these bigraphs are not very formal, but are designed to fit Milner's vision for a hierarchy-of-models [38]; thus, we need a mathematically precise description of what a bigraph is:

$$(V,E,\ ctrl,\ prnt,\ link) : \langle\ k,\ X\ \rangle \rightarrow \langle\ m,Y\ \rangle \tag{1}$$

where V is a finite set of nodes, E is the set of hyperedges, $ctrl$ is the control map that assigns controls to the nodes, $prnt$ is the parent map that defines tree structure (place graph) and $link$ is the link graph that defines the link structure [32].

This definition can be used to describe Figure 2 as:

$$/z.iSpace_z \mid /z.(key_{xz} \mid jenny_z) \rightarrow /z.iSpace_z(/z.jenny_z) \mid /y.(key_{xy}) \quad (2)$$

The dotted lines in Figure 2 represent regions; a region allows the bigraph to give a notion of locality for individual components (in this case, Jenny approaching the iSpace and uses the keypad on the wall of the iSpace to gain entry). We can represent more than one locality within a single bigraph.

Figure 3 shows that you may have multiple regions contained within the same bigraph which, in turn, means that the place graph will contain a forest of tree graphs, one for each region.

Figure 4 shows a scenario where Jenny is preparing a lecture in the iSpace, then physically travelling to the iClass (where the iSpace and iClass are geographically distal).

In the real world, it is likely that these environments will contain more than just a few members or a few devices; this could lead to the bigraphs becoming incredibly complex, very quickly. To counter this, it is possible to abstract away the intricate details, while keeping the notion that the space contains something. Consider that we wish to illustrate a member entering a space (as shown in Figure 5), but we don't need to know how the existing nodes within that environment are behaving. We can use a site to convey that the space contains other nodes, which will persist over the reaction rule, but are not relevant to the action performed. In Figure 5, there is a dotted square contained within S; this is the site. It provides enough information to say that S contains other nodes, but these nodes are not of interest to us in this reaction rule.

Figure 3. A user presents a key that enables an Intelligent Environment (IE) (the iSpace, in this case) to verify the user session. Upon success, the user is granted access to the space and the user profile is accessible directly from the trusted device. The encapsulated node TD depicts the user's trusted device.

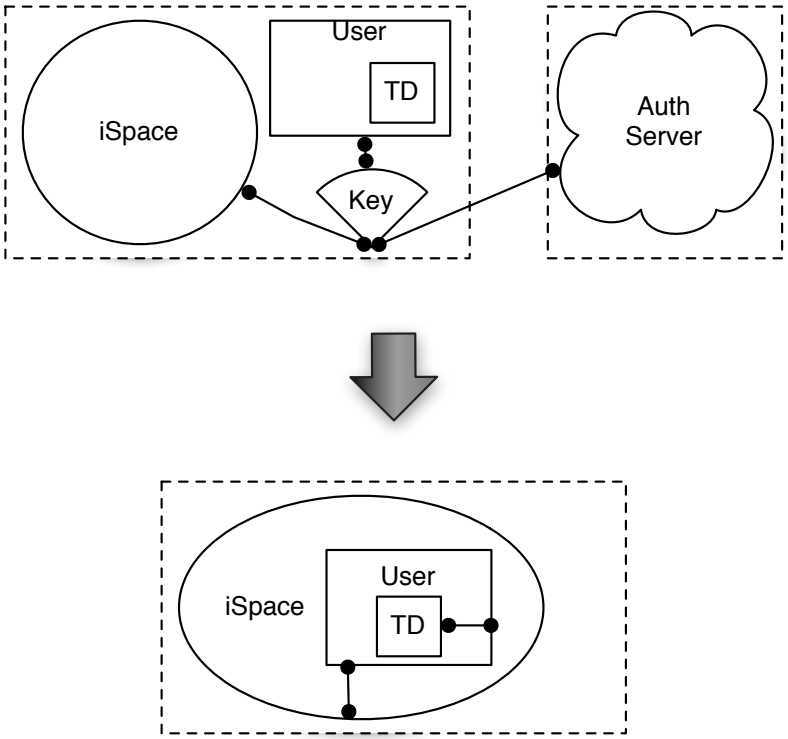


Figure 4. Jenny leaves the iSpace, walks over to the iClass and gets ready to use her one-time-use key to gain entry to the iClass.

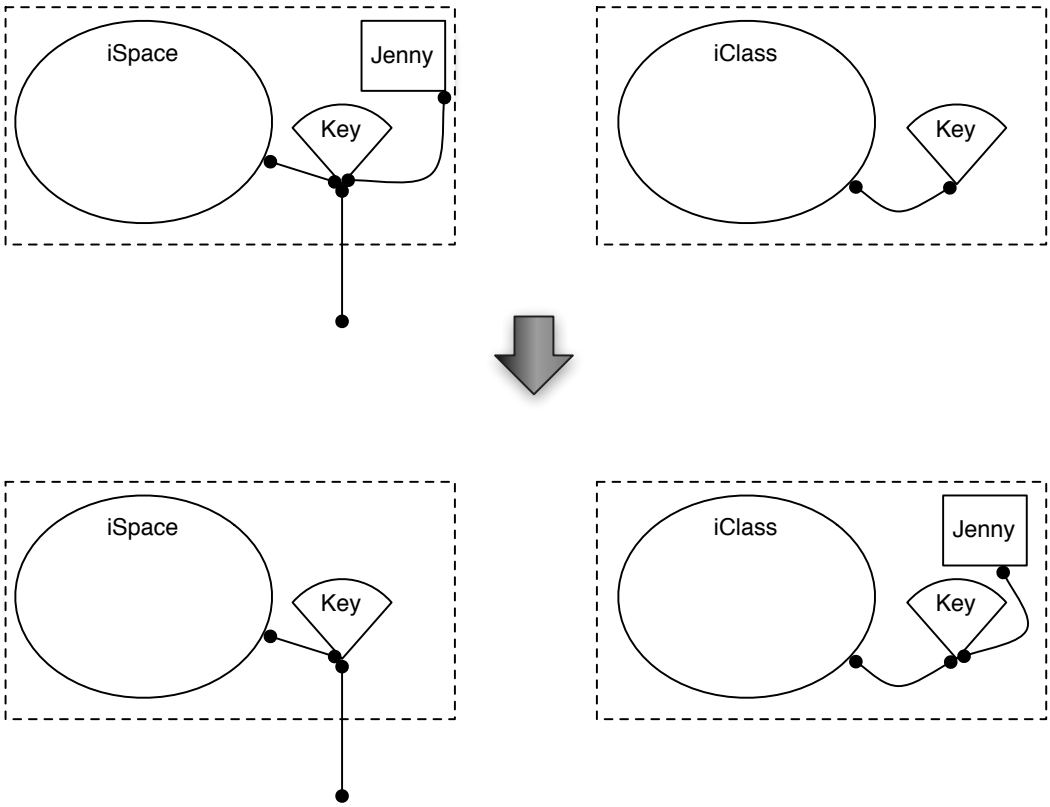
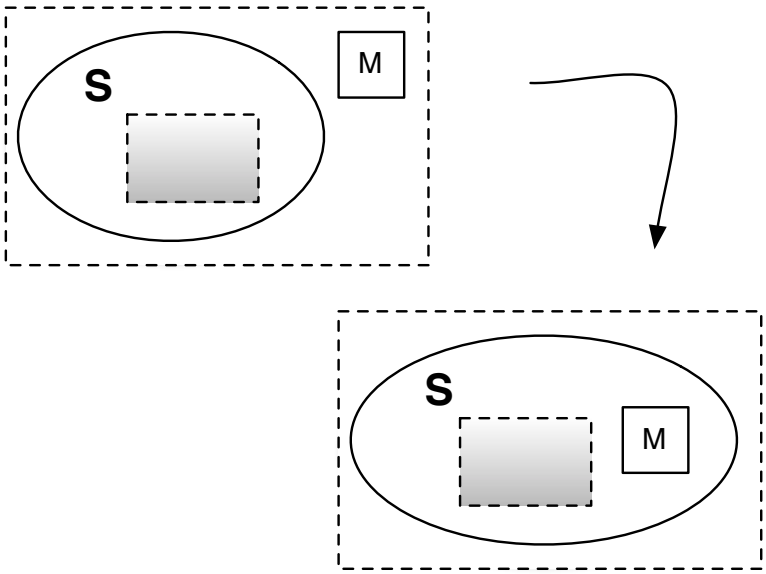


Figure 5. A reaction rule showing a member (M) entering a space (S).



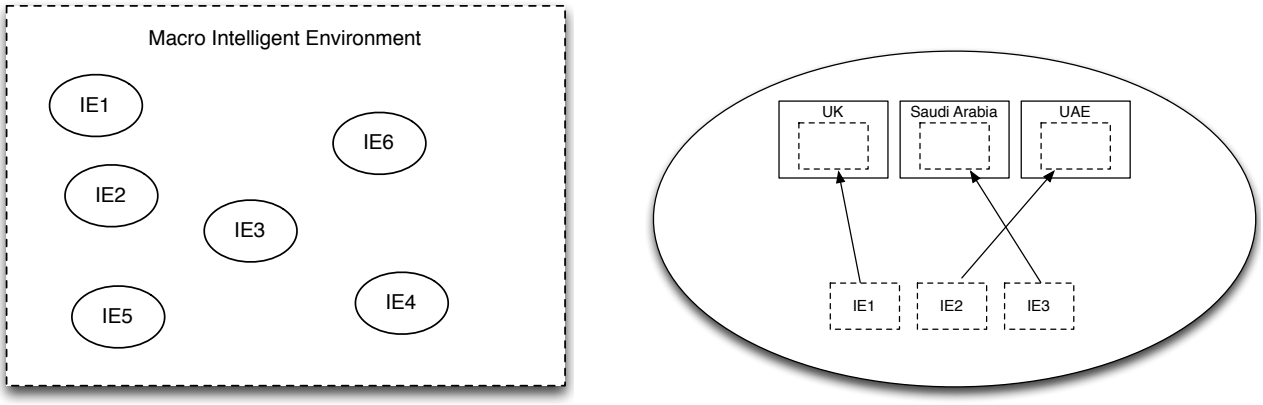
This section has provided a brief introduction to the basic bigraph concepts, such that the following sections are more easily understood. The use of bigraphs as a formal model for application within the IE field is an ongoing subject of research and is discussed at greater length in [39,40].

4. MacroIE: Realising an LSIE through Composition

In order to realise the creation of an LSIE, new approaches are required in order to account for scalability limitations of existing IE methods (from technical, security/privacy, management and usability perspectives). To solve these scalability problems, we propose the concept of a MacroIE (The word macro is used here as it is in the wider field of computer science—to define an input pattern that will create a larger, more complex output [41])—a single LSIE that is composed from a set of smaller IEs, rather than existing as a monolithic whole. This means each element of a MacroIE is autonomous, distributed and self-governing. This is in contrast to the Monolithic IE, which attempts a top-down approach to manage all the low-level details across the entire space.

Figure 6a shows the set of environments that are geographically co-located, but this need not be the case; the environments can be distal or proximal. The connection between each environment is electronic, so there is no requirement for them to be in the same campus, territory or, even, country. It is entirely feasible to interconnect several IEs from different countries to form a MacroIE that spans continents. Figure 6b illustrates that the regions represented in Figure 6a can be geographically sparse.

Figure 6. (**a**) Shows an Large-Scale Intelligent Environment (LSIE) that consists of a network of smaller, individual Intelligent Environments (IEs); (**b**) shows that these individual IEs have no requirement to be geographically co-located to be considered within a MacroIE.



(**a**) (**b**)

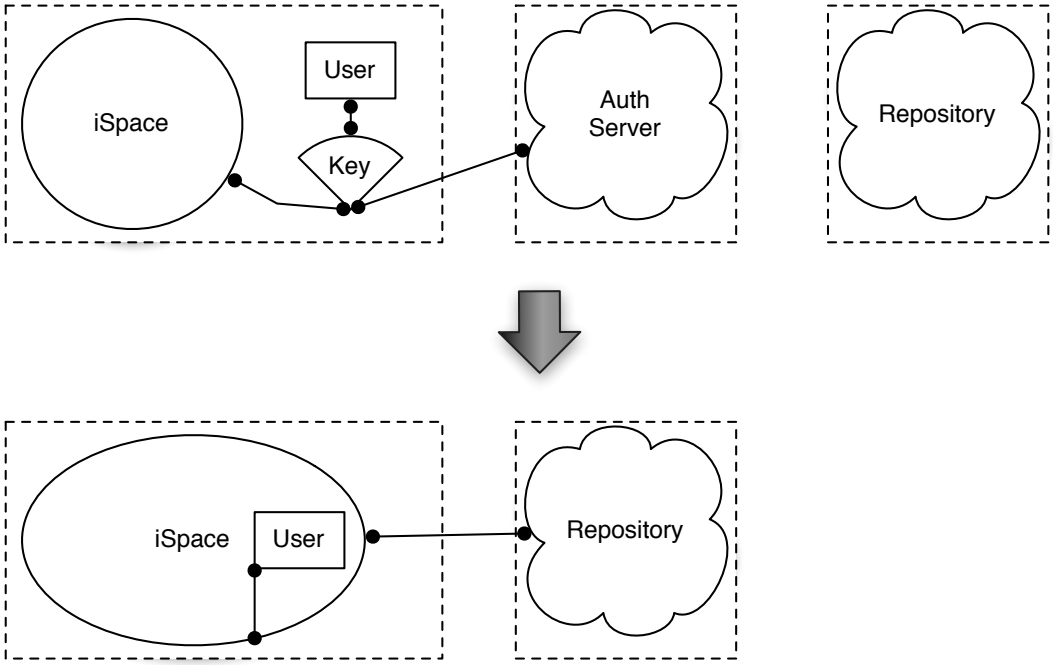
Let us consider an example in which an entire university campus is a MacroIE. Each department within that university could contain several IEs. It is intended that a user can roam freely among the component IEs that make up a MacroIE, while still enjoying technological transparency and continuity of experience. This is, of course, subject to security restrictions and access rights.

This suggests that there are multiple perspectives to the model; the user perspective, management perspective, and so on. The model is also designed to be dynamic, allowing individual environments to be added/removed with relative ease (i.e., a MacroIE has a modular structure). This introduces some interesting insights into the way security management would work; as each user may have a unique view of the overall MacroIE, the traditional role-based security or user-based security models may not fit, requiring an entirely new model to be created.

Traditional computer systems authenticate a user at initial session login [42]. This model can also be applied to an IE, requiring users to explicitly login, using some form of contextual credentials in order to access the assets within that environment. By realising a MacroIE, the login session can be shared among the component IEs, thus realising a continuity of experience without the obstructive need to create a new session when the user changes context (transitions from one IE to another). This is a concept we call FollowMe [43], as it enables a user profile to follow the user from place-to-place in an unobtrusive way. A continuity of experience is thus achieved by a user having continuous access to their digital assets and services, whilst also enjoying environment adaptation, where the spaces they inhabit are dynamically adjusted according to preference and context.

FollowMe forms a critical part of the MacroIE functionality/behaviour and influences both design and evaluation. We seek to realise FollowMe user profiles in an efficient, scalable and user-friendly way, similar to the way in which a mobile phone can roam between different cells, whilst maintaining minimal interruption to service. When a user roams between IEs, how does the environment obtain the relevant information (environment preferences, authentication and authorisation details, available applications) about that user? The abstraction of this problem is shown by the bigraph in Figure 7 and described below.

Figure 7. The key is used by the user to gain access to the Intelligent Environment.



The abstract solution shown in Figure 7 shows that once a user has gained access to the space, the profile for that user is acquired from somewhere and instantiated within that space. Although not explicitly mentioned, security has a vital role in the MacroIE model; thus, we have used a key to represent this security layer. This key could be any appropriate authorisation and authentication solution (something you know (e.g., username/password), something you have (e.g. Radio-Frequency Identification (RFID) tag) or something you are (biometrics) [44,45].) and is required by the user to gain access to the space (the need for that key disappears once the user has gained entry, as shown by the reaction rule).

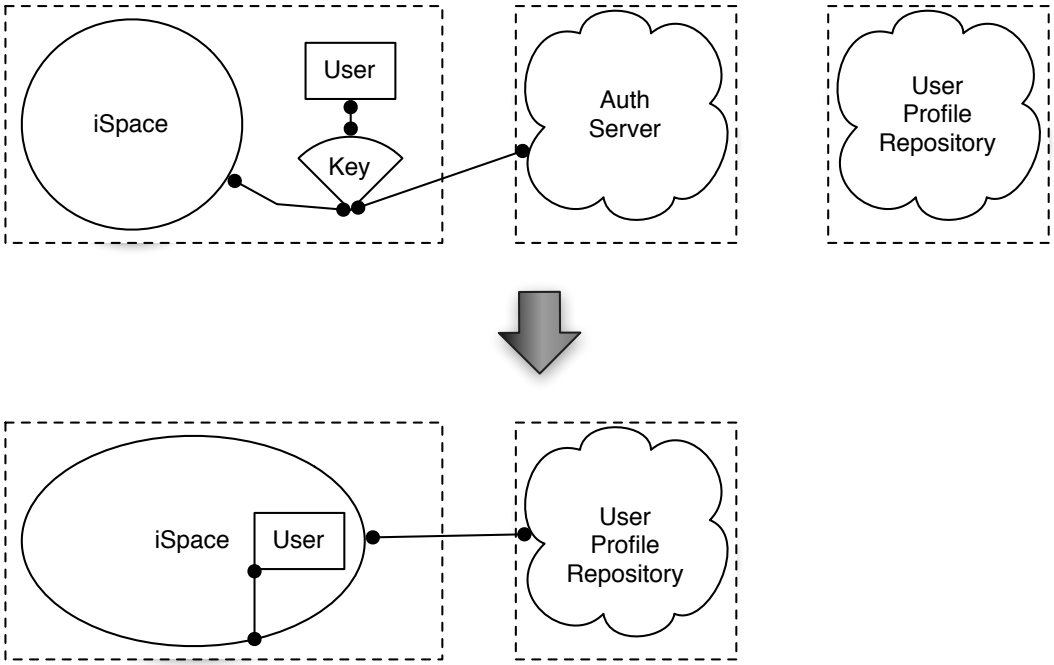
5. MacroIE Implementation

In this section, we propose three approaches to implementing the abstract MacroIE; the first relies solely on the storage of user profiles in a cloud repository, the second relies on the use of a trusted device that a user carries with them (for example, a mobile phone), and the final solution employs a combination of the two (utilising both cloud and trusted device functionality).

5.1. Implementation A: Cloud Repository

Cloud computing is a term used to describe the delivery and/or consumption of computing services over a network (usually the Internet) [46,47]. With cloud computing, the end user is not aware of the location of the service, but is merely concerned with the quality of the service. By using a cloud-based approach to store FollowMe user profiles, an authenticated user session will grant permission for an IE to retrieve certain portions of the user profile and instantiate them within the IE. As shown in Figure 8, the notion of some form of key is still required to identify the user and either trigger a login or session transfer. Upon success, the reaction rule shows that the key and authentication server play no further role. Each individual component in Figure 8 is shown to be contained within separate regions [48], but it is possible for them to be contained within the same region.

Figure 8. A user presents a key that enables an IE (the iSpace, in this case) to verify the user session by communicating with an authentication server. Upon success, the user is granted access to the space, and the user profile is accessible from the user profile repository.



Take this example: Jenny is a lecturer at the University of Essex. She is currently working in the iSpace preparing her next lecture. The time approaches for her to present the lecture, and so, she leaves the iSpace; everything she has been working on is being uploaded and stored in the cloud as she walks to the iClassroom. When she arrives at the iClassroom, she gains entry by waving her RFID tag over the

reader, prompting her profile be downloaded from the cloud, which recognises her status as the teacher and automatically adjusts the lights in the classroom for teaching mode. She pulls up the lecture slides she was previously working on in the iSpace and is almost immediately prepared for the students to arrive in the classroom.

This scenario immediately presents some potential issues that will need to be resolved; if a connection cannot be established to the server, how can the user be authenticated and the profile made accessible? Does the environment load a blank profile locally and attempt to synchronise at given intervals? The use of formal methods to model, discover and solve these problems is therefore essential in the design process.

Of course, a cloud-based user-repository can be used for far more than simply storing, syncing and serving user profiles as static data packages; one of the more useful benefits of using a cloud-based approach is that it can be used as a scalable resource to host the execution of applications and deliver services remotely. The effect of this would manifest in each IE that the user visits (through service delivery). These services would be synonymous with the services provided by the space itself; the user would be unaware of what was providing the services, just that the services are available for consumption. One such function could be a messaging service; take the previous example of our lecturer, Jenny. Fellow lecturer Ingrid wishes to contact Jenny immediately but does not know where she is. The server would know which—if any—IE Jenny was currently active in and, providing Ingrid was also in a similar environment, she could send a message to Jenny via the server—acting as a simple routing host. Perspectives of the MacroIE were mentioned earlier, and this holds true for the services; as a simple user, the instant messaging service would allow communication between all their friends, but from a management point of view it would be possible to see who is active in which space. This is similar to how instant messaging traditionally works, but illustrates how the cloud server could be used as more than just a file repository. This leads into the concept of one's entire presence following them around the MacroIE, their preferences, their documents, communications...the possibilities are endless!

5.2. Implementation B: Trusted Device

A trusted device is a personal device that the user would carry around with them the majority of the time and have a level of ownership over to entrust certain rights. The most obvious current example would be a smart-phone, given their ubiquity in society and ability to run powerful applications. This trusted device would contain all the relevant information about the user on the device itself; upon successful user authentication, the environment would then retrieve the information from the device via an *ad hoc* network connection. This method has the advantage of not requiring an external connection (or a complex network), thus being ideal for remote locations, where a strong internet connection is not available (e.g., the International Space Station).

Figure 8 shows that the user still requires a key to gain entry to the IE (that could be verified against an authentication server or could be based on certificate validation). Advances in mobile technology (particularly Near Field Communications (NFC)) mean that the trusted device could be used to also present the key.

We go back to our lecturer, Jenny. Again, she is working in the iSpace, ready for her lecture in the iClassroom. This time when she leaves the iSpace for the classroom, the information is stored on her trusted device (which, in this case, is her smartphone). She is logged out of the iSpace when her trusted device is out of range of the environment (either the environment's local area network or some location system, such as Ubisense). On the way to the classroom, Jenny makes changes on the trusted device that affect her profile. When she comes into range of the classroom, the trusted device automatically connects to the network and sends a handshake message; if accepted, she will be logged into the space and granted access. As before, the information will be pulled off the device (including the new changes), and she will be ready to teach.

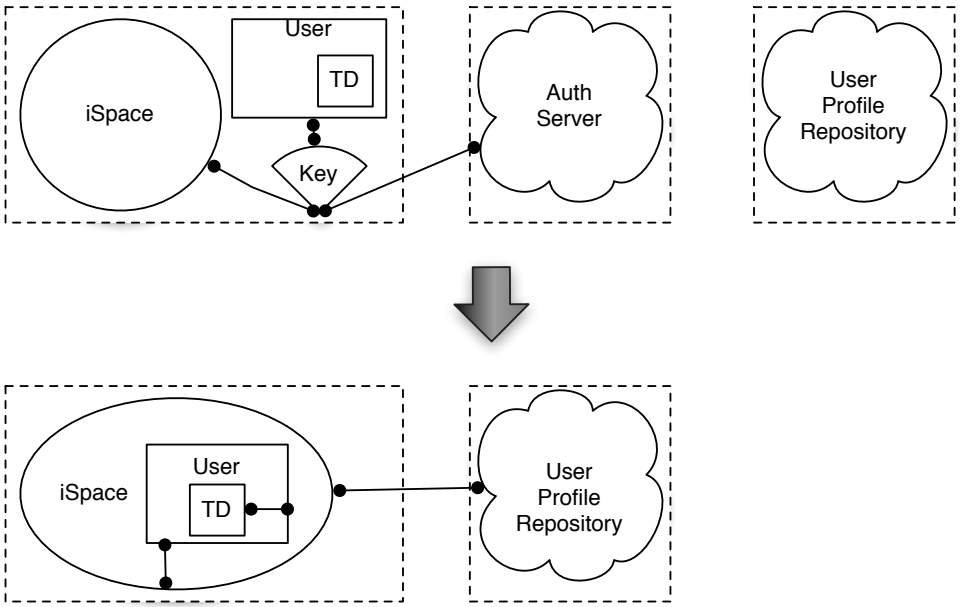
This presents some potential issues that will need to be resolved. In an ideal situation, every member would have the exact same trusted device, but in reality, the fragmentation of smartphone devices is somewhat prevalent in the consumer market, so it is of utmost importance that a suitable standard be established for this kind of interaction.

5.3. Implementation B: Hybrid

While there are situations in which each of the two previous implementation approaches are best suited, a hybrid approach could address the majority of use-cases and do so in a way that addresses the respective shortcomings.

Figure 9 shows that the user requires a key to gain entry to the IE. The reaction rule shows that once the user is authenticated, the space has an active connection to both the trusted device and the user profile repository. This allows access to both a user profile and cloud-hosted services.

Figure 9. A user presents a key that enables an Intelligent Environment (the iSpace, in this case) to verify the user session. Upon success, the user is granted access to the space, and the user profile is accessible from both the trusted device (TD) and the user profile repository.



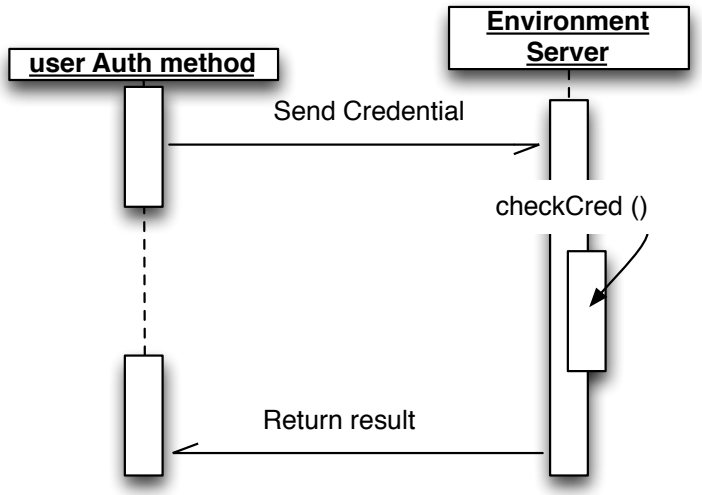
This hybrid approach provides a more complete system, as it inevitably cuts out some of the problems presented. The intent behind this approach is to use the trusted device to perform session-authentication and to store part of the user profile (for example, the more frequently used and static content), while the remainder is accessible from the cloud-based repository. The hybrid approach also permits the use of cloud infrastructure to host the execution of services and deliver them to the local user IE. By using this approach, a more efficient and fault tolerant experience can be delivered to the end-user.

6. Experimental Results

This section presents the results derived from initial experiments with prototype cloud and trusted device implementations. Each prototype provides functionality for the storage and retrieval of Extensible Markup Language (XML) encoded user profiles. Repetition was used to collect 100 samples for each experiment, where each sample records the amount of time required to acquire the user profile from the respective repository (cloud or trusted device-based). While there are several other metrics that we are also interested in, these initial experiments reflect the extremely important user requirement that these environments operate in a robust and real-time fashion.

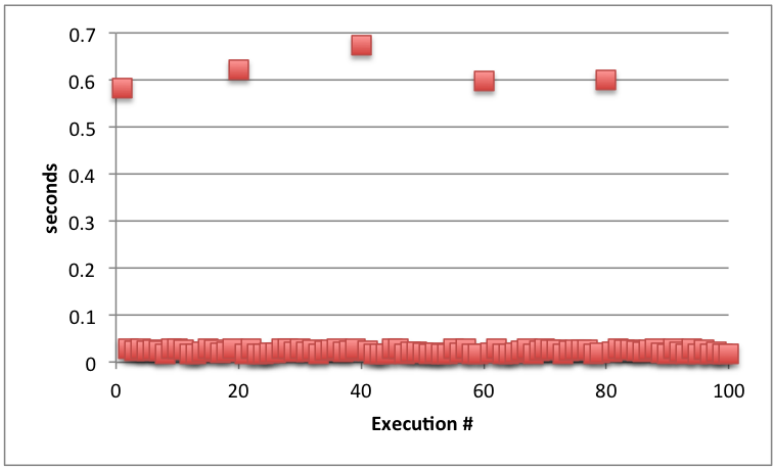
The trusted device implementation was built using an iPhone 4 (running iOS 5.1.1) and written in Objective-C, with an SQLite database to store the XML encoded user profiles. An Android implementation has also been developed and tested, producing results consistent with the iPhone. An application was developed to perform user authentication by sending encrypted credentials (a unique token from the device, which is registered to the user profile); the environment checks that the token is valid and, upon success, authenticates the user assigned to that token in the environment. This is not the most efficient method of performing authentication, especially as there are many use cases where this would not work; but for the proof-of-concept experimentation, it was adequate. Figure 10 shows a simple UML sequence diagram showing the abstract authentication method of both the trusted device and the cloud server.

Figure 10. Unified Markup Language (UML) sequence diagram of the user authentication in an environment.



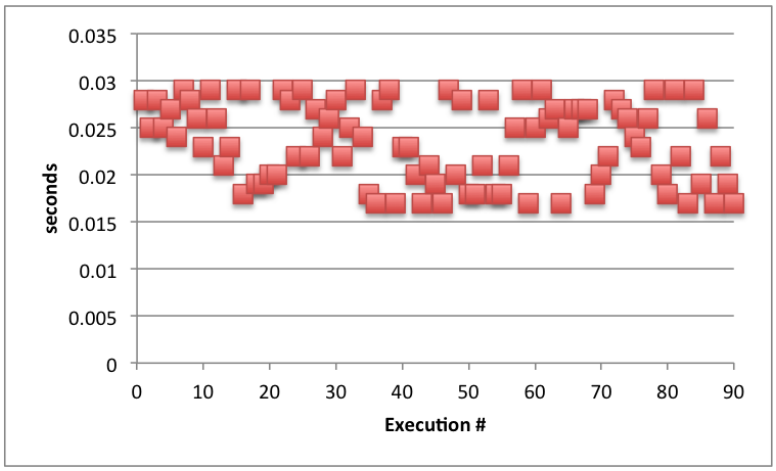
All experiments were conducted in the Essex iSpace and iClassroom [29,49]. The two implementations were identical in functionality, with the trusted device implementation having an additional module to perform dynamic discovery (this was done via the UPnP protocol). By comparison, the network address of the cloud server was hardcoded into the IE software (for the sake of simplicity). When recording the execution times, the broadcast and discovery of the trusted device was not taken into account. The results for the trusted device experiments can be seen in Figure 11 below and show a good cluster between 18 and 28 ms; in terms of ensuring that the environment would adapt quickly to a user's presence, this is certainly acceptable.

Figure 11. Scatter graph showing the raw results from the trusted device experiments.



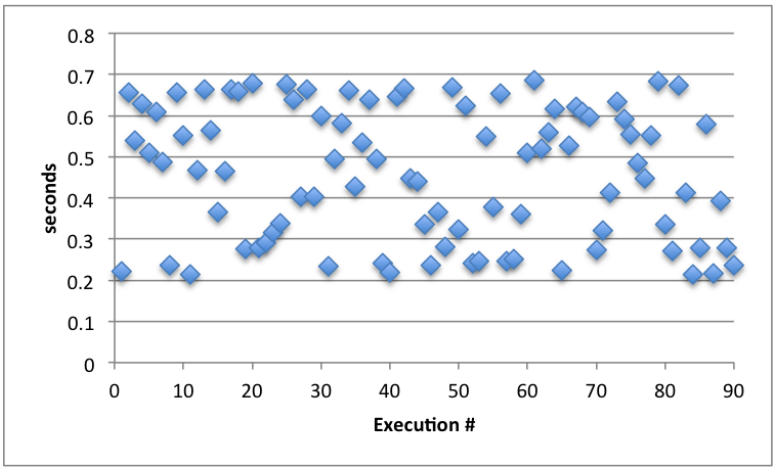
However, after each run of 20 samples, there is an abnormally high result. This was always the first execution of the device; we can account for these erroneous results due to software libraries being loaded, memory initialisation and network related delays. If we drop the top and bottom five percent of the results, the remainder lie between 18 and 28 ms (as shown in Figure 12). This results in an average time of 20 ms for the trusted device—with a standard deviation of 0.004—proving that the results are very consistent.

Figure 12. Scatter graph showing the results with the erroneous data points removed.



The results of the cloud-based experiments are consistent with the performance of the trusted device; however, in the case of the cloud repository, these times are spread over a wider range than the trusted device with an average time of 46 ms, spread over a standard deviation of 0.17, which is considerably larger than the deviation of the trusted device. This is attributed to the fact the user profile was being sent over a larger network (the Internet) in comparison to the trusted device (that was connected to the local network). Figure 13 illustrates this. As with Figure 12, the upper and lower five percent of the results were removed from Figure 13.

Figure 13. Scatter graph showing the sparse results of the cloud-based implementation experiments.



The cloud server was based around a Jersey server implementation (The Jersey framework implements the Java API for RESTful Web Services (JAX-RS) reference, but also provides its own Application Programming Interface (API) that extends the toolkit). This was supplemented by a simple MySQL database to allow the user profiles to persist between transactions. More technical information about the cloud server solution can be found in [50].

An overlay of the results from Figure 13 with Figure 12 is shown in Figure 14. It is clear that the performance of the trusted device solution is superior, as reflected by the quicker average performance time (23 ms *vs*. 46 ms respectively) and consistently performing within a tighter margin (highlighted by the standard deviation of the two; 0.004 and 0.17, respectively, again).

However, the results presented exclude the dynamic-discovery overhead incurred by the UPnP framework required to find the trusted device on the network. Figure 15 shows another overlay of the cloud and trusted device results, but this time, the trusted device results include the overhead incurred by dynamic-discovery. It is plain to see that the results are now similar, as reflected by the new average and standard deviation for the trusted device; 35 ms and 0.16, respectively. While the trusted device still performs faster on average, the spread of these results is very similar to that of the cloud server. This is an inherent problem of using a dynamic-discovery mechanism to locate resources at runtime [51] and provides an indication for an area of improvement in the trusted device implementation.

Figure 14. Scatter graph showing both the trusted device and cloud results.

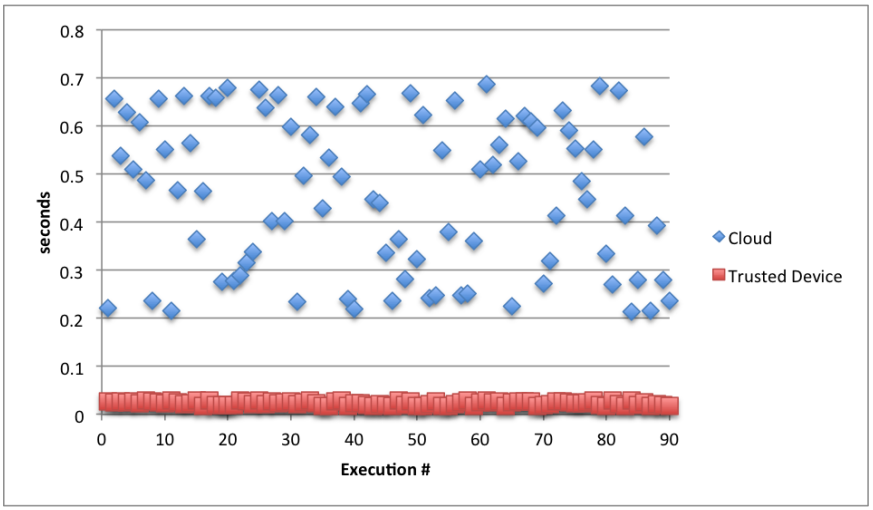
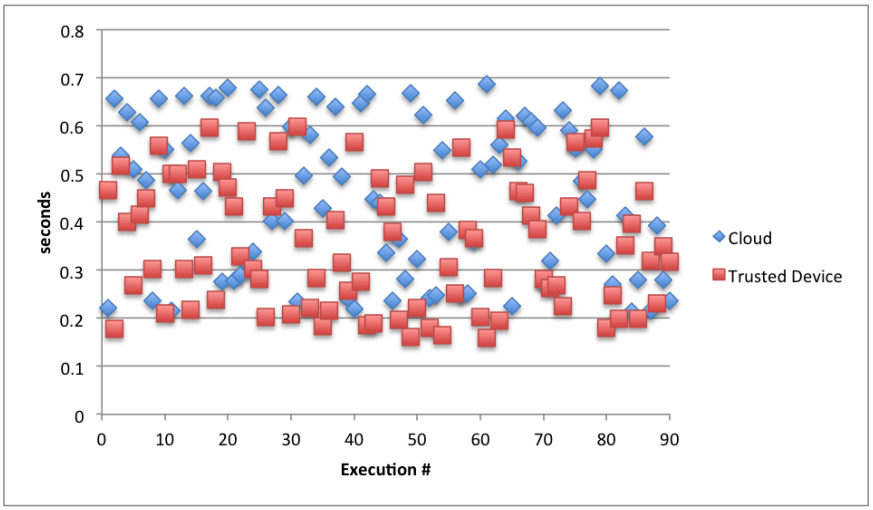


Figure 15. Scatter graph showing both the cloud and full trusted device results.



7. Conclusions and Future Work

As work in the field of Intelligent Environments continues towards large-scale real-world deployment, there is a clear and present need to not only solve the technical problems, but to also address the issues surrounding description and modelling of such systems. Through our numerous works, we are investigating these problems, whilst also remaining cognisant as to the critical nature of usability and social acceptance. This paper has introduced the area of Large-Scale Intelligent Environments (LSIE), in general, and has provided an introduction to the use of bigraphs as a formal method for description and modelling. We have also introduced the MacroIE model as a proposed solution to the scalability problems surrounding LSIE realisation. To provide some context to this model, we have also described implementation strategies, built two prototypes and provided experimental results regarding performance times of those prototypes.

The results that we have presented are promising and help identify the benefits of the implemented solutions, whilst also enabling us to select parts of the solution that need optimisation.

In addition to the many scalability benefits of the MacroIE model, it also addresses a key real-world problem of heterogeneity—the model is not concerned about the specific implementation of each individual IE, but relies on encapsulation to abstract an IE into a homogeneous user profile API. We are currently updating our experimental labs to conform to this model, so that we may conduct more in-depth studies beyond the proof-of-concept work reported here. We hope that this will enable us to verify and refine the model, whilst also allowing us to establish some best practices (in particular, which implementations work best under which circumstances).

Acknowledgments

This work has been undertaken as part of the ScaleUp project, which is funded by King Abdulaziz University, Saudi Arabia.

Conflict of Interest

The authors declare no conflict of interest.

References

1. Roalter, L.; Moller, A.; Diewald, S.; Kranz, M. Developing Intelligent Environments: A Development Tool Chain for Creation, Testing and Simulation of Smart and Intelligent Environments. In Proceedings of the 7th International Conference on Intelligent Environments (IE'11), Nottingham, UK, 28 July 2011; pp. 214–221.
2. Gellersen, H.; Beigl, M.; Krull, H. The MediaCup: Awareness Technology Embedded in an Everyday Object. In Proceedings of the 1st International Symposium on Handheld and Ubiquitous Computing, Karlsruhe, Germany, 27 September 1999; pp. 308–310.
3. Weber, W.; Rabaey, J.; Aarts, E. *Ambient Intelligence*; Springer-Verlag: Berlin/Heidelberg, Germany, 2005.
4. Miller, B.A.; Nixon, T.; Tai, C.; Wood, M. Home networking with Universal Plug and Play. *IEEE Commun. Mag.* **2001**, *39*, 104–109.
5. UPnP Forum Web Site. 2012. Available online: http://www.upnp.org (accessed on 5 December 2012).
6. Dooley, J.; Henson, M.; Callaghan, V.; Hagras, H.; Al-Ghazzawi, D.; Malibari, A.; Al-Haddad, M.; Al-Ghamdi, A. A Formal Model for Space Based Ubiquitous Computing. In Proceedings of the 7th International Conference on Intelligent Environments (IE11), Nottingham, UK, 28 July 2011; pp. 74–79.
7. Cohen, M. Towards Interactive Environments: The Intelligent Room. In Proceedings of the 1997 Conference on Human Computer Interaction, San Francisco, USA, 24 August 1997.
8. Hagras, H.; Callaghan, V.; Colley, M.; Clarke, G.; Pounds-Cornish, A.; Duman, H. Creating an ambient-intelligence environment using embedded agents. *IEEE Intell. Syst.* **2004**, *19*, 12–20.
9. Lopez-de Ipina, D.; Almeida, A.; Aguilera, U.; Larizgoitia, I.; Laiseca, X.; Orduna, P.; Barbier, A.; Vazquez, J. Dynamic Discovery and Semantic Reasoning for Next Generation Intelligent

Environments. In Proceedings of the 4th International Conference on Intelligent Environments (IE08), Seattle, USA, 21 July 2008; pp. 1–10.
10. De Ruyter, B.; Aarts, E. Ambient Intelligence: Visualizing the Future. In Proceedings of the Working Conference on Advanced Visual interfaces (AVI04), Gallipoli, Italy, 25 May 2004; pp. 203–208.
11. CERP-IoT. Internet of Things Strategic Research Roadmap, 2011. Available online: http://www.grifs-project.eu/data/File/CERP-IoT%20SRA_IoT_v11.pdf (accessed on 4 December 2012).
12. Whittington, L.; Dooley, J.; Henson, M.; Al-Ghamdi, A.A.M. Towards FollowMe User Profiles in Macro Intelligent Environments. In Proceedings of the 1st Workshop On Large Scale Intelligent Environments (WoLSIE), Guanajuato, Mexico, 27 June 2012; pp. 179–190.
13. Schaffers, H.; Sallstrom, A.; Pallot, M.; Hernandez-Munoz, J.; Santoro, R.; Trousse, B. Integrating Living Labs with Future Internet Experimental Platforms for Co-creating Services within Smart Cities. In Proceedings of the 2011 17th International Conference on Concurrent Enterprising (ICE), Aachen, Germany, 21 June 2011; pp. 1–11.
14. Ashton, K. That 'Internet of Things' thing, 22 June 2009. Available online: http://www.rfidjournal.com/articles/view?4986 (accessed on 5 December 2012).
15. Atzori, L.; Iera, A.; Morabito, G. The internet of things: A survey. *Comput. Netw.* **2010**, *54*, 2787–2805.
16. Magrassi, P.; Berg, T. A World of Smart Objects. Gartner Research Report R-17–2243, 2002.
17. Botterman, M. Internet of things: An early reality of the future internet, 10 May 2009. Available online: ftp://ftp.cordis.europa.eu/pub/fp7/ict/docs/enet/iot-prague-workshop-report-vfinal-20090706_en.pdf (accessed on 5 December 2012).
18. Uckelmann, D.; Isenberg, M.; Teucke, M.; Halfar, H.; Scholz-Reiter, B. An Integrative Approach on Autonomous Control and the Internet of Things. In *Unique Radio Innovation for the 21st Century: Building Scalable and Global RFID Networks*; Springer, London, UK, 2010; pp. 163–181.
19. Angulo-Lopez, P.; Jimenez-Perez, G. Collaborative Agents Framework for the Internet of Things. In Proceedings of the 8th International Conference on Intelligent Environments (IE'12). Guanajuato, Mexico, 27 June 2012; pp. 191–199.
20. Casagras. Casagras IOT Definition. 2011. Available online: http://cordis.europa.eu/search/index.cfm?fuseaction=news.document&N_RCN=30283 (accessed on 4 December 2012).
21. Research, S. SAP IOT Definition. 2011. Available online: http://services.future-internet.eu/images/1/16/A4_Things_Haller.pdf (accessed on 4 December 2012).
22. EPOSS, E. ETP EPOSS IOT Definition. 2011. Available online: http://old.smart-systems-integration.org/internet-of-things/Internet-of-Things_in_2020_EC-EPoSS_Workshop_Report_-2008_v3.pdf/download (accessed on 4 December 2012).
23. Cook, D.; Das, S. Pervasive computing at scale: Transforming the state of the art. *Pervasuve Mob. Comput.* **2012**, *8*, 22–35.
24. Guerrero-Ibanez, J.; Flores-Cortes, C.; Ramirez-Alcaraz, J.; Vizcaino-Anaya, H.; Mendoza-Robles, T.; Anguiano-Mancilla, A.; Pena-Cerdenas, E. Intelligent Signaling for

Prevention of Intersection Collisions in Urban Zones. In Proceedings of the 8th International Conference on Intelligent Environments (IE'12), Guanajuato, Mexico, 27 June 2012; pp. 200–207.
25. Augusto, J.; McCullagh, P.; Augusto-Walkden, J.A. Living without a safety net in an Intelligent Environment. *ICST Trans. Ambient Syst.* **2011**, *11*, 10–12.
26. Augusto, J.; Hornos, M. Using Simulation and Verification to Inform the Development of Intelligent Environments. In Proceedings of the 8th International Conference on Intelligent Environments (IE'12), Guanajuato, Mexico, 27 June 2012; pp. 413–424.
27. Baquero, R.; Rodriguez, J.; Mendoza, S.; Decouchant, D. MidBlocks: A Supervising Middleware for Reliable Intelligent Environments. In Proceedings of the 8th International Conference on Intelligent Environments (IE'12), Guanajuato, Mexico, 27 June 2012; pp. 389–400.
28. Habib, M.K. Collaborative and Distributed Intelligent Environment Merging Virtual and Physical Realities. In Proceedings of 5th IEEE International Conference on Digital Ecosystems and Technologies (DEST), Daejeon, South Korea, 31 May 2011; pp. 340–344.
29. Dooley, J.; Davies, M.; Ball, M.; Callaghan, V.; Hagras, H.; Colley, M.; Gardner, M. Decloaking Big Brother: Demonstrating Intelligent Environments. In Proceedings of the 6th International Conference on Intelligent Environments (IE10), Kuala Lumpur, Malaysia , 21 July 2010; pp. 324–327.
30. Jianhua, M.; Yang, L.; Apduhan, B.; Runhe, H.; Barolli, L.; Takizawa, M. A Walkthrough from Smart Spaces to Smart Hyperspaces towards a Smart World With Ubiquitous Intelligence. In Proceedings 11th International Conference on Parallel and Distributed Systems, Fukuoka, Japan, 20 July 2005; pp. 370–376.
31. Crandall, A.; Cook, D. Smart Home in a Box: A Large Scale Smart Home Deployment. In Proceedings of the 8th International Conference on Intelligent Environments (IE'12), Guanajuato, Mexico, 27 June 2012; pp. 169–178.
32. Milner, R. *The Space and Motion of Communicating Agents*; Cambridge University Press: Cambridge, UK, 2009.
33. IT University of Copenhagen, D. A Brief Introduction to Bigraphs. 2012. Available online: http://www.itu.dk/research/pls/wiki/index.php/A_Brief_Introduction_To_Bigraphs (accessed on 4 December 2012).
34. Perrone, G. Domain-Specific Modelling Languages in Bigraphs. Ph.D. Thesis, IT University of Copenhagen, Copenhagen, Denmark, 2013.
35. Birkedal, L.; Bundgaard, M.; Damgaard, T.; Debois, S.; Elsborg, E.; Glenstrup, A.; Hildebrandt, T.; Milner, R.; Niss, H. Bigraphical Programming Languages for Pervasive Computing. In Proceedings of the 1st International Workshop on Combining Theory and Systems Building in Pervasive Computing, 7 May 2006; pp. 653–658
36. Milner, R.; Parrow, J.; Walker, D. A calculus of mobile processes. *Part I. Inf. Comput.* **1992**, *1*, 1–40.
37. Cardelli, L.; Gordon, A. Mobile Ambients. In *Proceedings of FoSSaCS'98*; Springer-Verlag: Lisbon, Portugal, 1998; pp. 140–155.
38. Milner, R. Ubiquitous computing: Shall we understand it? *Comput. J.* **2006**, *49*, 383–399.

39. Henson, M.; Dooley, J.; Whittington, L.; Al-Ghamdi, A. Towards Simple and Effective Formal Methods for Intelligent Environments. In Proceedings of the 8th International Conference on Intelligent Environments (IE'12), Guanajuato, Mexico, 27 June 2012.
40. Henson, M.; Dooley, J.; Whittington, L.; Al-Ghamdi, A. FollowMe: A Bigraphical Approach. In Proceedings of the Workshop Proceedings of 8th International Conference on Intelligent Environments (IE'12), Guanajuato, Mexico, 27 June 2012; pp. 434–445.
41. Greenwald, I.D.; Kane, M. The Share 709 System: Programming and Modification. *J. ACM* **1959**, *6*, 128–133.
42. Niinuma, K.; Unsang, P.; Jain, A. Soft biometric traits for continuous user authentication. *IEEE Trans. Inf. Forensics Secur.* **2010**, *5*, 771–780.
43. Dooley, J.; Wagner, C.; Hagras, H.; Pruvost, G. FollowMe: The Persisent GUI. In Proceedings of the 1st International Workshop on Situated Computing for Pervasive Environments (SCOPE-2011) at 6th International Symposium on Parallel Computing in Electrical Engineering (PARELEC-2011), Luton, UK, 3 April 2011; pp. 123–126.
44. Federal Financial Institutions Examination Council, Authentication in an Internet Banking Environment. 2012. Available online: http://www.ffiec.gov/pdf/authentication_guidance.pdf (accessed on 5 December 2012).
45. Creese, S.; Goldsmith, M.; Roscoe, B.; Zakiuddin, I. Authentication for pervasive computing. In Proceedings of the International conference on Security in Pervasive Computing, 12–14 March 2003; Volume 2802, pp. 116–129.
46. Weiss, A. Computing in the clouds. *ACM Netw.* **2007**, *11*, 16 – 25.
47. Jadeja, Y.; Modi, K. Cloud Computing-concepts, Architecture and Challenges. In Proceedings of the 2012 International Conference on Computing, Electronics and Electrical Technologies (ICCEET), Tamil Nadu, India, 21 March 2012; pp. 877–880.
48. Hildebrandt, T.; Winther, J. Bigraphs and (reactive) XML—an XML-centric model of computation. IT University Technical Report Series, TR-2005-56; The IT University of Copenhagen: Copenhagen, Denmark, 2005.
49. Dooley, J. An Information Centric Architecture for Large Scale Description and Discovery of Ubiquitous Computing Objects. Ph.D. Thesis, University of Essex, Colchester, UK, 2010.
50. Bilgin, A.; Dooley, J.; Whittington, L.; Hagras, H.; Henson, M.; Wagner, C.; Malibari, A.; Al-Ghamdi, A.; Alhaddad, M.; Alghazzawi, D. Dynamic Profile-Selection for zSlices Based Type-2 Fuzzy Agent Controlling Multi-User Ambient Intelligent Environments. In Proceedings of the International Conference on Fuzzy System, Brisbane, Australia, 10 June 2012; pp. 1–8.
51. Dooley, J.; Callaghan, V.; Hagras, H.; Bull, P. Discovering the Home: Advanced concepts. In Proceedings of the ICADIWT '09. Second International Conference on the Applications of Digital Information and Web Technologies, London, UK, 4 August 2009; pp. 494–499.