Workshop Proceedings of the 8th International Conference on Intelligent Environments
J.A. Botía et al. (Eds.)
IOS Press, 2012

doi:10.3233/978-1-61499-080-2-179

Towards FollowMe User Profiles for Macro Intelligent Environments

Luke WHITTINGTON [a], James DOOLEY [a,b]. Martin HENSON [a] and
Abdullah AL-MALAISE AL-GHAMDI [a,b]

[a] *Department of Computer Science, University of Essex, UK. Email: lwhittl@essex.ac.uk*

[b] *Faculty of Computing and Information Technology, King Abdulaziz University, KSA*

Abstract. We envision an Ambient Intelligent Environment as an environment with technology embedded within the framework of that environment to help enhance an users experience in that environment. Existing implementations , while working effectively, are themselves an expensive and time consuming investment. Applying the same expertise to an environment on a monolithic scale is very inefficient, and thus, will require a different approach. In this paper, we present this problem, propose theoretical solutions that would solve this problem, with the guise of experimentally verifying and comparing these approaches, as well as a formal method to model the entire scenario.

Keywords. ambient intelligent environments, formal methods, mobile devices, cloud server, bi-graphs

1. Introduction

Intelligent environments are a wonderful example of ubiquitous computing[1], but ask two different researchers what an Intelligent Environment is and you'll probably get two very different answers. In our vision, Intelligent Environments tend to be common spaces (such as the home or the office) which contain a plethora of embedded computer devices that help enrich user experiences. These devices are generally interconnected by a network infrastructure (most likely a Local Area Network (LAN)), and controlled by a group of intelligent agents. Environments have a set of users, with each user profile containing their own set of preferences and applications.

Applications have a specific meaning in this context, for example: each light in the environment has a Universal Plug and Play (uPnP) wrapper which exposes functionality of the light across the network. This means that the light power can be changed, as well the dimming level of the light. Having control over individual lights, while beneficial, isn't exactly desirable. Ideally we would have control over *groups* of lights (e.g. *living room* lights, *kitchen* lights and so on). It follows that upon an event triggering (e.g. a DVD player powering on), this could trigger a rule in the intelligent agent, causing the living room lights to be dimmed and the curtains to automatically close. This would be called the "movie application" - users can create these applications as they need and/or desire.

These environments are able to recognise human occupants, reason with context and program itself to meet user(s) needs by learning from their behaviour [2]. The University of Essex has a purpose built Intelligent Environment, the iSpace, which is a fully functioning apartment (complete with bedrooms, kitchen, bathrooms etc.), which has been augmented by intelligent agents and a plethora of sensors. The iSpace contains false walls and false ceilings, allowing devices to be embedded directly into the framework of the apartment. This is used as a model deployment and is a template to shape new spaces such as the iClassroom and an office based scenario.

ScaleUp is a research project, led by the Intelligent Environments Group (IEG) at The University of Essex. The aim of the ScaleUp project is achieving large scale Intelligent Environments. This project has already been outlined in a previous paper published by the IEG [3]. The existing implementations tend to be the size of a single apartment, which in their current format present scaling issues when attempting to create monolithic sized environments. It is natural progression that these environments need to scale up so it is vital that these scalability issues are solved. In this paper, we introduce the problem in more depth (Section 3) and then move onto the proposed solutions, either a centralised user-profile respiratory, a quite literal ”mobile” user profile or a hybridisation of the two.

2. Initial Survey of existing Work

During the initial survey of existing work on large scale intelligent environments, two issues become apparent very quickly; while there has been a significant amount of publication on the topic of intelligent environments[4][5], there is significant fragmentation in the language used to describe these environments (even to the extent whereby the environment itself has different naming conventions[6]). This fragmentation causes difficulties ensuring the different research groups are talking about the same subject. In order to keep consistency with the IEG at Essex, we shall be using the language expressed in their published papers[7,8].

Some of the earliest examples of a purpose built intelligent environments are the ”Intelligent room” built in Bristol, UK [9] and the iSpace built at The University of Essex [10]. Since then, the field has made massive strides of progress, with other spaces being built; examples including workplace environments: the ”smart lab” at the University of Deusto and recreational purposes[4] and home place environments: the Phillips ”HomeLab”[11], which is a fully functional apartment similar to IEG’s iSpace. The Cisco ”Internet House”, while larger than an apartment, was built to show an environment with an ”always-on” internet connection and appliances which could be controlled via the internet. The theory 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 hopefully lead towards the unification of research in this area.

The other issues is that of all the major publications read, rather than how these environments interact and communicate between each other, the focus was on the *internals* of an intelligent environment[12]. Naturally, there is a lot of novel research still required for the internals of intelligent environments, ranging from the automation of these environments (through the use of embedded agents)[13,14], to the Human-Computer Interfacing[15,16]; it is vital this research continues as the model presented in this paper, means that each environment is self-governed and thus will still require a high level of individual intelligence. M. Habib published a paper [17] on bringing together geographically separated intelligent environments. However, this paper uses a virtual world in an attempt to bring together these geographically separated environments (similar to the concepts in [7,18]). We are aiming more at bringing together these environments in the physical world; though it would be feasible to say these ideas could be implemented in the virtual world as well. At the time of writing, this is the only paper to solely focus on merging smaller scale environments into a larger one. However, it has been noted that there is a need to start scaling up existing implementations of pervasive computing [19]. The fact that the majority of these publications only briefly touch upon the topic on the communication between intelligent environments and scaling, provides further emphasis on the originality and novelty of the proposed research area.

3. Problem Definition

Most computer systems authenticate a user at initial login session [20] and Intelligent Environments conform to this paradigm; requiring users to explicitly *"login"*, using some form of contextual credentials in order to access the assets within that environment. In their current format, scaling up the existing implementations presents issues; keeping track of multiple users in a large area, ensuring each user has continuity of experience and in particular the *"humanistic"* problem of continually being forced to re-authenticate oneself via entry of username and password. This is the equivalent of having to dig your front door key out of the bottom of your bag each time you wish to enter your house, it would ideal for the house to simply *"know"* who to grant access to. To solve these scalability problems, the project envisions creating a set of independent spaces that are treated as a traversable set rather than a monolithic environment.

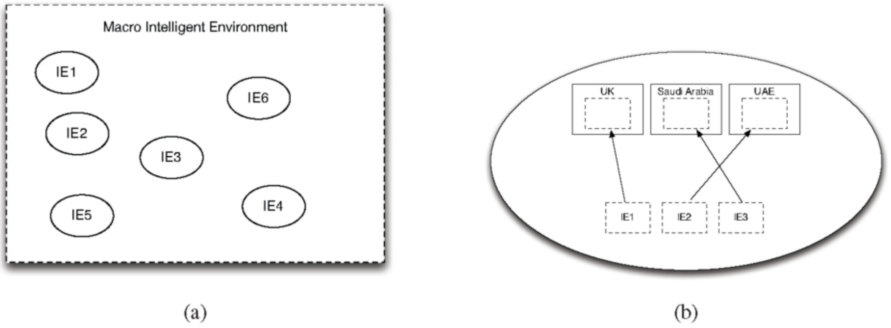

Figure 1. (a) shows a large scale IE actually consists of a network of smaller, individual IEs. (b) shows that these individual IEs have no requirement to be geographically co-located to be considered within a Macro IE

Figure 1.a 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 connections between each environment is electronic so there is no requirement for them to be on the same campus, territory or even country. You'll notice in figure 1.a that the border is dotted; this represents a "region" [21], a concept which is discovered in detail in section 5. It is entirely feasible to create these Large Scale Intelligent Environments (as shown in

Figure 1.a) in Essex, Saudi Arabia and the United Arab Emirates, and as long as they can communicate electronically then they may exist as a Macro Intelligent Environment (Figure 1.b illustrates that the ”regions” represented in Figure 1.a can be geographically sparse).

Users will have a set of environments that they are familiar with (there exists a mutual acceptance of data between the user and the space), and can roam freely between each environment (assuming they have the authorisation to do so). Solving this problem is significant because the core aspect of an ”Intelligent World” is the ability to roam freely between these individual ”islands” (individual environments), while enjoying technological transparency and continuity of experience[22,23,24]. In order to ensure the user experience is consistent across multiple environments, each user must have a profile that is mobile and can follow them around. In this way, the continuity of user experience can be maintained.

As a consequence, it is important that the users profile follows them between environments. When a user roams between multiple, individual environments, how does the environment obtain the relevant information (environment preferences, authentication and authorisation details, available applications) about that user?

Another advantage with using a set of traversable, individual environments is it allows ”fences” to be created within the Macro Intelligent Environment. Using a university as an example, the entire campus would be the Macro Intelligent Environment, M. Each department within that university could contain 2-3 of their own individual environments, IE. A student who studies Mathematics probably wouldn’t gain any advantage of using an environment in the English department, so it logically follows that the Mathematics student can only access the environments associated with his department; but you still want to keep that technological transparency and continuity of experience between the environments he has authorisation to use.

This creates ”virtual” large scale intelligent environments, within an existing large scale intelligent environment. The beauty of this solution is that it can keep going down as many layers as is required. These ”virtual” large scale intelligent environments augment the work started in [7]. The internals of these environments are not perfect either; the technology currently used to expose the assets within the environments has a large margin to improve[25]. While these are important problems that need to be addressed (especially if these environments are to be used in real time computing), they are out of the scope of this paper.

So far, we have talked about Macro Intelligent Environments from a *management* (top down) point of view, but there is another vital perspective that needs to be addressed; the user’s perspective. The model proposed for Macro Intelligent Environments is very dynamic; individual environments can be introduced and removed with relative ease. With the ability to create departments, as mentioned above, it also means that each user’s perspective of the environment may differ. This also introduces some interesting insights into the way security management would work for MIE; the traditional role based security or user based security models may not fit, forcing an entirely new model to be created.

4. Proposed Solutions

Generally, in computer science, the word macro is used to define an input pattern (usually abbreviated) that will create a larger, more complex output[26]. However, the word *macro* is from the greek ($\mu\alpha\kappa\rho\sigma$) for ”big” or ”far” and it is an amalgamation of these definitions we used when creating ”Macro Intelligent Environments”. A Macro Intelligent Environment is a composition of several smaller Intelligent Environment instances that are integrated to form the whole. This means each member of a Macro Intelligent Environment is autonomous and self governing, providing an abstraction to the Macro Intelligent Environment. This is in contrast to the Monolithic Intelligent Environment that would attempt to manage all the low level details for all the spaces from the top down. It is possible to create an abstract outline for all the proposed solutions:

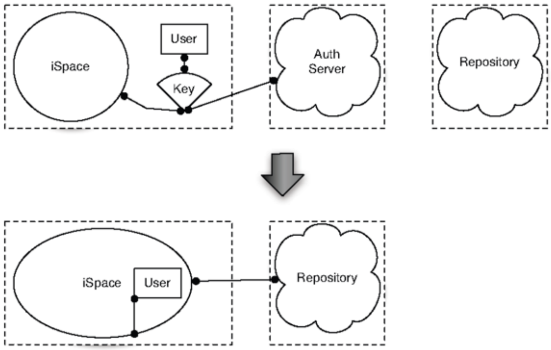

Figure 2. The key is used by the user to grant access to the Intelligent Environment

The similarities between all the solutions are such that once a user has gained access to the space, their user profile is acquired from *somewhere*. Although not explicitly mentioned, security is a vital role in Macro Intelligent Environments, thus we have used a ”key” to represent this security layer. This key could be any appropriate security solution: something you know (username/password), something you have (RFID tag) or who someone is (biometrics)[27,28]. There will need to be another server which contains these authorisation and authentication. Figure 2 shows that the user requires some form of key to gain access to the space (and the use of that key disappears once the user has gained entry, as shown in the reaction rule in Figure 2). Figure 2 looks very similar to Figure 3 presented in *4.1* - however one fundamental difference is the repository isn’t explicitly defined.

In order to create our Macro Intelligent Environments, there are a plethora of suitable solutions available, this paper proposes three solutions to the problem:

4.1. Cloud Server

Cloud computing is a marketing term used for the delivery and/or consumption of computing services over a network (usually the internet). With cloud computing the end user isn’t aware of the implementation of the server but is merely concerned with the delivery of the service. With this solution, the user information will be stored on the server, and upon successful user authentication, the environment will attempt to retrieve the user's profile from the cloud server. This method still requires the user to carry around a trusted device / token of some description but it would only be used to gain entry to the environments. As shown in Figure 3, the notion of some form of key is still required to gain entry to the environment, but upon successful entry, the reaction rule shows that the key and authentication server play no further role. Each individual component in Figure 3 is shown to be contained within separate "regions"[21] but it is possible for them to be contained within the same region.

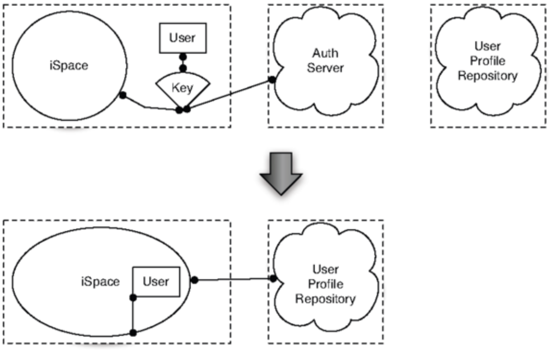

Figure 3. User 1 enters the iSpace and it immediately communicates with the Centralised User Repository.

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 said lecture 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, is the user not granted access? Does the environment load a blank profile locally and attempt to synchronise at given intervals? The advantage of using formal methods to model this is we can anticipate for this sort of problem, and hopefully come up with a suitable solution during the modelling stage (rather than hacking around at code in the last minute!).

This description of the cloud server sounds like a simple file host, serving and storing files when requested but this would be severely underusing the power of server side technology available. We envision the server actually providing services to the user, which would be available in most, if not all, of the environments that user has access to. 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 is a messaging service; take the previous example of our lecturer, Jenny. Fellow lecturer Ingrid wishes to contact Jenny immediately but doesn't know where she is. The server would know whether Jenny was currently active in an Intelligent Environment, 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. 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 onto the concept of ones entire presence following them around the Macro Intelligent Environment, their preferences, their documents, communications...the possibilities are endless.

4.2. Trusted Device

A trusted device is a device that the user would carry around with them the majority of the time. The most obvious example would be a smartphone, and seeing how smartphones allow the deployment of powerful applications, it makes them suitable to play this role. 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. This method has the advantage of not required an internet connection (or a complex network); thus being ideal for remote locations where a strong internet connection isn't available (e.g the International Space Station).

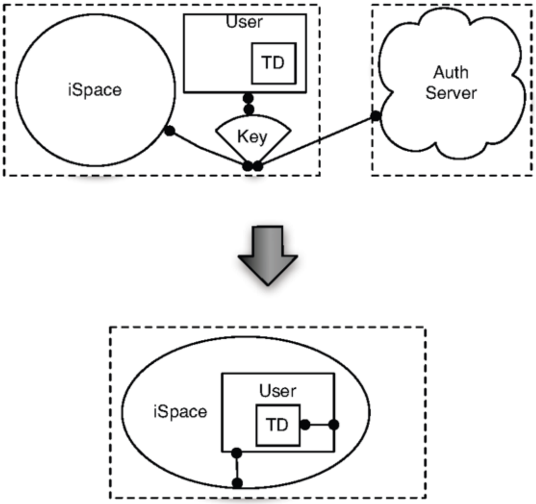

Figure 4. User 1 enters the iSpace and it immediately communicates with the Trusted Device to learn about the user.

Figure 4 shows the user still requires a key to gain entry to the IE (subsequently also requiring the server architecture for authentication); however, with the advances in mobile technology (particularly Near Field Communications) it may be possible to combine the trusted device and key. Once the user has been granted access to the space then no interaction occurs externally from the intelligent environment (as shown in the reaction rule of Figure 4).

We cut back to our lecturer Jenny. Again, she is working in the iSpace, ready for her lesson 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 environments 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, 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.

As before, 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 we keep the core logic written in a way that is supported by the majority of devices available, thus resulting in only having to develop a unique presentation layer for the different devices.

4.3. Hybridisation of A. and B.

Subsequently, comparing the two previous approaches to determine the most appropriate situation in which to implement each solution, could lead to a series of scenarios where one solution outshines the other and vice versa. Following on, the idea of a hybrid solution could solve most of the potential issues raised by each individual solution.

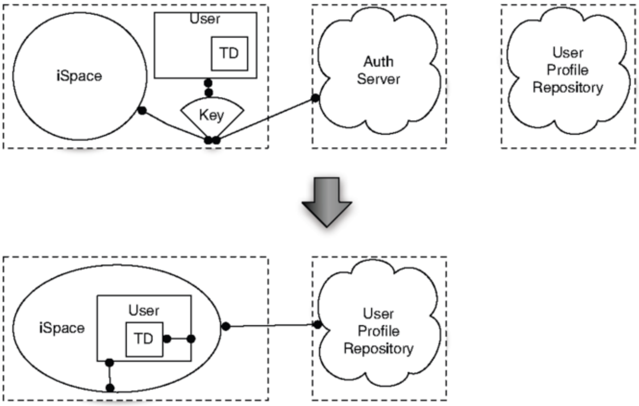

Figure 5. User 1 enters the iSpace and it immediately communicates with the Trusted Device to learn about the user, then sends this information to the Centralised User Repository.

As with Figures 2-4, the user requires a key to gain entry to the intelligent Environment. Figure 5's reaction rule shows that once the user is authenticated that the space has an active connection to both the trusted device and the Centralised User Repository.

The hybridisation of the previous two solutions provides a more complete system as it inevitably cuts out some of the problems presented. The idea behind this approach is to use the trusted device to store the user's profile and the server to also store the user's profile (as well as other files that may be too large to fit on a trusted device). When the user authenticates, the profile is pulled from the trusted device and any subsequent changes are saved to both the device and the server.

5. Introduction of Formal Methods

When designing a system, it is typically good practice to at least sketch a design first. For example, when designing a C++ program, you may use UML diagrams to show the class inheritance structure of the program. This gives you an insight into the properties of the system you're designing; how all the components interact, and wherein any potential problems may arise. The advantage of formal models is they usually give substantial evidence that these properties will hold.

Usually, Formal Methods are only used in the development of high-integrity systems (due to the associated cost with using Formal Methods)[29]. With Macro Intelligent Environment's vast size, it isn't only a case of designing a framework but essentially it is an entire eco-system being created; this requires extensive planning and modelling to ensure the proposed solution is going to work. Obviously, with a system of this size, not every problem can be anticipated at the design stage but it worth the time taken to do so.

Fragmentation of research into Intelligent Environments was mentioned earlier; different research groups have very differing opinions on what an Intelligent Environment actually compromises. Due to this, there hasn't been much research into creating a formal model for Intelligent Environments. This raises questions on whether the research will be into the formal methods of Macro Intelligent Environments or into the middleware; but that is synonymous with asking whether programming is about UML or software development. It is the entire model that we are trying to capture, at both the design stage and the implementation stage too.

The concept of using Bi-graphs as a formal framework for description, design and analysis is discussed in [30], though the ideas have been around for longer. As the name suggests, Bi-Graphs consist of two graphs; a *place graph* and a *link graph*. These two graphs share nodes. the place graph is restricted to a tree (no cycles), whereas a link graph tends to be a hyper-graph (a link can connect more than two objects)[31]. By representing both components of the bi-graph in one picture, we can get an impression of object locations and connections simultaneously[7].

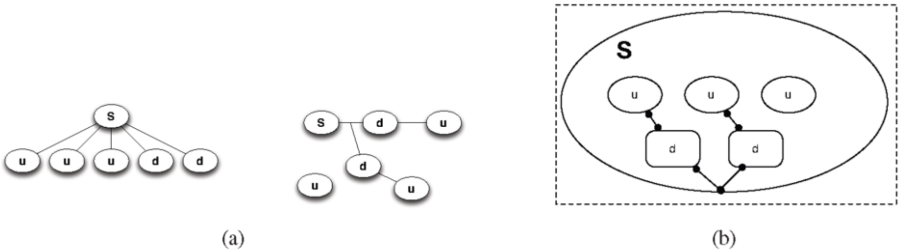

Figure 6. This figure shows the underlying bigraphs ((a) and then the result of representing both the graphs in one picture ((b).

Bi-graphs offer a simple understanding of newly introduced concepts and show possibilities for specifying behaviours at more than one level of abstraction. All the figures shown in this paper are themselves Bi-graphs. While Bi-Graphs are a high level, visual formal model, it is possible to break them down into their algebraic form (however this is not within the scope of this paper). To give an example of their use, figure 7 shows

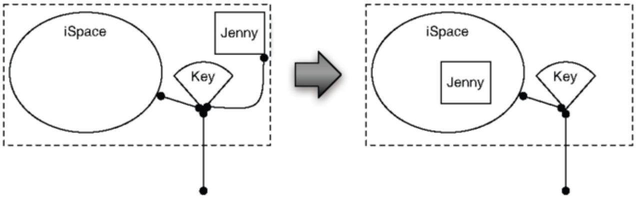

Figure 7. Jenny uses the keypad to gain entry into the iSpace. The key persist as it contains a link outside of the region.

our lecturer Jenny transitioning from outside an Intelligent Environment to inside, using a persistent key (which in this case, is a keypad on the wall next to the door).

You'll notice the dotted line in Figure 7, this represents a region. It allows the Bi-Graph to give a notion of locality for individual components (this case, Jenny approaching the iSpace and using the keypad on the wall of the iSpace to gain entry). Figure 8 shows that you may have multiple regions contained within the same bi-graph, which in turn means that the place graph will contain a forest of tree graphs, one for each region.

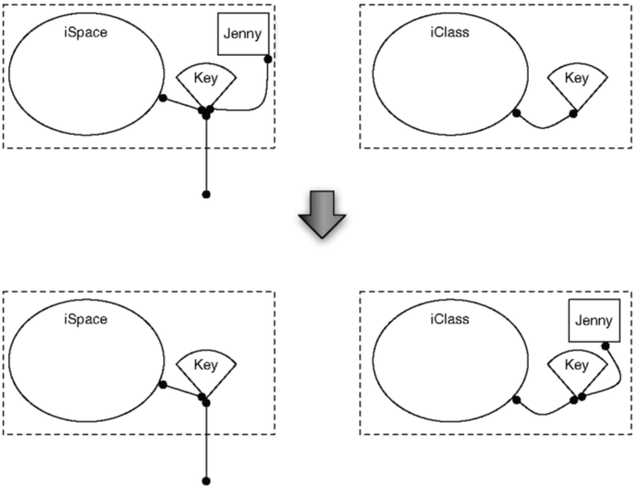

Figure 8. Jenny leaves the iSpace, walks over to the iClass and gets ready to use her one-time-use key to gain entry to the iClass.

Figure 8 shows the scenario explained *4.1* where Jenny is preparing a lecture in the iSpace, then physically travels to the iClass; which is geographically located elsewhere from the iSpace. We have only scratched the surface on using bi-graphs as a formal method of modelling Intelligent Environments but there is far more use than modelling the simple descriptions contained within this paper. It ranges from using a trace graph to predict how a system will react depending on which order reaction rules take place to using non-determinism to capture the complex details of a reaction rule. The use of bi-graphs as a formal model for Intelligent Environments is introduced in further detail in [32].

6. Conclusion and future work

It is the natural progression for these Intelligent Environments to start increasing in size, and this paper presents the one of the major problems with expanding the existing implementations. The solutions presented are only *theoretical solutions*, and while initial experimentation work has started (in the form of a paper published in the 2012 IEEE International Conference on Fuzzy Systems[14]), there is still much work to undertake implementing these solutions and experimentally verifying the outcomes. An in depth comparison of the approaches will also need to be undertaken, in order to fully appreciate with solution works best, and in what scenario. One interesting outcome will be whether the hybrid approach is a *"one-size fits all"* solution or whether the other approaches outshine it in certain situations. The connection between the theoretical practice of formal methods in Intelligent Environments and the practical implementation based off these models is a vaguely explored area, yet is proving to be an exciting one. We are hoping to use this work as a springboard to create interest and a community around this topic in order to further explore it.

Acknowledgments

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

References

[1] L. Roalter, A. Moller, S. Diewald, and M. Kranz. Developing intelligent environments: A development tool chain for creation, testing and simulation of smart and intelligent environments. In *7th Internation Conference on Intelligent Environments (IE'11)*, Nottingham, 2011.

[2] W. Weber, J. Rabaey, and E. Aarts. *Ambient Intelligence*. The Netherlands: Springer-Verlag Berlin Heidelberg, 2005.

[3] J. Dooley, M. Henson, V. Callaghan, H. Hagras, D. Al-Ghazzawi, A. Malibari, M. Al-Haddad, and A. Al-Ghamdi. A formal model for space based ubiquitous computing. In *7th International Conference on Intelligent Environments (IE11)*, Nottingham, 2011.

[4] D. Lopez-de Ipina, A. Almeida, U. Aguilera, I. Larizgoitia, X. Laiseca, P. Orduna, A. Barbier, and J.I. Vazquez. Dynamic discovery and semantic reasoning for next generation intelligent environments. In *4th International Conference on Intelligent Environments (IE08)*, Seattle, USA, 2008.

[5] R. Bose and A. Helal. Distributed mechanisms for enabling virtual sensors in service oriented intelligent environments. In *4th International Conference on Intelligent Environments (IE08)*, 2008.

[6] A. Kameas. Towards the next generation of ambient intelligent environments. In *In proceedings of 19th IEEE International Workshop on Enabling Technologies: Infrastructures for Collaborative Enterprises (WETICE)*, pages 1–6, 2010.

[7] J. Dooley, M. Davies, M. Ball, V. Callaghan, H. Hagras, M. Colley, and M. Gardner. Decloaking big brother : Demonstrating intelligent environments. In *6th International Conference on Intelligent Environments (IE10)*, Kuala Lumpur, 2010.

[8] M. Ball, V. Callaghan, and M. Gardner. An adjustable-autonomy agent for intelligent environments. In *6th International Conference on Intelligent Environments (IE'10)*, Kuala Lumpur, 2010.

[9] M. Cohen. Towards interactive environments: The intelligent room. In *Proceedings of the 1997 Conference on Human Computer Interaction*, Bristol, U.K., 1997.

[10] H. Hagras, V. Callaghan, M. Colley, G. Clarke, A. Pounds-Cornish, and H. Duman. Creating an ambient-intelligence environment using embedded agents. *IEEE Intelligent Systems*, 19(6), 2004.

[11] B. de Ruyter and E. Aarts. Ambient intelligence: visualizing the future. In *In proceedings of the Working Conference on Advanced Visual interfaces (AVI04)*, pages 203–208, 2004.

[12] P. Bull, P. R. Benyon, and P. R. Limb. Residental gateways. *BT Technology Journal*, 20(2):73–81, 2002.

[13] Y. Jin, R. Wang, H. Huang, and L. Sun. Agent-oriented architeceutre for ubiquitous computing in smart hyperspace. *Wireless Sensor Network*, 2(1):74–84, 2010.

[14] A. Bilgin, J. Dooley, L. Whittington, H. Hagras, M. Henson, C. Wagner, A. Malibari, A. Al-Ghamdi, M. Alhaddad, and D. Alghazzawi. Dynamic profile-selection for zslices based type-2 fuzzy agent controlling multi-user ambient intelligent environments. In *International Conference on Fuzzy System*, Brisbane, Austrailia, 2012.

[15] S.T. Shivappa, M.M. Trivedi, and B.D. Roa. Audiovisual information fusion in human-computer interfaces and intelligent environments: A survey. *Proceedings of the IEEE*, 98(10):1692–1715, 2010.

[16] C. Ressel, J. Ziegler, and E. Naroska. An approach towards personalized user interfaces for ambient intelligent home environments. In *2nd International Conference on Intelligent Environments (IE'06)*, Athens, Greece, 2006.

[17] M. K. Habib. Collaborative and distributed intelligent environment merging virtual and physical realities. In *In proceedings of 5th IEEE International conference on Digital Ecosystems and Technologies (DEST)*, pages 340–344, Daejeon, Korea, 2011.

[18] M. Jianhua, L.T. Yang, B.O. Apduhan, H. Runhe, L. Barolli, and M. Takizawa. 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*, Fuduoka, Japan, 2005.

[19] D. Cook and S. Das. Pervasive computing at scale: Transforming the state of the art. *Pervasuve and Mobile Computing*, 8(1):22 – 35, 2012.

[20] K. Niinuma, P. Unsang, and A.K. Jain. Soft biometric traits for continuous user authentication. *IEEE Transactions on Information Forensics and Security*, 5(4):771–780, 2010.

[21] T. Hildebrandt and J. Winther. Bigraphs and (reactive) xml - an xml-centric model of computation. Technical report, The IT University of Copenhagen, 2004.

[22] J. Dooley, C. Wagner, H. Hagras, and G. Pruvost. Followme: The persisent gui. In *1st International Workshop on SituatedComputing for Pervasive Environments (SCOPE-2011) at 6th International Symposium on Parallel Computing in Electrical Engineering (PARELEC-2011)*, Bedfordshire, UK., 2011.

[23] A. Ranganathan, C. Shankar, and R. Campbell. Application polymorphism for autonomic ubiquitous computing. *Multiagent Grid Systems*, 1(2):109–129, 2005.

[24] S. Pardha, M. Tungare, J. Holman, and M. Perez-Quinones. Continuous user interfaces for seamless task migration. In Julie A. Jacko, editor, *In Proceedings of the 13th International Conference on Human-Computer Interaction. Part III: Ubiquitous and Intelligent Interaction*, pages 77–85. Springer-Verlag, Berlin, Heidelberg, 2009.

[25] J. Dooley. *An Information Centric Architecture for Large Scale Description and Discovery of Ubiquitous Computing Objects*. Thesis, (phd), University of Essex, 2010.

[26] Irwin D. Greenwald and Maureen Kane. The share 709 system: Programming and modification. *journal of the ACM (New York, NY, USA: ACM)*, 6(2):128–133, 1959.

[27] Federal Financial Institutions Examinations Council. Authentication in an internet banking environment. `http://www.ffiec.gov/pdf/authentication_guidance.pdf`, 2012.

[28] S. Creese, M. Goldsmith, B. Roscoe, and I. Zakiuddin. Authentication for pervasive computing. *Proc. Security in Pervasuve Computing 2003*, 2803:116–129, 2004.

[29] M Archer, C. Heitmeyer, and E. Riccobene. Proving invariants of i/o automata with tame. *Automated Software Engineering*, 9:201–232, 2002.

[30] R. Milner. *The Space and Motion of Communicating Agents*. Cambridge University Press, 2009.

[31] Denmark IT University of Copenhagen. A brief introduction to bigraphs. `http://www.itu.dk/research/pls/wiki/index.php/A_Brief_Introduction_To_Bigraphs`, 2012.

[32] M. Henson, J. Dooley, L. Whittington, and A. Al-Ghamdi. Towards simple and effective formal methods for intelligent environments. In *8th International Conference on Intelligent Environmnets (IE'12)*, Guanajuato, Mexico, 2012.