Towards a Trust Based Approach to Security and User Confidence in Pervasive Computing Systems

J Basu and V Callaghan
Inhabited Intelligent Environments Group, Essex University, UK
Email: iieg@essex.ac.uk Website: http://iieg.essex.ac.uk

Abstract

In this paper we describe ongoing work into the development of a trust-based architecture aimed at increasing security and user confidence in pervasive computing systems. We begin by describing the issue of security and privacy in pervasive computing systems before introducing the notion of Role Based Access Control. Such access control methods depend on strict policies that can lead to a loss of flexibility and usability. To overcome this we show how by employing the concept of trust and reputation it is possible to strike a better balance between usability and security. This work is ongoing but we present an architectural diagram showing our novel Trust & Role Based Access control system together with reporting on initial results. Finally we discuss how this work is consistent with our wider approaches to intelligent environments such as evidential learning, promiscuous association, end-user programming and discuss the future directions we are planning.

1.0 Introduction

Pervasive Computing or Pervasive Intelligent Computer Technology (PICT) [16] typically depicts a scenario, where numerous networked devices interact with each other, essentially to improve quality of life of the people employing such technology. Lou Gerstner, CEO of IBM, has described pervasive computing as "…. a billion people will interact with a million e – businesses via a trillion interconnected devices" [12], a vision which throws light upon the potential of PICT. Today, pervasive computing is no longer just a fascinating technical vision of scientists and researchers but a reality that is slowly crossing boundaries from the laboratory to our home. Whilst the motivation behind the acceptance of such technology is to produce more comfortable living spaces and exciting life-styles, user acceptance is inseparably linked to the trust we have in such systems, both in terms of technical performance (e.g. reliability) and our privacy (e.g. shielding from others)

Privacy becomes an issue because pervasive systems utilise mechanisms that monitor our actions and movements through sensors, potentially communicating such information to others via the networks they are part of. Examples of such sensing can vary from relatively harmless data on the temperature of a refrigerator to more dangerous data from entry or occupancy systems disclosing who was in a house and what they were doing. Such systems raise serious concerns about the privacy of the individual [6].

From a practical point of view, security is risk management – assessing *threats*, *vulnerabilities* and *attacks*, estimating costs for the threats, estimating probabilities for the *attacks* given the vulnerabilities, developing appropriate *safeguards* and *countermeasures*; implementing where the uncertain loss that a potential threat entails are less than the costs incurred in the implementation [19]. Marc Langheinrich classifies privacy into four categories, namely *territorial privacy, communication privacy, bodily privacy and information privacy* [15]. In this paper we shall restrict our discussion to information privacy wherein an individual deems it undesirable to divulge personal information.

The two steps that essentially ensure security are authentication and authorisation. Authentication ensures identification of the user while authorisation protects us from malicious attacks by the identified user. This paper outlines the current problems related to security and privacy in PICT

systems and discusses current authorisation techniques and proposes a solution for Intelligent Environments.

2.0 Generalised Role Based Access Control Model

At Essex, we developed a simple model based on Generalised Role Based Access Control [7], henceforth referred as GRBAC, which has been developed from Role Based Access Control, abbreviated as RBAC [10]. Our contribution was to include context sensitive trust parameters that qualify a subject under a particular context. The concept of trust plays a pivotal role in determining access to resources and data.

The pervasive computing devices that are developed at Essex University are rule-based systems [2]. The pervasive Essex laws are as follows.

1. *Do not violate any safety constraints set by law or manufacturer*
2. *Do not violate any privacy constraints set by user of the environment or community (provided safety constraints are not violated)*
3. *Accept instructions (including configuration and training) immediately from stake holders of the environment (providing safety and privacy constraints are not violated)*
4. *Preserve the pervasive community (providing all the above are not violated)*

The above laws are generic and it is a pragmatic approach for ensuring safety of the subjects dwelling in the environment. However, we realize that pervasive computing environments are intrinsically complex, where different subjects with different privacy settings and preferences would be controlling the same environment. The status of subjects with respect to the environment would dynamically change. Moreover, the introduction of new subjects would require generation of dynamic rules. Under certain circumstances, there may be more than one subject trying to access the same resource. To accommodate the inherently dynamic nature of pervasive computing environments, it is necessary to deploy authorisation techniques, which determine access to resources. Our GRBAC model is a combination of user centric approach and agent centric approach and builds on well-developed work on Role Based Access Control (RBAC), which is described in the next section.

2.1 Role Based Access Control (RBAC)

RBAC is an access control mechanism where subjects are mapped to roles, and roles are mapped to transactions [10]. A subject can assume more than one role. Under certain circumstances role activation (which, essentially determines a subject's active role) and separation of duties (which prevents fraudulent activities) become very important issues.

The nature of resources, the type of interaction of humans with the resources and methods to share the resources is different in Intelligent Environments, henceforth abbreviated as IE, from that of traditional computer systems [17]. Due to such differences the RBAC model is considered static for Intelligent Environments. In organisations, where a person has a definite role, RBAC works perfectly well but the inherent dynamism of Intelligent Environments prevents qualifying RBAC as the model solution. However, we adopted the concepts of RBAC due to the inherent simplicity of the model.

The RBAC model is dependent on role and thus access is based only on the role a subject assumes under a particular context. However, in an IE, access would be dependent of the state of the environment that the subject is residing in. Thus we need a more flexible version; the Generalized Role Based Access Control model described in the next section.

2.2 Generalized Role Based Access Control (GRBAC)

2.2.1 Description

The Generalized Role Based Access Control has its foundations in the RBAC model described above, but it is more flexible that its precursor. The flexibility of GRBAC can be perceived from

the fact that security policies in this model are not only subject–centric but extended to accommodate object-centric polices as well, a feature that was missing in RBAC [7]. Such provisions clearly imply that objects and the environment in which the transactions take place also play a crucial role, thus allowing for more logical interaction between the user and the resource and also forces better access control. We shall discuss these features of the GRBAC model in the following section, mainly relating to object roles and environment roles.

Allowing objects to take up roles is an enhancement of this model over the RBAC model. Thus, an object like the heater can take up the role of an electrical appliance. Such a provision allows the system administrator to formulate policies that takes into account the roles played by an object, thus ensuring better access control.

Environmental Roles allow for better interaction between the object, which is essentially a resource, and the subject. An environment is generally defined with respect to time and location [7] but an environment's role can be captured with respect to the system state as well. Examples of location are "upstairs", "kitchen" and so on. Examples of time are "morning", "afternoon" etc. The state of the system can be anything that describes its condition such as "on", "loud", locked, "32c" etc

We now discuss how access decisions are made in the GRBAC model and then follow it up with an example that demonstrates the GRBAC model ability to provide security.

Access decisions in GRBAC are more complex than those in RBAC. In GRBAC, subject *s* possesses some subject roles while object *o* possesses some object roles [7]. In RBAC, for the subject *s* to access the object *o*, *s* must possess some role *r* that is authorized to execute transaction *t*, such that *t* can access *o*.
Additionally, the system keeps a track of a set of environmental roles. For *s* to access *o*, *s* must possess some subject role r_s, such that:

1. *There exists* some object role r_o, possessed by *o*.
2. *There exists* some environmental role r_e that is currently active.
3. *There exists* some transaction *t* that allows r_s to access objects in role r_o when r_e active.

2.2.2 An Example

In this section we offer an example to illustrate the suitability of the GRBAC model. For experimental purposes we have built a mock of such a room called the iDorm (see figure 1). Typically, in such accommodation, students are allowed to switch on lights and heaters and other appliances in their rooms. However, the state of the room or the kitchen acts as crucial factor.

Figure 1 – The iDorm

It is quite logical to assert that in case of an emergency situation arising due to a fire or similar causes, access would be denied to the appliances. In the example given above, the subject is the student and the objects are the appliances. It is clear that access control might differ in routine and emergency situations. For example, if the residence went on fire then the occupant might not be permitted to turn off the light or to switch on the heater, and thus bared from access to the

appliances. Thus, we observe that the environmental state plays a decisive role in controlling access. However, in the same situation firemen will be allowed to access the same resources in order to avert major losses, as their role demands such transactions under the critical state of the environment, which is that of emergency. Thus, we see how the GRBAC model is suitable for context dependent situations. We assert that to solve the same problem, RBAC would not been a useful choice.

We observe that for the same situation we could have two different policies for two different subjects:

1. *A student must not access the lights or heaters if there is an emergency.*
2. *A fireman is authorised to access the lights or heaters for rescue and safety purposes in a state of emergency but not risking his or her safety.*

Thus we see how policies can be implemented easily using the model.

3.0 Trust and Reputation Based Access Control

3.1 Trust

Consider the following scenario. Jemma intends to send some money to India to her parents. She has a bank account with Lloyds Bank. However, the bank is going to charge £20 for the amount that she is going to send. Jemma logs in and checks different web sites that facilitate remitting money to India which charge much less than £20. The offer looks exciting to Jemma but still she decides to send the money through her bank. She compromises a few extra pounds. Thus the question arises why did Jemma take such a decision? In this story the answer is due to her lack of trust in the web-based system. In Jemma's case she was not sure who was going to view her account details.

We come across the word, *trust*, almost every day in our life. Trust is one of the key philosophies behind the functioning of the world as a whole. Thus we have chosen to utilize the concept of trust in our endeavour to provide security in the world of pervasive computing. However, as in human terms, the concept of trust is somewhat subjective and difficult to quantify. One of the definitions of trust is *'qualified reliance on received information'* [18]. Another definition of trust is *'Trust (or, symmetrically distrust) is a particular level of subjective probability with which an agent will perform a particular action, both before [we] can monitor such action (or independently of its capacity to be able to monitor it) and in a context in which it affects [our] own action'* [11].

In an IE, we can provide security through the utilization of access control, but too much strictness can defeat the purpose of making the environment an intelligent one. Such measures might result in a loss of usability of the system. We shall show how trust can be used to strike a balance between usability and security. We have already mentioned that it is very difficult to quantify trust because it is an abstract concept, quite apart from being extremely subjective.

A trust relationship has the following characteristics, as stated in [1].

1. A trust relationship is exactly between two entities.
2. A trust relationship is non - symmetrical.
3. It is conditionally transitive.

It should be noted that the third characteristic is very important in the context of security and an intelligent agent's effectiveness can be judged from the way it can handle a situation where transition of trust takes place.

3.2 Reputation

Reputation of an entity can be defined as *'…some notion or report of its propensity to fulfil the trust placed in it (during a particular situation); its reputation is created through feedback from individuals who have previously interacted with the entity'* [13].

The notion of reputation is important in the context of an IE as the state of the environment changes with every action taken by the user. The intelligent agent must use the effective outcome of such actions to determine the reputation of the occupant and behave accordingly. Potentially harmful actions lower the reputation of the occupant and access to different resources must be regulated using reputation as the parameter of judgement. Mention must be made of the fact that trust and reputation are very closely related concepts.

In the human world, generally, an individual is trusted more if more than one person can trust him or her [14], that is he or she has *more* reputation. Similar arguments have been put forward in the pervasive world. In the pervasive world, a user or computing device (hereafter referred to as "an entity") is able to access a service if it has the right to do so or has been delegated the right by another entity. In a case, where an entity is delegated a right, it is assumed that the delegating entity trusts it. However, since it is difficult to quantify trust, it becomes very difficult to assert how much entities can trust each other in different contexts. Therefore, we aim to quantify trust in different contexts.

4.0 Architecture of Security Model

The architecture of the security model is based on the approach of Chin's TOP model, discussed below. We thus discuss the TOP model in brief. We plan to work towards integrating the two models as part of our future research.

4.1 Task Oriented Programming (TOP)

Smith introduced the Programming By Example (PBE) concept where algorithms were described by concrete examples and not in an abstract way [20]. PBE can be regarded as a software agent that writes a program based on mimicking actions that the user demonstrates are required in the target system. Computer Aided Design (CAD), children's programs are some examples where PBE has been applied. PBE principles are generic and thus have been transported into the pervasive computing world by Chin who has developed a methodology called Task Oriented Programming (TOP) which is the first application of PBE in the pervasive computing world [5].

Chin's TOP paradigm is especially interesting as it is based on a deconstructed appliance model where an appliance's functions are decomposed (or disaggregated) into their basic or atomic functionalities. Such disintegration allows creation of a richer pool of sub-functions, which opens up possibilities for re-combining sub-functions in conventional or novel ways, such that a virtual device is formed. Thus, in a networked environment, it is possible to create numerous virtual devices by sharing the sub functions of the appliances. One of the main achievements of TOP is that it enables non-technical lay-users to program pervasive systems.

4.2 Trust Based Architecture

The TOP Paradigm that was formulated by Chin at Essex [5][4] was intended to empower the end user by giving the user the ability to program the systems around the user and rather than the user having to adapt to the system. The idea was to enable an end-user to design the functionality of their own electronic space, creating their own "virtual appliances". Part of the motivation of Chin's work is to make the system operation both transparent and subservient to the user, not only supporting their creativity but also protecting their privacy. In other words, giving the user a level of control that instils a sense of trust in the system. Our work, provision of security is based on a somewhat similar approach. We give the end-user the power to input policies and roles. However, the framework also helps the user to form the best policies and evaluate trust parameters of the environment. Thus, the system should have the capacity of learning from the user and the effects of events on the environment.

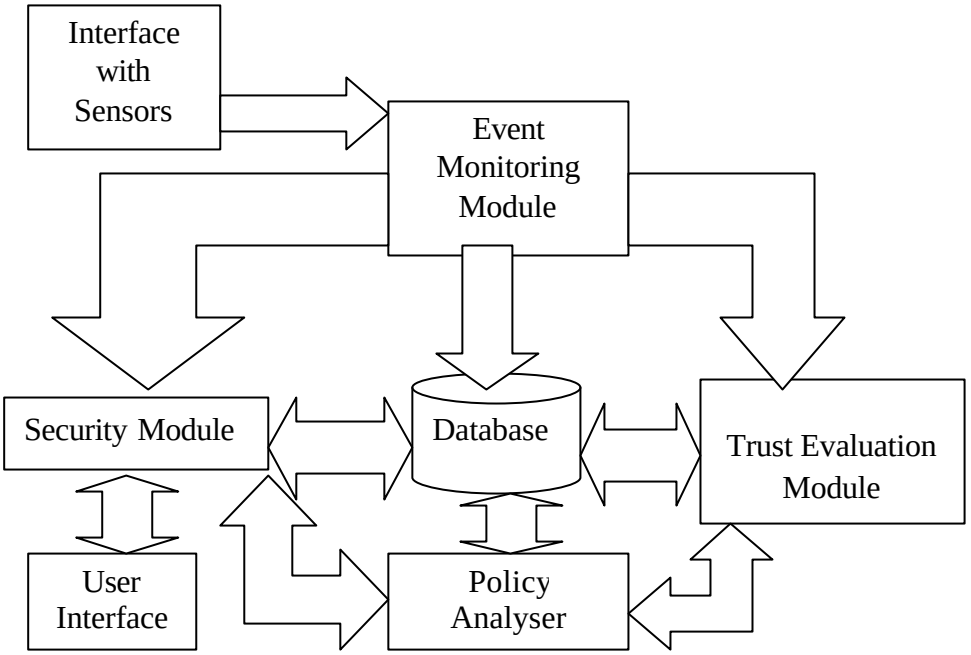

Figure 2 – Essex Trust Based Architecture

The trust-based architecture (ET for short), shown in figure 2, is based on the concepts discussed in this paper. The ET system contains four main modules, the security module (SM), the event-monitoring module (EMM), trust evaluation module (TEM) and policy analyser (PA). There are two interfaces, the user interface, and the sensory input interface. Through the user interface, the users can communicate with the system and vice versa. The central database stores the different policies, the trust values of different users and different roles users play under different environments. The different modules that have access to the database update it as well as retrieve data from it.

The SM is the most basic module. The functionality of this module is to take decisions related to authorising access to a resource. The SM works in a manner similar to the implementation described in section 5.0. The SM is responsible for accepting new roles, the trust values and the policies that the user inputs into the system. The SM is connected to the PA and the EMM. The main function of the PA is to analyse policies entered by the user and check the validity of the policies. The PA has the power to suggest new policies and also resolve conflicts between polices.

The EMM interacts with the environment and sends the events to the SM to aid in taking decisions related to authorisation of requests. The EMM informs the SM of the various events taking place in the environment. The EMM also logs events in the database. The TEM acts in the background interacting with the EMM and the policy analyser. The EMM sends the different events that take place and the results of the events (which are also events) to the TEM. The TEM evaluates the trust between all entities and between subjects and entities using some complex function and attempts to justify the occurrence of the events within a certain domain of validity defined by the user under the Essex laws for pervasive computing. If the evaluation gives a positive result then the TEM takes no further action. However, if the result is negative, then the TEM tries to access the database searching for previous logging of events in order to compare the results and variance between the actual and expected results. This feature is very important, as pervasive computing environments are dynamic in nature. It also sends a signal to the policy analyser asking it to perform an analysis on its present set of policies. The policy analyser analyses the current policies and the rejected policies and tries to form a combination of them to check if such a combination of policies justifies

the events. Such actions essentially try to optimise the nature of the policies and the policies themselves.

5.0 Implementation and Evaluation

The work carried out at Essex is based on the combination of concepts. We have augmented the GRBAC model to utilise trust as one of the parameters for making decisions. Since GRBAC is based on RBAC, we allow for the definition of roles. In our agents we utilise a promiscuous association model, which initially has a high level of trust needed to strike up relationships [8][3].

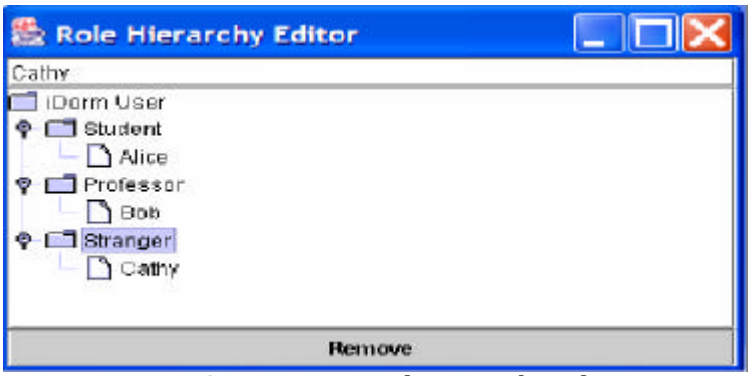

Figure 3 – Prototype Role Hierarchy Editor

We use an evidential learning mechanism that uses its experience of decisions to determine the values of relationships. A graphical user interface is used to define the roles. A database keeps track of the roles a subject takes. We also have a user interface to define policies. Policies are based different parameters, such as the state of the object, the time, the state of the environment, trust value, roles on which the policy is applicable and so on. Such policies are quite simple in nature. Moreover, we also included a trust editor where the trust of different subjects can be defined under different circumstances (e.g. time, location).

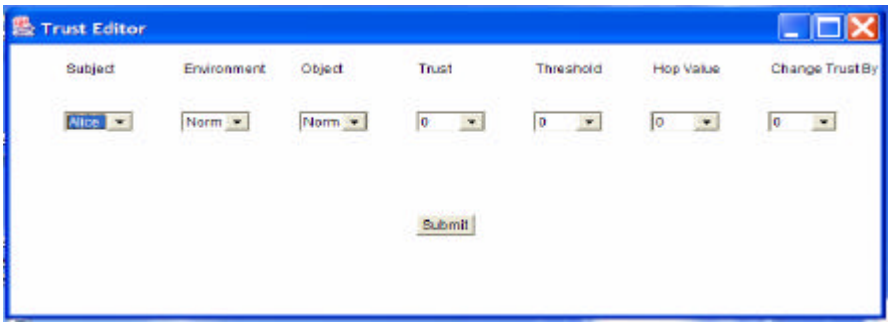

Figure 4 – Prototype Trust Editor

The operation of the model is relatively straightforward. When access to a resource is requested, the system checks the subject's credentials from a database, relating to the current circumstances. The system iterates through the policies that are applicable for the current request. If any of the policies do not allow access to the resource, then access is denied. Such measures are justified because it indicates that the subject has an attribute that would not qualify him to gain access. Apart from the object state, location, time etc., we use trust to define a subject's credential. Moreover, since trust is quantified, the system decreases trust if the person's interaction with the system is not acceptable under the policies. In this context, the relevance of a threshold value becomes important. If the trust value for the person goes below a threshold value that we define for the subject, the access to the resource would not be allowed. Thus, not only are we dependent on environment parameters, but also on a value that is attached to the person.

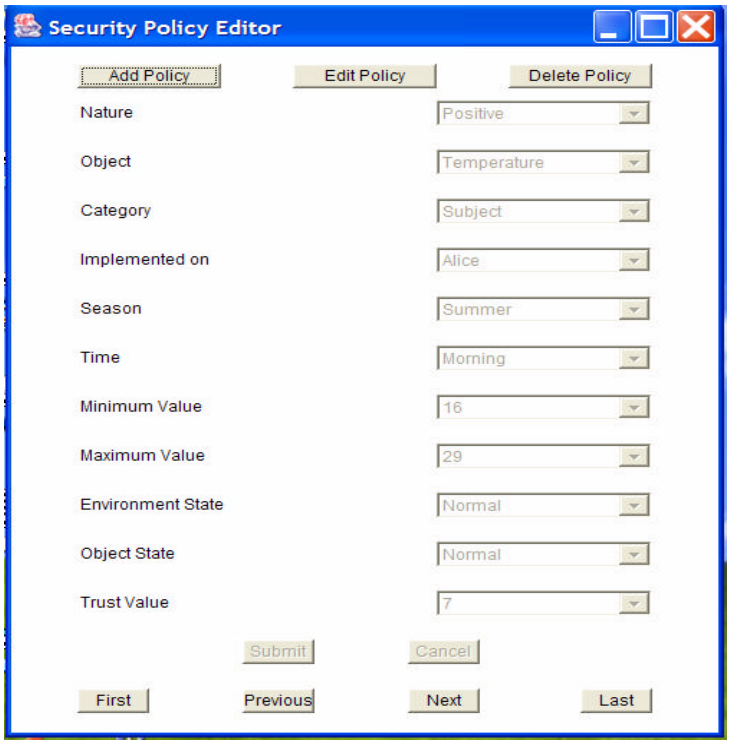

Figure 5 – Prototype Security Policy Editor

Shankar argues that in pervasive computing, ad hoc trust relationships are based on some common grounds such as location, whereas in the real world it would be based on the person's identity [18]. Similarly, in Intelligent Environments we have a similar notion. However, question arises as to how we make intelligent environments take decisions based on identity or location. In environments such as the iDorm, we can introduce a subject to the IE based on a trust value that the person is assigned. The trust value acts as the credential of the person. Thus, Alex (say, would initially introduce his friend Barry (say) with a higher trust value than a stranger.

The user himself frames the security policies. Thus the privacy of the environment is actually dependent on the type of the security policies the system gets from the user. A security policy takes account of all the different parameters that we have defined so far. The parameters can be considered as the different attributes of the policy. Our model actually allows the user to define policies. This gives the user the power to define the security of the model.

Microsoft Access - [SubjectTrust : Table]

File Edit View Insert Format Records Tools Window Help

SubjectName	StatCopy	EnvState	ObjState	TrustValue	ValueChangedBy	Threshold	HopValue	TrustChanger
Alice	T	Normal	Normal	7	0	3	5	2
Alice	F	Normal	Dangerous	0	0	0	0	0
Alice	F	Emergency	Normal	0	0	0	0	0
Alice	F	Emergency	Dangerous	0	0	0	0	0
Bob	F	Normal	Normal	0	0	0	0	0
Bob	F	Normal	Dangerous	0	0	0	0	0
Bob	F	Emergency	Normal	0	0	0	0	0
Bob	F	Emergency	Dangerous	0	0	0	0	0
Cathy	T	Normal	Normal	5	0	4	4	3
Cathy	F	Normal	Dangerous	0	0	0	0	0
Cathy	F	Emergency	Normal	0	0	0	0	0
Cathy	F	Emergency	Dangerous	0	0	0	0	0
				0	0	0	0	0

Figure 6 – Trust on Subjects under Different States of the Environment

The GRBAC model was chosen as it allows for defining an environment more explicitly and thus allows defining it with finer granularity making it easier to match together the diverse traits of people and their environments. With the addition of the trust factor to the model, we enhance the dynamism of the policies and decisions based on the notion of value. Since this value is changeable, it acts as the dynamic factor, which defines the relationships between the subject and the environment and its entities. Moreover, since the trust factor is also embedded into policies, the policies become dynamic in nature. A similar notion is used for objects. When more that one subject tries to access the same resource, the system iterates through the policies to check that the subjects are allowed to access the requested resource, the subject with the higher trust value would be granted access to the resource. Thus we ensure that the stakeholders of the IE have more access rights than others. However, a question may arise as to how we should decide on granting access to the subjects with the same credentials. In such case, we use common policies. The policy analyser can also change existing policies or generate new policies with evaluation of different behaviours of subjects under certain circumstances. Another relevant aspect of trust – based architecture is the delegation of rights; we can delegate rights by just assigning a trust value to a subject.

The notion of trust maps well to our task-oriented-programming approach as the policies can be defined by users and encoded in an ontology such as dComp; it also maps well to our behaviour-based agents as a subject's behaviour under different circumstances (being monitored by the agent) can be used as a measure of the trust that can be put on the subject (and the agent can, potentially, learn / adapt the trust value).

The system was built and implemented as described as part of an MSc project. The aim of the work was to prove the practicality of building and incorporating a trust based GRBAC into an IE. The project was successful in that it succeeded in producing a working prototype. Clearly this is just a first stage in this research and longer-term goals are to conduct a more extensive user evaluation in an environment such as the iDorm over an extended period. Given the complementary relationship to other work, such as TOP it would also be productive to consider how such a trust model could be combined with TOP, and perhaps integrated into a common architecture.

6.0 Summary and Future Directions

In this paper we have reviewed the issues of trust and security in pervasive computing and intelligent environments. We have presented an augmented version of a trust based GRBAC model, implement as a holistic trust based architecture that we refer to as ET, as the basis of a solution that

would meet the needs of inhabited intelligent environments in which numerous users share common facilities. We have argued that this provides a flexible approach that would allow for the formation of a variety of policies and roles under the Essex pervasive laws which attempts providing security to the environment while maintaining usability simultaneously. The seamless integration of the trust evaluation module and the policy analyser makes the system very dynamic in terms of new policy generation. The system is adaptive in nature and access decisions are based on dynamic features such as trust. The quantification of trust is somewhat difficult but the system has a definitive approach to decision making as we attempt to quantify trust. Such an approach is well suited to intelligent environments, as ambiguities are removed from the policies. The updating of policies makes the system scalable for future changes.

Our plans for taking this work forward include issues such as considering whether our trust architecture could be synergistically combined with elements of the TOP end-user paradigm. For example, could the dComp ontology be extended to include the policy, environment and people descriptions that the augmented GRBAC model and ET architecture require; could the TOP Engine run-time environment be adapted to support ET (ideally ET and TOP could be executed on a virtual machine to give it portability akin to Java applets); could the TOP Editor be extended to include the end-user setting up trust policies? We are also interested in considering whether our trust model could be integrated with our autonomous agents, such as the ISL model, (perhaps static or quasi static rule sets). Other issues we would like to explore include ideas like ad hoc trust [9] and embedding trust factors into the roles the subjects and object plays (e.g. objects with more than one role, such as a mobile device being used to receive telephone calls as well as an alarm clock). In the case of the latter, for accessing an entity, both the subject and the object would require trusting each other. The subject would essentially acquire trust by delegation of a trust value by the stakeholder of an environment. In this way, privacy of the individual can be protected.

References:

[1] Alfarez Abdul-Rahman, Stephen H. *A Distributed Trust Model,* ACM New Security Paradigms Workshop, 1997

[2] Callaghan V., Clarke G., Colley M., Hagras H., *"A Soft Computing DAI Architecture for Intelligent Buildings*" Journal of Studies in Fuzziness and Soft Computing on Soft Computing Agents, *Physica - Verlag – Springer,* November 2000

[3] Callaghan V, Colley M, Hagras H Chin J, Doctor F, Clarke G "*Programming iSpaces: A Tale of Two Paradigms*" Chapter 24, Springer-Verlag Book July 2005

[4] Chin J, Callaghan V, Colley M, Hagras H, Clarke G, *"Pervasive Information Systems: Issues for the Individual and Society",* in Book Pervasive Information Systems published by M.E.Sharpe New York, August 05

[5] Chin J, Callaghan V, Colley M, Hagras H, Clarke G. "Virtual Appliances for Pervasive Computing: A Deconstructionist, Ontology based, Programming-By-Example Approach", IE05, Essex, 28-29th June 05

[6] Chin JSY, Callaghan V, *"Embedded-Internet Devices: A Means Of Realizing The Pervasive Computing Vision"*, IADIS International Conference, Algarve, Portugal, 5-8 November 2003

[7] Covington M, et al, *Generalized Role-Based Access Control for Securing Future Applications*, Proceedings of the National Information Systems Security Conference (NISSC), October 2000.

[8] Duman, H., Hagras, H.A.K., Callaghan V., Clarke, G.S., Colley, M.J., *'Intelligent Association in Agent-Based Ubiquitous Computing Environments'*, International Conference on Control, Automation, and Systems, Muju, Korea, (2003)

[9] English C., et al, *Dynamic Trust Models for Ubiquitous Computing Environments*, Workshop on Security in Ubiquitous Computing, 4th International UBICOMP, 2002.

[10] Ferraiolo D., Kuhn R, *Role Based Access Control*, Proceedings of 15th National Computer Security Conference, 1992.

[11] Gambetta D. *Can we trust trust*? In Gambetta, D (editor) *Trust: Making and Breaking Cooperative Relations*, electronic edition, Department of Sociology, University of Oxford, chapter – 13, pages 213-237, http://www.sociology.ox.ac.uk/papers/gambetta213-237.pdf , 30th August, 2003

[12] Gerstner L, *"Presentation to Joint Economic Committee's first National Summit*

[13] Goecks J., Mynatt, E, *Enabling Privacy Management in Ubiquitous Computing Environments through Trust and Reputation Systems*, GVU Center, College of Computing, Georgia Tech, Atlanta GA 30332.

[14] Kagal L. et al, *"Moving from Security to Distributed Trust in Ubiquitous Computing Environments"*, IEEE Computer, December 2001.

[15] Langheinrich M. *Privacy by Design – Principles of Privacy – Aware Ubiquitous Systems*, In Gregory D. Abowd, Barry Brumitt, Steven Shafer, Proceedings of 3rd International Conference on Ubiquitous Computing (Ubicomp 2001), Pages 273 – 291, 2001.

[16] Lyons M, *"Pervasive Computing: Control and Freedom in Cyberspace"*, 2002

[17] Tuchinda R, *Access Control Mechanism for Intelligent Environments*, http://www.mit.edu/activities/ieee/bitstream/Access.pdf, 29th July 2003.

[18] Shankar N, Arbaugh, W.A. *On Trust for Ubiquitous Computing*, Workshop on Security in Ubiquitous Computing, 4th International UBICOMP, 2002.

[19] Stajano F, *Security for Ubiquitous Computing*, John Wiley and Sons Ltd., Chichester, England 2002, ISBN 0470 84493 0.

[20] Smith, D.C., *'Pygmalion: A Computer Program to Model and Stimulate Creative Thought'*, Basel, Stuttgart, Birkhauser Verlag, 1977.

