Building Access Controls for Intelligent Environments

Buddhika Kottahachchi
MIT CSAIL
32 Vassar Street
Cambridge, MA 02139
Email: buddhika@csail.mit.edu

Robert Laddaga
MIT CSAIL
32 Vassar Street
Cambridge, MA 02139
Email: rladdaga@csail.mit.edu

***Abstract*— Projects involving the investigation and construction of Intelligent Environments have had reasonable success in demonstrating their potential applicability in the real world. However, most of these efforts pay little or no attention to their security and privacy implications. These are key issues that need to be addressed in order for Intelligent Environments to be deployed broadly. In this paper we present an approach towards addressing the issue of access controls within the Hyperglue platform for constructing Intelligent Environments.**

I. Introduction

Applications of ubiquitous computing promise a world where computing resources around us work transparently towards improving the quality of our daily lives. In order to achieve this goal, communities of software agents representing these computing resources must work together. Research efforts like UIUCs Gaia [8], Stanford's iROS [7] and MITs own Metaglue [6] are platforms aimed at facilitating such interactions. These projects work toward the common objective of allowing interactions between software agents within a particular, self-contained intelligent environment. In general, they also make the assumption that agents within a space are trusted and have free access to other resources within the space. However, it seems incorrect to assume that any agent can and should always be allowed to work with another agent.

Recently, iSecurity [9] was developed to provide security within the iROS platform. iSecurity handles authentication via a centralized model, but more interestingly provides for decentralized security policy enforcement. Cerberus [10], is another recent development designed specifically for use with Gaia. Cerberus also relies on a centralized authentication mechanism, but is interesting because it takes context into account when enforcing security policies. It is important to note that both iSecurity and Cerberus are designed to work within self-contained intelligent environments (as required by the platforms within which they function). However, to truly realize the potential of ubiquitous computing we need platforms that are highly scalable and capable of encompassing multiple intelligent environments.

As intelligent environments evolve and expand to encompass multiple distinct spaces and support multiple users simultaneously the issue of access control becomes increasingly prominent. Yet, platforms such as iROS, Gaia and Metaglue do little to address this. Even attempts like iSecurity and Cerberus have their own shortcomings, particularly within the context of the kind of deployments we're interested in.

For example, iSecurity focuses on iROS enabled interactive workspaces. Since it has such a well-defined domain, the semantics of iSecurity are relevant primarily for applications within interactive workspaces. Therefore, we see iSecurity as a specialized solution which is not extensible for use in other applications. Furthermore, iSecurity focuses on lower level implementation specific details which make it difficult for it to be evolveable to something useful in other areas and applications. Cerberus on the other hand avoids these pitfalls, but unfortunately has the characteristics of being designed for a self-contained space. Thus, while it can support a range of applications and is extensible within the context of a single space it faces difficulties in supporting the kind of Intelligent Environments we are interested in.

In this paper we aim to provide insight to an alternative approach towards access controls in Intelligent Environments. In particular, we're interested in a system that is easy to extend and evolve. Since we situate our work in the realm of ubiquitous computing, we're also interested in providing a system that feels natural and transparent to the end-user. We discuss a Role-Based Context-Aware Access Control system that (1) utilizes a semantic representation of the roles, resources, contextual cues and the permissions and relationships that inter-connect them (2) performs inferencing on this representation to determine access rights (3) exposes access controls via an intermediary software agent, thereby providing a level of abstraction that allows access controls to be evolved independent of the larger system they are deployed in. We begin by providing a brief overview of Metaglue and Hyperglue [5] - the agent infrastructures on which our work is based - and the motivations behind our design. Then we proceed to discuss and define the semantics involved and frame our design goals within those semantics. We follow this with a description of our implementation design and finally offer some concluding thoughts.

II. Metaglue/Hyperglue

Metaglue is a distributed agent infrastructure developed here in our lab. It provides brokering and directory services

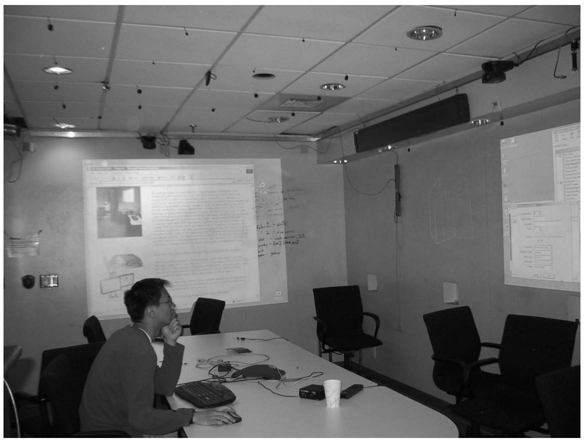

Fig. 1. The e21 conference room: Contains a collection of projectors and a sound system that can be shared based on user needs, built in software infrastructure to intelligently support meetings, presentations and handle ancillary tasks like lighting.

that enable agents to interact with each other. Metaglue has been deployed in a number of conference rooms [12], offices [13] and common spaces [11] and continues to be used on a daily basis. Our e21 conference room space is generally the preferred location within the lab for both research group meetings and presentations (Fig. 1). A few of the faculty and research staff have deployed Metaglue in their offices and have come to rely on it and the convenience it brings in going about their daily tasks (Fig. 2). Kiosks based on Metaglue and resulting from the k:info project [18] are being deployed across the lab in common public spaces. Thus, while it is apparent that Metaglue has reached a level of maturity in its intended domain - ie. well defined, self-contained spaces - in its current form we find it unsuitable for use in the construction of Intelligent Environments that encompass multiple spaces.

As such, we're in the process of building Hyperglue: an extension to the existing Metaglue infrastructure to support multiple agent societies working across multiple spaces. As we build Hyperglue we have come to realize that the domain of issues pertaining to privacy and security grow significantly when we allow Intelligent Spaces to grow unbounded. As such a fresh look at how we address security issues was needed.

III. Controlling Access in Intelligent Environments - Conceptual Requirements

Here at the MIT Computer Science and Artificial Intelligence Lab, we have built a number of Intelligent Environments as part of our ongoing research in the area [1]. In doing so, it has become apparent that we are concerned with four primary types of entities: *People, Places, Devices and Data*. In our Intelligent Environments, these types of entities have agent societies associated with them. Furthermore, agent-based interactions occur between these entities both within a type and across types (ie. between People and People, People and Places, People and Data etc.). The entities we're concerned about can also share resources within their control during these interactions. Furthermore, we are now interested in controlling

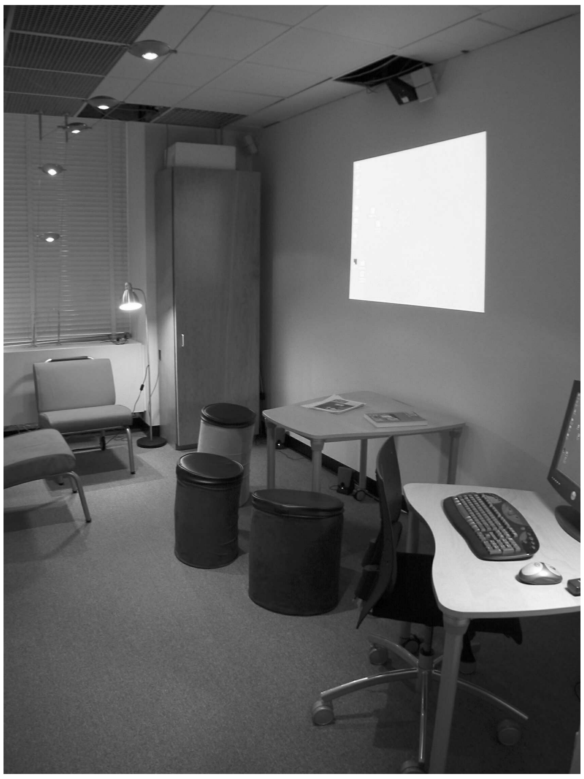

Fig. 2. A Metaglue enabled office: Applications include, handling lighting based on user preferences and reminding the user of calendar events etc. In particular, the blinds are controlled by the system based on user preferences and current weather conditions. Also selects and uses relevant interfaces like projectors, the user's computer screen or the sound system to remind the user of relevant appointments, events etc.

access to these resources. This leads to a complex set of relationships and constraints, which we would like to build an evolvable conceptual representation for. As such we're interested in defining a set of basic concepts on which an ontology specifying our conceptualization of entities and their relationships to one another can be built.

A. Roles

We assert that when such entities interact, they assume a Role for the purposes of that interaction. We also assert that these Roles can be and are often relational. That is, the roles are defined in terms of the relationships between the entities concerned.

For example, from the perspective of a Person Entity, another Person Entity could fit any of a range of roles such as:

1) Student - Fellow Student, My Student
2) Professor - My Adviser, Colleague
3) Friend
4) Administrator
5) Unknown Person etc.

Also, from the perspective of a Place Entity, a Person Entity could fit a role such as:

1) User

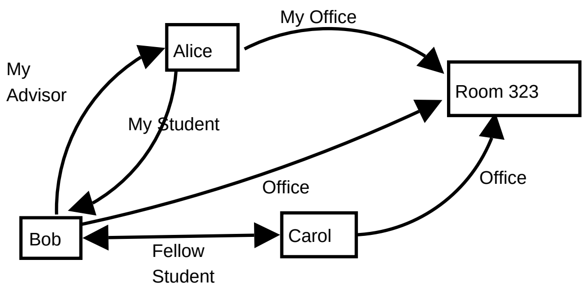

Fig. 3. Role Relations

2) Presenter
3) Administrator
4) Unknown Person etc.

And, to a Data Entity a Person Entity could be a:

1) Owner
2) Subscriber
3) User etc.

For example, Bob could be a *Fellow Student* from Carol's perspective but *My Student* from Alice's Perspective. Similarly, Alice could be a *Professor* from Carol's perspective but *My Adviser* from Bob's perspective. Furthermore, room 323 - a Place Entity - could be *My Office* from Alice's perspective and an *Office* from Bob or Carol's perspective. Thus, it is apparent that different entities can take on different roles from the perspective of other entities. (Fig. 3)

B. Resources

These entities can also be in control of different resources. For example, a Place Entity such as a Conference Room can have Projectors, Lights, Sound Systems, Room Capacity Information as resources within its control. Similarly, a Data Entity like a Personal Information Management Server can have individual user Address books and Personal Calendars as resources within its control. A Device Entity such as a mobile phone can have such resources as a phone book, a speaker, a microphone and a display in its control. However, while these resources are within the control of particular entities there can be instances when they are useful to other entities as well. For example, Alice could desire to check the Room Capacity Information of a conference room before she schedules a meeting for her Research Group. Once a suitable conference room is found Alice would desire access to the Personal Calendars of Bob and her other students on the Personal Information Management Server in order to schedule the meeting.

C. Permissions

It is apparent that resources within the control of a given entity can't be given out at will. For example, when Alice is scheduling his Research group's meeting should she be allowed to make an entry in Bob's calendar? What about Bob's right to privacy?

Thus, we see the need for Permissions. Bob should be able to define that calendar access should only be given to those Person-Entities that assume the *My Adviser* role from his perspective. If Carol were able to do the same then Alice, even though she is a *Professor*, would only have access to Bob's Calendar and not to Carol's. We would like to define permissions with respect to Roles that entities can assume and the Resources they have access to. They could take the following form:

```
An entity assuming role X has access to Resource Y.
```

By access, we mean complete access. This may seem non-ideal. For example, one might desire to give *Write* access to one's Calendar for those assuming one role and *Read/Write* access to those assuming another role. However, cases belonging to this class can be covered by defining the resources at a more granular level: Calendar Write resource, Calendar Read/Write resource etc.

D. Context

The basic concepts defined thus far are sufficient to describe a set of static permissions. But, are static permissions sufficient to encapsulate and model the world we're interested in? Consider the case where Bob takes time off from school to deal with a family emergency. What he does during this time maybe of a very personal nature to him, and even though Alice has static permissions to his calendar the current circumstances affecting Bob should be considered in making a decision on whether Alice should be allowed access or not. Thus, a need to model context also becomes apparent.

Certain aspects of context can be described by simply extending our notion of Roles to include context. For example, we could define a role called student-on-leave and define permissions restricting access to the personal calendar of a person entity assuming that role. However, we believe that this approach may not be desireable from a scalability perspective. Instead, modelling context seperately was a more attractive approach where we had a seperate notion of context associated with each entity. We assert that each entity should maintain a model of the context in which it currently finds itself. Thus any decisions that require contextual information can be correctly determined.

IV. Design Goals

Our goal is to build a system of Access Controls intended for use in applications of ubiquitous computing environments. Such environments require that all computing tasks occur in the background with minimal intervention from the end user. Thus, computing resources become ”pervasive, like batteries, power sockets, and the oxygen in the air we breathe” [2]. Therefore, our Access Control system needs to feel as natural to the user as possible and ideally function with little or no intervention from the user. For inspiration, we turned to the real world and human society. How do ”access controls” work in human society?

Consider the case where Alice meets Bob on the road and Bob asks Alice for her phone number. Would she give it out?

Assume she knows and identifies Bob as a friend. In this case it is likely that Alice would release her phone number to Bob. But, now consider the case where the person Alice meets on the road is Carol (who is unknown to Alice). Would Alice still give out her phone number? Probably not. Assume Carol tells Alice that she is a friend of Bob's. Is that enough to satisfy Alice that Carol is indeed a friend of Bob? Usually the proof of friendship will be provided implicitly with Carol sharing bits of information with Alice that only someone who was friends with Bob would know. Once the friendship is proven, based on Alice's relationship with Bob, Alice might consider releasing her phone number. But, what would Alice do if she knew her phone number was changing soon? In that case, it doesn't seem to make sense to give out her phone number.

While these may seem trivial observations, they have important implications for the system we're building. This scenario exhibits the following properties that we found desirable for replication in our system:

- **Scalability** Ubiquitous computing environments and applications are inherently unbounded: they can grow and change at will. Thus, a high degree of scalability is desired. Therefore, a centralized authority that makes decisions on access controls wouldn't be appropriate.
- **Local Relevance** The access control decisions must be made based on local relevance. Alice's perception of Bob or Carol is key to Alice making a decision on the level of access she allows. Furthermore, contextual information about the circumstances affecting Alice at that moment are also relevant. Therefore, allowing a third-party authority to make this decision is inefficient and may also lead to inaccurate decisions.
- **Extensibility** It is impractical to assume that all relevant access controls can be predetermined and defined. Thus, it is desirable to have a mechanism by which new knowledge can be obtained and used to augment the existing model.

V. Access Controls in Hyperglue

Based on these observations, we propose a design and implementation of a Role-Based Context-Aware Access Control mechanism suitable for ubiquitous computing applications. Our focus is in providing a *robust*, *scalable*, *secure* and *dynamic* mechanism.

A. Software Platform

The Access Control mechanism we present in this paper is intended for use within Intelligent Environments based on the Hyperglue platform currently being developed in our lab. Yet, the basics concepts put forward are generally applicable in this domain. Hyperglue is a distributed agent infrastructure that provides lookup and brokering services to agents. These agents are organized into societies and Hyperglue facilitates inter-society communications. Furthermore, these societies provide an excellent mapping for the real world (Place, Person etc.) entities that we are interested in. Also, the agents provide an ideal mapping for the real world resources we care about.

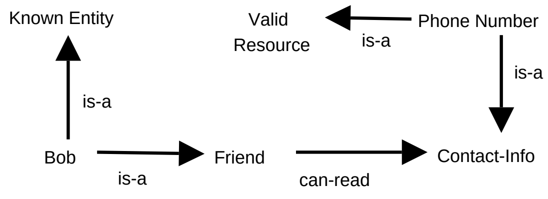

Fig. 4. K-Base

B. Access Manager

We provide Access Controls within Hyperglue via an Access Manager agent. Hyperglue has a notion of a Security Manager that is responsible for all security related decisions within an agent society. In our initial implementation the Security Manager defers all access control related decisions to the Access Manager. The Access Manager contains the following components:

1) K-Base: This is where we store the roles of entities we know about and the types of permissions they have with respect to resources within our control. The effect of various contextual cues on these permissions are also modelled within the K-Base. We chose a semantic net representation for this knowledge. Our representation involves having entities, roles, resources and context cues that are linked to one another with relationships and permissions.

This allows us a great deal of flexibility when manipulating the knowledge and augmenting it based on new findings. It also provides easy access to perform inferencing on. For example, Alice's K-Base with respect to the scenario described above could be represented as shown in Fig. 4.

In the trivial case above, *Known Entity* and *Friend* are roles, *Bob* is an entity, *Contact-Info* and *Phone Number* are resources and *Valid Resource* is a context-cue. They are inter-related by *is-a* relationships and *can-read* permissions.

Our internal representation is done using SEMANTIC [3], a set of tools developed in our lab, which provide a Java based framework for semantic net representation and storage. Since SEMANTIC provided interfaces for and had previously been used in the Metaglue/Hyperglue framework, it was an ideal choice to enable rapid prototyping.

2) Context Keeper: In meeting the need for a means to model context, we modify and extend the Context Keeper module originally developed for the Ki/o project [11]. Again, this proved an ideal starting point since it had been previously applied successfully within the Metaglue/Hyperglue framework. The Context Keeper module is based on a blackboard architecture [16], [17] and provides a society specific repositary for contextual information. It allows any agent within an agent society to make assertions about knowledge obtained regarding contextual cues that effect that society. Thus, a Personal Information Management Agent knowing that Alice's phone number is about to change could assert to the Context Keeper that Alice's phone number is now an *Invalid Resource*. Thus, when the Access Manager has to make a decision about granting access to Alice's phone number it has the required contextual information to make a correct decision.

We believe that of these context cues, Location is a key component. Consider that, the universe we've defined has the notion of entities that can move - ie. Person Entities, Device Entities etc. Also, an entity's Location has implications on the role it can assume. Let us consider a common Conference Room which has no statically defined owner. If Alice has made the appropriate reservation for the room, and she conducts a meeting in it - then she assumes the Role of *Owner* with respect to the Conference Room. This gives her access to the resources like Projectors, Lights etc. within the Conference Room. However, she should only be allowed to assume this role if she is present in the Conference Room. If she is not in the Conference Room and Bob is using it to discuss some research ideas with Carol - then Alice should not be allowed to assume a role that provides her access to the resources in the Conference Room.

Thus, we see that Location can be an important contextual cue required to make a decision about access rights. As such, it provides the primary context cue in our initial implementation. We model the requirement for location sensitivity relationships as context cues in our K-Base. Determining Location information is done via the PLACE [15] system built in our lab. PLACE handles location detection via sensor fusion and provides an interface accessible via the Metaglue/Hyperglue framework. Using PLACE we can determine the location of a given entity - provided we can obtain the rights to that entity's location information. We make this information available to our inferencing engine as an assertion made in the Context Keeper.

3) Extensibility Rule Base: While we desire to allow extensibility of our K-Base representation, we want to control it such that it does not occur in a haphazard manner. Thus, we include a notion of an Extensibility Rule Base that represents the manner in which the K-Base can be extended based on new knowledge. For example, in the scenario described above - Alice may in general consider that any friend of a friend of hers is also her friend. Thus if an unknown entity U tells Alice that U is a friend of Alice's friend Bob, Alice would consider U to be a friend of hers as well. An extensibility rule describing this would take the following form:

```
(EXTEND-RULE-1 if  (?requester friend ?my-friend)
               add (?requester is-a Friend)
```

However, this raises security concerns about whether a given entity's claims can be trusted at face value. We get around this issue using proven Public Key cryptographic techniques [4]. Each entity that is a part of this system has a unique digital ID in the form of a Public/Private Key Pair. A given entity knows the Public keys of each entity known to it. Thus, when an unknown entity makes a claim it first asks the commonly known entity to provide a token proving its relationship to the unknown entity. The unknown entity then presents that token as the relationship claim. The token itself is a piece of data representing the relationship which is signed using the known entity's private key. Thus, the claim can be validated simply by validating the signature on the claim provided the signing entity is a known and trusted one.

4) Common Ontology: This isn't explicitly defined in the Access Manager module. However, it is an essential part of the system. For Access Managers on different entities to be able to exchange information about relationships, they must exchange information that adheres to the same ontology. That is, we desire that all entities adhere to an ontological commitment. Therefore, each instance of an Access Manager running in our system represents its own knowledge based on a common ontology we have defined. In particular we first use OntoGen - a tool developed here in our lab for use with SEMANTIC - to describe the common ontology we desire, and proceed to build our knowledge representation upon that.

5) Inferencing Engine: Given the knowledge we store, we need a mechanism by which to reason on it when making access control decisions. For this purpose, we use a Kawa [14] based forward-chainer also built here in our lab. It was built to perform inferencing on Semantic Networks represented using SEMANTIC and as such was the favored approach.

In Hyperglue, when a request for a resource is received it also carries an identifier of the Requestor. We use these Requestor ID's as entity identifiers within our K-Base. This allows us to perform inferencing to determine if the requester should be allowed access to the desired resource. We make this decision in the following manner:

Step 1: Determine if the Requestor is known. This is achieved simply by checking if the Requestor is a known entity in our K-Base. If the Requestor is unknown, then we challenge the Requestor with a set of known entities and ask the Requestor to demonstrate a relationship with one of the known entities. If the Requestor can prove a relationship in the manner described above, that information is used to augment our existing K-Base. For the case where the Requestor remains unknown, we handle it in one of two ways:

1) We define the Role of an Unknown Entity and define a set of permissions associated with it.
2) We query the user for guidance, if one is available.

Step 2: Determine the Requestor's role. We do so by chaining on our K-Base to find the node associated with the Requestor's role.

Step 3: Determine if the Requestor has permissions to access the desired resource. Since the node associated with the Requestor's role has mappings with respect to the resources available to it, we examine those mappings to determine the Requestor's access rights. At this point, assertions about relevant contextual cues in the Context Keeper are also considered.

C. User Studies

While we feel this system is an appropriate representation of the properties we desire, validating it is an important task. We intend to test the validity of our approach by conducting a set of User Studies within our existing Intelligent Spaces. In particular we are interested in the system's performance under the following scenarios:

1) Single user interacting with a single intelligent space.

2) Multiple users interacting with a single intelligent space.
3) A Single user interacting with multiple intelligent spaces particularly, the change in access rights in different spaces.
4) Multiple users interacting with multiple intelligent spaces.

Being able to determine access rights correctly in these scenarios is our ultimate goal. We believe a set a of appropriately designed user studies will help us fine tune our design towards the end of achieving this goal.

VI. Future Work and Conclusions

We have highlighted the need for more focus on the security and privacy issues related to Intelligent Environments. In particular we have focused on Access Control issues. We have defined semantics with respect to Access Controls in Intelligent Environments and provided discussion regarding the properties desired in these Access Controls. An approach to address the issues raised here has also been proposed.

Future work will initially focus on useability testing. Obtaining data about the validity of our assumptions is essential to guide us in our design. In order to do so, we need to evolve our existing applications to be security conscious. However, like many other systems built before, security has been an afterthought in the Metaglue/Hyperglue framework. As such we have been forced to focus our attempts on retrofitting security within an existing and significantly large system. We would like to highlight that this is a non-ideal approach and encourage any future work in this area to consider addressing security issues at an early stage. Also, our work has so far ignored issues like Authentication. Approaches that may seem suitable for this domain like biometrics don't currently provide the level of trust that is seen in conventional cryptographic mechanisms. Therefore, authentication efforts are an area needing more effort.

This is undoubtedly a complex a problem. A perfect solution would involve solving the general AI problem. Our goal is simply to implement a useable system. As such, to deal with the problems that arise we desire a flexible and evolveable system. Approaches that focus on specifics and target particular classes of applications are difficult to use in the manner we desire. Therefore, we have found that a higher level modelling of the problem and related issues provides a useful framework to work in.

Acknowledgment

This research is funded by MIT's Project Oxygen.

References

[1] Hanssens, N., Kulkarni, A., Tuchinda, R., Horton, T.: Building Agent-Based Intelligent Workspaces. In: ABA Conference Proceedings. June 2002.

[2] Oxygen: Pervasive, human-centered computing. MIT Laboratory of Computer Science, Cambridge, Massachusetts, May 2002.

[3] Peters, S., Shrobe, H.: Using Semantic Networks for Knowledge Representation in an Intelligent Environment. In: PerCom '03: 1st Annual IEEE International Conference on Pervasive Computing and Communications. Ft. Worth, TX, USA, March 2003.

[4] Rivest, R. L., Shamir, A., Adleman, L. A.: A method for obtaining digital signatures and public-key cryptosystems; Communications of the ACM, Vol.21, Nr.2, 1978, S.120-126.

[5] Peters, S., Look, G., Quigley, K., Shrobe, H., Gajos, K.: Hyperglue: Designing High-Level Agent Communication for Distributed Applications. Originally submitted to AAMAS'03.

[6] Coen, M., Phillips, B., Warshawsky, N., Weisman, L., Peters, S., Finin, P.: Meeting the Computational Needs of Intelligent Environments: The Metaglue System. In: 1st International Workshop on Managing Interactions in Smart Environments (MANSE'99), pp.201–212. Dublin, Ireland, December 1999.

[7] Johanson, B., Fox, A., Winograd, T.: The Interactive Workspaces Project: Experiences with Ubiquitous Computing Rooms IEEE Pervasive Computing Magazine 1(2), April-June 2002

[8] Romn, M., Hess, C.K., Cerqueira, R., Ranganathan, A., Campbell, R.H., Nahrstedt, K.: Gaia: A Middleware Infrastructure to Enable Active Spaces. In: IEEE Pervasive Computing, pp. 74-83, Oct-Dec 2002.

[9] Song, Y.J., Tobagus, W., Leong, D.Y., Johanson, B., Fox, A.: iSecurity: A Security Framework for Interactive Workspaces Technical Report, Stanford University, September 3rd 2003

[10] Al-Muhtadi,J., Ranganathan, A., Campbell, R., Mickunas, M.D.: Cerberus: A Context-Aware Security Scheme for Smart Spaces. In IEEE International Conference on Pervasive Computing and Communications (PerCom 2003), Dallas-Fort Worth, Texas, March 23-26, 2003.

[11] Van Kleek, M.: Intelligent Environments for Informal Public Spaces: the Ki/o Kiosk Platform. Master's Thesis. MIT, Cambridge, MA. February 2003.

[12] Hammond, T., Gajos, K., Davis, R., Shrobe, H.: An Agent-Based System for Capturing and Indexing Software Design Meetings. In: Proceedings of the International Workshop on Agents in Design (WAID'02). Cambridge, MA, August 2002.

[13] Gajos, K., Kulkarni, A.: FIRE: An Information Retrieval Interface For Intelligent Environments. In: Proceedings of International Workshop on Information Presentation and Natural Multimodal Dialogue IPNMD 2001. Verona, Italy, December 2001.

[14] P. Bothner.: Kawa: Compiling Scheme to Java. In: Proceedings of the Lisp Users Conference, November 1998.

[15] Lin, J., Laddaga, R., Naito, H.: Personal Location Agent for Communicating Entities (PLACE). In: 4th International Symposium on Human Computer Interaction with Mobile Devices. Pisa, Italy, 2002.

[16] Nii, H.P.: Blackboard applications systems from a knowledge engineering perspective. In: The AI Magazine, 7(3), 1986.

[17] Nii, H.P.: The blackboard model of problem solving and the evolution of blackboard architectures. In: The AI Magazine, 7(2), 1986.

[18] Van Kleek, M.: k:info: An Architecture for Smart Billboards for Informal Public Spaces. UBICOMP 2003. Interactive Poster. Seattle, WA. 2003