ACCESS: Access Controls for Cooperatively Enabled Smart Spaces

Buddhika Kottahachchi BUDDHIKA@CSAIL.MIT.EDU
MIT Computer Science and Artificial Intelligence Laboratory, Stata Center, 32 Vassar Street, Cambridge, MA 02139, USA

1. Introduction

Smart spaces and the infrastructure they need have seen significant research interest. Yet, as the infrastructure matures and becomes applicable in real-world settings, a lack of focus in their privacy and security implications prevents widespread adoption. This is particularly important in applications that encompass multiple distinct entities.

1.1 Hyperglue

At the AIRE Research Group, we are currently building Hyperglue (Peters et al., 2003), an agent-based middleware infrastructure supporting interactions between multiple entities. Hyperglue organizes agents representing real-world resources (eg. telephones, projectors, addressbooks) into agent societies associated with real-world entities (eg. people, places). Furthermore, Hyperglue allows these agent societies to share resources amongst one another, thereby supporting multi-entity interactions. Previous work on smart spaces such as Metaglue (Coen et al., 1999), iROS (Johanson et al., 2002) and Gaia (Roman et al., 2002) have focused on interactions within a single distinct space. iROS and Gaia in particular focus on distinct interactive workspaces. However, Hyperglue opens up opportunities to explore more interesting applications similar to those orignally envisioned by Project Oxygen (Dertouzos, 1999).

1.2 Motivation

Yet, as we build Hyperglue, we have come to realize that we are opening up a range of unaddressed security and privacy issues. In truly pervasive applications encompassing multiple distinct spaces, trust cannot be taken for granted as is implicitly or explicitly done in the current crop of middleware platforms supporting smart spaces. Situations where interacting agents know little about each other can easily occur. For example, an arbitrary person entering a conference room for the first time may attempt to use the projector in that room. Whether access to this resource should be granted is a function of the level of trust that can be established between the person and the conference room which have no prior knowledge about each other. Therefore, a mechanism for determining trust and the corresponding level of access becomes a necessity. Efforts such as the iSecurity (Song et al., 2003) extension for the iROS project and Cerberus (Al-Muhtadi et al., 2003) for the Gaia project are early steps in addressing these issues. iSecurity handles authentication via a centralized model, but more interestingly provides for decentralized security policy enforcement. Cerberus also relies on a centralized authentication mechanism, but is interesting because it takes context into account when enforcing security policies.

Our own work is motivated by the need for such a mechanism in Hyperglue and driven by the notion that entities should not be able to share resources arbitrarily, but rather, they should only share resources when respecting the personal preferences of the entities involved. Based on the work done in iSecurity and Cereberus, we believe these preferences are best described by a combination of dynamic roles and dynamic context which are enforced in using a decentralized architecture. We also assert that the framework of our knowledge model should consist of very basic building blocks that do little to constrain the type of policies that can be defined by the end user. Previous efforts in this domain such as iSecurity and Cerberus require Administrators to define access control policies. This is a constraint we wish to move away from.

2. Design Challenges

Smart spaces are dynamic in nature and any access control mechanism applied to them must be adaptive. Furthermore, traditionally access controls have been enforced unidirectionally between a user and a system. Smart spaces require them to be bi-directional (eg. individual agent societies can serve as both a resource requestor and a resource provider) and therefore support a structure that is closer to peer-to-peer architectures. In keeping with the spirit of pervasive computing, we also require that access control mechanisms work with minimal active participation from the user. Determining access rights correctly requires us to consider a complex set of drivers representing relationships between entities and the current context governing those entities. With these in mind, we propose ACCESS: a framework for deploying access controls in Hyperglue-enabled smart spaces.

3. System Description

In ACCESS, we model and use both dynamic roles and dynamic context to determine access rights. We consider interactions between two types of Hyperglue Agent Societies: *Provider* Societies that have control over a resource of interest, and *Requestor* Societies that wish to use a resource. We assert that resources should only be released when the Requestor can assume a role having permission to use it, and the Provider is governed by an appropriate context. Furthermore, the Requestor's role needs to be defined and determined relative to the Provider. ACCESS is able to accurately determine access rights within these constraints. To enable this, ACCESS maintains a knowledgebase in every agent society that can function as a Provider. Our knowledge representation is modeled and maintained using the SEMANTIC(Peters & Shrobe, 2003) framework. In our model, each resource has associated with it a set of Requestor Roles and a set of Provider Contextual States.

Requestor Roles are defined in terms of other Requestor Roles and/or Evidence Attributes. Together, they are used to model the different roles an external agent society can assume and the constraints under which a role is applicable. Evidence Attributes can be digital IDs, delegation credentials or other attributes that can constrain the scope of when a role can be assumed. Since Evidence Attributes are provided by external agent societies, they are authenticated and validated before being accepted. Requestor Roles are dynamically assigned based on evidence provided and available at the time of a given Resource request.

Contextual States, like Requestor Roles, are hierarchical in structure and defined in terms of other Contextual States and/or Context Attributes. Context Attributes are individual context cues (eg. mobile phone status) that are known within the Provider society. Therefore, our context representation allows us to consider the entire spectrum of contextual cues available when necessary. A diagram of this representation applied to a simplified scenario is shown in Figure 1. Here, our model is such that Alice's mobile phone can only be released when the entity requesting it is a friend, and Alice is contactable. If the Requester can provide an Evidence Attribute that authenticates itself as Bob, then it is allowed to assume the role of *friend*. The contextual state *contactable* occurs when the Context Attribute determining the status of Alice's mobile phone is "idle" and the contexual state "awake" is applicable. Thus, eventhough Bob is a friend, Alice's mobile phone would be released to him only when Alice is awake and her phone is idle.

In ACCESS, resources can only be released when predefined pairs of Requestor Roles and Provider Contextual States are deemed applicable. This allows us to model access rights in a form that requires the Requestor to assume

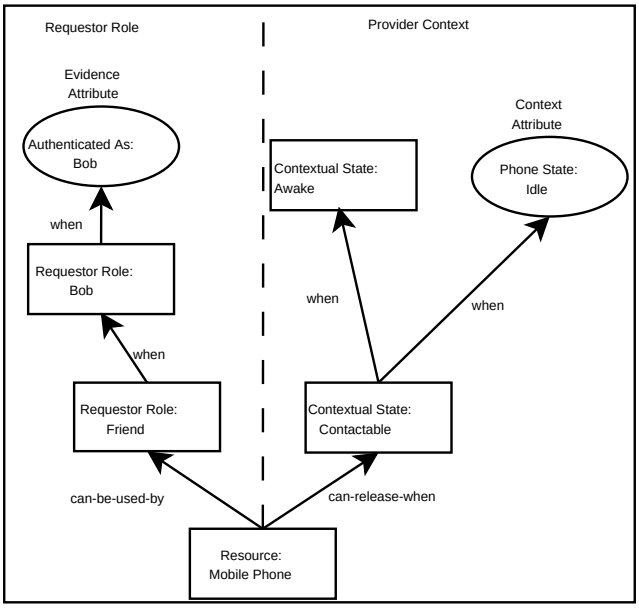

Figure 1. ACCESS Knowledgebase in Alice's Society

a role with permission to use a particular resource, but also where the Provider only releases it when the context affecting the Provider allows its release. This allows access rights to be both dynamic and adaptive as required by the smart spaces they are applied in. In order to determine which roles and states are applicable, we backward chain on the requested resource until a suitable Role/Context pair is found to be valid and applicable. This keeps the amount of reasoning that needs to be done at a manageable level.

Furthermore, grouping the Context Attributes into Contextual States and the Evidence Attributes into Requestor Roles allows us to create hierarchical structures that are easily evolvable and reusable. This is a very useful property that can be exploited in supporting end-user configuration of access control policies on different entities. Allowing end-user configuration is an important step towards having these technologies applied in real-world applications.

4. Current Progress and Future Work

At present, we are putting together the required infrastructure for this system. The requirements of the system and its current design has evolved over time as requirements were better understood and existing design issues were resolved. For example, we are currently also examining means for dynamic truth maintenance within the SEMANTIC framework since existing means weren't sufficiently flexible for the context modelling we desired.

Striving to maintain a design that is highly evolvable enables us to effectively address issues that arise along the

way. Our efforts are grounded on the notion that solving the problem perfectly may require a solution to the general AI problem, but keeping our system evolvable will enable us to make incremental progress as our understanding of the problem and therefore our representation of it improves.

Next, we plan to deploy and test this system's ability to correctly determine access rights amongst a set of Hyperglue enabled entities that interact with one another. A number of candidates for this purpose exist within the current corpus of work taking place in the AIRE group. Thereafter, we plan to introduce and evaluate an interface that allows end-user configuration. Both of these are important factors in determining the success of the system and we look forward to re-examining our work based on the feedback we collect.

References

Al-Muhtadi, J., Ranganathan, A., Campbell, R., & Mickunas, M. D. (2003). Cerberus: A context-aware security scheme for smart spaces. *Proceedings of the First IEEE International Conference on Pervasive Computing and Communications (PerCom '03)* (pp. 52–58). Dallas-Fort Worth, Texas, USA.

Coen, M., Phillips, B., Warshawsky, N., Weisman, L., Peters, S., & Finin, P. (1999). Meeting the computational needs of intelligent environments: The metaglue system. *1st International Workshop on Managing Interactions in Smart Environments (MANSE'99)* (pp. 201–212). Dublin, Ireland: Springer-Verlag.

Dertouzos, M. L. (1999). The future of computing. *Scientific American*.

Johanson, B., Fox, A., & Winograd, T. (2002). The interactive workspaces project: Experiences with ubiquitous computing rooms. *IEEE Pervasive Computing*, *1*, 67–74.

Peters, S., Look, G., Quigley, K., Shrobe, H., & Gajos, K. (2003). Hyperglue: Designing high-level agent communication for distributed applications. http://www.ai.mit.edu/projects/aire/papers.shtml.

Peters, S., & Shrobe, H. (2003). Using semantic networks for knowledge representation in an intelligent environment. *PerCom '03: 1st Annual IEEE International Conference on Pervasive Computing and Communications*. Ft. Worth, TX, USA: IEEE.

Roman, M., Hess, C., Cerqueira, R., Ranganathan, A., Campbell, R. H., & Nahrstedt, K. (2002). Gaia: a middleware platform for active spaces. *SIGMOBILE Mob. Comput. Commun. Rev.*, *6*, 65–67.

Song, Y. J., Tobagus, W., Leong, D. Y., Johanson, B., & Fox, A. (2003). *isecurity: A security framework for interactive workspaces* (Technical Report). Stanford University.