

State of the Art: Policy Techniques for Adaptive Management of Smart Spaces

Kevin Carey, Kevin Feeney, Dave Lewis
Knowledge and Data Engineering Group
Trinity College Dublin

1. Introduction

Policies represent an important existing approach to providing runtime flexibility in the operation of management components and systems. Smart Space will require management systems that are highly adaptive and which can accept modification to their behaviour at runtime from a number of appropriately authorised roles. Policies will therefore be a key mechanism in providing the runtime configuration of component behaviour needed in any adaptive smart space management system. This paper aims to give a brief overview of the state of the art in the structure of policy languages and their application, the latter with a focus on access control and management of quality of service.

2. Overview

A policy is a rule that can be used to change the behaviour of a system. They can be considered as declarations of business rules that an organisation wishes to apply to the operation of its systems. In general policies are expressed in terms of an *event* which triggers the evaluation of a policy rule, a set of *conditions* that must be met for a policy rule to be enacted and a set of *actions* that are performed upon such enactment. A policy management system is tasked with interpreting policies to enact behaviour on a set of devices. Policy events are mapped to the requests made on those devices or specific state change events, conditions are mapped to specific device states and actions to specific device operations. As policies are declarative and interpreted by policy management systems, they can be updated at runtime to flexibly control the behaviour systems. Policies are therefore being increasingly widely used in a variety of network and system management applications to provide an element of adaptability and run-time configurability in the behaviour of networks and information systems. Policies are useful in applying a common set of operational rules to a large set of distributed managed nodes and/or to an information system with a large set of users. The ultimate aim of policy based system is to derive policies from business goals, so that the operation of an organisation's systems can respond dynamically to changes in those goals.

Policy based management rests on the assumption that sets of policies can be applied to classes of devices, users or services. This allows easier management by grouping individual units into classes with common requirements and obligations with respect to the overall system while still retaining the ability to exercise fine-grained control over the choices in the behaviour of a system. Policies have many areas of application including the following. When policies are applied to network bandwidth and routing, they provide a mechanism for specifying Quality of Service (QoS) rules for classes of service (Lymberopoulos 2002). When policies are applied to users, they provide a mechanism for specifying access control for classes of user (roles), known as role based access control (RBAC) (Sandhu et al 1996). When policies are applied to nodes they provide a means for distributed configuration management (Crane 1995). By enabling decisions to be made closer to where the event and condition are detected, policies allow a less centralized and more flexible management

architecture, which could be a particularly important feature in the complex and heterogeneous environment of Smart Spaces.

3. Analysis

This analysis first addresses the structure and capabilities of various policy languages and the challenges presented by developing policies in these languages, e.g. conflict between different policy rules addressing the same resource. An understanding of the expressiveness and limitation of policies is required if they are to be used in engineering Smart Space management systems. The section then goes on to analyse the application of policies in role-based access control and resource management. Smart space management requires multiple people to monitor and exert control over resources, such as bandwidth, display real-estate and application services, without the benefit of boundary-based grouping of resources or people. A finer grained and more flexible mechanism is therefore required to control access to resources, for which role-based access control using policies is a strong candidate. The process of managing and sharing resources in smart spaces must react quickly to changes in the physical and system environment and so the intelligence needs to perform such reactive adaptation must be placed as close to the resource as possible. Policy-based resource management provides a mechanism for achieving this, and we examine the state of the art with respect to network resource management and its relationship to delivering quality of service to users.

3.1 Policy Languages

There are a number of approaches to the definition of policies, and accompanying policy languages, which represent a number of different levels of policy expressiveness and policy enactment semantics. There is therefore no single accepted policy languages, with many languages being proprietary in nature and tied to particular system management products. Policy languages broadly split between ones addressing access control and ones addressing resource management (Sloman and Lupu 2002). The most widely accepted standardised model for resource management is the joint policy model from the Internet Engineering Taskforce (IETF) (Moore et al 2001) and Distributed Management Task Force (DMTF) (Rafalow 2002), which has been adopted by both organisations for Internet and enterprise management applications. PONDER is one of the more expressive policy languages addressing both access control (i.e. authorisation policies) and management (i.e. obligation policies). PONDER supports the notion of domain membership for applying policies to specific subject and target objects. Authorisation policies define which subject groups are authorised to perform which actions of which target groups. Obligation policies define under which condition a subject object must perform on certain actions on a target object. Figure 1 gives some example of authorisation policies in PONDER. Work with PONDER (Lupu and Sloman 1999) has highlighted some of the broader engineering problems with policies such as the sort of conflicts that can arise through its use.

```
inst auth+ switchProfileOps {
subject /NetworkAdmin ;
target <ProfileT> /Nregion/switches ;
action load(), remove(), enable(), disable() ;}
Members of the NetworkAdmin domain are authorised to load, remove, enable or disable objects of type ProfileT in the Nregion/switches domain.
```

```
inst auth- /negativeAuth/testRouters {
subject /testEngineers/trainee ;
action performance_test() ;
target /routers ;}
Trainee test engineers are forbidden to perform performance tests on routers. The policy is stored within the /negativeAuth domain.
```

Figure 1: Example of Policies in PONDER

Conflict detection is a crucial area if policies are to be used on any scale. Conflicts can be modal conflicts, for instance where a positive and negative authorization apply to the same objects, or application specific conflicts related to the semantics of the resources and roles in the target and subject domains of policies. Specification-time conflict detection is important since different people in an organisation may author different policies at different times and because policies are typically interpreted at run-time, computationally expensive conflict detection at that point would be impractical. Detection of overlaps between the tuple set of subjects, actions and targets for a set of policies helps detect modal conflicts, and if one tuple set is more specific in one of those domains than another, its policy is given precedence. Meta-policies are policies about policies, and may be used to detect some semantic conflicts between policies and to guide the resolution of those conflicts. Overall, resolving policy conflicts is an unsolved problem, largely still relying on manual policy-rewriting, including the redefinition of subject and target domains.

The application of policy to role based access control and QoS management is discussed in more detail below.

3.2 Policy and Role Based Access Control

Most research in the area of Smart Space Management has focused on providing a limited number of wireless services to a clearly defined user base in a specific environment. For example, the provision of location-aware multimedia content to visitors to museums (Semper and Spasojevic 2002), or providing a small number of location-aware services to students on a college campus (William et al 2002). In general these projects can make assumptions about access control and security that are not applicable to envisaged real world smart spaces since:

They assume a single role, i.e. every user has the same rights and privileges as the others with respect to the smart space services. Roles are a means of modelling classes of user in terms of what the user wants or needs to do.

The smart space services only apply to particular mobile computing devices, which are configured by the smart space managers and distributed to users. Thus there is no variability in terms of potential devices within the space.

There is a limited number of smart space enabled services which do not vary dynamically.

For these reasons, many of the current smart space research projects have been able to adopt relatively simple security and access control models, which can rely upon the fact that the mobile devices and users which utilize the smart space services are pre-defined and static. However, there are several current projects that have started to look at the type of problems associated with more universal and interoperable smart space management, such as GLOSS (Gloss 2001) and OXYGEN (Oxygen 2002).

When attempting to devise general security models for real world smart space management systems, there is a need to accommodate a much greater diversity in terms of the users, devices and services available. This can be done with traditional Access Control Lists (ACL's), whereby each resource is associated with a particular ACL and only users included in the ACL have access to that resource. Policies provide several advantages over traditional ACL's since they are much more expressive and several powerful policy description languages exist (eg PONDER), and they allow for much more

convenient distributed management. In addition there are a number of very good reasons, specific to smart spaces, for using policies and role based access control in any smart space security model.

Firstly, it is almost always the case that access control rights are not randomly variable with respect to individuals, but are clustered around certain classes of user or roles. For example ordinary users are commonly granted certain rights, to access printers, network resources, etc... Other classes of users such as administrators are granted certain extra privileges. In general rights are related to the role of a user rather than to the individual identity of the user, thus it is much more convenient to manage these rights with respect to roles rather than on an individual basis, especially when most systems will have far more users than roles defined on them. This is the traditional justification for policy and role based access control.

The majority of RBAC research has focused on defining models that allow decentralisation of role creation and administration. One of the most influential initiatives is the ARBAC 97 (Sandhu et al 1997) (Administrative Role Based Access Control) family of models which proposes 4 different types of RBAC models with special administration roles which allow creation of further roles. However this approach is problematic for dynamic collaborative environments since there is no guarantee that participants will be able to fill pre-defined administrative roles (Crook et al 2002). The type of RBAC model used also depends on the policy language. Some languages, like PONDER are very expressive, including obligation, refrain and delegation policies alongside environmental role activation constraints, while others opt for simplicity in pursuit of comprehensibility, like OASIS (Yao et al 2001).

In smart spaces, a user's access rights and obligations towards resources is not necessarily static. Access rights can vary due to context. For example a user may have different rights depending on what other users are present in the space. This dependence of rights on context presents a difficult problem for traditional ACL mechanisms. However, when using policies and RBAC, these contextual changes can be easily modeled as role changes. The roles available to a particular user can change according to the context. Therefore, as long as we can manage this type of role variability, the access control system does not need to be changed in any way to take account of context – a significant advantage – however, the question of role mapping with changes in context remains a difficult problem and little work has been done in the area.

Furthermore, smart spaces present particular problems to traditional authentication systems. Smart spaces can be composed of a variety of services, some of which will not be particularly sensitive from a security point of view, while others may require high levels of confidence in the authentication of users. Ideally we would like to allow users to authenticate themselves to the smart space only to the required level for those services which they will use during their presence in the smart space. By associating different types of authentication with different roles and allowing users to choose which roles to activate upon entering the smart space, we can provide a seamless, integrated means by which a user can authenticate herself to the smart space to the minimum required level. For example, a typical user might be able to authenticate herself to a space by means of an active badge (Want 1992), but if the user wants to activate her 'administrator' role, a password or biometric identification mechanism may be required. Thus variable levels of authentication can be associated with different roles in a seamless manner.

The Gaia project (Roman and Campbell 2000) at the University of Illinois has adopted a powerful model for security and access control using roles, policies and different authentication methods with variable confidence levels among other measures. Gaia uses credentials, similar to Kerberos tickets, as a basis for authentication and access control (Viswanathan et al 2001). In Gaia, all resources in the system are associated with policies. Users present credentials to the system which are compared with policies in order to validate particular resources. Credentials are closely linked to roles. Users can activate any subset of their valid roles and receive credentials associated with these roles, which then allow them to access resources. This work demonstrates the flexibility and power of policies and role based access control in the realm of smart spaces. However, it is a very new research area and much work remains to be done.

Projects like Gaia show how the latest techniques in access control mechanisms can be very useful in solving some of the particular problems associated with Smart Spaces. Roles, in conjunction with policies, provide a convenient mapping for access rights and authentication methods that vary with context. However, there are still several areas that have yet to be addressed in the realm of Access Control for smart spaces. Policy refinement, role delegation, collaborative management and management of shared resources are all areas that have yet to be touched on in the smart space context.

3.3 Policy-based Resource Management

To manage a smart space with all its devices and their interactions, a Policy-based QoS management system is considered in order to assure best end to end QoS for the individual user as well as to the smart space group.

IETF and DMTF have jointly produced a set of standards on policy and policy-based management systems. The two main elements in their model of a Policy-based management system are Policy Decision Point (PDP), a logical entity that makes policy decisions for itself or for other network elements that request such decisions; and Policy Enforcement Point (PEP), a logical entity that enforces policy decisions (Westerinen 2001). PDP is likely to store its policies in a repository, such as a Lightweight Directory Access Protocol (LDAP) directory service.

The basic interaction between the components begins with the PEP. The PEP will receive a notification or a message that requires a policy decision. Given such an event, the PEP then formulates a request for a policy decision and sends it to the PDP. The PDP returns the policy decision and the PEP then enforces the policy decision by appropriately accepting or denying the request (Yavatkar 2000). Common Open Policy Service (COPS) (Fernandez 2001) can be used as a policy transaction protocol between the PDP and PEP for transporting the policy requests and decisions.

Optionally, there can be a local PDP (LPDP) in each network domain and a high level PDP for the overall the network. PEP will first use the LPDP to reach a local decision. This local decision and the original policy request are next sent to the PDP, which renders a final decision for the good of the overall network (Yavatkar 2000). This way gives a better scalability to the policy management system.

One of the main application areas of policy research in network management is the management of Quality of Service guarantees. According to (Ponnappan 2002), there are two models used in Policy Based Network (PBN) for policy management, namely outsourcing and provisioning. In outsourcing model, when PEP has to make a decision regarding an event, it outsources the decision-making to the

PDP. It is typically used with RSVP requests, where PDP receives a request from PEP for an admission control decision regarding an RSVP request for resource reservation. Based on high-level policies found in the repository, PDP either allows or deny a RSVP signalled packet to enter the network. If using COPS for the policy transaction, COPS-RSVP (Herzog et al 2000) is a specification of functionalities for this model.

Whereas in provisioning model, the PDP typically predicts future configuration needs, and proactively pre-provisions for them ahead of time. It is most commonly used for controlling network policy for non-signalled protocols, such as Diffserv, where PDP decides, based on various criteria, whether the newly added policy should be installed in the PEP by sending policy rules as configuration commands that the device can interpret, so that policy decisions are enforced by classifying data packets as they enter the network and processed accordingly. COPS specifies functionality COPS-PR (Chan et al 2001) suited for this model.

The policy server, in most cases, is the PDP but it might have other components to help in its decision-making, such as a bandwidth broker to manage the bandwidth usage information in the network. There would probably be a policy repository connected to the policy server to store the policy information and a policy management tool for creating/modifying/deleting the policies. Policy server might optionally contact an external/remote entity, such as Authentication, Authorization and Accounting (AAA) server to check whether a user is authorised to use the service requested in order to make its decision. E.g., the AAA server could be part of the ISP.

Policies are either created by an administrator using the policy management tool or they are translated from a Service Level Specifications (SLS) (Lymberopoulos et al 2002). After validation and static conflict tests, the new policies are stored in the policy repository. High-level policies are mapped down to lower-level policies or device specific commands.

PDP is the heart of the system and its decision algorithm is vital for the system performance. There are several different type of algorithms developed, such as a bacterial/genetic algorithm (Marshall et al 2001) that autonomously removes policies that degrade its performance, or a fuzzy logic algorithm (Fernandez et al 2001) to offer better QoS in a Diffserv domain because of the uncertainty and inaccuracy characteristic of the data flow estimate. Also, the PDP can have an algorithm for an adaptive policy framework (Lymberopoulos et al 2002) that dynamically changes the parameters of the QoS policies (high-level) at run-time or enable/disable policies from a pre-defined set. Still there is plenty of scope for more research on this area.

4. Research Directions

Currently Policy-based management suffers from fragmentation of approach, partly due to differences in semantics between access control policy languages and resource management policy languages. As a consequence there is no commonly accepted policy language and no common approach to the engineering of policy based systems. The validation of policies prior to their deployment is not a solved problem, with some policy conflicts still only detectable through unwanted runtime behaviour. The promise of policies being driven from business goals remains largely unfulfilled, with policy decomposition and hierarchical translation still presenting difficult challenges.

Some policy-related research directions that have been identified with relation to adaptive smart space management are:

- One of the strengths of policy based management is that it is essentially a decentralised activity, allowing different people to distribute their policies onto a distributed system. However this raises the problems of conflicts between policies, especially when generated by different people in different roles. Though the use of role and domain abstractions in languages like PONDER help constrain the policy conflict detection problem, a fuller conflict resolution approach may be, as suggested in (Lupu and Sloman 1999), may result from deriving policies from higher level user goals and use analysis of these goals to shape the definition of non-conflicting requirements. As adaptive Smart Space management required some model of user context and goals, this may integrate well with the generation of conflict free policies. Application specific conflict detection may also be aided by better semantic descriptions of subjects, targets and actions, using ontology-based mark-up languages such as DAML+OIL and DAML-S.
- Current policy-based management schemes assume that policies operate on a population of existing objects, with most applications addressing fine grained managed objects that directly represent resources to be managed. There seems however, little attention paid to the potential role of policies in providing flexibility to coarser grained object in a generic way, in particular COTS components. This could involve the vendor of a software component specifying the policy events, conditions and actions that may be combined to modify the behaviour of that component at run-time. The responsibility for ensuring the expression of policy semantics for a component and the correct behaviour of different policy combination within the components is delegated to the COTS developer, arguable the best placed person to perform this task. This approach also raises the issue of adaptive techniques that merge service composition of services with configuration of behaviour of the COTS components that offer those services.
- Projects like Gaia show how the latest techniques in access control mechanisms can be very useful in solving some of the particular problems associated with Smart Spaces. Roles, in conjunction with policies, provide a convenient mapping for access rights and authentication methods that vary with context. However, there are still several areas that have yet to be addressed in the realm of Access Control for smart spaces. Policy refinement, role delegation, collaborative management and management of shared resources are all areas that have yet to be touched on in the smart space context.
- There are also several areas that remain to be researched in the complex area of access rights and roles. There are several decidability problems relating to accumulation of roles and the attendant matrix of access rights. We may find mutually contradictory policies associated with different concurrently held roles, or we may have a combination of roles which gives unintended access rights to the holder of specific combinations. The analysis and resolution of these decidability problems is a difficult problem which has thusfar barely been touched upon by the research community.

5. References

Chan, K., Seligson, J., Durham, D., Gai, S., McCloghrie, K., Herzog, S., Reichmeyer, F., Yavatkar, R., Smith, A.(2001), "COPS Usage for Policy Provisioning (COPS-PR)", *IETF RFC3084*, March 2001.

Crane, S., Dulay, N., Fossa, H., Magee, J., Sloman, M.(1995), "Configuration Management for Distributed Software Services", *Proc. IFIP Int. Symposium on Integrated Network Management (ISINM 95)*, Santa Barbara, Chapman Hall, May 1995, pp. 29-42.

Crook, R. Ince, D., Lin, L., Nuseibeh B. (2002), "Security Requirements Engineering: When Anti-requirements Hit the Fan" Security Requirements Group Department of Computing, The Open University Walton Hall, Milton Keynes.

Durham, D., (Ed.), J. Boyle, J., Cohen, R., Herzog, S., Rajan, R., Sastry, A., (2000), "The COPS (Common Open Policy Protocol) protocol", *IETF RFC2748*, January 2000.

Herzog, S. (Ed.), Boyle, J., Cohen, R., Durham, D., Rajan, R., Sastry, A.(2000), "COPS Usage for RSVP", *IETF RFC2749*, January 2000.

Fernandez, M.P.; Pedroza, Ade.C.R.; de Rezende, J.F., (2001) "QoS Provisioning across a Diffserv Domain using Policy-based Management", *Proceeding of Global Telecommunications Conference (GLOBECOM '01)*, IEEE , Volume: 4.

"Gloss", (2001), (www.gloss.cs.strath.ac.uk), Available: <http://www.gloss.cs.strath.ac.uk/project.html> (accessed: 2003, March 4).

Lupu, E., Sloman, M., (1999) "Conflicts in Policy Based Systems Management", *IEEE Transactions on Software Engineering*, Vol.25, No.8, November/December, pp 852-869.

Lymberopoulos, L., E. Lupu, E., Sloman, M. (2002), "An Adaptive Policy Based Management Framework for Differentiated Services Networks", *Proc. 3rd IEEE Workshop on Policies for Distributed Systems and Networks (Policy 2002)*, Monterey, California, June 2002, pp147-158.

Marshall, I.W.; Gharib, H.; Hardwicke, J.; Roadknight, C., (2001) "A Novel Architecture for Active Service Management", *International Symposium on Integrated Network Management (IM'01)* IEEE/IFIP.

Moore, B., Rafalow, L., Ramberg, Y., Snir, Y., Strassner, J., Westerinen, A., Chadha, R., Brunner, M., Cohen, R. (2001), "Policy Core Information Model Extensions", Internet Draft <draft-ietf-policy-pcim-ext-01.txt>.

"Oxygen Project, Pervasive, Human-Centred Computing" (2002), (oxygen.lcs.mit.edu), Available at: <http://oxygen.lcs.mit.edu/index.html> (accessed: 2003, March 4)

Ponnappan, A.; Lingjia Yang; Pillai, R.; Braun, P.(2002), "A Policy based QoS Management System for the InServ/Diffserv Based Internet", *Proceedings of Third International Workshop on Policies for Distributed Systems and Networks, 2002*.

Rafalow, L. (2002), "CIM Policy Model Whitepaper", DMTF document DSP0108, v2.6.0, Available at: <http://www.dmtf.org/standards/documents/CIM/DSP0108.pdf>

Román, M., Campbell, H. (2000), "GAIA: Enabling Active Spaces", *Proceeding of 9th ACM SIGOPS European Workshop*, September 17th-20th, 2000. Kolding, Denmark, pp. 229-234,.

Sandhu, R., Coyne, E., Feinstein, H., Charles Youman, C., (1996) "Role-Based Access Control Models" *Laboratory for Information Security Technology, George Mason University, , IEEE Computer*, Volume 29, Number 2.

Sandhu, R., Bhamidapati, V., Munawer Q. (1999), "The ARBAC97 Model for Role- Based Administration of Roles", *ACM Transactions on Information and System Security*, Vol. 2, No. 1.

Semper, R., Spasojevic, M., (2002) "Exploratorium - The Electronic Guidebook: Using Portable Devices and a Wireless Web-based Network to Extend the Museum Experience - Overview and Findings to Date", *HP Labs. Paper presented at Museums and the Web Conference*, April 2002

Sloman, M., Lupu, E., (2002) "Security and Management Policy Specification", *IEEE Network*, vol.16 No. 2, March/April 2002, pp.10-19.

Viswanathan, P., Gill, B., Campbell, R.H. (2001) "Security Architecture in Gaia", *Technical Report UIUCDCS-R-2001-2215 UILU-ENG-2001-1720*, University of Illinois at Urbana-Champaign.

Want, R., Hopper, A., Falcao, V., Gibbons, J. (1992), "The Active Badge Location System", *ACM Transactions on Information Systems*, Vol. 10, No. 1, January 1992, pp 91-102

Westerinen A. et al. (2001), "Terminology for Policy Based Management", IETF RFC3198.
Griswold, W.G., Boyer, R., Brown, S.W., Truong, T.M., Bhasker, E., Jay, G.R. Shapiro, R.B.(2002), "ActiveCampus - Sustaining Educational Communities through Mobile Technology?", *UCSD CSE technical report #CS2002-0714*

Yao, W., Moody, K., Bacon, J. (2001), "A Model of OASIS Role Based Access Control and its Support for Active Security", *Proceedings of SACMAT'01*, May 34, 2001, Chantilly, Virginia, USA.

Yavatkar, R. Pendarakis, D. Guerin, R. (2000), "A Framework for Policy-based Admission Control", *IETF RFC2753*.