Beyond the role model:
Organisational modelling in policy based management systems

Kevin Chekov Feeney
Knowledge and Data Engineering Group,
Trinity College Dublin, Ireland
353-87-7939931
Kefeeney@cs.tcd.ie

ABSTRACT
In this paper, we discuss recent developments in the field of Policy Based Management (PBM) and Role Based Access Control (RBAC). We postulate that PBM tends towards a total description of an organisation's operations. We examine the organisational models used in PBM and analyse them from the point of view of organisational theory. We conclude that the current organisational models and engineering methodologies employed in RBAC systems suffer from a simplistic, Taylorist, conception of the organisation. We propose a new approach to modelling organisations for PBM systems. Our approach replaces roles with communities (which may be granted autonomy in specific realms) as the fundamental unit of organisational analysis and integrates them with organisational decision making mechanisms to build a more accurate model of the real-world organisation.

Keywords
Role Based Access Control (RBAC), Policy Based Management (PBM), Organisational Studies (OS), Organisational Modelling.

1. INTRODUCTION
Policy-Based Management (PBM) has become a widely employed and promising solution for managing enterprise-wide networks and distributed systems. Such systems are driven by business needs, which require management solutions that are both self-adapting, in that they respond to changing conditions, and that dynamically change the behaviour of the managed system accordingly. Security and access control is an area where the application of PBM is particularly attractive. In today's Internet-based environments security concerns tend to increase as programmable mechanisms are introduced to enable such adaptation. As systems expand in size and become every more distributed, the task of managing the subjects and objects in the system becomes ever more difficult and we need convenient ways of grouping them together for efficient management. Role Based Access Control (RBAC) has recently become the dominant paradigm in the research community for regulating access to information systems. Role-based models regulate the access of users to the system on the basis of the activities the users execute in the system, rather than being purely based on the identity of the user.

Central to the idea of PBM is that policy drives management procedures. The person who defines policy exercises control within the organisation - as policy dictates those choices that will be made in the behaviour of the system. The distribution of rights to affect changes to policy is essentially the management structure of the system. Some work has been done in defining models for administration of RBAC systems. However, these models generally create independent administrative hierarchies alongside user hierarchies and do not reflect the complexity or inter-relatedness of the roles and groups within an organisation. The role hierarchies are based on simplistic, Taylorist notions of how power is arranged in real organisations and there is little or no provision for modelling of teams and workgroups. This is one of the major reasons that, although RBAC models have flourished in academia, there have only been minimal attempts to introduce the concepts into commercial systems. In this paper, we examine the current RBAC models from the viewpoint of the latest research in organisational studies. We propose a model for managing and distributing policy definition rights among groups and individuals within an organisation in such a way so that it will dynamically model the real power structure of the organisation. We show how this model can be used to provide access control to the resources that are controlled by the organisation, while helping to analyse and improve its existing structure.

Such an approach is seen as vital to applying policy based management to ubiquitous computing. The dynamically adaptive nature of ubiquitous computing environments means that there is a significant blurring to the traditional binding between the owners and operators of resources and the organisational allegiances of the people who use those resources. Currently, within an organisation, resources are operated for the benefit of people employed by and working for that organisation or for their customers. Access restricted to resources and the control over those resources is restricted to clearly defined groups of employees and customers. However, ubiquitous computing offers the possibility of using resources in whatever surrounding we find ourselves, so we need a very flexible way for the owners of resources to managing access and control to those resources by a very dynamic and fluid collection of people. Policy-based management provides a flexible mechanism for managing access and control of resources, but exhibits complexity limitation when dealing with rich organisational structures that prevent if being administered in a highly dynamic way. By linking the administration of policies to the changing community structure that drive the need for dynamic resource management, this research aims to provide a mechanism where the management of ubiquitous computing resources are matched closely to the organisation structures it must support.

2. Policy

A policy is a rule that can be used to change the behaviour of a system. Policies can be considered as declarations of the 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 resources. Policy events are mapped to the requests made on those resources or specific state change events; conditions are mapped to specific resource states and actions to specific resource operations. As policies are declarative and interpreted by policy management systems, they can be updated at runtime to flexibly control the behaviour of 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.

PBM rests on the assumption that sets of policies can be applied to classes of nodes, users or services. This allows easier management by grouping individual units into classes. 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 [1]. 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) [2]. When policies are applied to nodes they provide a means for distributed configuration management [3]. By enabling decisions to be made closer to where the event and condition are detected, policies allow a more decentralised management architecture, which is a particularly important feature in the dynamic, complex and heterogeneous environments that are increasingly common in today's networked world.

The main benefits of using policy are improved scalability and flexibility for the management system. *Scalability* is improved by enabling the uniform application of a single policy to large sets of devices and objects, while *flexibility* is achieved by separating the policy from the implementation of the managed system. Policy can be changed dynamically, thus changing the behaviour and strategy of a system, without modifying its implementation or interrupting its operation. PBM is supported by standards organisations such as the Internet Engineering Task Force (IETF) and the Distributed Management Task Force (DMTF), and most network equipment vendors.

Bureaucratic organisations strive to achieve control over their resources, to allow them to implement organisation-wide standard rules. In particular, organisations seek technologies that can provide uniform security guarantees across the organisation. Current practices commonly see a variety of independent subjects with practical responsibility for managing the diverse information resources of organisations. The dissemination and implementation of organisation-wide policies, depending as it has on independent subjects interpreting natural language policy directives, often results in haphazard dissemination of policy and uneven implementation using a variety of incompatible technologies. Thus, bureaucratic organisations see policy-based management as a tool for achieving uniformity and standardisation of management and security measures across their information resources.

In addition to the classic bureaucratic organisation, the information revolution has seen the rise of new organisational paradigms. The widespread access to networked information systems and the ubiquity of the Internet has facilitated organisations that do not operate in the traditional, hierarchical, centralised, bureaucratic manner. Internet communities often operate without any formal hierarchy, with a fluid membership and constantly evolving goals within a rapidly changing technical framework. In many cases, these unstructured organisations manage as broad a range of information resources as do their traditional counterparts. In these new, less structured environments, PBM is seen as a way of providing general guidelines for the provision of personalised services, tailored to the context, in a potentially chaotic environment.

2.1 Policy Specification

In order to realise the potential of PBM, there is a fundamental requirement for a means of precisely, explicitly and unambiguously specifying implementable policies. This requirement has been addressed by the development of policy languages that can be used to precisely describe the rules governing the behaviour of a system and can be deployed to enforce particular policies. In general, these languages are not applicable to high-level, *abstract policies* such as service level agreements or natural language policies like mission statements and organisational goals. These high-level policies must be refined to lower *specification level* policies before they can be implemented. There must also be a means to deploy these policies to the target devices before they can be enforced.

There are a number of distinct 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 widely accepted policy language. Many languages are proprietary in nature and tied to particular system management products. Policy languages broadly split between ones addressing access control and security and ones addressing resource management [4]. Broadly speaking, access control approaches focus on supporting roles and *authorisation policies* while resource management approaches focus on *obligation policies* (expressed in this domain as *event-condition-action*), configuration management and enterprise modelling. The following sections give a brief sketch of the various approaches.

2.1.1 Security policy approaches

Security specification derives from work on formal security models and thus many of the approaches in this area use formal logic languages concentrating on proving properties of the security system. The Authorisation Specification Language (ASL) [5] is an example

of a formal logic language for specifying access control policies. Although formal specifications are particularly useful in the security realm because they allow reasoning about the specified policies to enable the detection of conflicts or inconsistencies, they are generally non-intuitive and they cannot be directly translated into an implementation due to their abstract nature [6].

While most work in security policy specification has concentrated on the use of formal logic, there have been some proposed approaches to high-level security languages. In general, high-level languages sacrifice formal provability of security concepts for ease of comprehension and can be implemented directly. Their goal is to provide the user with a notation for expressing policies in an environment-independent way. An example of this type of language is the Security Policy Language (SPL) [7]

2.1.2 Resource management policy approaches

The most widely accepted standardised model for policy-based resource management is the IETF/DMTF policy core information model (PCIM), which has been adopted by both organisations for Internet and enterprise management applications. The IETF Policy Core is only a conceptual information model for describing policymanagement and is not bound to a particular implementation. This model considers policies as rules that take the form of *if condition(x) then action(x)*. This differs from the obligation policy specification above (*event-condition-action*) in that the event is implicit and is assumed to trigger the particular action by a particular traffic flow or user action. The IETF model does not define a language but rather a generic information model for representing policy information according to the rule based approach above. The IETF information model defines two hierarchies of object classes: structural classes representing policy information and control of policies, and association classes that indicate how instances of the structural classes are related to each other. The model also defines an associated policy management architecture, which defines policy decision points and policy enforcement points (PDPs & PEPs), to enable the separation of the verification and enforcement of policy rules [8].

2.1.3 Ponder: a general policy language

Ponder [6] is a complete PBM framework which includes a specification language. It is derived from over 15 years of work in the area of policy at Imperial College, London. It is one of the more expressive policy languages addressing both access control (authorisation policies) and management (obligation policies). As both obligations and authorisations can be applied to roles, Ponder is an example of Role Based Management (RBM), a superset of RBAC. Ponder also includes many of the more recent developments in policy research, including support for relationships and management structures (discussed further below). It also supports the notion of hierarchical domains for applying policies to specific subject and target resources and for grouping managed objects into convenient collections to enable scalable enterprise-level management.

3. Role modelling in PBM

The concept of role is well established in the literature of sociology and management science. It can be defined as "the set of rights and duties associated with a position, which are assigned to person who occupies that position" [9]. The utility of roles in access control models is not new [10], but has been given new impetus with the arrival of PBM systems. The use of roles in access control rests upon the observation that in most organisations access control decisions are based upon the role that the individual is acting in, rather than their identity. This observation, and the RBAC approach that springs from it, allow us to take advantage of several extra efficiencies in the specification of a policy-based system.

- A single policy applies to all members of a role, rather than needing policies for each individual member of the role.
- When an individual leaves or joins a role, there is no necessity to change the policies associated with the role.
- Policies that are common to many large organisations such as separation of duties can be implemented conveniently through roles, by declaring separate roles with constraints upon individuals concurrently activating these roles.

3.1 Role relationships

It is self-evident that roles within an organisation do not exist in isolation from each other. Roles are related to each other in a variety of different ways and these relationships provide opportunities for realising greater efficiencies in the management and specification of roles. Most approaches to role relationships in RBAC apply the concepts of role hierarchy and inheritance derived from the object-oriented paradigm popular in computer programming. The motivation for the creation of such role hierarchies is similar to the motivation in object-oriented programming. In object-oriented programming, inheritance allows us to reuse methods and attributes. Policy inheritance allows us to reuse sets of policies between roles that, although not identical, have a set of common policies. The benefits of such policy reuse are obvious as they allow us to apply universal policy changes across all roles in an organisation by changing a single policy. Without such reuse of policies, a large distributed PBM system would quickly become impossible to manage and full of contradictions and inconsistencies, which might undermine the integrity of the system. However, despite early optimism, the concept of role hierarchies has proved to be problematic with respect to real world organisations

In [11] Moffett identifies three distinct role hierarchies, based on different relationships between roles, all of which form partial orders, namely:

- Generalisation/specialisation, or the *isa* relationship. This relationship is useful for describing hierarchies based upon specialisation. For example, a neurosurgeon *isa* surgeon.
- Supervision hierarchy, or the *supervises/is supervised* relationship. This hierarchy is based upon the organisational hierarchy. For example, the department manager role *supervises* the team leader role.

- Activity hierarchy, based on aggregation of activities. The hierarchy is based on the *more/less activities than* relationship and the roles are related to activities by the *responsible for* relationship. For example, *financial control* is responsible for *financial forecasting* and *financial accounts* which places the financial controller higher up the hierarchy than the financial forecaster role and the financial accountant role with respect to the *more activities than* relationship.

However, Moffet does not claim that these three hierarchies form an exhaustive classification. He allows that "there may be other useful role hierarchies". The multiplicity of useful models for defining role hierarchies within organisations creates serious problems for designers of RBAC systems, since if a hierarchy is improperly used, we may get undesirable inheritance of access rights which may break the security constraints on the system. For example, a separation of duty constraint will be broken if the two exclusive roles are derived from any single role higher in the hierarchy. On the other hand, if we limit role activation with overly cautious constraints, we may arrive at deadlocked situations where the security restrictions forbid tasks that are fundamental to the operation of the organisation. We outline some of the more sophisticated approaches to dealing with these problems in PBM systems in the following sections.

3.1.1 RBAC96 role relationships

In [12] Sandhu et al. propose a family of models for RBAC rather than a single model, due to the fact that RBAC requirements can range from the very simple to the very sophisticated. In the most complex case, roles are hierarchically organised and may have constraints applied to limit role activation. These constraints can express many typical requirements of RBAC systems such as separation of duties, prerequisite roles and cardinality of groups. These models, known as RBAC96, are in the process of being standardised by the US NIST [13]. However, this simple hierarchical role model suffers from various problems; in particular, if superiors automatically inherit the authorisations of their inferiors the principle of least privilege will be broken. To get around this problem, the idea of private roles and limited inheritance is introduced. This is achieved by creating a private role as a complement to each inherited role. The private role is used to group the non-inherited permissions together and it is not inherited. In the case where we want certain superiors in the hierarchy to inherit private permissions, but not others, we can create private sub-hierarchies. However, these devices can lead to excessively complicated role hierarchy structures that are difficult to engineer and even to comprehend. For these reasons, several researchers have started to question the utility of all role hierarchies in RBAC systems [14].

3.1.2 OASIS role relationships

The Open Architecture for Secure Interworking Services (OASIS), developed at Cambridge University [15] eschews the notion of role hierarchies, due to the problems mentioned above. They argue that: "one of the main reasons for using RBAC is that it provides a natural way to model constraints such as separation of duties [and] role hierarchies complicate the specification and enforcement of these constraints". They believe that the use of such hierarchies in RBAC "arises mainly through the influence of object oriented modelling" and "are not convinced of their utility" in this domain.

Instead of utilising role hierarchies OASIS uses role activation dependency as the fundamental role-role relationship. Individuals must meet a set of conditions in order to activate roles before being granted the permissions and obligations associated with the role. Role activation is governed by a set of rules, specified in logic, which are parameterised and dynamic. The conditions for role activation may include prerequisite roles that must have been activated by the user prior to activating the current role. Prerequisite roles offer much of the same ability to reuse policies that are used by role hierarchies. While this approach allows for both reuse of policies and enforcement of a wide range of constraints, it does increase the complexity of role engineering as, without careful analysis of role activation rules, some rules may be impossible to activate due to conflicting constraints on some of the prerequisite roles in the role activation conditions.

3.1.3 PONDER role relationships

In Ponder roles are "semantic groupings of policies with a common subject" [6]. Ponder's role based management (RBM) is an extension of RBAC since it incorporates obligation and refrain policies, enabling a more complete specification of a managed system than is possible with the positive and negative authorisations that make up RBAC. In Ponder, roles can be related through three separate relationships. The *type specialisation* relationship relates roles through the mechanism of inheritance, using the keyword *extends*. This relationship is directly based on object oriented programming and creates a role hierarchy which corresponds to the *isa* hierarchy identified by Moffet [11]. Inherited roles can add new policies and override policies with the same name, thus allowing for limited inheritance. Ponder also supports two other means of relating roles together, *relationships,* which group the policies defining the rights and duties of roles with respect to each other, and *management structures*, to model organisational structures.

3.2 Administration of RBAC

Most organisations are dynamic to a greater or lesser extent. Employees come and go, are promoted and demoted, managed objects are added and decommissioned, organisational units are created, restructured and eliminated. Furthermore, as systems evolve, problems with policy specification become clear and require redesign. Therefore, it is not sufficient to define a set of policies and roles a single time, then sit back and let the system work. We need a mechanism for updating the roles, policies and other elements that go to make up a policy driven management system. It is one of the widely acknowledged shortcomings of the NIST standard for RBAC is that there is no provision for the standardisation of administration of RBAC systems [16].

3.2.1 RBAC96 & ARBAC97

At the same time as putting forward the RBAC96 family of models, Sandhu et al proposed the idea of distinct administrative roles with administrative permissions. In a later paper, known as ARBAC97 (Administrative Role Based Access Control), this concept was fleshed out, defining a set of standard administrative actions, grouped into User-Role Assignment (URA97), Permission-Role Assignment

(PRA97) and Role-Role Assignment (RRA97). These administrative actions are carried out by users who belong to a separate administrative role hierarchy, which is generally envisaged to consist of the information security officers of the organisation. The ARBAC97 model provides a framework for enforcing administrative policies and delegating responsibility for certain areas of the administration of a RBAC system to roles lower down the administrative hierarchy.

3.2.2 Administration in OASIS.

Like ARBAC97, OASIS uses a separate administrative set of roles to administer the system [17]. These roles are constrained by meta-policies, which ease the task of policy administrators since they constrain policy design and allow administrators to adjust policies without breaking the fundamental requirements of the policy system. OASIS allows controlled self-management. Users can manage a subset of the resources in the system, including the access control policies for the resources that they manage, for example their personal files. However, this self-management is subservient to a system administrator who enjoys full control. The administrator is conceived of as indispensable for ensuring the overall consistency of the system. Users have no control over the actions of the administrator who can modify their access control policy and their rights within the system. Therefore, the concept of self-management and delegated administration is limited, since the authority of the specialised system administrator is absolute and there is no mechanism for constraining it.

3.2.3 Administration of PONDER

Ponder policy objects, roles and other composite policies are managed objects like any other. This enables the creation of policies whose objects are other policies, roles or relationships and provides the ability to specify, through policy, which roles are permitted to add, delete and edit policies. In addition, meta-policies allow for constraints on the final form of edited policies. Finally, obligation policies can be used to specify what actions must be performed on policy objects when certain events occur. These features combine to provide a set of tools for a potentially powerful, self-managed system. There is no need for specialised administration roles, since any normal role can be granted permission to update policies - in exactly the same way that roles can be granted permission to access non-policy resources. However, although the Ponder framework does provide the building blocks for enabling a self-managed organisation, it is only a framework. It provides the technical tools, like policy authoring toolkits and policy enforcement engines, from which we can construct a robust model of an organisation that has the ability to manage itself.

3.3 The practical problems of policy

Clearly, it is impossible to impose a unique model that can apply to all organisations. The traditional approach to this problem is that each organisation must be modelled in advance in a phase of requirements engineering. This phase includes provisions for roles that have responsibility for creating, deleting and editing roles and the policies that apply to them. However, this is an extremely difficult task, particularly when we are trying to achieve a balance between an accurate model of how an organisation works with the freedom for the model to develop and restructure itself according to changing business goals and environmental forces.

Initially, it was assumed that PBM was a simple means of realising efficiencies in the operation of organisations. That by grouping individuals into roles, resources into domains, and so on, we could prescribe simple rules and apply them to automate much of the management of a company's electronic resources. However, despite being a prominent research subject for most of the last decade, PBM has been slow to take root in commercial projects and, where it has, it has often been used in a simplified, limited way, more as a tool to create greater efficiency in traditional administration, than as a generalised approach. The one area where it has achieved a certain penetration into industry is in network management, particularly QoS contract enforcement products, where vendors have introduced simplified, limited implementations of policy driven management approaches. Yet, even here it has proved much more difficult to apply than was ever anticipated:

"Policy-based network management (PBNM) turned out to be difficult to put into practice. Early adopters have found that developing and deploying policies is not simple, cheap or quick. Instead, PBNM has been a time-intensive, complex, expensive process. Additionally, it has demanded that the enterprise organisation mutate to match the technology—rather than the technology meeting the enterprise's management needs." Michael Jude, March 2001 [18].

This quote is instructive in that it highlights the fact that those organisations that have attempted to use PBM, even in the relatively limited and controlled sphere of PBNM, have found that the organisational models that the technology has been based upon do not correspond to the way that the organisation works in practice. This fact is also reflected in the fact that Policy languages and role models have tended to become increasingly complex and multi-featured as experimental evidence has shown the need for ever greater flexibility to deal with real-world organisations and the multitude of exceptions which are part of their operation. The increase in complexity of the models has been necessary in order to adequately model real world organisations, but it has also increased the difficulty of the 'time-intensive, complex, expensive process' of developing PBM systems. In the next section we will seek answers to why this difficulty has come about.

4. Policy, roles and organisational theory

The concept of policy is well understood in its commonly used sense. Organisations adopt policies to define the decisions that subjects within the system should make in particular situations. Whenever a subject needs to make a choice from a set of possible actions, the subject refers to the policy rather than whatever internal decision making mechanisms the subject possesses. The accumulation of organisational policies allow organisations to collect institutional wisdom and experience and use it to guide future decisions without the necessity for every subject within the organisation to have a complete understanding of all of the factors that influence each decision.

The set of policies of an organisation, both implicit and explicit, is equivalent to the description of the organisation itself, its structure and its operation. In addition, we can observe that the ability to define policy equates to the ability to exercise managerial control over an organisation and that possession of the rights to change policy is in essence power within an organisation. The definition of policy is the management of the organisation, and the implementation of policy is its operation. Although there are clearly limits to what policy can describe, and many areas are difficult or impossible to precisely define, we can still attempt to describe as much of the system as possible in terms of policy. It is to be expected that, as our understanding of business processes increases, we will be able to describe an ever-increasing subset of the organisation's processes in terms of policy. There is also no conceptual problem in leaving specific areas undefined in terms of policy or in using other modelling approaches, like workflow analysis, where the complexity of the system, or our lack of knowledge of the important factors at play, makes the process un-amenable to PBM approaches. In these cases, we can still use policy to bound the results that we expect from the sub-system and integrate it into the overall PBM system.

To understand why there has been such difficulty in implementing real-world policy driven management, we need to look at how policy is imposed and implemented in real-world organisations and contrast this to the often simplistic approach of the RBAC models. Early approaches to RBAC conceived of it as a subset of security management. However, the development of general-purpose policy specification languages, like Ponder, has illustrated how PBM can approach a total description of a system's operation. With the combination of permissions and obligations, both positive and negative, we can conceivably describe, in minute and precise detail, exactly how an organisation should work and exactly how each subject should react to any given situation. Although policy languages only really relate to information resources, they could be conceivably used to describe and record the duties of each subject within the organisation even in situations where there is no interaction with information systems. In any case, most organisations are moving towards a situation where every resource has an electronic representation and it is increasingly feasible to manage access to all the resources, electronic and physical, through computer-based interfaces. For example, booking of meeting rooms, control of inventories and even interpersonal communication are increasingly performed through networked computerised systems. Again, we should emphasise that we are not claiming that these electronic systems will ever entirely replace physical interactions, and there are limits to their application [19]. However, it is the goal of PBM systems to discover these limits and to describe as much as possible within them.

4.1.1 Who defines policy?

With the appreciation of the power of policy and its ability to approach a holistic prescription of the operation of an organisation, it seems obvious that the definition of policy should encompass much more than an organisation's security specialists, or in the realm of PBNM, the network engineers. For example, there can be few organisations that would believe that access to the company accounts should be regulated purely by the Chief Security Officer. In most real world organisations, the rules and procedures for accessing the financial resources will involve the financial controllers, accountants and general managers, while the security officer will be merely responsible for implementing their decisions. Any role model should reflect this, or else it will be necessary for the organisation to *"mutate to match the technology"*. Thus, we can claim that the responsibility for the definition of a policy whose target is a particular resource, should reflect the real-world responsibility for that resource, rather than being hived off to a security officer, or some other specialist. In effect, this implies that the administrative structure of an RBAC system should be identical to, or as close as possible to, the decision making structure of that organisation.

However, the formal structure of an organisation does not necessarily correspond with the way that decisions are actually made. In bureaucratic organisations, the individuals who implement policy can be given the ability to take decisions when there are policy conflicts using their ability to refine high level policy into concrete policy in complex situations. In most traditional organisations, a key management goal is to get the best balance between decisions driven by concrete policy and decisions taken by individuals. If there is too much emphasis on concrete policy, situations of deadlock can ensue when unanticipated factors create a situation in which no subjects have the authority to take the required actions. If there is too much emphasis on individual decision making, contradictory decisions could ensue leading to a chaotic situation. Thus, organisations strive to come up with a balance of power between autonomy and authority.

Furthermore, as well as the capacity for autonomous policy conflict resolution and refinement, real-world organisations consist of informal decision making structures in addition to the formal positions. In many organisations, key decision makers may, in practice, be personal assistants, secretaries or system administrators. Due to their access to information and their monopoly of crucial skills, these individuals, despite holding relatively lowly positions with respect to the organisation's formal hierarchy, may exercise an enormous amount of control over what choices are made in the behaviour of the system. We can say that, in practice, they define organisational policy.

4.2 Organisational theory: Taylor to HRM

To relate the above discussion to organisational theory, we can say that organisations need to achieve a balance between classic Taylorism and newer human resources approaches to management. Taylor pioneered "scientific management", primarily as a means for management to gain control over the production process. Taylor's top-down approach to organisation and management was geared towards the fast growth of industrial corporations in the early years of the 20th century, particularly within the factories and large workplaces that were the hallmark of the industrial revolution. Taylor's system was aimed towards taking traditional knowledge and "classifying, tabulating and reducing this knowledge to rules, laws and formulae". The aim being that "all possible brain work should be removed from the shop floor and centred in the planning and lay-out department" [20]. Taylor's systematic approach to the production process was developed by theorists such as Weber and Fayol, who introduced motivational elements which were notably absent from Taylor's simple conception of 'economic man', to form the theoretical bedrock of the modern bureaucratic organisation. Richard Edward's research on companies such as Polaroid, IBM and General Electric [21] reveals just how much of these theories had been absorbed into the new bureaucratic corporations.

Although Taylorism was a totalitarian system, intended to be implemented in its entirety, in practice "it never quite worked like this" [22]. There is a negative side to the rigid enforcement of the "standardisation of work processes, outputs and skills" through detailed rules prescribing company policy. "Standardisation and predictability could easily degenerate into rigidity and defensive behaviour…resistant to innovation" [23]. Blau's [24] studies of a state employment agency and a federal law enforcement agency in the US show how the 'work to rule' is in fact an extremely effective form of industrial action, rather than a tool for more efficient production!

Organisational theory continued to develop from the apparent failure of Taylorism. The new approach, which can be loosely classified as human relations (HR) or its modern equivalent Human Resources Management (HRM), quickly came to the realisation that "it is as well to recognise that informal organisation is not 'bad' as it is sometimes assumed to be" [25]. The school of HR, arising out of the famous Hawthorne researchers of the 1930's, is often seen as unrelated to the organisational structure and even in some cases as a "gigantic, if dangerous, con-trick" [26], intended to provide the appearance of involvement in decision making structures, more than the reality. However, HR did at least acknowledge that micro-control was not necessarily the most effective approach and that organisational models had to take account of some form of autonomy. HR also crucially first focused on the workgroup, later to become the 'team', as an important unit for organisational analysis. Although HR approaches went out of fashion for several decades, "the idea that the internal dynamics of the small group could be turned around so that a degree of self-governance could favour management resurfaced in the substantial wave of interest in teamworking from the 1980's onwards" [27].

Generally, the evolution of organisational theory in the 20th century was the story of an attempt to reconcile the seemingly contradictory demands of work organisations. Management needs to control the operations, since there is a contradiction between the needs and goals of the organisation as a whole and the motivations of the individuals and groups that make up the organisation. On the other hand, there is a need to involve the individual subjects in the overall management in order to gain their goodwill and take advantage of the inherent human propensity for innovation. This dialectic has given rise to many insights, management fads and behaviourist theories. For example, contingency theory was the dominant approach of the 1970's. It held that different levels of autonomy and direct control depended on the particular environmental factors within which an organisation found itself and that different approaches should be applied in different industries, divisions and market sectors. So, for example, the research of Lawrence and Lorsch (1967) [28] into plastics manufacturing companies revealed that whereas the research departments of plastics companies operated in dynamic, innovative environments that were reflected in non-hierarchical structures with minimum bureaucracy, the production division had a stable, technical environment and was dominated by short-term concerns and more bureaucracy. They found that high-performing firms were those that manifested a high degree of differentiation of structures and goals.

Overall, we can say that the development of organisational theory has been dominated by the conflicting forces of Taylorist, scientific management and notions of autonomy and self-management, generally embodied in teams. These forms of structure and organisation exist in parallel in all organisations, depending on the particular environment and history. In general, we can see all management innovations and fads in terms of this conflict. Modern production methodologies such as Total Quality Management (TQM) and Just In Time (JIT) are modern Taylorist trends, in that they aim to precisely control the production and labour processes. However, they also include elements of self-managing, self-supervising, team-working paradigms. The trend towards flat hierarchies, team-working and autonomous groups is most obviously seen in many 'new economy' companies.

5. Limitations of current role based models

Current role-based models in PBM systems can be seen to adopt an extreme Taylorist position. The organisation is specified, in advance, through an extremely complex requirements engineering phase. Adjustments to this specification are made during operation by a specialised class of security officers. This is analogous to Taylor's maxim that "all possible brain work should be removed from the shop floor and centred in the planning and lay-out department". However, although a Taylorist approach can work within traditional bureaucratic organisations, it is much more problematic when the agents making the policy enforcement decisions are automated rather than human. Automated software agents take decisions that are, by their nature, deterministically applied and based upon precise, non-ambiguous rule definitions. Even in the most hierarchically organised bureaucratic organisation, the (human) policy enforcement agents have the capacity to take autonomous decisions and are influenced by informal power structures as well as the formal role hierarchy. If computer models of organisations are to be accurate, they need to reflect the way in which organisations actually work, rather than their formal structure. Furthermore, models such as those that have been discussed above, which posit a tremendous amount of power in the hands of a specialist, such as a RBAC security administrator, is unlikely to prove attractive to current management. They will inevitably see such a position as undermining their authority. If management resists the introduction of PBM systems they are unlikely to proliferate.

5.1 Requirements engineering and RBAC.

Requirements engineering for RBAC is a field that is in its still in its infancy. Current methodologies from fields such as software engineering are inadequate for dealing with the complexities of organisational structures. There have been some attempts to derive an analytic role modelling framework on the part of RBAC researchers in the last few years, but there is no evidence of their utility in practice. Current approaches in [29] do go some way towards recognising the complexity of real world organisations, in modelling roles under a number of different criteria, based on seniority, qualification, function, work process, market and various environmental factors. However, despite the complexity of the engineering process, they concentrate heavily on organisational structure to define and are limited by their narrow view of organisational theory. Crook et al. use the starting point that "the organisational structure of an enterprise is designed by the top management of that enterprise and defines the lines of authority and the division of work" [30]. This is rooted in their concentration on organisational behaviourism (OB), rather than the broader field of organisational studies. OB takes a prescriptive approach to management. It does not ask how the organisation works, rather it asks how it can work better from its current state. OB and

management writing in general "serve a variety of functions, virtually all of which have a strong component of ideology and values" [31]. On the contrary, organisational studies, which stems from sociology rather than management science, recognises that "formalised power (the kind implied by organisational charts, for example) may well differ markedly from actual power relations" and that "the exercise of power by organisational members may not be sanctioned by their position in the organisational structure" [32]. This has been well documented in empirical studies by Pfeffer and others [33]. Although this can often be safely ignored by organisational behaviourists, since their theories can be applied alongside the tacit acceptance of informal structures of authority, we do not have this luxury when designing computer based PBM systems. While we can have a word in the ear of the firewall administrator, we can't have a word in the ear of the firewall!

5.2 Modelling teams and other groupings

In addition to a simplistic, Taylorist approach to organisational structures, current role based models contain significant oversights in terms of what they are capable of modelling. "It has long been widely recognised that many of the decisions that affect the policies of organisations are made in groups. Even in hierarchical organisations with powerful control from the centre, there has to be a flow of information to the centre from formal and informal 'assemblies' of individuals working within formal and informal groupings" [34]. This acknowledgement of the central position of groupings to the functioning of organisations is not reflected by the almost exclusive concentration on roles and role theory by most PBM systems. Roles are generally modelled in isolation, without giving them context within particular groups, which may have their own distinct policies. If we are to build models that accurately describe an organisation, we need to recognise that roles cannot be understood in isolation but have a strong dependence on their particular context within a division, department, project group or team.

Although there has been some discussion of this problem in building PBM systems, it is tenuous and separated from the mainstream of role based models. Ponder does provide a 'domain' construct for grouping resources and a 'management structure' construct which allow roles, policies and relationships to be grouped together into functional units, however these constructs are very much secondary to the primary focus, which is on defining individual roles. The requirements engineering literature does also make some mention of organisational groupings, however these are seen mainly as a way of conveniently grouping roles together rather than as the basis for the organisational analysis. There have also been some research efforts which have specifically focused on Team based access control methods. The TMAC model was formulated by Thomas [35] and more recently there have been attempts to integrate it with RBAC models to form C-TMAC. Although this research does contain many interesting insights, the fundamental basis of permission still remains the role, as the "effective permissions of a user are always derived from permission types defined for roles that the user belongs to" [36]. This limits the autonomy of the group, and although it may be a correct limitation in most cases, it amounts to an unnecessary restriction on the theoretical model. We believe that, rather than being a syntactical convenience in the construction of organisational models, groups should form the basic unit of analysis and of the resulting model.

5.3 Decision making mechanisms

Another area that has been almost totally overlooked by current role based models is decision making and conflict resolution, not in terms of conflicting policies, but in terms of conflicting agendas by groups and individuals within the organisation. We believe that any model of an organisation needs to be integrated with decision making mechanisms which reflect the way that decisions are made in the real-world organisation.

5.4 Modelling collaborative communities

The proliferation of the Internet and the resultant connectivity of enormous numbers of people has given rise to new organisational paradigms. In particular the 'collaborative community' has become an extremely common type of organisation, existing in many guises. Although there is no single working definition of what constitutes a collaborative community, they are recognisable by contrasting them to the traditional bureaucratic organisation: a tendency to favour informal, egalitarian structures over hierarchical bureaucracy; voluntary participation in pursuit of collective goals; membership that is geographically dispersed.; fluidity in terms of membership, structure and even goals; and heterogeneity of structure.

While current RBAC role models and methodologies face real problems in modelling the working structure of bureaucratic organisations, the problems are even more severe in the context of collaborative communities. Many collaborative communities have no formal structure whatsoever. Even in cases where there are formal positions, as in a virtual enterprise, the roles with respect to the collaboration will not necessarily have any relation to the formal positions. This naturally creates a very difficult situation for the requirements engineering phase, which is central to most current role based modelling methodologies. In addition, the fluidity and heterogeneity mean that any engineered solution will be unlikely to remain stable for any length of time before re-engineering is required. Nevertheless, roles and policies can be very useful concepts for creating order in this potentially chaotic environment.

5.5 Potential of PBM systems

Despite the various problems mentioned above, we think that PBM remains a promising approach to the problem of automation of the routine elements of an organisation's operations. However, we believe that, if they are to be used in practice, PBM systems need an accurate internal model of the organisation. This model must have an inherent capacity for flexibility and heterogeneity. Different parts of the organisation must be able to adopt different structures, different decision-making mechanisms and be granted variable degrees of autonomy for the management of the resources which they are responsible for. The model must have the innate capacity for evolution, reorganisation and redistribution of resources.

6. Community based model

In this section we will outline, in general terms, our proposed solution to the problem of modelling organisations for PBM systems. We base our model on the community, rather than the role. We feel that, at the most general level, any organisation can be best modelled as a community with a particular goal (or set of goals). In our model, communities, rather than individuals, take actions whose targets are particular resources. The set of actions that a particular community may take on a particular resource is defined by the policy rules whose subject is the community and whose target (or object) is the resource. Therefore, in our model, communities, rather than roles, are the subject of policy rules. The term community is chosen, as this concept of a community with a particular goal is similar to the community, defined by a contract, as described in the RM-ODP Enterprise Language. However, our model of the community has a number of particular features which render it quite different to all existing models.

6.1 Community resource management

In addition to the community, our organisational model includes a set of resources that it the community is responsible for. Being *responsible for* a resource, means that the community can define policy rules whose target is the particular resource. The ability to define policy is equivalent to being authorised to take the action of modifying this set of policies. So for example, if the community is responsible for a database, the community can create policies which define who is authorised to access the various tuples of data in the database, by defining authorisation policies whose object is the database or a subset of it. The community can also define obligation policies on the database, which can specify actions that will be taken when specific events occur. So far, our model is relatively straightforward and uncontroversial. It is obvious that any organisation which manages resources should be able to define rules for the utilisation of the resources and define how the resources should be affected by events.

6.2 Community decision making

However, this model remains at an extremely abstract level. What, in practice, do we mean when we say that the community can define policies? The community is not a single entity and we require a means of deciding if a community wishes to define a policy. This is why the community needs decision making mechanisms. Decision making mechanisms define the process through which a community reaches a decision. Communities may have multiple decision making mechanisms, corresponding to actions which are deemed more or less important by the community. So, for example, a community may decide that some actions may be taken on behalf of the community by any member of the community, while other actions may require agreement by several members, or even a majority vote, before they can be taken on behalf of the community.

6.3 Mandated Communities

If our model remained on this general level, it might be interesting theoretically but would have minimal practical use in any organisation of significant size or complexity. The organisation would either have to relinquish control to the whims of members or become bogged down with an enormous number of decisions concerning the various resources that it is responsible for. Therefore, we introduce to our model the concept of the mandated community.

The *mandated community* is created as a subset of an existing community. Any community can create a mandated community by defining a community qualification policy. The creating community becomes the *parent community* of the mandated community. This policy rule is similar to the role activation constraints of constrained RBAC [13] or OASIS [15], in that it allows us to specify the conditions that must be met before an individual is admitted into the community. So, for example, we could create a mandated community called ‘engineers’ and define a qualification rule in the form of a policy stating: ‘must be employed by the company in an engineering role’. The community is described as *mandated* due to the fact that the parent community may create policy rules whose subject is the mandated community. That is to say that the parent community may define what the mandated community is authorised to do (by defining authorisation policies) and what it is obliged to do (by defining obligation policies). The set of policies that have been defined by the parent community and whose subject is the mandated community, is the *mandate* of the mandated community. There are no restrictions on the content of mandates, allowing mandated communities to have a variable degree of self-management. In certain cases, the parent community will wish to precisely describe the operation of the mandated community, through a detailed and fine-grained definition of its mandate. In other cases, the parent community may specify more general policies in the mandate and allow the community a degree of self management in terms of refining these policies into more specific policies. For example a community’s mandate may consist of no more than an abstract policy, consisting of a SLA or natural language statement defining the community’s “mission statement” with a set of authorisation polices, authorising the community to define policy on the resources that the community is responsible for.

The ability to create mandated communities gives our model an extra degree of flexibility, since we can now define policies whose subject is the mandated communities. This means that, if we have the right to define policy for a particular target resource, we can grant authorisations to, or impose obligations on, specific groupings in the organisation. Thus, our community structure has many of the features of a role, since it is used as the basis by which access control rules are made, but its simultaneously the basis for the distribution of policy definition rights.

Mandated communities operate in exactly the same way that their parents do. They have decision making mechanisms, may create their own mandated communities and may be responsible for particular resources. Granting a mandated community responsibility for a resource is simply achieved by creating an authorisation policy which states that the mandated community may define policies whose target is that resource. In our model the creation of a policy of this specific type is known as *delegating responsibility for* the resource. In fact, our model allows delegation of much finer granularity than granting responsibility over an entire resource. The parent community may delegate responsibility for a specific aspect of the resource, by means of specifying meta-policies which constrain the subject, or

target of the policy rule. Meta-polices allow us to describe the *sphere of responsibility* of mandated communities. For example, our organisation may create a mandated community called 'database administrators' and delegate responsibility for the organisations' database servers to this community. We may decide to define a meta-policy, constraining the subject of policy rules that grant root access to these servers to members of the mandated community. Thus, we forbid the database administrators from granting root access to anybody who is not a database administrator.

6.4 Hierarchical policy enforcement

In addition to specifying meta-policies on delegated resources, the parent community can itself continue to define policies on the target resource. This creates potential policy conflicts on particular resources due to conflicting policies being defined by parent communities and their mandated communities (which may occur several steps of delegation away). To get around this problem, policy is hierarchically enforced. Thus, policies defined at the more general levels of the organisation have precedence over the policies imposed in more specialised communities. This is an important feature since it allows us to define organisation-wide policies at the most general levels, policies that will continue to be enforced irrespective of any policies that are applied in mandated communities. For example we can impose organisation wide security guarantees at the most general level without having to depend upon each specific community to enforce them on the resources that they are responsible for.

The hierarchical enforcement of policy depends upon the policy enforcement system having a certain amount of semantic knowledge about the target resource. In particular the policy deployment and enforcement mechanisms must be able to identify when the targets of two policy rules overlap. The complexity of the required semantic knowledge is one of the unknowns in our model but early experiments suggest that it should be a relatively straightforward task with regards to most typical information resources.

6.5 Policy conflicts

Another obvious problem of our approach to delegated responsibility is the occurrence of policy conflicts where the communities defining the policies have no direct hierarchical relationship. In this case the conflict cannot be automatically resolved, as there is no means of knowing which policy should have precedence. However, we can go some way towards automating the solution. The policy management system can identify the community within which the conflict must be resolved by choosing the closest common parent of the conflicting communities. In general, a conflict of this nature will point towards a problem in the delegation of responsibilities, either an improperly specified meta-policy, an ill-conceived community structure, or an inappropriate delegation. A conflict event will be generated in this parent community and we may be able to generate suggested policy rewrites which can be adopted to overcome the conflict.

This treatment of policy conflicts, as a useful aid in policy and organisational debugging, is not confined to situations where the conflict is not automatically resolvable. When we encounter a policy that is wholly or partially nullified by a policy defined at a higher level in the community hierarchy, we also generate a conflict event. These conflict events can also be useful in terms of identifying areas where the community structure or resource policy set may need to be modified in order to allow the mandated communities to carry out their mandate. A parent community may decide to relax some policies in order to allow the mandated community more autonomy in fulfilling its mandate, or on the contrary may strengthen certain policies to specify more exactly the mandate.

6.6 Modelling hierarchical organisations

Another obvious problem with our model so far is that, although it might reflect the way in which collaborative communities work, with responsibility delegated from the overall collective to groups with specific areas of responsibility, most organisations are more hierarchical. Authority in many organisations resides in the upper echelons of the organisational structure rather than in the mass of members. In fact, this apparent structural difference is relatively minor. In hierarchical organisations certain groups take decisions *on behalf of* organisational groupings. So, the board of directors takes decisions on behalf of the overall company, while the head of marketing takes decisions on behalf of the marketing department. In our model, we allow for this type of hierarchical decision making by allowing a community's decision making mechanisms to be delegated to a mandated community. We say that the mandated community takes decisions *on behalf of* the parent community. So, for example, a university community could create a mandated community called 'university council', and then delegate decision making to this community, which would then act on behalf of the overall university.

6.7 Modelling change

One of the great strengths of this model is that it does not require a detailed process of requirements engineering to create a model of the organisation. We can create the most basic structure and allow the detailed divisions of responsibility and the organisational groupings to evolve in an organic manner. Thus, for example, we can introduce a PBM system by merely modelling the entire organisation as a single community which is responsible for the full set of resources to be managed by the system. As the needs arise, we can create mandated communities and delegate to them responsibility for specific resources. The structure of the organisation can remain in constant flux, as we strive to more fully understand the key rules in its operation. The analysis of policy conflicts can be used to signal structural problems in our model and in the underlying real-world organisation, thus giving us constant feedback and impetus to refine our model. In the case of a merger of two PBM organisations, we again have an organic path towards integration. We create a parent community, comprising the membership of both organisations. This parent community creates two mandated communities, which are equal to the original communities, and delegates responsibility over all of their respective resources to them. The two organisations will continue operating exactly as before. From this starting point, we can attempt to merge the organisations by introducing new policies at the top level, and creating new mandated communities whose membership is drawn from across the organisational divide, eventually leading to a single, integrated system, achieved in an organic way, at minimal expense in terms of requirements engineering.

6.8 Conclusion

We can at this stage claim that our model fulfils the requirements that we identified in the previous section. The organisation is modelled in terms of groupings rather than roles. The definition of mandates allows us to grant various degrees of self-managed groupings. Responsibility for resources can be moved around the organisation through delegation. The structure of the organisation can evolve rapidly as communities create mandated communities and delegate responsibility for resources to them. As any community may have many mandated communities, with overlapping membership, we can build a model of an organisational structure with multiple hierarchical relationships existing side by side. Conflict events are utilised to allow us to engage in a constant process of organisational debugging and policy fine-tuning, so that we can constantly strive towards more exact specifications of the organisation's functions and automate ever more of the routine operations. We believe that this approach has the ability to create adequate models of real-world organisations for the purpose of realising the potential of PBM.

7. Future directions

We are currently engaged in the construction of a PBM system that operates according to the model described in the previous section. Our prototype will be built on top of the Ponder policy management framework. Although our community based PBM system has significant differences to the role based model assumed by Ponder, we feel that, in most areas, our system will require minimal changes to the Ponder framework. A role in Ponder corresponds to a community decision in our system. Thus, for each decision making mechanism adopted by a community, there will be a distinct Ponder role. These roles will be activated once a decision has been reached and will persist only until the decision has been carried out. These temporary roles will be implemented by means of single-use certificates issued by the decision making mechanism.

The one area of the Ponder framework that will require substantial modification relates to policy enforcement. In Ponder, policy is enforced by means of a deployment phase which maps Ponder policies to the underlying security mechanisms, like firewall configuration files. In order for us to impose policy in a hierarchical manner, we will need to create deployment software which will ensure that policies which have been defined higher up the community hierarchy can not be negated by ones defined beneath them. We also need to construct this software to enable it to identify conflicts in the policies imposed on resources to generate appropriate events.

Finally, we are engaged in constructing conflict analysis tools to enable us to analyse policy conflicts and automatically generate alternative organisational structures and policies to overcome the source of conflict and to refine the existing organisational structure itself. We feel that the potential for such an empirical tool to analyse the structure of organisations is one of the most exciting prospects of our research.

8. ACKNOWLEDGMENTS

Many thanks to Dave Lewis, Department of Computer Science, TCD, Aileen O Carroll, Department of Sociology, TCD, for useful discussions and advice on the content of this paper. Thanks to Tony O Donnell and Declan O Sullivan, both KDEG, TCD for reviews. This research was conducted with funding from the Irish Higher Education Authority under the M-Zones Programme.

9. REFERENCES

[1] Lymberopoulos, L., E. Lupu, E., Sloman, M., An Adaptive Policy Based Management Framework for Differentiated Services Networks, 3rd IEEE Workshop on Policies for Distributed Systems and Networks (Policy 2002), Monterey, California, June 2002,147-158

[2] Sandhu, R., Coyne, E., Feinstein, H., Charles Youman, C. Role Based Access Control Models., IEEE Computer, Volume 29, Number 2 / February 1996

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

[4] Sloman, M., Lupu, E., Security and Management Policy Specification, IEEE Network, vol.16 No. 2, March/April 2002. 10-19

[5] Jajodia, S., Samarati P, Subrahmanian.V. A Logical Language for Expressing Authorisations. Proc. IEEE Symposium on Security and Privacy, May 4-7, 1997. 31-42

[6] Damianou, N., A Policy Framework for Management of Distributed Systems, PhD Thesis, Imperial College, University of London, 2001

[7] Ribeiro, C., Zuquete A., Ferreira. P., SPL: An access control language for security policies with complex constraints. In Proceedings of the Network and Distributed System Security Symposium, San Diego, California, February 2001.

[8] Westerinen A. et al., Terminology for Policy Based Management, IETF RFC3198, November 2001

[9] Thomas, E. and Biddle B., The Nature and History of Role Theory, in Role Theory: Concepts and Research, Krieger Publishing, 1979

[10] Ting, T.C., A User-Role Based Data Security Approach, in Database Security: Status and Prospects, (C.E. Landwehr, Ed). Elsevier, 1988.

[11] Moffett, J. D., Control Principles and Role Hierarchies. Proceedings of the Third ACM/NIST Role Based Access Control Workshop, Fairfax, Virginia, USA, ACM Press, October 1998. 22-23

[12] Sandhu, R. S., Coyne, E. J., Feinstein H. L., Youman C. E. (1996). *Role-Based Access Control Model*s. IEEE Computer, vol. 29(2), 38-47.

[13] Sandhu, R., Ferraiolo D., Kuhn R., The NIST Model for Role-Based Access Control: Towards A Unified Standard. Proceedings of the 5th ACM Workshop on Role-Based Access Control, Berlin, Germany, July 2000. 47-61, 26-28

[14] Goh, C., Baldwin, A., Towards a more complete model of role. Proceedings of the third ACM workshop on Role-based access control, 1998. 55 - 62

[15] Hine, J., W. Yao, J. Bacon and K. Moody. An Architecture for Distributed OASIS Services. Proceedings of Middleware 2000, New York, USA, Lecture Notes in Computer Science, Springer-Verlag, 4-8 April 2000. 107-123

[16] Trent Jaeger, Jonathon E. Tidswell Rebuttal to the NIST RBAC model proposal, Proceedings of the fifth ACM workshop on Role-based access control, 2000. 65 - 66

[17] Belokosztoloszki, A., Moody, K. Meta-Policies for Distributed Role-Based Access Control Systems. Proceedings of the IEEE 3rd International Workshop on Policies for Distributed Systems and Networks, Monterey, California, USA, 2002. 106-115

[18] Jude, M., Policy-based Management: Beyond The Hype. In Business Communications Review, available from http://www.bcr.com/bcrmag/2001/03/p52.asp, March 2001. 52-56

[19] Grudin, J. Group Dynamics and Ubiquitous Computing. Communications of the ACM, December 2002

[20] Braverman, H. Labour and Monopoly Capital: The Degradation of Work in the 20th century, Monthly Review Press, New York, 1974.

[21] Edwards R. Contested Terrain: the Transformation of the Workplace in the 20th century, Heinmann, London 1979.

[22] Thompson P. and McHugh, D., Work Organisations, Palgrave, New York, 2002. 32

[23] Ibid. 39

[24] Blau P. The Dynamics of Bureaucracy, University of Chicago Press, Chicago, 1955

[25] Roethlisberger, F. and Dickson, W. Management and the Worker, Wiley, New York 1964.

[26] Mayo, E. Human problems of an Industrial Civilization, Macmillan, New York. 1946

[27] Thompson P. and McHugh, D., Work Organisations, Palgrave, New York, 2002. 58

[28] Lawrence, P. and Lorsch, J. Organisation and Environment, Harvard University Press, Cambridge, Mass, 1967

[29] Crook, R. Ince, D., Nuseibeh, B., Towards an Analytical Role Modelling Framework for Security Requirements

[30] Crook, R., Ince, D., Lin, L., Nuseibeh, B. Security Requirements Engineering: When Anti-requirements Hit the Fan. Security Requirements Group Department of Computing, The Open University, Milton Keynes. 2001

[31] Pfeffer, J. Power in Organisations. Pitman, Marshfield MA. 1981

[32] Kearins, K. Power in Organisational Analysis: Delineating and Contrasting a Foucauldian Perspective. Electronic Journal of Radical Organisation Theory, Vol. 2, no 2. Available from http://www.mngt.waikato.ac.nz/ejrot/, 1997

[33] Pfeffer J. Managing with Power: Politics and Influence in Organisations, Harvard Business School Press, Boston, 1992.

[34] Gear T., Minkes, L., from Call for Papers, Co-opereative Working, Group Decision Making and Philosophy of Management. The Journal of Philosophy of Management, see http://www.managementphilosophers.com/Getting%20Published.htm. 2003

[35] Thomas, R. K., Team-Based Access Control (TMAC): A Primitive for ApplyingRole-Based Access Controls in Collaborative Environments, Proceedings of the 2nd ACM Workshop on Role Based Access Control, Fairfax, VA, 1997

[36] Georgiadis, C., Mavridis, I., Panglos, G., Thomas, R. Flexible Team-Based Access Control Using Contexts. Proceedings of SACMAT, Chantilly, VA, 2001