

# Policies as Signals in Collaborative Policy Engineering

Kevin Feeney, Christos Tsarouchis, David Lewis

**Abstract—** Policy Engineering is the process of authoring policies, detecting and resolving policy conflicts and revising existing policies to accommodate changing resources, business goals and business processes. In operations of any scale, policy engineering is a collaborative task undertaken jointly by multiple stakeholders with different concerns across an organization. Moreover, the operation of an organization is increasingly likely to be part of a broader value-chain involving other organizations. Therefore operational policies in one organization must be engineered with some understanding of the interaction of the behavior those policies engender and the desired behavior across the value chain. Furthermore, operations of many organizations are increasingly subject to external regulation which may be enacted as business policies with a necessarily transparent and traceable impact on operational behavior. It is therefore clear that any form of policy engineering not only involves multiple groups and individuals within a single organization, but also groups in other organizations. This view of policy engineering as a collaborative activity results in complex interdependencies that need to be analyzed and understood whenever making changes to an organization's policy set. In this paper we propose representing policies as signals between authors as a means to performing this analysis and thereby enabling the required understanding and applying it in a practical way to policy engineering tasks. The approach is based on semiotic theory. We apply this to an existing policy-based application managing dynamic spectrum access, a rich example of multi-stakeholder policy engineering. We outline how a semiotic analysis of policy authoring can assist in designing the semantics and structuring of policies in an organization, and also how it can inform the design of PBM application interfaces. This is performed within a business modeling framework known as the viable system model, which captures recursive layers of management and thus assists in addressing meta-management policy analysis.

This material is based upon work partially supported by Science Foundation Ireland under Grant No. 03/CE3/I405 as part of the Centre for Telecommunications Value-Chain Research (CTVR) at Trinity College Dublin (TCD), Ireland and by the Irish Higher Education Authority under the PRTL3 programme M-Zones.

D. Lewis is with the Knowledge and Data Engineering Group, School of Computer Science and Statistics and is participating in the CTVR. (corresponding author phone +353 1 896 2158, fax +353 1 6772204, dave.lewis@cs.tcd.ie)

K. Feeney is with the Knowledge and Data Engineering Group (KDEG), School of Computer Science and Statistics at TCD and is participating in the CTVR (kevin.feeney@cs.tcd.ie)

C. Tsarouchis is with the Knowledge and Data Engineering Group, Department of Computer Science and Statistics and is participating in the CTVR

## I. INTRODUCTION

The field of policy based management is relatively young and thus there has been little in the way of specific research into human interaction with such systems. In the words of one researcher in the field: “little is known about policy-based interaction among people and between people and systems” [maglio]. However, there are a few notable exceptions. Dijker [dijker] and Barrett [barrett, campbell] have produced studies of how humans traditionally interact with policy systems for the purpose of identifying requirements for policy based management systems. Their research identified some major requirements of such systems.

Several of these requirements closely reflect the features of policy based management that are generally accepted: policies should be specified in a human-friendly vocabulary; specifications of policy rules should separate goals from the methods used to attain these goals; policies should embody the accumulated expertise of the organization in their area of application and should make that expertise available. In addition to these basic features, they identified several requirements which are not commonly viewed as being an essential part of policy based management systems but are important in the context of deploying policy based systems within human organizations.

- The formation of policy should include input from the stakeholders in policy decisions as well as domain experts and the author of the policy. As Djiker notes, “there are too many potential consequences of establishing a policy to leave its development to a single individual” [dijker]. This means that policy based management systems need to be able to identify who these stakeholders are in any decision and provide a means for them to have input into the policy formation.
- It should be possible to specify policy rules with different levels of generality and specificity. Thus, it should be possible to create general rules which will serve as behavioral boundaries of the managed system. Within these boundaries, it should be possible to apply more specific rules to govern specific actions, operations or events.
- Policy rule sets should be adaptable over time to add more specific details to rules. Barrett refers to this as progressive grounding. This reflects the fact that work organizations are complex entities and many of the policies are implicit and form part of the employees

‘common sense’. So, rather than assuming that it is possible to completely translate an organization’s operations into a set of policies at one time, it is necessary to allow policy rules to be built up over time, by adding exceptions and more specific rules as problems with existing policy sets become apparent.

Another piece of work to note is an analysis of the interaction between humans and policy systems, described in [maglio]. This study conducted ethnographic studies, focusing on decision making in IT organizations. It concluded that human interactions with policy systems tend to follow a pattern of negotiation and that this negotiation was particularly valuable in trying to debug or fix problems such as policy conflicts. It further concluded that policy systems need to be flexible, lightweight, and support reversible and incremental updates.

The problem we begin to address in this paper is how may we begin to evaluate the effectiveness of a particular policy-based management system by viewing it as a tool for what is essentially a collaborative engineering task. Such evaluation requires the integrated analysis of both the usability of the tool itself and its effectiveness as a tool for communicating policy engineering decisions between individuals involved in the collaborative policy engineering process. This process is an on-going and conducted over long period of time, often involving people with only intermittent opportunity for direct social contact. Also, the drivers that cause individuals to engage in the process may be different, for different people at different time. Thus we must support a collaborative engineering process with possibly sporadic involvement by people separated in distance and time. We are therefore interested in the policy engineering tool increasingly providing the means by which different policy engineers exchange knowledge with the aim of converging on a stable policy set over time.

In the next section we introduce the problem domain in which we are addressing collaborative policy engineering. This is Dynamic Spectrum Access, which is interesting as a policy engineering problem because of the wide range of actors involved in setting policy. The policy system we are applying to this domain, and which its therefore the focus of our analysis is the Community-Based Policy Management system [lewis]. This is specifically designed to support collaborative policy engineering, and it thus provides a sound basis for studying the interactions involved in this process. In section III we introduce elements that we are integrating for the analysis of such interactions. In section IV we introduce a new user interface that has been developed for the policy system in question and discuss how the analytical framework presented in sections V can be applied refining its design.

## II. POLICY ENGINEERING FOR DYNAMIC SPECTRUM ACCESS

Dynamic Spectrum Access (DSA) is currently attracting much interest internationally as a future alternative to the current command and control approach to allocating usage right for the electro-magnetic spectrum, and in particular the portions of it useful for Radio-Frequency (RF) communications. Regulators in North America and Europe are actively

considering DSA, driven in part by opportunities for spectrum usage reallocation made possible by the release of bands from analogue TV usage. The two main technological drivers are the promise of Software Defined Radio (SDR) and of Cognitive Radio (CR). SDR has the potential to break the traditional linkage between the characteristics of an RF transceiver and the spectrum regulation under which it is intended to operate. This may lead to much more general purpose RF transceivers that are able to reconfigure themselves to operate in different bands, under different regulatory regimes and using different encodings. The production of more general-purpose RF transceivers in high volumes has the potential to dramatically reduce the long lead time and high level of investment needed to roll-out a new RF service. Thus regulators will no longer be under the same pressure from RF technology vendors and service operators to maintain the status quo in the allocation of spectrum bands to specific applications so that they can recover the investment made in developing and deploying the new services. CR, with its ability to monitor spectrum usage and intelligently plan transmissions, allows unused areas of the spectrum to be used opportunistically, but without impacting the needs of occasional priority users, e.g. emergency services, from using those bands when needed. This therefore reduces the impact of claims by these classes of users for exclusive access to certain bands. DSA still involves a large degree of uncertainty, and its management will need to exhibit agility at many level of organization from national and international regulators, secondary markets and RF commons communities.

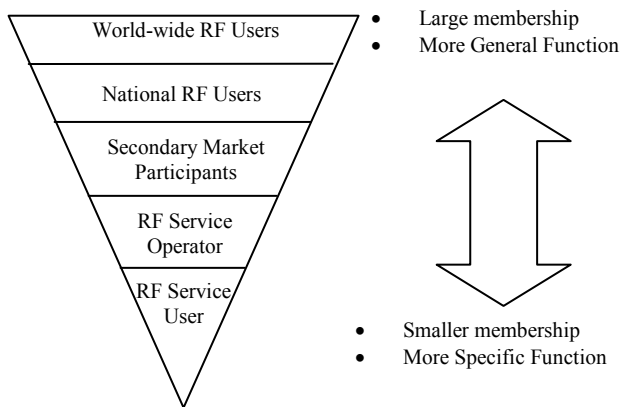
Policy-based management is widely seen as a suitable mechanism for supporting such agility in DSA management. The U.S DARPA Next Generation Communication (XG) initiative has proposed a management framework for executable policies for cognitive radio by the U.S DARPA Next Generation Communication (XG) initiative [xg-vision]. This proposal includes the specification of a specific, executable policy language, the DAPRA XG policy language (XGPL). In [feeney] we propose a specific extension to XGPL based on abstractions from a scheme called Community Based Policy Management (CBPM). The CBPM scheme, as detailed in [lewis], uses a tree-like hierarchy of communities for structuring the meta-management of executable policies for DSA-aware RF devices [xg-policy].

The benefits brought by integrating CBPM with XGPL stem from the organizational modeling that CBPM uses as a basis for enforcing policies. CBPM uses the notion of *community* as the grouping abstraction with the aim of allowing groups within the organization itself to define communities to naturally reflect the changing nature of decision making (i.e. policy setting) authority. This contrasts with modern role-based access control systems [sandhu][lupu], such as that used in IBM’s Workplace, which typically require the services of external consultants to define roles and their policies. This approach is both expensive and brittle in handling frequent organizational change [jude], as will inevitably be the case of a multi-organizational setting such as DSA. Communities towards the top of the hierarchy have the wider membership and more general function, while those toward the bottom have

more narrow membership and more specific function. The hierarchy is designed to support organizational change, allowing new sub-communities to be formed and encouraging the delegation of decision making authority as far down the hierarchy as possible.

This structure was in part inspired by observation of how on-line communities operate, where there is no legal seat of authority, but where chains of authority emerge over time. It is our contention that this is a feature of all successful organizations, i.e. that all organizations are subject to continuous organization change from within and without and it is those that manage that change well that succeed and prosper. In this sense we view the set of relationships between regulators, operators, secondary markets and authorities managing spectrum commons as a multi-domain organization. For this reason CBPM has been specifically designed to accurately reflect the structure of decision making authority within an organization. More importantly, however, the scheme is designed to be agile in reacting to changes in an organization's structure.

The key insight upon which the CBPM model is based is that of modeling the organization independently of its decision making structure. The organization is conceived as an inverted pyramid, with each level representing a different functional and structural unit within the organization and the entire organization forming the base of the pyramid, as in figure 1.

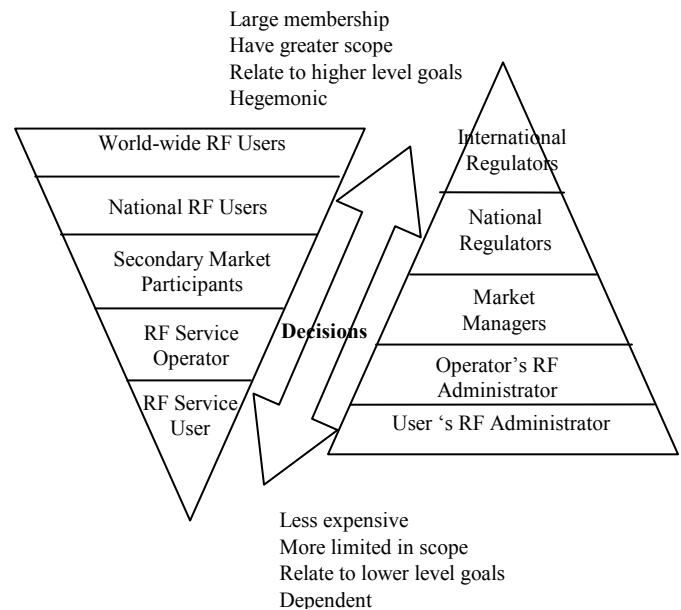


**Figure 1 Organization as a Hierarchy of Authority**

As we descend the pyramid, each level is more specific in function than the level above and has a subset of its membership. By specific, it is meant that the function of the unit is to carry out a specific function which has been identified as assisting in the attainment of the general goal of the organization. This pyramid is termed the hierarchy of authority. Although the diagram above only shows a single path through the organization, the multi-domain dynamic spectrum management organization will have multiple branches and include federations between organizations at various levels. The diagram illustrates the hierarchy of authority between organizational units at various levels of generality, rather than being a model of the organization.

This hierarchy of authority is used as the basis of an

*authority map* of the organization. Each organizational unit is associated with a decision-making method. An example, of this mapping is shown in figure.2, where the hierarchy of authority in a traditional monolithic organization is mapped to the equivalent groups and individuals who make decisions on behalf of the functional and structural units. It should be noted that policies are considered to be “decisions about choices in the behavior of a system” and are merely a means of specifying decisions that have been made. The notion of decision-making as it is used here is entirely identical to the authoring of policies.



**Figure 2 Decision Making Hierarchy**

Each level in the pyramid on the right represents the decision-making methods of the organizational units in the pyramid on the left. In moving up the hierarchy of authority, it can be observed that decisions:

- *Have Greater Scope.* Decisions of a regulator at the top of the hierarchy may be binding on the entire membership of the organization, i.e. all RF users, and may relate to any of the organization's resources – they have global scope. Meanwhile, decisions made by a local wireless network administrator normally only apply to the users of that network and to the resources that have been allocated to it – they have a scope that is limited to that group.
- *Relate to Higher Level Goals.* Normally, the decisions of the regulator will relate to the high level goals of the multi-domain DSA organization - they will be defining strategies for the entire organization to follow, rather than concerning themselves with the low level workings of the various groups that make up the organization. Similarly, the local administrator's decisions will reflect the particular goals of their network operation activity and they may not be directly derivable from the organization's high-level goals.

- *Are More Expensive.* Decisions at the higher levels of the organization are considered more expensive in general than those further down the hierarchy of authority in two ways. Firstly, it is normally the case that the higher in the hierarchy one goes, the slower and more time-consuming the decisions are. For example, regulators have decision making meeting relatively infrequently, while a local administrator can often take an instantaneous decision without consulting anybody else. Secondly, the cost of mistakes in decisions generally becomes more expensive as one ascends the hierarchy.
- *Can Overrule Decisions at a Lower Level.* Finally, it should be noted that the further up the hierarchy one goes, the greater the weight of the decisions. If the regulator makes a decision on behalf of the entire body of RF users, all of the units further down the hierarchy are bound to follow it. Individual operators or administrators require the explicit permission of the regulator before they can take decisions which conflict with the latter's decisions. In the diagram, decisions taken at the higher levels are described as hegemonic while those taken further down are dependant.

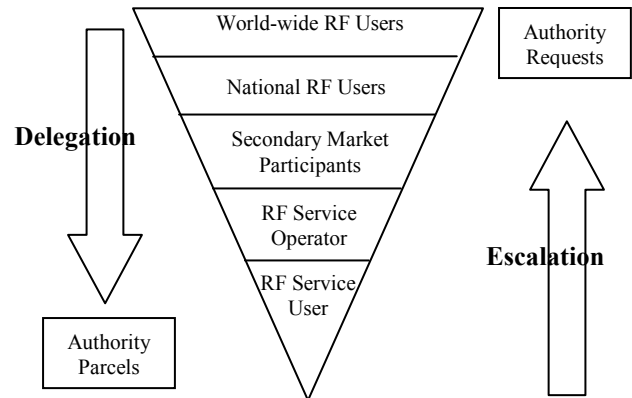
It should be noted that these characteristics generally hold, regardless of the decision-making hierarchy that is associated with the organization's hierarchy of authority. For example, if an organization is run according to the principles of direct democracy, where every person who is affected by a decision has a vote in the decision, this can be modeled by configuring communities so that a decision within any unit of the organization requires a referendum of all the members of that unit. Decisions made for local set of wireless users might often be relatively inexpensive, since the group will have few enough members working together to be able to take a vote quickly, and the decisions will still have the same scope and relationship to the goals of the organization as if the organization was managed by a traditional hierarchical bureaucracy. Conversely, in any sizeable organization, it can be expected that the time and effort involved in organizing a referendum of the entire organization will be considerable. However, once a decision has been taken by universal referendum, all of the members of the organization, and all of the units within the organization will be bound to follow it according to the general principles of this organizational model.

Based on these characteristics of the hierarchy of authority, the following rule of thumb for the distribution of decision making throughout the organization is adopted.

*Decisions should be made as far down the hierarchy of authority as is possible*

Due to the increasing expensiveness of decision as one ascends the hierarchy, it is desirable to distribute decision making authority to the lowest possible point – it makes much more sense for the details of the workings of a local wireless network to be decided by the team local administrator than by

the regulator. In order to fulfill this rule, the CBPM model is designed to be inherently dynamic. Each unit in the organization aims to push decision-making responsibility downwards wherever possible and units can be dynamically created whenever existing units manage to decompose their own structure and function into sub-functions. The process of pushing decision making down the hierarchy is based on delegation. However, this notion of delegation differs from that used by most PBM systems. Rather than the temporary delegation of rights from one individual to another, this delegation is a semi-permanent transfer of authority between organizational units.



**Figure 3 Delegation / Escalation Cycle**

Figure 3 illustrates the basic dynamic of the CBPM model. Higher-level units in the hierarchy of authority delegate Parcels of Authority to the lower level units. A parcel of authority refers to a set of resources whose use is constrained in various ways. Through delegation, the grantee unit gains authority to make decisions about the resources that have been granted, but those decisions are bound by the constraints specified in the parcel of authority. Receipt of an authority parcel amounts to being given responsibility to make decisions about the management of certain resources in order to fulfill the goals of the unit. This notion of the distribution of authority in constrained bundles to units which serve as a representation of the stakeholders of the potential decision, reflects research into human interactions with policy based systems [barrett].

In order to respect the principle of least privilege, these parcels of authority should be constructed in such a way so as to minimize the scope of authority to that which is strictly required. The dialectical goals of respecting this principle, while maximizing the amount of responsibility that is pushed downwards, provide one of the motors which drive the evolution of the system. This is a manifestation of the fundamental tension in all management systems between the certainty of centralized control and the convenience of decentralization.

The other factor that serves to shape the dynamic evolution of the system is escalation. Escalation deals with situations

where a functional unit needs to make decisions about a resource for which it has not been delegated sufficient authority. It amounts to a lower unit asking a higher unit to delegate it authority, or to take a decision on its behalf with regards to the relevant resource.

In the hierarchy, the authority to author management policies flows from the top of the hierarchy to its leaf communities. Information flows up the hierarchy concerning policy conflicts and situations where there is a lack of sufficient policy guidance to make decisions in particular contexts. It is these two flows up and down the hierarchy of communities that forms a dialectic that can be harnessed to improve the operation of the systems.

### III. CBPM IMPLEMENTATION

A high-level outline of the implementation of the CBPM System (CBPMS) is given in figure 4. At this level, the workings of the CBPMS can be divided into two different functions. Firstly, it serves as what is termed a Policy Decision Point (PDP) in the IETF architecture. In the most general terms, this function can be described as responding to events by evaluating the appropriate policies and returning the result of those policy evaluations if required. Secondly, it serves as a policy management system, corresponding with the Policy Management Tool of the IETF architecture, the function of which can be described, again in the most general terms, as responding to management requests by updating the state of the system in accordance with the rules for the operational semantics of the community based model and returning the result of that management request if desired.

A service oriented paradigm is adopted, where the system is divided into a number of discrete providers of well-defined and stateless services and these services are orchestrated in order to provide the required functionality of the system. A service-oriented architecture eases the integration of the CBPMS with existing policy systems and network architectures, as well as allowing other systems to gain selective access to information about the state of the policy system. A top down approach to the definition of the services that make up the system is adopted, defining each component in the system in terms of simple services that it provides and the functions that it provides to the consumers of the service. The first level of analysis of the functioning of the CBPMS - as a service which acts as a PDP - is known as the Community Policy Decision Service (CPDS). An event occurs somewhere in the domain covered by the CBPMS, causing the service consumer to issue an appropriate policy decision request to the policy decision service provider. For instance a user may wish to use a cognitive radio in a new area, the device senses transmission opportunity in an unused RF band, but before making use of it requests a policy-based decision on the permissibility of this action from the CPDS.

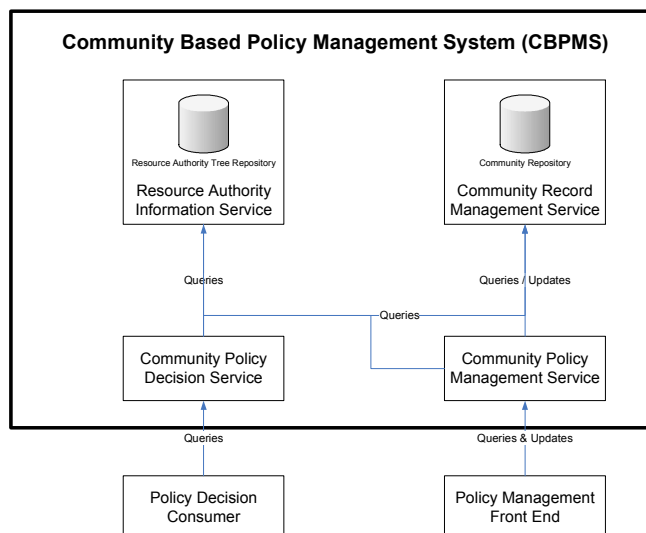


Figure 4: CBPM System Overview

The primitives offered by the Community Record Management Service (see Table A) can be used by policy authors to provide meta-policy management for DSA. This involves interaction with the system from the Policy Management Tool.

CBPM Primitive	Effect
genesis / expel	Create a new organizational model or remove an existing one
spawn / cull	create or destroy a sub-community
delegate / recall	Delegate authority from a community to a sub-community
policy / revoke	Define a policy for a community or remove one.
merge / liberate	Merge two organizations together or split an organization in two
federate / withdraw	Join a federation or leave one

Table A: Community Management Primitives

By making it easy to continuously change the organizational structure represented in the CBPM system to reflect changes in the real organization, the CBPM system can be used to document the structure and reason about potential changes. A potentially powerful benefit of this use of the CBPM system is that it enables conflicts that are detected between separately authored access control policies to be mapped to the point in the organization where the divestiture of the authority was sanctioned. In this way the CBPM system can be used to resolve problems in the organizations authority structure as they are encountered through the detection of conflicting policies.

In CBPM, communities and the actions that can be performed upon them, are included as a hierarchical resource model, normally consisting of a hierarchical tree of targets and a hierarchical tree of actions. Delegation involves the allocation of a pair of nodes from these two trees from one

community to another. This provides fine grain management of the delegation of authority to make both structural and policy changes in an organization.

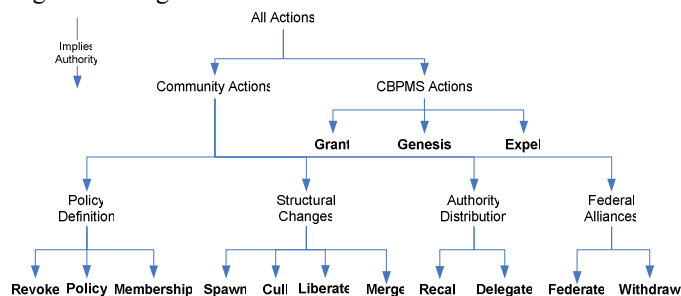


Figure 5: Action tree for the community resource

Figure 5 shows the built in action tree related to the community resource. The structure of the resource target tree in this case, is the structure of communities forming the hierarchy of authority in the organization concerned. This therefore provides inherent manageability of whom can write policies about what by controlling access to the community management primitives (see Table A) offered by the web service implementation of community Policy Management Interface and which map the leave of the community action tree. A more detailed description of the operation of CBPM is given in [feeney07]. In the remainder of this paper we examine in more detail the design of the Policy Management Front end and the impact this has on the collaborative policy engineering process within the CBPM scheme.

#### IV. CBPM POLICY MANAGEMENT TOOL

Our initial implementation of a policy management front end for the CBPM implementation was a form-based web interfaces [lewis06]. Though this was adequate for general manipulation of the CBPMS, it was not designed with ease of use and productivity of the policy engineer in mind. To address this, a more sophisticated CBPMS client was developed to fulfill the policy engineering tool role. This client software architecture is intended to run at the end-user's PC and will communicate with a server running the CBPMS implementation. The CBPMS is accessed via a web service implementation of the primitives in Table A together with a general purpose interface for browsing community, resource, context and policy structures. Access to these interfaces is managed in accordance to policies governing access to this information for different policy tool users.

The client presents the user with a management interface adhering to the CBPM principles. This means that the user will use communities as the grouping abstraction in order to construct the desired management system. We aim to provide an interface that exploits direct graphical manipulation of models. For this, the client is implemented using the Eclipse Graphical Modeling Framework (GMF). GMF provides the component and runtime infrastructure for developing graphical editors based on Eclipse Modeling Framework (EMF) and the Graphical Editing framework (GEF). In our case the GMF diagram was created using annotated Java. This provided both a rich graphical interface as well as a meta-model driven

structure to that interface, making it possible to rapidly prototype varying representations of the CBPM model being manipulated. The challenge this flexibility presents, however, is in developing a form of interface that best suits the collaborative policy engineering tasks for which it is used.

As a proof of concept, a simple scenario was created that makes use of the client's interface for DSA management. Here we present some screen shots from this proof of concepts to convey a possible interactive framework within which specific interface design decisions must be made. In the scenario presented, it is assumed that a part of the spectrum which we call Band A is available for allocation. Two Sub-Communities (named KDEG and NTRG in our example) have acquired the resource authorities from a Root Community (named here TCD) and can therefore write policies on the above resource (Band A). This management structure is presented in figure 6.

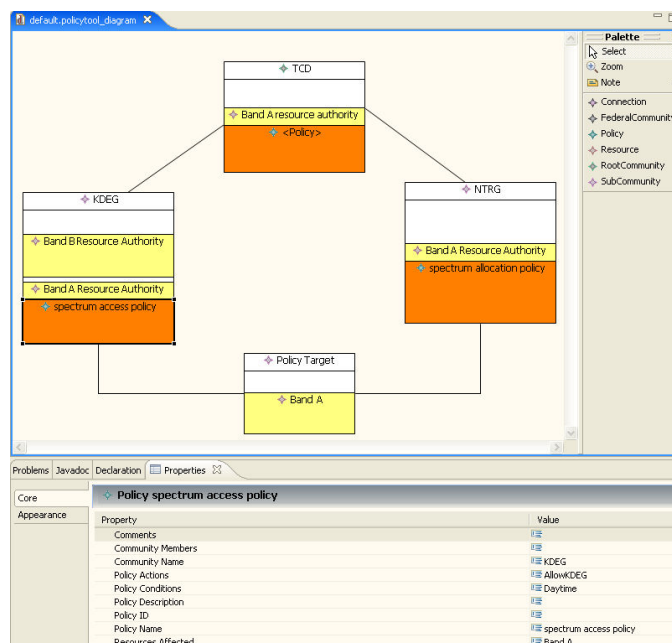


Figure 6: The community management structure using GMF

The end-user creates all these communities as well as the resources upon which the policies will be installed. This can be done by selecting any of the available model elements that are shown on the right hand side of the GMF diagram.

The user then writes policies on behalf of the two sub-communities. A sample spectrum allocation policy template is shown in figure 7. This view is presented when the user double-clicks in any of the orange boxes that represent policies in the GMF diagram. Alternatively, the user can edit the policy properties that appear at the bottom of the GMF diagram. In order for the policy to be installed, the Submit Policy button shown at the bottom of Figure 7, needs to be pressed.

If the policies conflict with each other, the server will notify the user with a window pop-up in the client that a conflict has occurred. At the same time, it will present the user with feedback about the likely cause for the conflict (figure 8).

In our example a conflict occurs when policies are submitted allowing simultaneous transmission of telemetry info on Band A for both KDEG and NTRG sub-communities.

Since only one sub-community at a time should transmit, the last one attempting to transmit will be notified that a conflict has occurred. When the user changes the policy action accordingly, the conflict is resolved and the user is notified again with a window similar to Figure 8, that no conflicts exist in the system any more.



Figure 7: A template for spectrum allocation policies

For the purposes of this presentation, the policy rules and conflicts are produced directly from a JBoss Rules (drools) rule engine, unmediated by the CBPMS and its community based meta-policy enforcement mechanism. Though the CBPM integration is under development, the direct JBoss Rule output serves to illustrate the operation of the interface. It provides a detailed analysis of the rules invocation at runtime is available. Using the Audit View available in drools we can see when every object is asserted and when this leads to a rule activation and to an activation execution. Figure 9 presents the case when a policy conflict is detected, whereas figure 10 presents the case when a conflict is resolved.

It should be noted that JBoss Rules uses its own language which is in non XML textual format to represent rules. When a Domain Specific Language (DSL) is used, a high level natural language (written English) is used to represent rules, which can then be mapped to the native JBoss Rules language.

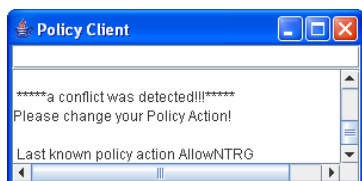


Figure 8: Conflict detected at the client

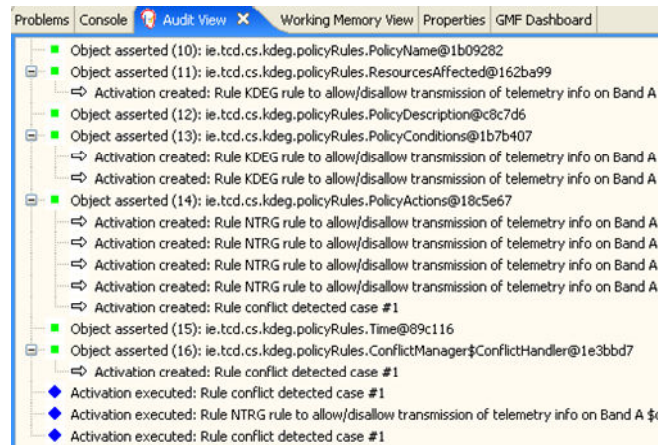


Figure 9: The Audit View of JBoss Rules showing conflict detection

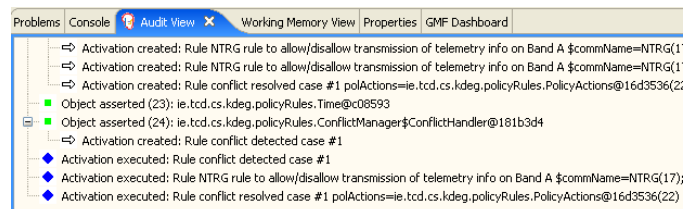


Figure 10: The Audit View of JBoss Rules showing conflict resolution

This user interface, however, presents only one approach to presenting a policy-based management interface to a user. Typically users are network or IT administration staff, who are few in number and therefore affordably subject to specific training in using a new administration tool. There is often little attention paid to the usability aspect of PBM interfaces since shortfalls in this respect can be addressed by specific training. However, the underlying assumption of CBPM is that policy-authoring is democratized throughout an organization, whether it be a single enterprise or a federation involved in DSA management. Therefore, the body of users using any CBPM interface will be diverse and interfaces will have to be designed with specific classes of non-specialist users in mind. Furthermore, the design of an appropriate user interface may also be dependent on the size and structure of the communities and resources being managed. The Eclipse/GMF approach to the current GUI design will support such variety to an extent through rapid meta-data driven GUI generation. However, guiding the design of a variety of CBPMS GUIs to suit different user communities and their policy authoring needs requires a systematic approach to analyzing these needs. Indeed, these needs may only be detected once instances of CBPM are undertaken, so such a systematic approach must also suit post deployment meta-data driven adaptation of GUI operations. Such an approach should therefore operate in concert with the policy authoring delegation-escalation control loop which, though progressive grounding of meta-policies, also provides runtime adaptation of how policy authoring operates. The remaining section proposes a unified mechanism for the progressive refinement of usability coupled with that of policies. This is based on the analysis of policy-authoring problems using semiotic analysis, within an organizational

modeling framework that supports recursive analysis of operations, management and meta-management (i.e. governance) in context.

## V. AN ANALYTICAL FRAMEWORK FOR POLICY SIGNALS

As discussed in section II, the primary benefits of applying CBPM to a multi-domain application such as dynamic spectrum management, is that it provides a mechanism for managing the flow of authority between different parts of an organization or composition of organizations, as well as handling the escalation of problem discovered within the shared policy set. Though the example in the previous section was a simple resource-centered modal policy conflict, in a scenario such as DSA where there are large number of organizations, all with authorized personnel contributing the shared policy set, meta-policy conflict must be dealt with. Such a conflict could be between separately authored policies that give conflicting permissions related to the resources, context and actions for which personnel in a particular part of the organization (modeled as a community) can write policies.

Analyzing interaction design decisions for such scenarios is problematic because it is often difficult to separate issues related to the collaborative application from issues about how the users use it to communicate. In our scenario the user interaction is conducted disjointly in time between one policy author and another. Nevertheless, each policy, especially meta-policies related to policy authoring capabilities can be regarded as a form of inter-user communication.

The approach we aim to adopt is based on the work of deSouza and Preece in applying Semiotic Theory to online community support systems [deSouza]. This work forms part of an online community framework that attempts to separate issues of usability, which are software based, from issues of sociability which are based on how people communicate, the rule that use to govern their interaction and the aims or purposes of those interactions. Semiotic theory involves the analysis of signals in a wide sense. The benefit of applying this analysis to collaborative systems is that a single technique can be used to analyze both the signals exchanged between people using the system and signals exchanges between the designer of the system and its users. The former therefore allows analysis of the system's sociability while the latter allows analysis of its usability.

The approach used is to apply a specific test to interactions that are raised as problematic in usage studies of the system. This test, called the Communicative Adequacy Test (CAT), consists of a simple set of questions posed to a user who has identified some unsatisfactory interaction feature of the system. The questions relate a signal or message experienced by the user. Broadly, they determine whether the user understood the message. Secondly, could they detect from whom it was from, i.e. from another user or from the system, as the runtime proxy of the designer. This is broken down to determine if the user could discern who had established the semantics of the message and who had actually generated it. Finally, the user is asked to assess the level to which the system is interfering with the message.

This approach seems well suited to analyzing policies as signals, though they represent a different form of communication from the user-user or user-designer/system communication to which semiotic analysis is applied in [desourza]. Specifically, the signals exchanged in the CBPM are the policies drawn up by one group that impact the policies that can be developed by another. This corresponds to the delegation flow in figure 3. Equally, the escalation flow could be treated in the same way. Currently the CBPM does not have a formal system for this flow up the hierarchy of authority, though some form of trouble ticketing system would seem appropriate in a practical system.

To give an example for the scenario set out in the policy management client description given in the previous section, the proforma given in [desouza] for conducting the CAT could be completed as follows:

**Identifier** = conflict message in figure 8

**Speaker** = a previous policy author (traceable through policy authorship meta-data)

**Listener** = current policy author

**Topic** = a conflict between a new policy and an existing one

**Content** = rule engine description of logical conflict detected

**Form** = rule conflict listing

**Speaker intent** = to prohibit specific use of this frequency band in circumstances that include those specified for the new policy

**Listener Understanding** = expects listener to understand rule engine conflict listing

**Listener Response** = expected to either rewrite new policy or escalate the conflict to be addressed higher up the authority tree

**Pre-condition** = expects the user to understand the semantics of the terms used in the previous policy and to understand

**Post-condition** = expect the user intent in writing the policy to be fulfilled or for the user to understand why this intent cannot be accommodated (or in the less ideal case have it confirmed simply that it cannot be accommodated)

The above represents the ideal response, reflecting perhaps the designer's intention, however the aim of administering these questions would be to determine where the users response is at variance with this ideal. For instance is there a difference in the assumed post-conditions, or a misunderstanding of who is producing the message. As this mechanism can be used to both analyze the design of the interface and of the policy logic, it can be used to differentiate the problems associated with each. Thus we can more readily diagnose where usability problems arise from the user interface design, when it arises from how policies and conflicts are conveyed and when it arises from the policies themselves being poorly constructed, e.g. by selecting a confusing model of the resources being managed. A round of usability tests is planned for the CPBM client and such a semantic analysis will be used to drill down into the causes of problems experienced and make modifications in the appropriate areas.

While the approach in the previous section shows promise for the analysis of problems identified with policy engineering interfaces, when problems relate more to the semantic of the policies involved, correct analysis requires a systematic understanding of the broader organizational setting in which policy engineering decisions take place. This is particularly

important when dealing with problems that arise in resolving conflicts between policy authoring decisions and meta-policies. Our experience indicates these types of problems present themselves as particularly convoluted even to an experienced policy author. This is often because an individual policy author views their problem from their position in the authority hierarchy of the organization, whereas a more holistic view of the interaction between levels of authority is needed to debug these problems. A candidate framework we are examining to provide a more holistic view of flows of authority in an organization is the *viable system model* (VSM). The VSM was developed by the operations research theorist Stafford Beer [beer]. A viable system is one that is able to survive change. The VSM was developed to analyze how human organizations, e.g. corporations and governments, are best able to structure themselves as viable systems in order to adapt to change over time. The model broadly breaks down a system into the Operation of an organization, the Environment in which that operation is conducted and the Management that is exercised over the operation. The VSM places a focus on analyzing the channels that exist between the Environment and Operation and between Operation and Management. The analysis of these channels uses the concept of Variety. This is a measure of the number of states in which a system can occupy. The essential rule applied to by VSM to these relationships is that to be viable a regulating system must possess as least as much variety as the system it regulates. This premise is taken from Ashby Law of Variance [ashby]. However, an organization's Operation typically has less variety than the broader Environment in which it exists, while organizational Management typically has less variety than the Operation it manages. This problem is addressed by allowing the channels between Environment, Operation and Management to amplify or attenuate the Variety of one system as perceived by another. For instance the channel from Operation to Management will attenuate the variety that needs to be perceived, e.g. summarizing performance metrics.

The VSM breaks down the Operation and Management functions into a set of systems. This breakdown is recursive, such that each element can represent a viable system. The Systems defined are as follows:

- *System 1* represents the primary activities of a viable system, i.e. the functions that justify the system existence. Crucially, these systems are regarded as recursive, i.e. each can be a viable system, made up of the same composition of systems. The viable system will often consist of several System 1 elements working together.
- *System 2* represents the information channels and processes that enable System 1 entities to communicate with each other in achieving the aims of the overall system
- *System 3* represents the structures and controls that define the operational behavior of System 1. It uses System 2 to monitor and co-ordinate the activities of System 1. System 3 is complemented by a further system, System 3\*, which monitors the working of System 1 directly, bypassing the coordinating filter of System 2.
- *System 4* represents the outward looking aspect of an organization, monitoring the environment and also

assessing potential future changes. It interacts with System 1 via System 3.

- *System 5* represents the policy decisions that balance and steer the organization as a whole, capturing its mission and ethos. Part of this for a viable system is the goals of homeostasis, i.e. reaching a state of equilibrium.

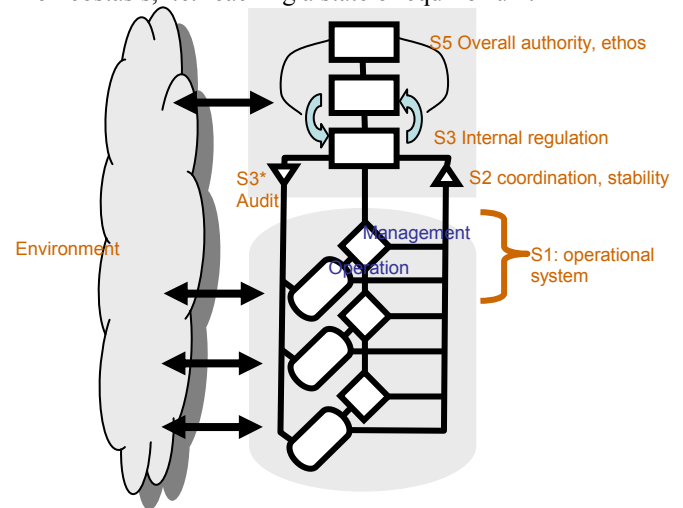


Figure 11: VSM Overview

Figure 11 illustrates the overall structure of the VSM. It shows the different communication channels between the Environment (the left-hand area), Management (the large square) and Operations. It also shows the pattern of recursion, where System 1 represents the Management portion of the next level of recursion. It also indicates the feedback loops that enable a system to become viable through self-management. This is primarily performed by System 3, but with feedback provided from System 4 which, coupled with intelligence gathered from the environment and predictions of the future provide meta-management imperatives to System 3. The interaction between System 3 and System 4 is in turn regulated by the overall ethos of the organization.

In considering the application of the VSM to DSA we could surmise that the outermost recursive loop would be structured such that the Management systems are enacted at the international level. International agreements on the aims of DSA could be enshrined by de jure bodies such as the International Telecommunication Union – Radio-communications, while bodies such as the IEEE and the Wireless World Research Forum could conduct the activities of System 4, guiding the adaptation of the system as technology advances and political and economic changes occur. System 3 at this level may most likely be an enabling activity, supporting the implementation of international standards through certification of technologies and processes by industry sponsored bodies. It could however be the position of an international market in trading spectrum rights [doyle]. In the main however, we expect most of the management functions to be conducted by national and regional bodies embedded in the next level of recursion. At this next level the management systems would largely be operated by the local regulator, e.g. the Federal Communications Commission,

Office of Communications in the UK or the Commission for Communications Regulation (COMREG) in Ireland. However, System 5 and to some extent System 4 would involve elements of national and regional government, e.g. the European Commission, as they seek to exercise oversight and influence over the management of the spectrum, regarding it as a valuable national resources and a crucial part of the infrastructure of modern economies. At this level of recursion System 1 would be made up of particular spectrum users, e.g. the military and cellular operators, as well as newer bodies given sub-regulatory powers, e.g. national spectrum trading markets or metropolitan authorities changed with handing spectrum commons. Within these System 1 instances, management would be exercised within the constraints of the authority passed down from the higher level, but would again operate to an ethos, e.g. maximize trading volume and spectrum value, or social inclusion and fairness of access. Again these systems would have to involve System 3 and System 4 activities in order to stay viable. The System 1 instances at this level would come closer to specific spectrum user organizations, individuals and cognitive radio devices.

Such a holistic analysis of the flow of authority and enacted through the definition of policies exchanged between the different systems, provides an organizational level framework which can be overlaid on the semiotic analysis presented earlier.

## VI. CONCLUSION AND FUTURE WORK

In this paper we highlight the problems associate with developing effective tools for collaborative policy engineering. We describe how the CBPM can be applied to possible future dynamic spectrum management scenarios, which is a demanding example of inter-organizational collaborative policy engineering. We then describe the problem involved in developing suitable interfaces for such engineering and propose a novel application of semiotic theory to provide a more systematic approach to analyzing problems with the interface design. Underlying this approach is a shift to regarding policy-authoring decision and feedback on their impact on other policy authoring decision as flows of signals that pass through an organizational structure.

Our future work will involve the application of this approach to a more comprehensive evaluation of the policy engineering client. More broadly we will formalize the integration of semiotic engineering with the VSM to provide an analytical framework for policy engineering interactions that exposes the organizational and inter-organization management interactions within which individual policy engineering decisions are made. In addition we plan to examine using ontology based semantics for capturing the resource and context models used to define policies, so that these can be extended and refined more readily as part of an on-going analysis and improvement cycle for a collaborative policy engineering systems based on the CBPMS.

## REFERENCES

[ashby] Ashby, W.R., "An Introduction to Cybernetics" (Chapman and Hall, 1956)

- [barrett] 11. Barrett, R. People and Policies: Transforming the Human-Computer Partnership, 5th IEEE International Workshop on Policies and Distributed Systems and Networks, IEEE, 2004
- [beer] S. Beer, The viable system model: Its provenance, development, methodology and pathology, Journal of the operational research society, Volume 35, Issue 1, pp 7-25, 1984
- [campbell] Campbell, C.S., Kandogan, E.; November, A.; Barrett, R.; Maglio, P.P. Policy: an experimental evaluation of policy-based administration in a city simulation Sixth IEEE International Workshop on Policies for Distributed Systems and Networks, 2005. 6-8 June 2005 Page(s):119 – 122
- [desouza] De Souza, C., S., Preece, J. (2004) A framework for analyzing and understanding online communities. Interacting with Computers, The Interdisciplinary Journal of Human-Computer Interaction.
- [dijker] Dijker, B. L., "A Guide to Developing Computing Policy Documents," SAGE, <http://sageweb.sage.org/resources/publications/policies/>.
- [doyle] Doyle, L., Forde, T., Argyroudis, P., Nolan, K., O'sullivan, D., "Towards a Fluid Spectrum Market for Exclusive Usage Rights", DySpan 2007
- [jude] Jude, M., Policy-based Management: Beyond the Hype. In Business Communications Review, available from <http://www.bcr.com/bcsmag/2001/03/p52.asp>, March 2001. pp 52-56
- [lewis06] D. Lewis, K. Feeney, K. Foley, L. Doyle, T. Forde, P. Argyroudis, J. Keeney, D. O'Sullivan, Managing Policies for Dynamic Spectrum Access, 1st International IFIP TC6 Conference on Autonomic Networking (AN2006), Paris, France, September, 2006, LNCS 4195, Springer, 2006, pp285 - 297
- [lupu] Lupu, E. C. A Role-Based Framework for Distributed Systems Management. PhD Thesis, Department of Computing, Imperial College. London, U. K., July 1998
- [maglio] Maglio, P. Campbell S. Kandogan E. On the Need for Negotiation in Policy-based Interaction Proceedings of the Second International Conference on Autonomic Computing, Seattle, WA, USA between 13-16 June 2005 [xg-vision] DAPRA XG Working Group, "The XG Vision Request for Comments". Version 2.0, prepared by BBN Technologies, Cambridge MA, USA, January 2004
- [feeney07] K. Feeney, D. Lewis, P. Argyroudis, K. Nolan, D. O'Sullivan, "Grouping Abstraction and Authority Control in Policy-based Spectrum Management", IEEE Dynamic Spectrum Access Networks (DySpan 2007), Dublin, Ireland, 17-20 April 2007
- [sandhu] Sandhu R., Coyne E., Feinstein H., Youman, C. "Role-Based Access Control Models," Computer, vol. 29, no. 2, pp. 38-47, Feb., 1996.