Patterns of Trust and Policy

Daniel J. Essin[1]

[1] University of Southern California
essin@usc.edu

This paper proposes a new paradigm of trust and policy that provides a unified treatment of organizational and data system policies. Policy is the programming language of organizations and just like any other language must be formally specified or specifiable. This paper attempts to demonstrate that it is specifiable. Trust is a major component of policy. Trust is presented as a function of specific elements - identity, reputation, capability, stake and benefit. These elements are defined and presented in the form of a trust equation. The points at which trust enters into the formal definition of policy are identified. The trust equation provides a useful way do describe trust in general that is not circular (unlike many previous definitions). The resulting constructs can be sufficiently non-technical that both systems people and those without a technical background can understand them. The availability of a common language to guide analysis of policy requirements, policy formulation and policy execution may provide a way for organizations to break out of a recurring cycle of policy failures.

1. Introduction

"At the beginning of God's creating of the heavens and the earth ... the earth was wild and waste." [1] In rapid succession, God created light, plants and animals, and mankind - to subdue the earth and have dominion over its assets. God then created the first policy: "...but from the Tree of the Knowing of Good and Evil - you are not to eat from it, for on the day that you eat from it, you must die..." [2] In the process of establishing this policy God performed the first risk assessment, established the first trust relationship and indicated that information was an asset deserving of special protection. As it turned out, the trust relationship was weak and the policy unenforceable because the proposed sanction conflicted with others that had equal precedence.

The focus of this paper is the socio-technical workplace often created by service organizations. Their goal is to produce a work product that must be of high quality, is often highly regulated and where there is a potential for catastrophic loss if adequate quality is not realized. The cost of attaining the goal is important but often it is not the predominant concern. Examples include airlines, banks, and police crime labs. Organizations use a variety of "systems." Some are sociological (often called "manual" systems) and some use computers to perform some or all of the work (frequently called "automated" even when the only thing automatic is the movement of characters from keyboard to screen and then to disk). In this setting, people are trusted to discharge their duties (as defined in the policy and procedure manuals, regulations and laws) and to refrain from doing those things that are proscribed. The people doing the work, in turn, trust the systems they use to facilitate (or at least not impede) their attempts to do the right thing.

As organizations evolve, work is frequently restructured. Computer systems are introduced, modified, replaced and eliminated but the policy framework and the trust relationships must remain consistent and functional. More importantly, these are frequently domains in which trust is a key element but in which data systems and traditional data security mechanisms do not figure prominently. This paper proposes to explore how organizations can implement policy and assess trust in an environment in which data systems and calculating machines are not the predominant activity and how this might be done in a quantitative and structured manner that could support the future automation of portions of the policy and trust machinery.

The following discussion will draw its definitions from the opening example. The terms, **people**, person, individual and one will be used to represent mankind. That which mankind is to subdue and over which it is to have dominion will be referred to as items, objects, assets or **resources**. This will include any derivative items (including information) produced as a function of mankind acting on resources. It is assumed that People engage in **goal**-directed activities. The term **actor** may be used to describe either an individual or an object in a setting where either can perform a similar action. A group of actors that can be considered equivalent in some context, when combined with the similar action that they can all perform define a **role**. As actors observe each other's actions, and experience conflict between their goals and the goals of others, they find it necessary to attribute **value** to various objects and to create policy. A **policy** proscribes or prescribes behavior in an authoritarian context but without the force of law, i.e. the promulgator of the policy has some power to require compliance and sanction violations. The objective of policy is to regulate access to objects and/or dictate the behaviors deemed necessary to achieve the specified goals. It is important to note that it is common for actors, goals and actions to go unnoticed, or to be perceived as not conflicting, and therefore not to become a topic of a policy. Policy invocation and **policy instance** will be used to describe the series of events and decisions that is carried out after the policy is triggered by a

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
1997 New Security Paradigms Workshop Langdale, Cumbria UK
Copyright ACM 1998 0-89791-986--6/97/ 9...$5.00

particular set of circumstances. The term **outcome** or objective will be used to describe a set of end results that may be either positive or negative. When assessing outcome or developing policy to influence it, the term **outcome probability** will be applied to the estimate of the probability of gain or loss, i.e. the probability that the outcome in question will be successfully realized or avoided. The terms threat and risk will be applied more specifically to situations that have a negative effect on outcome; result and opportunity will be applied to positive effects. Used in this way, opportunity cost, projections of market share, the likelihood of causing or sustaining an injury and concern for reputation are all risk assessments. The terms **trust** and trustworthiness will be used to represent a conclusion (at a point in time) as to whether an outcome will be positively affected (or not adversely affected) by allowing an actor to interact with resources in the context of the risks involved.

2. Background

Healthcare presents a socio-technical workplace in which trust figures prominently. Healthcare organizations manage vast quantities of information - some of it with computer systems - to which access must be controlled. The legal and ethical environment which surrounds medical practice requires that special rights and authorizations be granted before individuals can access this information or perform many of their daily activities, such as psychiatric evaluation or surgery. In response, organization need to do three things: 1) define specific criteria that must be met before concluding that an individual is trustworthy 2) establish and adhere to numerous policies and 3) employ some means of evaluating outcome probability when dealing with policy and trust issues.

Unlike temperature and weight, trust is not a physical property of objects or entities. It cannot be measured directly. It is commonly understood to be an opinion. Computers at Risk [3] offers a somewhat circular definition of trust as "the belief that a system meets its specifications" - the implication being that one already "trusts" the specifications. Faced with what appears to be an imponderable, the common approach is establish policies, i.e. define conditions in which trust would be deemed to be absent or violated and what should be done under those conditions without first defining trust.

3. Related Work

Stepney and Lord [4] have dealt with trust in the context of a formal model of access control. They used the Z language to specify an access control system and schemas to structure and modularize the specification. One of the advantages cited for this approach is that the specification is readable and conveys its general intent even to those that may not understand the details. In their model, an access control decision may be made by a single "local authority" or may be influenced by trust information, in the form of predicate statements, obtained from other (trusted) authorities. Statements take the form " **FRED has UPDATE access to PAYROLL_FILE**". Their formulation states explicitly that establishing and changing trust relationships and, by implication, the security policy itself, are outside of the model and are to be decided and set in place by human administrators. In this model, trust is asserted, not calculated or derived. Any variation in an access control decision from moment to moment would presumably b the result of combining different sets of trust assertions from different authorities, not because a single authority changed its trust assessment dynamically based on local circumstances.

Lampson, et.al. [5] describe using a reference monitor to guard each object (stored data or process) and to make a case by case decision as to whether there is sufficient trust or authority to allow the request to be fulfilled, based on the source of a request, information in an access control list (ACL) and logic (rules). Although their scheme is based on determining if a request originated from a principal that is trusted as much or more than a principal listed in the ACL, the fact that their model admits roles, groups, delegations and conjunctions allows for considerable indirection in how access is actually granted. The focus of Lampson, et.al. is on authentication. Other aspects of trust are left to the domain of the trusted computing base including the communication channels and, like Stepney and Lord, their task is completed when the request is either granted or denied.

Hauser [6], while discussing the need to integrate an allocative licensing scheme with a computer operating system, introduces the concept that activity must be monitored beyond the time at which initial access is granted. Various policies or other operational or accounting tasks may be invoked, depending on circumstances, at any time while a resource is in use. He also suggests that the state of each access needs to be maintained, for the life of that activity, in a data structure that contains entries relating to other active users, priorities, and factors that might influence early or forced termination of a user's interaction with a resource.

4. Conceptual Framework

4.1 Trust

Returning to the Garden of Eden - God trusted Adam and Eve to adhere to the policy of not eating from the Tree of the Knowing of Good and Evil, but on what basis? Their identity was not in question nor could they hide it. There were just the two of them; there was no crowd of individuals already clothed in grape leaves into which they could blend. This was, however, a new situation for all involved. There is no indication in the story that Adam and Eve had proven themselves trustworthy on other occasions nor was there anyone of whom to inquire into their reputation. Furthermore, not having eaten of the Tree, Adam and Eve were ignorant of Good and Evil. How then could they be expected to make an informed decision regarding the temptation offered by the snake? And finally, did they understand the potential consequences of not complying with the policy? Did they have a concept of death? Did they understand that the Garden of Eden was a special place, in which they had a stake? Had God warned them adequately of the potential threat to their interest in the Garden?

Service organizations, of the type described here specifically evaluate these four areas before concluding that an acceptable degree of trust has been established. First they verify an individual's identity. Physicians and peace officers are fingerprinted as part of the process of licensing or recruitment. Social Security Numbers are checked; national databases are consulted. Second, the individual's capability is assessed. This is often done indirectly by equating training with knowledge and knowledge of the field in which they will work with capability. Therefore training is verified. Physicians must produce evidence

that have completed accredited schools and programs. Recognizing the gap between documentary evidence of training and substantive capability, most organizations supervise new staff members during a probationary period as an additional validation of their capability. Third, inquiries are made into an individual's reputation. This may involve letters of recommendation, phone calls or formal background investigations. Lastly, some organizations attempt to verify that an individual has a sufficient and appropriate stake in the outcome of the organization that they can be expected to contribute to that outcome. This may involve requiring statements of outside employment, disclosure of investments in competing organizations, etc.

Trust assessments are not static. Each potential action or policy invocation must be judged in context. There may be different levels of trust, based on differences in knowledge, reputation or stake of the actors relative to the specific policies that are involved. A common example of this is when judges or legislators excuse themselves from a case or proceedings on the basis that they have a conflict of interest based on the specifics of the case. The fact that they voluntarily take such an action may actually increase their reputation in other settings.

In a socio-technical work environment, the granularity of trust assessments is not as uniform as it is in traditional access-controlled systems. It extends both to a lower level within computer systems and beyond the boundaries of systems to other areas of the enterprise that are, nevertheless governed by policy. The trust-related decision may not be whether to grant access or permit action in general but whether to allow it on a case-by-case basis depending on specific circumstances. The actor, the actor's intended actions, the subject, and the nature of the relationship between them may lead to a decision to block or permit an instance of an action without necessarily affecting the ability of the same actor to perform the same action under other circumstances. In the extreme, there may be circumstances where trust in an individual has been significantly damaged but the organization's human resources policies or other exigencies nevertheless require that the individual in question be given a work assignment. It is commonplace in work situations for individuals to be given assignments that they may not be fully trusted to perform because there is no other option. In practice, each one of these situations is resolved whether emotionally or quantitatively (using some combination of boolean or fuzzy logic).

These examples suggest that different patterns of trust will emerge depending on which factors have the greatest influence on the outcome probability. In highly sensitive and restrictive situations, identity will figure prominently. In settings where authenticated individuals are working semi-autonomously, the emphasis may be on knowledge and stake, whereas when making purchase decisions the major factors may be capability/knowledge and reputation.

4.2 Policy

In service organizations, work typically consists of three classes of assignments or duties and is governed by two classes of policy. Some assignments are explicit, some are implicit to the nature of the work as defined by the training and preparation of the worker and some are spontaneously and self-generated by the worker. Some policies are proscriptive, intended to prevent or restrict action while others are prescriptive, intended to influence action and achieve specific results or outcome.

Few policies exist in isolation. Policies tend to occur in swarms that surround specific, sensitive or critical areas of an organization. Some topics are sensitive because they relate to highly confidential subjects, some because there are severe fiscal or legal consequences associated with policy failure and others are sensitive because they affect the quality of the work product and thereby the reputation and bottom-line of the organization. It is common for an assignment to encompass a number of steps or processes, each of which is subject to different policies, some prescriptive, some proscriptive and some potentially undefined.

The policies that are related to a specific outcome form a network; each affected by and affecting others. Since an organization has multiple goals there are multiple policy networks. The networks are not totally independent because there are certain policies and trust patterns that appear in and influence more than one network.

A thoroughly specified policy should allow the intended user to make unambiguous decisions that the framers of the policy would consider to be consistent with their original intent. It should contain a number of elements: 1) the precedence of this policy relative to others in the same network, 2) a list of policies that may or must effect the policy in question and whether the effect is triggering, inhibiting or facilitating; 3) a list of other trigger events that may cause the policy to be invoked; 4) a list of preconditions that must be fulfilled and without which the invocation could not complete including the trust parameters required of any individuals who are to involved with each invocation instance; 5) a declaration of informational elements that will be required by the body of the policy and any retrieval instructions needed to retrieve or acquire that information; 6) the body of the policy which specifies the logic necessary to decide what is to be done under the circumstances of invocation; and the specific actions (including the triggering of other policies) that are to be taken as a result of evaluating the logic; and 7) conditional (fuzzy) logic that uses trust parameters and the state of a policy definition to generate a measure of the contribution that it will make to the outcome of the policy network. This level of specification goes far beyond the mere adoption of a standard page layout for the policy manual and provides detailed guidance about the content of each policy.

The required information lends itself to a formal representation as a frame or loosely structured policy definition (PD) which contains information bearing slots or elements. PDs can then serve as prototypes for generating executable policy components (EPC). A similar construct, called the Medical Logic Module (MLM) [7] has been incorporated into several healthcare information systems. MLMs have been used in a very focused manner to monitor clinical events and changes in a patient care database. As dictated by their logic, the system either alerts practitioners that circumstances require their attention or initiates the specified actions directly such as ordering a repeat test or scheduling a patient for a follow-up examination. The frame or document oriented policy definition approach, with its requirement for a quantitative description of the decision logic, leads to less ambiguity and easier maintainability even if only used as a format for printed policies

One characteristic of policies that are difficult to implement is that they address multiple situations and inadequately specify the details necessary to guide decision-making. Treating each policy as a group of PDs narrows the focus. In effect, a single broad policy is replaced by a policy network of PDs, each with clearly defined decision logic, sequence of invocation and interconnection to other PDs. Using a healthcare example, Dr. Jones is due for her biannual reappointment to the hospital staff. Her application requests renewal of her privileges to perform " Craniotomy with elevation of bone flap; for transection of corpus callosum" (a form of brain surgery). The broad, traditional policy simply states that the department chairperson must approve renewal of surgical privileges. A corresponding policy network states that before privileges can be renewed, in addition (and prior) to departmental approval, the hospital must have documentary evidence that the physician has performed a specific minimum number of procedures of this type during the past two years and achieved a complication rate below a specified level. In the absence of this evidence of qualification, the privilege will only be granted provisionally and the next specified number of cases must be proctored by another member of the department who has unrestricted privileges to perform the procedure in question. Furthermore, any request by Dr. Jones to schedule a case of this type must include the name of the proctor (whose availability will be verified) and the case will not be allowed to start if the proctor is not present.

The simplest forms of PD exhibit patterns of activity that are analogous to digital logic circuits. One can therefore expect to identify PDs that function as inverters (NEG) and gates (AND, OR, NAND, NOR, XOR). Unlike their digital counterparts, a single PD can encapsulate the logic that could only be realized with an entire array of gates. Additional, more complex patterns are possible: 1) multiple outputs, 2) collateral inputs that modulate conditional logic, 4) conditional logic to address absent or ambiguous inputs, and 4) persistent PDs that alter their subsequent behavior based on state information.

Once invoked, a policy network may be active over a prolonged span of time dictated by the nature of the underlying events. Furthermore, the conditional aspects of individual PDs may be influenced by concurrent instances of the same network or PD. For these reasons, each policy invocation may require the ability to maintain persistence of state at the level of the policy network.

4.3 Outcome Probability

The motivation for creating policy is the desire to obtain a specific desired outcome from an activity or in response to a stimulus. Since most objectives involve a policy network and an invocation of the policy may span a period of time, it would be useful to have a periodic estimate of the eventual outcome probability and the probability that a resolution will actually reached.

The process of developing and implementing policy can be separated into three phases: planning, invocation and evaluation. Each must be guided by the concept of outcome probability. There is little value in planning a policy without some objective evidence that: 1) the policy network will accomplish its objective expressed as a prediction of outcome probability, 2) there is at least one complete pathway through the network, and 3) the network is free from non-terminating, cyclic pathways. The policy invocation mechanism should include monitors that feed the conditional logic within an EPC. The objective is to alter the current estimate of outcome probability, perhaps favoring early termination of the instance. Lastly, there must be a *post hoc* determination of actual outcome for comparison to initial predicted outcome both to assess organizational performance and to facilitate the iterative improvement of the policy.

Policies that are well designed are likely to produce the desired outcome if executed. The existence of such policies is not sufficient. The individuals or systems involved in the activity must be aware of the fact that applicable policies exist before they are in a position to be influenced by them. The weak link in most policy environments, whether sociological or computer-based, is recognizing that an event or situation is occurring/has occurred for which there are one or more applicable policies. This shortcoming automatically lowers the outcome probability especially if the policy is invoked infrequently or there are overlapping and conflicting policy networks or PDs. In constrained settings where there are well defined information structures and the bulk of the information that falls within the decision space is in an electronic or machine readable format it may be possible to accomplish the detection function. The task is essentially impossible in the absence of specialized surveillance systems or in cases where much of the information is not in electronic form. Examples of effective surveillance systems, though not targeted specifically at organizational policy are the healthcare systems that implement Medical Logic Modules (see above) and the systems employed in the intelligence community to scan news services looking for specific events or patterns of activity.

Exceptions and ambiguity have the potential to reduce the likelihood of a good outcome. Circumstances will certainly arise that are not addressed by the PD's. A well-designed policy should identify and classify potential sources of exceptions and provide mechanisms to gracefully interrupt the execution of the policy when it becomes necessary. Each policy should also provide a fail-safe pathway to be invoked if the level of ambiguity reaches unanticipated levels. It is not the intent of this formulation to suggest that people not be allowed to exercise judgement and override the dictates of a policy, rather that policies which explicitly address potential sources of conflict are better than policies that don't. In the case of machine execution, it especially important to identify those circumstances under which a fail-safe mechanism should be invoked or in which the final decision should be reserved for a person.

5. Conceptual Model

5.1 Modeling Trust

In the future organizations and systems will be challenged to explain the rationale behind their decisions to permit or deny various activities. The purpose of modeling trust is to provide a vocabulary to describe the elements that make up a trust decision and to animate the decision making process.

In this formulation, trust is a function of context, identity, reputation, capability and stake. Trust is also conditioned by social and cultural factors; in certain cultures tradition may provide a strong influence. The model presented here identifies the major elements that contribute to an evaluation of trustworthiness. In some settings, based on the needs of the

evaluators, one element may predominate. For example, it may be possible to permanently damage a reputation.

Each setting in which trust plays a role has a default behavior (whether explicitly recognized or not). In a computer system the default behavior, if trust cannot be established, may be to prevent access. In a socio-technical workplace, the default behavior may be that every employee has unimpeded ability to act. Regardless of how simple or complex the decision making process, trust always comes reduces to a yes/no answer. It is up to the implementation to decide what type of default behavior will occur in the event that all of the required element factors are not available or quantifiable.

This level of detail is unnecessary if clear grounds can be established for denying access completely. The formulation therefore assumes that trust evaluations will be performed frequently, perhaps each time access is requested, rather than infrequently. The elements of a trust pattern are:

Element	Notation
A context for the activity	$c \in C$
A subject or object (entity) that is being evaluated for trustworthiness	$e \in E$
The activity in which E will engage	$a \in A$
The subjects, objects, property, resources or assets that will be affected by A	$d \in D$
The valuation of D	$v = V(d, c)$
The benefits or risks that attends A	$\{b_1, \ldots, b_n\} \in \beta$ $B = \{b_{1(a,v,e)}, \ldots, b_{n(a,v,e)}\}$
The capability or knowledge that E has about A and D	$K(e, a, d)$
The reputation of E with respect to A or in general	$R = \{r_{1(e)}, \ldots, r_{n(e)}\}$
The certainty that the true identity of E is known	$I(e)$
The stake that E has in C and D	$S(e, c, d)$
The trust assessment	T

The component sets are left unspecified. A subject or object may appear as a member of the entity set E or as a member of the affected resource set D. Activities are those actions that consume resources or have the potential to permanently alter the state of the organization and are therefore subject to policy considerations. Activities can also be actions that must occur in order to preserve or protect resources. Valuation is a quantitative or qualitative assessment of monetary value, good will, repair cost, etc. Benefits assess the cost (either positive or negative) associated with performing the activity and is different from the valuation of the resources involved. The context is the particular association of the entity(s) who will engage in a specific instance of an activity. Reputation attempts to take into account the effect that the entity has had on others. In addition, the total contribution that reputation makes to the trust equation is composed from individual components that may reflect either historical information about past performance or specific information need to asses reputation in the current context. This aspect of the model allows an implementation to address asymmetry between the effects that positive and negative historical information has on a specific evaluation of trust. The degree of certainty with which the identity of the entity(s) is known must be determined since it will condition the interpretation of the other factors. Stake is the degree to which the entity(s) proposing to engage in the activity has a vested interest in the outcome. Capability is measurable expertise that the entity possesses about the activity and/or demonstrable access to the resources and authority necessary to act. Capability may be demonstrated or verified in ways that contribute to establishing reputation and identity as well.

The general forms of the resulting trust equation are:

1) $T = f(B, S_{(e,c,d)}, K_{(e,a,d)}, R_{(e)}, I_{(e)})$

2) $T' = f(S_{(e,c,d)}, K_{(e,a,d)}, R_{(e)}, I_{(e)})$

$T = T' \bullet B$

5.2 Modeling Policy

The purpose of modeling policy is to establish a representation of a policy that could 1) drive an implementation and show that the implementation upholds the policy and/or 2) facilitate a debate between stakeholders about what the policy is or should be. It may also be possible to use the model to demonstrate internal contradictions in a policy or the nature of a conflict between competing policies [8]. It is hypothesized that the notation suggested below can accommodate both implementation (within computer systems) and debate and implementation at the sociological level.

Policy execution may begin because a policy-triggering external event has occurred or because a currently executing policy instance has generated a side-effect. The following discussion is hypothetical but is expressed in terms that might be used to refer to an actual implementation. The metaphors used are chosen specifically because they represent activities and functions that could be implemented either as a computer system or as a sociological system. In the following discussion the term policy coordinator object will be used. While the language clearly suggests a machine environment, many organizations assign people to act as policy coordinators. For example, there is a large hotel chain that has instructed its staff that whoever receives a customer complaint or request "owns" that issue and is responsible for mustering the necessary organizational resources to address the customer's need.

Policies initially exist as a cluster of related PDs. When triggered, a policy coordinator object (PCO) and at least one EPC (executable policy component) must be created. EPCs contain or point to conditional logic, provide storage for any instance variables that may be declared, and provide persistence, if necessary. In the limiting case, policy networks are virtual, defined by the references that the related EPCs may make to each other. These references can be thought of as hyperlinks if

they are to already instantiated EPCs but have the effect of causing the instantiation and invocation of a EPC if it does not yet exist. There may be specific benefits, either for policy modeling and testing, or for certain schemes of policy execution, to actually instantiate an entire network of EPCs.

The model of coordination uses the paradigm of a blackboard system in which there is a shared resource to which information can be posted and from which information can be read or removed. This model is chosen for the following reasons: 1) it can be implemented using people or machines as the execution vehicle, 2) it lends itself to managing and solving complex, multifactorial problems in which time may be a limiting factor and in which it is important to know when an incomplete solution may be complete enough to take action [9], and 3) the nature of the computations and the need for speed and high capacity favor an approach in which parallel computing can be applied. Carriero and Gelernter [10] described a parallel programming language called Linda that is uniquely suited to the type of blackboard manipulations* needed to execute and track policies.

A diagrammatic representation of a PD, drawn with an SGML tool†, appears in Figure 1. The corresponding SGML definition appears in Appendix 1. This could easily be replaced with a different notational convention such as ASN.1. Using the document oriented format makes this representation equally useful as a specification for a data structure and as a format for creating more quantitative, human readable policy manuals. An example of a Medical Logic Module expressed in Arden Syntax [4] is shown in Appendix 2 to demonstrated the feasibility of creating a single data structure that is readable by both humans and machines. Using SGML as the representation offers the opportunity to use a wide variety of COTS products to create, manipulate and publish policy definitions. A similar representation of the possible structure of a Policy Control Object is shown in Figure 2.

In order to use this approach in a blackboard implementation, additional data structures would be necessary and would most likely include request objects, alert objects, condition and situation monitors and message controllers.

6. Conclusions and Future Work

The motivation for this paper was the observation by the author that large healthcare and other service organizations are being rendered non-functional by a never-ending proliferation of policies that fail to achieve their desired results. The typical reaction to policy failure is the endless revision of existing policies and the creation of more policies aimed at correcting the situation. When the systems that are afflicted with failed policy are sociological systems, those failures, although irritating, are rarely life threatening either to individuals or to the organization. When the people who are supposed to carry out non-working policies are confronted by ambiguous or conflicting sets of instructions, they do what people do best - they survey the situation, make a decision and live with the consequences. Sometimes they decide incorrectly and there is a price to be paid: embarrassment, reprimand, law suits, financial losses. More often the decisions are acceptable, if not optimal.

The same is not true when the policies are enforced with dogmatic rigidity or when they affect things that go on inside computer systems. First of all, a policy that may be specified in sufficient detail for an experienced person to understand cannot be comprehended by a data system. The steps must be small, more finite and there must be a clear definition (even if it uses fuzzy logic) about what to do at each step of the process.

In order to create an automated policy execution environment that provides some semblance of flexibility, it is necessary to:

1) Provide more detailed specifications of policy than are common now
2) Express that detail using relatively simple data structures that can be combined in recursive and hyperlinked patterns and
3) Provide inhibitory and facilitory collateral inputs, in the form of trustworthiness assessments, benefit assessments and fuzzy weighting factors, at each node within the policy network.

Using the model presented here it may be easier to isolate the causes of policy failure. The overall policy or its individual steps may be ambiguous - the decision logic is absent or inconsistent. The analysis of factors such as trustworthiness, value, and benefit may be absent or insufficiently detailed. The policy may lack defined endpoints. There may endless loops, unanticipated points where execution terminates abnormally, non-terminating or unresolved pathways or multiple and unpredictable outcomes even when the starting conditions are similar. All of these policy failures occur, even in sociological systems. The impact is more immediate and potentially greater if the failure occurs in an automated policy execution environment. Appropriate access may be denied, inappropriate access granted and actions initiated all of which conflict with the original intent of the developers and users of the policy.

Awareness is an important defense against policy failure. In order to reduce the frequency of conflicting or silent policies, organizations should attempt to document all policies whether sociological or computerized using a common notation. This documentation should be cataloged and readily accessible throughout an organization (with appropriate access controls - of course) and should ideally be machine readable, executable and available to any automated systems intended to monitor or control policy invocation. Repeated assessment of outcome probability is the best defense against policy failure. This should be done by simulation during the initial formulation and development of the policy and every organization that relies heavily on policy should develop facilities to monitor individual instances of policy invocation for effectiveness, whether on a concurrent basis or statistically. Secondly, the policy execution environment should attempt to periodically assess the probability that active instances of a policy have not deviated from the initial target outcome to the extent that their execution should be interrupted or evaluated.

---

* There is no indication that these authors have mentioned the term blackboard in conjunction with any discussion about Linda or parallel computing or have drawn any association between blackboard systems and parallel computing.

† Near & Far Designer, Microstar Software, Inc.

The final defense against policy failure is common sense. Today people lack the means to convince themselves that certain policies are not worth the effort even though their instincts tell them so. The material presented here provides a vocabulary that can be used to discuss policy in more detailed and quantitative manner than is familiar to service organizations of the type described. As an organization gains familiarity with the new techniques of policy formulation, analysis and testing they will observe that certain patterns [11] emerge that are characteristic of their business and socio-technical environment. Their familiarity with these patterns will make it possible to assemble policies that work with less effort. Given a better means to discuss policy objectives, the means of realizing them and mechanism to assess the likelihood of success - organizations may be able to reduce the number of policies that they attempt to implement to those that have a chance of working. This will intrinsically improve outcome by allowing the available resources to be concentrated on a smaller number of tasks.

Future work will focus on creating a Policy Workbench [12] equipped with document management and discrete event simulation tools to provide a platform on which the policies can be test run and the assumptions of outcome probability validated. Referring back to the three purposes for modeling policy:

1) To drive an implementation
2) To facilitate debate and
3) To demonstrate internal conflicts within and between policies

It remains, as an open research issue, to determine if the data structure proposed here to represent policy definitions will appropriately facilitate discussion between participants in the policy formulation process, or if it is overly formal for that purpose. It has been suggested that the later purpose may need to be cast in terms of a language that supports formal reasoning [13]. In the healthcare environment, the most immediate applicability appears to be in the area of facilitating debate about the details of policies and helping to ensure that that are formulated and documented in a way that can promote understanding and compliance. An implementation attempt would make a excellent research and development project. Since most healthcare organizations use proprietary systems that are commercially obtained, a policy machine would undoubtedly need to be implemented as a server. It would need to accept requests and provide responses using protocols and messaging standards that are already in place such as HL7 [14], a healthcare standard that has already identified the possibility of incorporating an external "Rule Engine" in an overall system architecture. Once implemented as a proof of concept demonstration healthcare organizations would then be in a better position to evaluate the utility of adding such a server to their computing environment.

Acknowledgements

The author extends thanks to the anonymous reviewers for their constructive criticism of the original manuscript. Their input is greatly appreciated.

References

[1] Genesis 1:1 (Biblical references are from The Shocken Bible: Volume I, The Five Books of Moses, a new translation with introductions, commentary and notes by Everett Fox, 1995.) This translation was prepared directly from Hebrew and Aramaic texts and yields a translation that differs from the more familiar translations such as the King James Version.

[2] Genesis 2:17, ibid.

[3] National Research Council: Computers at Risk: Safe Computing in the Information Age. National Academy of Sciences, 1991.

[4] Stepney, S and Lord, SP: Formal specification of an access control system. *Software—Practice and Experience*, 17(9):575-593, September 1987.

[5] Lampson B, Abadi M, Burrows M, Wobber E: Authentication in distributed systems: theory and practice. *ACM Transactions on Computer Systems*, Vol. 10, No. 4 (Nov 1992), Pages 265-310.

[6] Hauser, R: Does Licensing Require New Access Control Techniques. *Commun ACM*, Vol. 37, No. 11 (Nov 1994), Pages 48-55.

[7] ASTM E 1460-92: Standard Specification for Defining and Sharing Modular Health Knowledge Bases (Arden Syntax for Medical Logic Modules), ASTM Committee on Standards, latest revision 2/95.

[8] Suggested by one of the anonymous reviewers.

[9] Englemore, R and Morgan, T: Blackboard Systems, Addison Wesley, 1988.

[10] Carriero, N and Gelernter, D: Linda in context. *Commun ACM*, Vol. 32, No. 4 (April 1989), Pages 444-459.

[11] Alexander, C: A Timeless Way of Building, Oxford Press, 1979.

[12] Sibley EH, Michael JB, and Wexelblat RL: "Use of an Experimental Policy Workbench: Description and Preliminary Results" in Landwehr CE and Jajodia S (eds.): Database Security, V: Status and Prospects. Results of the IFIP WG 11.3 Workshop on Database Security, Shepardstown West Virginia, USA, 4-7 November, 1997. North-Holland, 1992, IFIP Transactions A-6, ISBN 0-444-89518-3.

[13] Suggested by one of the anonymous reviewers.

[14] HL7, Version 2.3 Standard, Health Level Seven, 3300 Washtenaw Avenue, Suite 227, Ann Arbor, MI 48104-4250

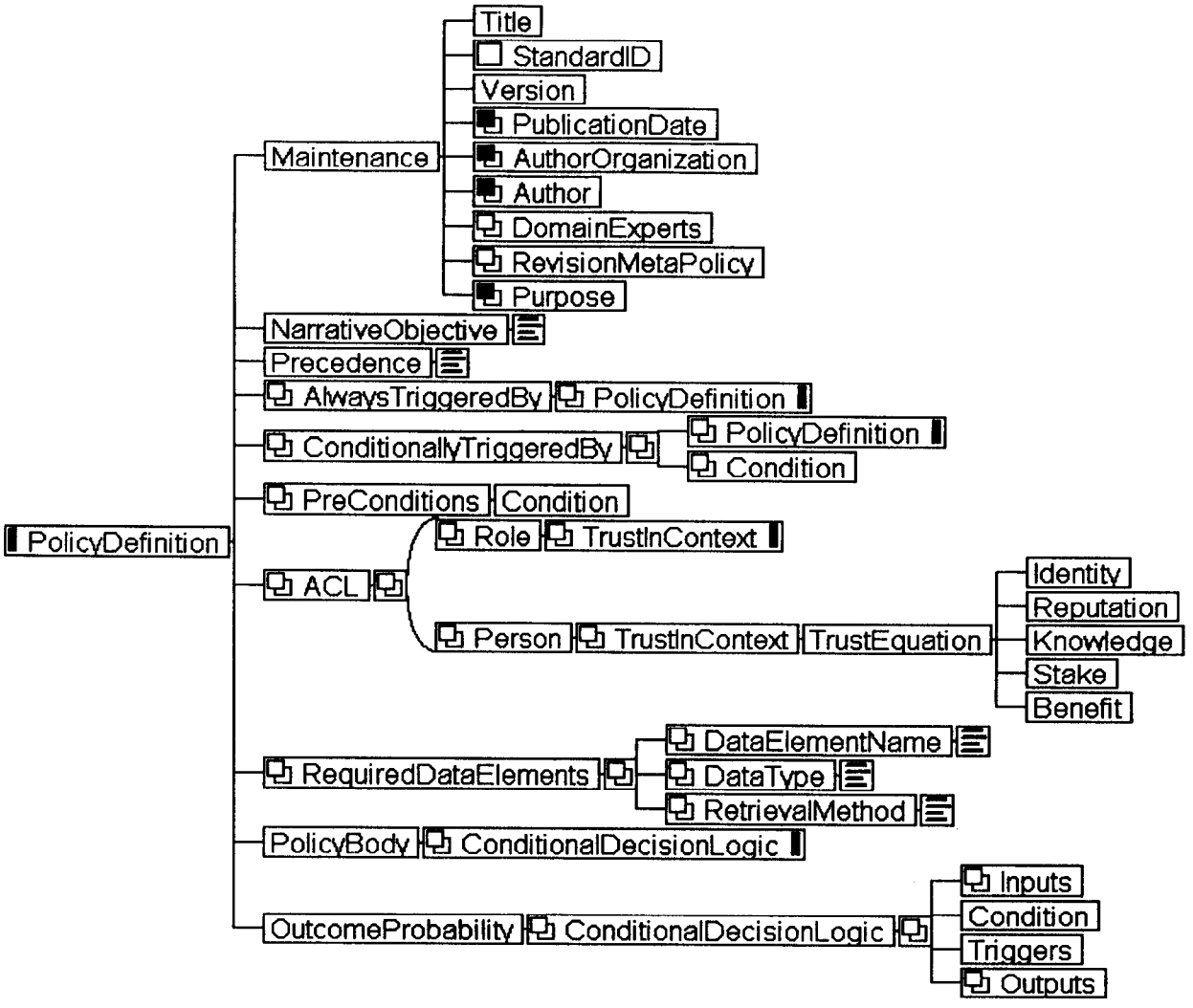

Figure 1 - Example of a Policy Definition Data Structure

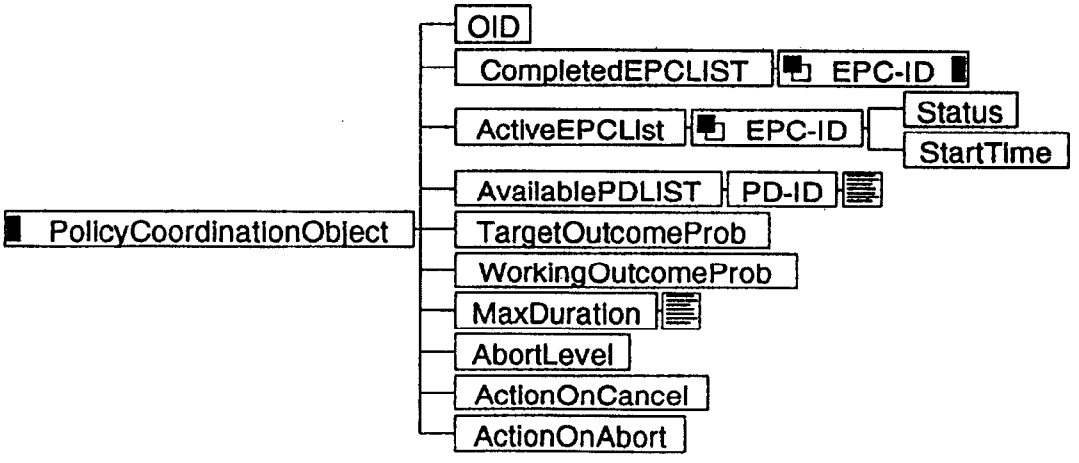

Figure 2 - Example of a Policy Control Object Data Structure

Appendix 1 - Example Policy Definition (Structure only) in SGML

```
<!--<Title>Policy Definiton-- >
<!ELEMENT PolicyDefinition - - (Maintenance, NarrativeObjective, Precedence, AlwaysTriggeredBy*,
        ConditionallyTriggeredBy*, PreConditions*, L*,  RequiredDataElements*, PolicyBody,
        OutcomeProbability) --<Title>NFDoc-- >
<!ELEMENT ConditionallyTriggeredBy - - (PolicyDefinition*, Condition*)*  --
        <Title>ConditionallyTriggeredBy-- >
<!ELEMENT AlwaysTriggeredBy  - - (PolicyDefinition*) --<Title>AlwaysTriggeredBy-- >
<!ELEMENT Precedence  - - (#PCDATA) --<Title>Precedence-- >
<!ELEMENT PreConditions  - - (Condition) --<Title>PreConditions-- >
<!ELEMENT ACL  - - (Role* & Person*)* --<Title>ACL-- >
<!ELEMENT TrustEquation  - - (Identity, Reputation, Knowledge, Stake, Benefit) --<Title>TrustEquation--
        >
<!ELEMENT TrustInContext  - - (TrustEquation) --<Title>TrustInContext-- >
<!ELEMENT Person  - - (TrustInContext*) --<Title>Person-- >
<!ELEMENT Role  - - (TrustInContext*) --<Title>Role-- >
<!ELEMENT RequiredDataElements  - - (DataElementName*, DataType*, RetrievalMethod*)* --
        <Title>RequiredDataElements-- >
<!ELEMENT DataType  - - (#PCDATA) --<Title>DataType-- >
<!ELEMENT RetrievalMethod  - - (#PCDATA) --<Title>RetrievalMethod-- >
<!ELEMENT DataElementName  - - (#PCDATA) --<Title>DataElement-- >
<!ELEMENT ConditionalDecisionLogic  - - (Inputs*, Condition, Triggers, Outputs*)*  --
        <Title>ConditionalDecisionLogin-- >
<!ELEMENT Condition  - - (#PCDATA) --<Title>Condition-- >
<!ELEMENT Outputs  - - (#PCDATA) --<Title>Outputs-- >
<!ELEMENT Inputs  - - (#PCDATA) --<Title>Inputs-- >
<!ELEMENT PolicyBody  - - (ConditionalDecisionLogic*) --<Title>PolicyBody-- >
<!ELEMENT OutcomeProbability  - - (ConditionalDecisionLogic*) --<Title>OutcomeProbability-- >
<!ELEMENT Benefit  - - (#PCDATA) --<Title>Benefit-- >
<!ELEMENT Stake  - - (#PCDATA) --<Title>Stake-- >
<!ELEMENT Knowledge  - - (#PCDATA) --<Title>Knowledge-- >
<!ELEMENT Reputation  - - (#PCDATA) --<Title>Reputation-- >
<!ELEMENT Identity  - - (#PCDATA) --<Title>Identity-- >
<!ELEMENT NarrativeObjective  - - (#PCDATA) --<Title>NarrativeObjective-- >
<!ELEMENT Triggers  - - (PolicyDefinition) --<Title>Triggers-- >
<!ELEMENT Maintenance  - - (Title, StandardID?, Version,  PublicationDate+, AuthorOrganization+,
        Author+,  DomainExperts*, RevisionMetaPolicy*, Purpose+) --<Title>Maintenance-- >
<!ELEMENT RevisionMetaPolicy  - - (PolicyDefinition) --<Title>RevisionMetaPolicy-- >
<!ELEMENT Purpose  - - (#PCDATA) --<Title>Purpose-- >
<!ELEMENT PublicationDate  - - (#PCDATA) --<Title>PublicationDate-- >
<!ELEMENT DomainExperts  - - (#PCDATA) --<Title>DomainExperts-- >
<!ELEMENT Author  - - (#PCDATA) --<Title>Author-- >
<!ELEMENT AuthorOrganization  - - (#PCDATA) --<Title>AuthorOrganization-- >
<!ELEMENT Version  - - (#PCDATA) --<Title>Version-- >
<!ELEMENT StandardID  - - (#PCDATA) --<Title>StandardID-- >
<!ELEMENT Title  - - (#PCDATA) --<Title>Title-- >
```

Note: #PCDATA stands for processable character data. In this model this data-type provides a container to hold data, lists and conditional logic, etc. As the model is refined, applications that use this structure will map components of the #PCDATA into application-specific complex data-types.

Appendix 2 - Example Policy Definition in Arden Syntax

X1.10 *MLM Translated from HELP:* [4]

```
maintenance:
        title: Ampicillin for Pneumonia (HELP p. 81);;
        mlmname: help_amp_for_pneumonia;;
        arden: ASTM-E1460-1995;;
        version: 1.00;;
        institution: LDS Hospital;;
        author: Peter Haug, M.D.; George Hripcsak, M.D.;;
        specialist: ;;
        date: 1991-05-28;;
        validation: testing;;  library:
        purpose:
        Recommend the use of ampicillin for pneumonia.;;
explanation:  If the patient has pneumonia, then suggest treatment with ampicillin
              unless there is a penicillin allergy.;;
        keywords:
        pneumonia; penicillin; ampicillin;;
        citations:
        1. HELP Frame Manual, version 1.6. LDS Hospital, August 1989, p. 81.;;
capability:
        type: data-driven;;
        data:
        let diagnosis_storage be event {STORAGE OF DIAGNOSIS};
        let penicillin_allergic_reaction be read {PENICILLIN_ALLERGIC_REACTION};
        let penicillamine_allergic_reaction be read
              {PENICILLAMINE_ALLERGIC_REACTION};
        let penicillin_allergy be penicillin_allergic_reaction merge
              penicillamine_allergic_reaction;
        let (diagnosis,status) be read last {DIAGNOSIS,STATUS where status =
              "active"};
        ;;
evoke: diagnosis_storage;;
logic:
        if any (diagnosis = "pneumonia")
        and no (penicillin_allergy is present) then let dosage be 1000;
              conclude true;
        else
              conclude false;
        endif;;
action:
        write "Suggest initial treatment of pneumonia with Ampicillin," ||dosage|| "mg
              IV qid.";
end:
```