

# A Policy Framework for Security and Privacy Management

John Karat<sup>1</sup>, Clare-Marie Karat<sup>1</sup>, Elisa Bertino<sup>2</sup>, Ninghui Li<sup>2</sup>, Qun Ni<sup>2</sup>, Carolyn Brodie<sup>1</sup>, Jorge Lobo<sup>1</sup>, Seraphin Calo<sup>1</sup>, Lorrie Cranor<sup>3</sup>, Ponnurangam Kumaraguru<sup>3</sup>, and Robert Reeder<sup>3</sup>

IBM TJ Watson Research<sup>1</sup>, Purdue University<sup>2</sup>, Carnegie Mellon University<sup>3</sup>

---

## 1.0 Introduction

This paper describes the notion of end-to-end policy management and advances a framework which can be useful in understanding the commonality in information technology security and privacy policy management. Policies which address security and privacy are pervasive parts of both technical and social systems. In IT, there is a general notion of policy-based systems as those whose behavior is guided by rules of the general form “If *condition* then *action*.” Collections of rules are considered policies, and policies can be developed for various aspects of system behavior. In social systems, organizations have policies covering proper conduct to protect the safety of people and effective use of resources. Information technology systems have policies which govern who can access what resources aimed at protecting the integrity and confidentiality of the information and resources. Individuals have policies guiding their behavior towards others formed with the intention of guiding how they live their lives. These policies might be expressed in text (common in organizations), code (common for IT systems), or might be implicit (common for individuals). Such policies might be seen as including high level guidance (e.g., “to insure a safe workplace”) and more specific operational rules (e.g., “don’t run with scissors”).

We see several aspects common across a wide range of policy types in technical and social systems. First, high level policies – generally expressed in human language – are refined into operational rules while attempting to keep the intent of the high level policy. This process is difficult – often subject to differences in interpretation or context. Second, the existence of multiple, possibly conflicting, policies must be accommodated. This process is also difficult, as comparison across policies requires detailed understanding of the meaning of each policy rule and is rarely straightforward. For human or technology systems there is a resulting gap – sometimes referred to as the *gulf of execution*<sup>1</sup> – between human intentions and technology capabilities. We believe developing approaches to closing the gulf of execution would be valuable in many domains. For example, most organizations store sensitive business and personal data in heterogeneous server systems. They do not have a unified way of defining or implementing security and privacy policies regarding the storage and use of that data throughout their organization. Changing legal requirements, social pressures and technologies are making these issues increasingly critical to organizations and society at large.

While we are interested in policies in many areas, the research reported on here focuses on security and privacy policies. There are several reasons for selecting a focus on security and privacy. First, there are a growing number of strict security and privacy audit and compliance requirements for healthcare, banking/finance, and government. This creates a practical need for improving the management of such information through policy-based systems. Second, there is considerable similarity between some aspects of privacy policies and some aspects of security policies. Specifically, rules for a major component of security policies, namely, access control rules defining who can have access to what resources, are nearly identical to generally accepted rules which are used in the formation of privacy policies<sup>2</sup>. This similarity in rule structure leads us to focus on bringing together research rooted in security policy analysis with research in privacy policy authoring and implementation. Third, privacy – when viewed as appropriate use of information - relies on the security of that information in a system, and it is difficult to talk about privacy without considering security. Our principal research objective is to create an integrated privacy and security policy management framework which builds on the commonalities between the two and encompasses end-to-end solutions for use across heterogeneous configurations covering all data. This includes mechanisms and tools for supporting policy authoring, analysis, enforcement, and auditing.

In this paper, we outline our directions and our progress to date, along with remaining research challenges in developing a framework for policy management. The framework helps us to harmonize work across the range of activities related to the creation and analysis of policies. The term “policy” will be used here in a number of contexts with different levels of specification. We will describe how we view security and privacy below, as well as addressing how we see policy as relevant to security and privacy systems (Section 2). We then describe the different levels and the refinements which take place between the levels in our framework (Section 3). Policy can refer to high-level human-understandable statements of an intent expressed in natural language (e.g., “Our policy is that only people over 17 can be admitted” or “Only authorized physicians will have access to a patient’s sensitive medical information”), but we will focus on the refinement of such policies into rules and sets of rules which can be implemented in computer systems. In covering this range of abstraction, we will sometimes refer to a single rule as a policy, and sometimes refer to a collection of rules as constituting a policy. In general, policies can have different structure and syntax depending on the domain (e.g., security access control, privacy, networking, systems management, business operations) the policy applies to. The framework is presented with examples of policies involved in security access control and privacy (Section 4). Our intention is to provide a framework which is comprehensive enough to enable collaboration among researchers working on a range of topics related to policy development and implementation.

## **2.0 Relating Security and Privacy**

The rapid advancement of the use of information technology in industry, government, and academia makes it much easier to collect, transfer, and store personal information (PI) around the world. This raises challenging questions and problems regarding the protection and use of PI<sup>3</sup>. Questions of who has what rights to information about individuals for what purposes become more important as we move toward a world in which it is technically possible to know just about

anything about just about anyone. As stated by Adams and Sasse<sup>4</sup>: “Most invasions of privacy are not intentional but due to designers’ inability to anticipate how this data could be used, by whom, and how this might affect users.” Deciding how we are to design privacy considerations in technology for the future includes philosophical, legal, and practical dimensions – any or all of which can be considered as within the domain of the field of human-computer interaction (HCI). We see considerable overlap between issues related to access to resources and issues related to appropriate use of resources. We see the former as being primarily in the domain of IT security (access control), and the latter as being primarily in the domain of IT privacy.

Security in information technology refers to many aspects of protecting a system from unauthorized use (e.g., authentication of users, information encryption, firewall policies and intrusion detection). For our purposes here, we will limit our treatment of security to the concepts associated with how well a system can protect access to information it contains (i.e., information security). While there are many other factors associated with security in IT systems, we think that focusing on issues of policy management in access control represents an important and significant part of the overall security research agenda.

The concept of Privacy goes beyond security to examine how well the use of information that the system acquires about a user conforms to the explicit or implicit assumptions regarding that use associated with the personal information. There is an important distinction that we would like to draw when discussing privacy from an information technology view. In general, research involving privacy and IT has looked at the intersection from two different perspectives. From an end user perspective, privacy can be considered as **preventing storage** of personal information, or it can be viewed as **ensuring appropriate use** of personal information. In the former, the user expresses a “wish to be left alone”; in the latter the wish is to “use my information according to expressed wishes.” Much has been said about the end of privacy in the pervasive computing world in which so much is known about each of us that it is futile to worry about privacy. This view is an attempt to highlight the difficulty people face in trying to remain anonymous (the first consideration above). While electronic surveillance is increasingly common, this does not mean that people have to give up their rights to control the use of information collected about them. While it might be possible to imagine a world in which governments and companies decided not to collect information on people, it is our assumption that collection of more and more data about us is a trend that will continue. However, we do think that increased data collection will be balanced with rules concerning data use. We assume that legislation, industry best practices, and the judicial system will support people’s rights to ensure that appropriate use is made of data collected. In considering privacy, we generally need to assume that security in a system is adequate to protect information against unauthorized use. Thus, we view data protection failures associated with unauthorized access to information to be security failures or breaches, and those associated with non-compliance to stated privacy policies to be privacy breaches. We will explore the privacy and security access control issues in more detail below. For the purposes of this chapter, a simple but useful definition of privacy is:

“The ability of individuals to control the terms under which their personal information is acquired and used<sup>5</sup>.”

In summary, security involves technology to ensure that information is appropriately protected. Security enables and is a required building block for privacy to exist. Privacy involves mechanisms to support compliance with some basic principles<sup>6,2,7</sup> and other explicitly stated policies. Basic principles suggest that people should be informed about information collection, told in advance what will be done with their information, and given a reasonable opportunity to approve of such use of information. Trust in a system is seen as increasing when it is perceived that security and privacy are provided for<sup>8</sup>. Without trust, it is perceived that people will be less likely to use systems.

## **2.1 Privacy Policies in Information Systems**

A view of privacy as “control over use of personal information” leads us to prioritize managing privacy policies (either at an individual or organizational level) over considering privacy as a desire to remain anonymous. We understand that privacy is not entirely about “setting rules and enforcing them”<sup>9</sup>, but do not see a world in which there is less personal information stored in systems as likely. Privacy is an important social issue and technology can enable flexible, reliable and verifiable privacy policy enforcement to preserve individual rights. We agree that technology design generally does reflect concerns of society in general<sup>10</sup>, and believe that we are experiencing a shift toward a greater concern for privacy in IT design.

Organizations commonly provide a description of what kind of information they will collect and how they will use it in privacy policies. In some areas (e.g., the collection and use of health care information in the US or movement of personal information across national boundaries in Europe), such policies are required by legislation, though the content of the policy to address legislative mandates is left open for organizations handling the data to define. Additionally, there are considerable differences in privacy legislation in different regions of the world<sup>11,12</sup>. Similarly, organizations in different fields (e.g., healthcare, banking, government) need to tailor policies to their domains and needs<sup>13,14</sup>. To implement privacy within an organization, the coordination of people, business processes, and technology is required<sup>15</sup>. We believe that such policies are essential when interacting with technology and organizations in that they enable people to better understand the boundaries between public and private information and technology<sup>16</sup>.

It is interesting to note that while privacy policies are not new to most organizations, very little has been done to implement the policies through technology<sup>17</sup>. There are standards for privacy policies on websites<sup>18</sup>, but these address machine readable policy content without specifying how the policy might be created or implemented. The reality is that there is very little capability to have technology actually implement access and disclosure limitations that we might expect from a policy statement like “We will not share your information with a third party without your consent.” The emerging focus is on how organizations could create a wide range of policies, and how technology might enable the policies to be enforced and audited for compliance. Karat et al.<sup>15</sup> focus on technology to enable usable privacy policy authoring and enforcement, rather than trying to directly address what privacy rights people should have<sup>19</sup> or how to de-identify information such as video stored in systems<sup>20</sup>.

Central to this view of privacy is the notion that the parties involved in information exchanges have implicit or explicit policies with regard to the use of the information. This applies both to the person whom the information is about and to the organization collecting and using the information. In the privacy literature on organizations, while some attention has been given to the generally implicit policies of end users whose PI is being collected and used (often called data-subjects), the main focus is on the policies of the organization collecting the information. Smith<sup>17</sup> described such organizational policies, and also noted the lack of technology in enforcing the policies. He described the rather unstructured ways in which organizations develop privacy policy, a characterization that has changed little in the nearly 15 years since his research was published, in spite of the increased legislation of the past few years. Future research must address both the needs of data subjects and organizations by addressing the gap between policy and practice.

## ***2.2 Security Policies in Information Systems***

It is broadly recognized that one of the major challenges to the effective deployment of information security systems is getting people to use them correctly<sup>21</sup>. As far back as the 1970's, usability with specific reference to security mechanisms was identified as a key principle in the design of secure systems<sup>22</sup>. Even beyond the domain of electronic information systems, there are many examples of the fact that overly complex security systems actually reduce effective security. For example, Kahn<sup>23</sup>, cited by Anderson<sup>24</sup>, suggests that Russian military disasters of the Second World War were partly due to the fact that Russian soldiers abandoned the official army cipher systems because they were too hard to use, and instead reverted to simpler systems that proved easier to crack. Scheiner<sup>25</sup> sums up the situation: "Security measures that aren't understood by and agreed to by everyone don't work." However, as with many areas in the design of complex systems, recognizing that there is a potential problem does not necessarily create a rush of work to resolve it. Work on making security usable – and balancing the complex relationship between "secure" and "easy to use" is just beginning to become a major topic. We see contributing to making security policies more understandable as an important element of this challenge.

De Paula et al.<sup>26</sup> suggest that it is important to separate theoretical security (the level of secure communication and computation that is technically feasible) from effective security (the level of security that can practically be achieved in everyday settings). Levels of effective security are almost always lower than those of theoretical security. A number of reasons for this disparity have been identified, including poor implementations of key security algorithms<sup>27</sup>, insecure programming techniques<sup>28,29</sup>, insecure protocol design<sup>30,31</sup>, and inadequate operating systems support<sup>32,33</sup>. We believe that the overall problem is so large, that it must be approached in many directions, and one which has received considerable research attention is policy-based access control.

One important source of the disparity between theoretical and practical security is the extent to which users can comprehend and make effective use of security mechanisms. Approaches that attempt to make the provision of system security "automatic" or "transparent" essentially remove security from the domain of the end-user. However, in situations where only the end user can

determine the appropriate use of information or the necessary levels of security, then this explicit disempowerment becomes problematic.

## 2.3 Similarities and Differences

At the rule level, there are many similarities between privacy policy rules, and security access control rules (as described in EPAL and XACML). As specified in these standards, privacy policies can have up to six elements (Data user, action, data, purpose, conditions and obligations). Access control policy rules include corresponding elements, but do not explicitly identify purpose, and only sometimes allow conditions and obligations.

Conventional access models, such as Mandatory Access Control (MAC), Discretionary Access Control (DAC), and Role Based Access Control (RBAC)<sup>33,34</sup>, are not designed to enforce privacy policies and barely meet privacy protection requirements<sup>35</sup>. Privacy aspects that are particularly lacking in these approaches include purpose binding (i.e. data collected for one purpose should not be used for another purpose without user consent) and conditions and obligations. The significance of *purpose*, *condition*, and *obligation* elements originates from OECD Guidelines (Organisation for Economic Co-operation and Development) on the Protection of Privacy and Transborder Flows of Personal Data, current privacy laws in the United States, and public privacy policies of some well known organizations. The OECD guidelines are the most well known set of private information protection principles, on which many other guidelines, data-protection laws, and public privacy policies are based. The *purpose* element is directly addressed in the OECD Data Quality Principle, Purpose Specification Principle, and Use Limitation Principle. The purpose element is also widely used for specifying privacy rules in legislative acts and actual public policies. The HIPPA (Health Information Privacy Protection Act) legislation clearly requires policy rules to clearly state the purposes for the use of personal health information. The majority of public privacy documents posted at well known sites also specify purposes of data use. It is important to note that security policies can be automatically enforced while privacy policies often rely on external enforcement mechanisms because they have requirements that cannot be determined before access is granted (e.g. that data will be used only for a particular purpose)

There are many situations in which access to private data imposes *obligations*, that is, actions to be performed after an action has been executed on data objects. For example, the OECD Accountability Principle states that “A data controller should be accountable for complying with measures which give effect to the principles stated above.” A common approach to implement this principle in operating systems or DBMS is to log each data access as an event. Executing logging actions are obligations required by the majority of privacy policies. *Conditions*, that is, prerequisites to be met before any action can be executed, are critical in some cases. One of these cases is related to children information. Policy rules based on the Children’s Online Privacy Protection Act of 1998 (COPPA) apply to the online collection of personal information from children under 13<sup>36</sup>. Here “age under 13” is a condition that must be satisfied before the COPPA rules are applied.

Despite its limitations, existing access control technology can be used as a starting point for managing personal identifiable information in a trustworthy fashion<sup>37</sup>. Because both security policies for access control and privacy policies usually control access to the same set of resources they should not conflict and having a single model or integrated models for both kinds of policies will simplify management<sup>38</sup>. One such model has been developed in the context of our framework. The model, known as the Privacy-Aware Role Based Access (P-RBAC) model, extends the well known RBAC Model<sup>34</sup>, with information needed to control accesses to privacy-sensitive information. P-RBAC, which we describe in the next section, includes constructs to directly represent purposes and conditions found in privacy act and regulations<sup>39</sup> and has been recently extended to support privacy-related obligations<sup>40</sup>. Because it extends RBAC, P-RBAC policies are compatible, at least from the syntactic point of view, with RBAC policies and as such support a seamless introduction of privacy policies to organizations that already use RBAC as the access control model.

### 3.0 End-to-end Policy Management Framework

One of the challenges in creating an effective policy management framework is to enable the logical progression from the high level statement of security and privacy policies to low-level implementable policies with sufficient and accurate detail to govern system behavior as intended (see Figure 1).

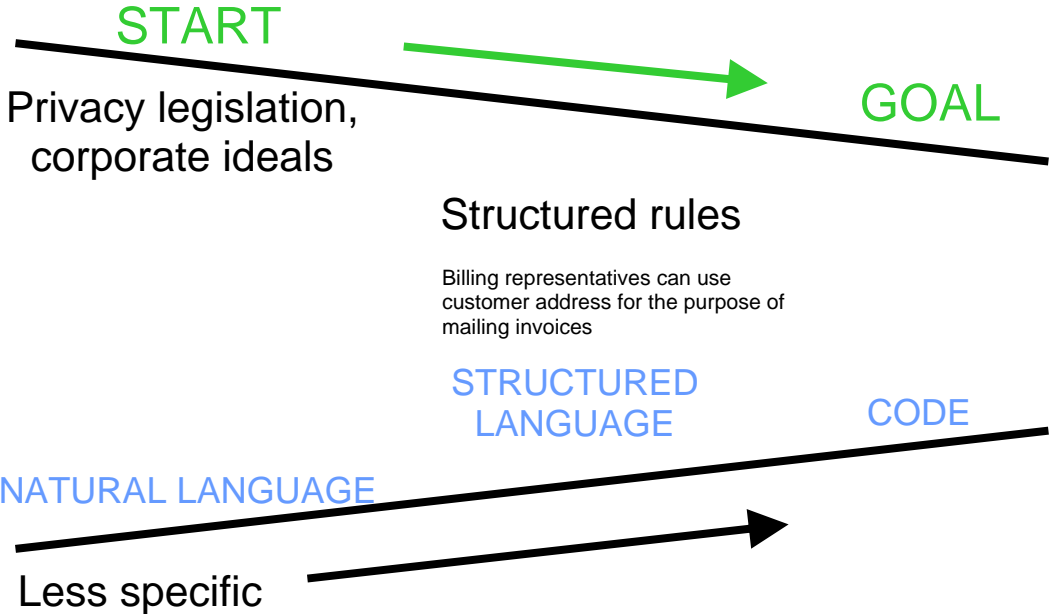


Figure 1 Linkage of policy intent from high to low-level expression.

We have formulated a layered policy model to provide a basis for reasoning about dynamic security and privacy policies. These abstraction and transformation models are a key enabler for providing end-to-end mechanisms for adapting system behaviors to meet high-level user-specified security and privacy goals through the enforcement of low-level controls in distributed

computing systems. These models provide a basis for automated policy processes including policy specification, verification, transformation, deployment, and auditing.

Policy specification requires more than a simple policy editing tool that enables users to express policies and perform syntax analysis on them. Policies can interact with each other, often with undesirable effects, and a policy author needs to be made aware of such interactions between policies as well as possible resolutions of these issues. Furthermore, policies are usually deployed in distributed systems where it is likely that a policy author may have a partial view of the entire system. Multiple authors may write policies applicable to the same set of resources, and processes must identify issues across these authors. An end-to-end policy management framework must provide for the full range of necessary refinements or transformations of policies from the stated human intention through system execution and monitoring.

### 3.1 Levels of Policy Refinement

The layered policy model describes multiple levels of representation: the Specification Layer, one or more Abstract Layers, and the Executable Layer (see Figure 2). The **Policy Specification** Layer is concerned with methods of authoring policies and capturing their structure and syntax in a formal manner. There are different interface techniques that might be used to do this, such as item selection from structured lists<sup>41</sup>, graphical rule selection methods<sup>42</sup> or constrained natural language authoring<sup>15</sup>. By constrained natural language, we mean natural language which can be understood or parsed by a system. The policies are specified by the user and automatically transformed into a formal format at this level.

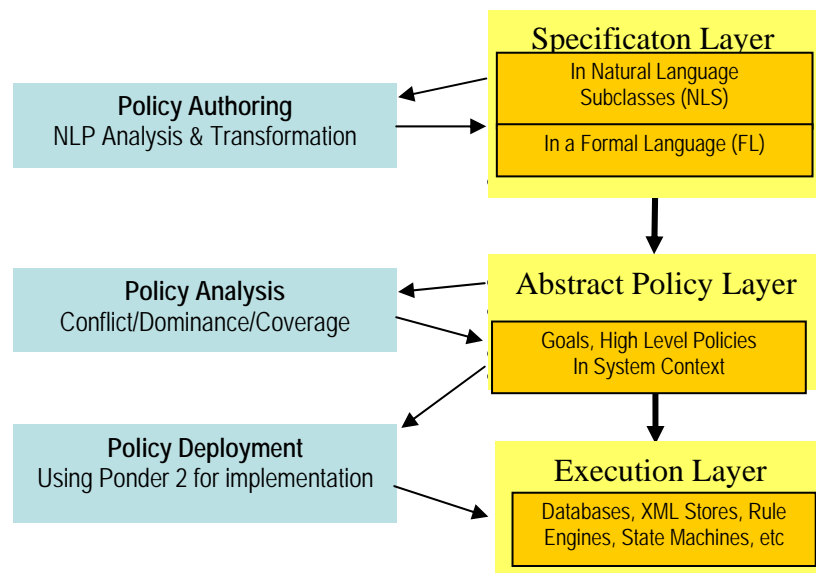


Figure 2 Security and privacy policy framework

The **Abstract Policy Models** Layer captures the semantics of the policy or policy sets that express goals and high level objectives for system behavior. Additionally, one or more of these

layers may delineate the policies that must be upheld by the different components of the distributed system to meet the policy goals, and incorporates, for example, explicit models for data, and classes of users and risk. At this more formal level of representation, the policies can be analyzed to find issues such as: conflict, dominance, and coverage. Part of this automated processing includes creating suggestions for resolving identified issues. Policy transformations occur within the Abstract Policy Model Layer and as the policies are passed to the Executable Policies Layer.

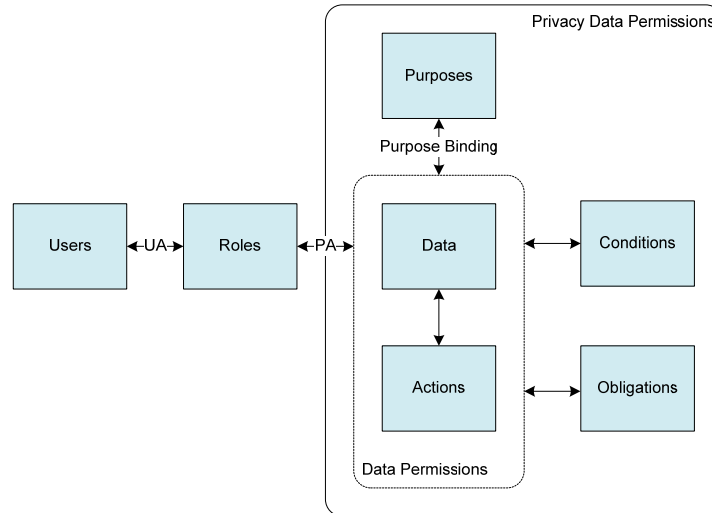
The **Executable Policies** Layer determines the precise constraints on resources that must be enforced by the existing security mechanisms in the system, and expresses policies explicitly in terms of the formats required by those mechanisms. Policies are deployed and executed at this level, decision logs are kept, various types of real time monitoring can be initiated, and post-transaction audits can take place.

Research continues on the development of the models required to support the management of policies, and the identification of suitable abstractions for relating the security and privacy policies at each layer. The intent here is to identify the characteristics of each of the layers, their functions, inputs and outputs; and, to specify the elements of the refinement process. The refinement process is meant to capture the automatic (with minimal human input) transformation of policies defined at the constrained natural language level into the enforceable security and privacy policies required in the dynamic distributed system through a series of steps that preserve their semantic intent. Each refinement step must be:

- *correct* in that the set of refined policies correctly implements the higher level policy;
- *consistent* in that the refinement must not lead to conflicts between the derived policies or the other policies existing in the system;
- *valid* in that the policies must be able to be enforced in the system context to which they will be applied; and
- *minimal* in that all policies in the derived policy set must be required for the correctness of the refinement.

### **3.2 P-RBAC – A Structured Language for the Execution Level**

P-RBAC is a family of Privacy-aware RBAC models that extend RBAC with support for privacy policies<sup>39,40</sup>. P-RBAC supports the notion of “privacy-aware” permissions, that is, authorizations taking account privacy-relevant information, such as the purpose of data usage. Core P-RBAC, the base model, is at bottom. There is a tradeoff between expressivity and complexity in the design of Core P-RBAC. On the one hand, Core P-RBAC has limited expressive power which is, however, sufficient for representing public privacy policies, privacy statements and privacy notices in Web sites, and policies based on privacy related acts, such as HIPPA, COPPA, and GLBA, in the US. On the other hand, conflicts detection in Core P-RBAC remains tractable. Advanced models in the family extend Core P-RBAC with additional modeling constructs, e.g., Conditional P-RBAC supports a full-fledged conditional language<sup>43,44</sup>. Core P-RBAC, illustrated in Figure 3, includes several sets of entities: Users, Roles, Data, Actions, Purposes, Obligations, and Conditions expressed by using a customized language, referred to as LC<sub>0</sub>.



**Figure 3 Core P-RBAC Components**

A user in P-RBAC is human being, and a role represents a job function or job title within the organization with some associated semantics regarding the authority and responsibility conferred on a member of the role. Data refers to any information related to an identified or identifiable individual. An action is an executable image of a program, which upon invocation executes some function for the user. The types of action and data object that P-RBAC controls depend on the type of system in which they are implemented. The motivations for introducing purposes, conditions, and obligations in Core P-RBAC are discussed in the Section 2.3 above, and Core P-RBAC directly models these notions.

In Core P-RBAC, as in classical RBAC, permissions are assigned to roles and users obtain such permissions by being assigned to roles. The distinctive feature of Core P-RBAC lies in the complex structure of privacy permissions, which reflects the highly structured ways of expressing privacy rules. The model captures the essence of OECD principles and privacy acts. Therefore, aside from the data and the actions to be performed on it, privacy permissions explicitly state the intended purposes, along with the conditions under which the permissions can be given, and the obligations that are to be finally performed if permissions are granted.

### **3.3 System Perspectives**

Enterprises specify policies for the operation of their internal processes, as well as for interaction with other entities. These policies are authored and maintained by different individuals, playing different roles within the organization. Typically there are multiple kinds of Policy Administrators (Security, Database, etc.) and System Operators (Network, Servers, etc.), as well as Resource Owners, e.g., those who control and can update specific information. In general, the higher level policies are established by Business Process Administrators on behalf of the managers of the enterprise.

The different roles tend to interact with the policy management system at different levels of abstraction. Business Process Administrators might author policies at the Specification Layer, while Security and System Administrators tend to author policies at one of the Abstract Policy Model Layers. System Operators would interact with the system at the Executable Layer.

Mechanisms should be provided to maintain transparency and consistency of the various policies within the overall system.

A critical aspect of policy management within the enterprise is monitoring and logging of policy controlled transactions, such as a history of transactions that were allowed or denied. The log information provides the basis for auditing and compliance checking, and allows the enterprise to determine whether the desired policies are actually being carried out as intended.

### 3.3.1 Algorithms and tools - Relationships between Levels

The three layers of the policy model in Figure 2 capture the policy refinement process. Policies are defined at the *Policy Specification* layer in constrained natural language, and transformed into a structured format (e.g., a privacy policy rule would include User Category, Action, Data Element, Purpose, Condition, and Obligation). Sets of policies can be analyzed at this level of abstraction with respect to conflicts and inconsistencies, using grammars and logical operations on the structured elements. In order to elaborate the semantics of the policies further, ontological models are applied to give formal meanings to the elements of the policies so that they can be interpreted in the context of the system. They then become *Abstract Policy Models*.

Between each of the *Abstract Policy Models* and the *Executable Policy Model*, policy transformations are performed in order to derive the concrete policies for the components of the distributed system from the higher level goals. Policy synchronization must also be performed, keeping track of the relationships between the policies at each level. The lowest level Abstract policies must be mapped into the formats required by the mechanisms that enforce the system policies.

For example, a policy may state that doctors can view patient information (*Specification*). At the next layer of refinement (*Abstract Policy Models*), the process for identifying a user as a “doctor” (e.g., by inclusion in a particular access control list or association with a specific role), the meaning of the action “view” (e.g., through identification with specific applications or system functions), along with the definition of “patient information” (e.g., through identification of specific electronic files) would be defined. When the policy is further refined the different policies for each of the different categories of patient information will be elaborated. Finally, these must be transformed into the rules and constraints on the system components that control access to the sources of information (*Executable Policies*).

One good example to show the relationships between levels is how SPARCLE assists policy authors by supporting a natural language specification of P-RBAC policies and by automatically translating these natural language specifications into P-RBAC permission assignments. As illustrated in Figure 4, P-RBAC policies are first authored in natural language by using SPARCLE; from these natural language specifications an XML specification of P-RBAC permission assignments is automatically generated by SPARCLE. An analysis is then performed to determine possible conflicts and feedback is returned to the policy authors.

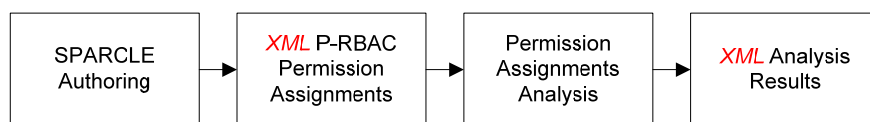


Figure 4 The policy assignment workflow

### 3.3.2 Policy Analysis and Ratification

Analysis of policies can be performed at each of the different levels of the framework. A single policy rule can be found to be in error if the authoring component detects a syntax error. For example, in natural language authoring we could highlight when a required component is missing from a privacy policy statement. It is possible to carry out syntactical analyses on the high level policies (e.g., on the parsed syntactic level in natural language policies we can detect that a purpose is missing in the policy rule “Doctors can view medical history information”), or on policies refined into formal representations which include semantic detail for the objects and actions in the policy rule (e.g., to detect that an object in an XACML policy is not of the proper type or class).

Broadly speaking we can classify the analysis in three categories<sup>45</sup>:

- *Policy validation* to verify that the system or systems where a policy needs to be enforced have the capabilities to implement the enforcement
- *Policy ratification* to provide a formal process to policy managers/authors to certify the appropriateness of a policy before the policy is activated
- *Policy run-time analysis* to provide monitoring, auditing and dynamic conflict resolution to ensure policies are carried out as intended

Policy validation can be done at each level of the framework. At the specification level, if policies are written in natural language there must exist a mapping from terms in the policy specification to structures in the abstract model. For example, if a policy refers to encryption mechanisms, the formal model must understand what encryption is and what encryption mechanisms exist to rule out as invalid a policy that tries to enforce an encryption mechanism that is not supported by the concrete layers below. Validation continues to happen at the lower layers since policies get transformed and possibly split into multiple policies that may be enforced by different end-devices or elements. Some mechanisms may be supported by some but not all end-devices, and from the specification, before transformation, it might be difficult to detect how the policy will be transformed and what devices will be affected to be able to decide if the policy will be valid after going through all the transformations to get to the enforcement point.

We can identify three primitive operations for Policy ratification: conflict detection, dominance and coverage<sup>46</sup>. Policies are in conflict if they cannot be enforced simultaneously. A policy is dominated or redundant if the policy when added to a system does not modify the behavior of the system. Coverage refers to the determination of whether a policy set covers all the cases of interest, e.g., all users in the system have rights to at least one resource in the system. In general the implementation of these operations is computationally hard but several special cases (see <sup>43,44</sup>, and the references therein) and approximations<sup>47</sup> have been identified. In addition to these primitive operations there are application-dependent properties that policy authors need to verify. Typical application-dependent properties are conflict of duty and conflict of interest. Conflict of duty occurs when a subject is not supposed to have the right to apply two operations to the same object, e.g., an employee that makes a payment request should not be able to approve it. Conflict of interest may arise when a subject is not permitted to do an operation to two different objects, e.g., a bank providing advice to competing clients.

Policy analysis techniques developed for P-RBAC are in accordance with this classification, e.g. conflict detection and redundancy checking in Conditional P-RBAC<sup>40</sup>. Policies in P-RBAC are represented by permissions. The introduction of obligations in P-RBAC raises several issues with respect to policy validity and policy ratification: indeterminism of obligation enforcement, validity of permission assignment, and coverage of obligations. Indeterminism in obligation enforcement arises because multiple policies can apply to the same access request. Given a request to which two or more policies apply, the choice of the obligations to be executed can be non-deterministic if policies are not well written, e.g., the relationship between policies is an “OR-relationship” and the policies have different obligation sets. Policy normalization<sup>40</sup> has been developed to detect indeterminism in obligation enforcement.

The validity of permission assignment results from the interaction between permissions and obligations. An obligation may invalidate its containing permission. For instance, there is no permission allowing the subject of an obligation to perform the action of the obligation, and the conjunction of the condition of an obligation and the condition of the permission containing the obligation is not satisfiable. Obligation execution may result in *obligation cascading*, that is, the execution of an obligation can trigger the execution of another obligation which in turn may trigger other obligations.. Ill-written permissions can result in non-terminating obligation execution. The notion of cascading bag<sup>40</sup> has been proposed to efficiently handle such the problem.

The coverage of obligation characterizes the different “heaviness” or “strictness” between two obligations that is similar to the notion of policy dominance. Some requests may have several applicable policies that could lead to a large number of obligations. Therefore reducing the number of obligations to be executed may have significant practical impact. Obviously, the remaining obligations should not modify the “duties” required by the original policies. On the other hand, we can imagine that many of these obligations are similar to each other since they are obligations associated with similar permissions. An efficient algorithm for the minimization of obligations, that takes into account temporal constraints and both pre-obligation and post-obligation, has been recently developed<sup>40</sup>.

Validation and ratification are meant to be applied before policies are deployed. The third category of analysis techniques is performed at run-time. Run time conflict resolution can take into account the state of the system when the conflict occurs to decide how to resolve it. The resolution mechanism itself can be specified by a meta-policy. Monitoring traces the system behavior to determine if the policies are having the expected effects. One very common use of monitoring is intrusion detection.

There are situations in which the best way to discover an incorrect policy is to monitor what subjects are accessing what resources because the policy granted the wrong access rights to a certain class of user. Monitoring can also help to detect the abuse of privileges of a user by discovering unusual patterns of the use of rights. Auditing is also important. Because of government regulations, enterprises may need to demonstrate compliance to the rules set by regulatory agencies. In addition, as legal protection for potential private litigation, auditing systems may help prove that there were no violations of policies by individuals. One important aspect of monitoring and auditing modules is that they get their input from the bottom layer in the framework, but they will need to produce their output or results in some way connected to the policies at higher levels of abstraction to make the explanations understandable to policy administrators who have overall responsibility for policy enforcement.

### 3.3.3 End-User Issues

It is important to develop technologies for effectively conveying security and privacy policies to the end-users who interact with those policies. These users include both *data users* - the individuals within an organization who may request access to resources, and *data subjects* - the individuals (e.g., customers, employees, patients, citizens) who wish to understand and possibly limit an organization's practices before interacting with the organization. Data users, in order to comply with a policy, need to understand those portions of policies that apply to them. Data users may also require explanations of policy decisions, in order to understand and possibly correct situations in which requests for access were denied when the data user expected it to be allowed<sup>48</sup>. Like data users, data subjects have a need to understand those portions of a policy that apply to them. Studies reveal that if people are not comfortable with an organization's security and privacy practices, they take their business elsewhere<sup>49</sup>. Furthermore, in cases where individuals are given options to limit the organization's practices, for example, an option to "opt out" of receiving advertisements, they require mechanisms for understanding and exercising these options. Thus, technologies to help data users and data subjects fall into three classes: *policy presentation* technologies, which convey the content of policies; *policy explanation* technologies, which explain policy decisions; and *processing technologies*, which provide mechanisms for conveying options and eliciting the individual's choices. These three classes of technologies are described in the sections below.

#### 3.3.3.1. Policy Presentation

Approaches to policy presentation generally include two stages: first, finding a language in which to represent a policy; and second, developing a means to present the policy to users.

Languages for representing policies may be natural or formal languages. Natural language representations of policies may be prone to the contextual problems of prose in which author and reader might not share a common context or understanding of the terms in the policy. Thus any natural language policy can be ambiguous or inconsistent (just as poorly written programming code is problematic). One approach to making natural language representations clear and consistent is to constrain the language by providing a fixed vocabulary in which to author a policy, in which the meaning of each term is carefully defined and known by all users. Systems which author rules by selecting terms from categories take this structured list approach. Another approach is to allow natural language authoring, and augment the policy with definitions for terms used by the author. For example, the SPARCLE system developed at IBM provides both structured list and constrained natural language methods for authoring<sup>15,41</sup>. SPARCLE transforms the policies into formal languages, maps the policy elements onto system elements for automated enforcement, and draws on the decision logs for auditing. This tool provides a logical and verifiable link from policy specification to the compliance audits of enforcement decisions. Another approach to making clearer natural language representations is to standardize their organization and layout, as the multi-layered notice approach does<sup>52</sup>.

Formal languages for representing policies remove the ambiguities and inconsistencies of natural language, allow for consistent presentation of many different policies to users, and can allow for automated comparison of different policies to each other. For example, the Platform for Privacy Preferences (P3P) provides a formal representation of websites' privacy policies that can represent any website's policy, but allows for any website's policy to be presented to a user in a simple, short format<sup>18</sup>. P3P also enables a Web browser to compare a user's privacy preferences

to a website's P3P policy, so that if a website's P3P policy does not match a user's preferences, the browser can alert the user, or prevent the user from visiting the site. There is no verifiable link between the web policies and internal business operations at this time. Besides P3P, other formal languages for representing access control and privacy policies include the Enterprise Privacy Authorization Language (EPAL)<sup>50</sup> and the eXtended Access Control Markup Language (XACML)<sup>51</sup>.

There have been several approaches to presenting policies to users. The multi-layered approach to presenting natural language privacy policies is intended to provide a short, consistent, comprehensible structure to policy presentations so that users find it easy to search for specific information in a policy<sup>52</sup>. The PrivacyFinder, a search engine that ranks websites by how protective their P3P policies are of a consumer's personal data, provides Privacy Reports that present P3P policies in sentences that have been vetted in user tests for consistency and readability<sup>52</sup>. PrivacyFinder can also represent P3P policies graphically, by showing a graphical meter that indicates how well a website's P3P policy matches a user's privacy preferences. While the PrivacyFinder's graphical representation of P3P policies aggregates the details of a P3P policy into one simple graphical meter, IBM's SPARCLE system provides a simple visualization of policy coverage in a user configurable matrix format with policy content presented for the key policy elements<sup>15</sup>. For enterprise systems, there is an expandable grid visualization that provides a meta-view with drill-down capability<sup>42,53</sup>. Simplified, aggregate policy representations may be most appropriate for customers, while detailed overviews are often needed by requestors in order to understand policy decisions in specific situations.

Desirable characteristics of good policy presentation technologies include:

- The ability to convert the machine readable policies into human understandable policies and vice versa, and
- The ability to provide high level views of policies and access to further levels of detail as desired.

### *3.3.3.2. Policy Explanation*

Policy explanation technologies help users to understand the decisions made by privacy and security enforcement mechanisms. When a requestor has tried to access data to which they believed they should have access, it is helpful to understand why that access was denied. When an access is denied, it may be that there is an error in a policy. Therefore, explaining why access was denied may enable a requestor to work with administrators to work out errors in policies<sup>48</sup>. While there has been a fair amount of research in providing explanations in rule based systems<sup>55</sup>, there has been little work to date to connect this to policy management systems. We see this as an important area for further research in the development of end-to-end policy management systems.

### *3.3.3.3. Processing Technologies*

Systems should also have mechanisms to enable users to specify preferences so that the systems can provide information and decisions according to the users' needs. One solution to this challenge is A P3P Preference Exchange Language (APPEL)<sup>56</sup>. APPEL is used to represent a user's preference with respect to privacy policies; user agents like Privacy Bird and Internet Explorer compare a user's preferences against a website's P3P privacy policies. A user can

consider the output of such a comparison to decide whether to continue interaction with a website.

When organizational policies allow customers to opt in to or opt out of certain organizational practices (telemarketing or sharing data with third parties, for example), some technology should exist for eliciting user preferences and incorporating them into the backend access control mechanisms.

Most of the languages mentioned in this section are fundamentally designed on the basis of eXtended Markup Language (XML); XML can be used to represent users' preferences, system preferences and organizations' preferences in a formal language which can be used by the users in different contexts (e.g., while making a decision whether to share one's credit card number or not) to make informed decisions.

Since user preferences are fundamentally policies, users have a need to author policies just as enterprises do, albeit on a smaller scale. It is therefore possible, and perhaps likely, that some of the same authoring mechanisms, such as user interfaces and visualizations, that support enterprise policy authors would also help end-users in authoring preferences.

In this section we discussed technologies that are available or need to be developed to satisfy user needs in the security and privacy policy framework. While some of the needs are fairly general to usable systems, we believe that there is a considerable need for additional work to tailor the technologies to policy management issues.

## 4.0 Healthcare Scenario-based Security and Privacy Policies

We will briefly illustrate the different levels of refinement that are needed for policy management through the use of a scenario which includes some sample privacy and security policies. In our healthcare scenario, a patient named Mary Smith sees her primary care physician regarding a medical situation, is then referred to a specialist, and finally enters a hospital for a successful operation to remove a cancerous tumor. She agrees to have her medical information forwarded to a national research database for medical research purposes. Examples of relevant privacy and security policies are presented at the three levels of the Security and Privacy Policy Framework and the types of transformations and synchronization that would occur are illustrated (see Figure 2). The security policies provide the base which enables the privacy policies.

Privacy policy rules have a structure which calls for identification of user category, action, data category, and purpose elements in all rules, and additionally can have conditions or obligations as optional elements<sup>57</sup>. Thus syntax for privacy policies is given, but semantics for each element remain to be specified. Note that security access control policies have a similar syntax, but lack a purpose element<sup>58</sup>. At the first level of representation, we do not yet have clear specification of what the individual elements refer to. For example, one healthcare organization might define "healthcare staff" differently than another.

### **Policy Specification Layer**

#### **Privacy Policy Rules**

1. Doctors can collect and use patient medical information for the purpose of treatment.
2. Patients can view patient medical information for the purpose of informed consent to

treatment if the information is about them.

3. Specialists can collect and use patient medical information for the purpose of consulting on medical treatment.
4. Healthcare staff can forward patient medical information for the purpose of national medical research if the information is anonymized.

### **Security Policy Rules**

1. Doctors can access medical applications.
2. Administrative staff can access billing applications.
3. Patients can access medical information applications.
4. Healthcare staff can access test results databases.
5. Healthcare staff can access upload and email applications.

For illustration, we will consider the fourth privacy policy rule above. In the Policy Specification Layer policy rule elements for this policy would be identified as shown below. The output of the Specification Layer is a machine-readable version of the rule.

- Users == Healthcare staff
- Actions == can forward
- Data == medical information
- Purpose == national medical research
- Condition == information is anonymized

### **Abstract Policy Model Layer (transforming the fourth and fifth policy rules from above)**

#### **Privacy Policy Rules**

- 4) Healthcare staff (user group A ) can upload (upload application) patient test results (DB table patient info, column results) to the NIH DB (NIH DB Study Results) if patient identity is not disclosed (Do not use DB table patient info, column name).
- 5) Healthcare staff (user group A ) can transmit (Send application ) patient test results (DB table patient info, column results) to NIH registered university researchers (email address list) if patient identity is not included (Do not use DB table patient info, column name).

#### **Security Policy Rules**

- 4) Healthcare staff (user group A) can access (read/write/modify) test results databases (DB table patient info, column results).
- 5) Healthcare staff (user group A) can access upload (App 1) and email applications (App2).

### **Executable Policy Layer (transforming from the policies in the layer above)**

#### **Privacy Policy Rules**

- If request(upload(DBname)) && DBName==NIH\_Records then Set DB = anonymize(PatientDB,NameAttribute,AddressAttribute); upload(DB,DBname)
- If request(transmit(destination\_address,testData)) && member(destination\_address,RegisterUniversityList) then
- Set DB = Select BloodTest From PatientTable; transmit(destination\_address,DB)
- If request(transmit(destination\_address,Type)) && (Type != testData OR NOT(member(destination\_address,RegisterUniversityList)))
- then deny(transmit(destination\_address,Type))

#### Security Policy Rules

- If user(member group A) && Read(PatientDB) then allow.
- If user(member group A) && Access(App1) then allow.
- If user (member group A) && Access (App2) then allow.

**Figure 5 Rules at different policy layers.**

## 5.0 Global Considerations

### 5.1 Context

In the description of the refinement process it was noted that each refinement step must be *correct*, *consistent*, *valid* and *minimal*. These characteristics apply with respect to a given system model. At each level of abstraction there must thus be an understanding of the system model within which the policies are to be interpreted. At the specification layer this is captured in the vocabulary and grammars of the particular domain. One cannot expect to be able to write meaningful policies that reference subjects that are not relevant within the domain of discourse. For healthcare, policies will typically involve doctors, nurses, pharmacists, patients, etc., and deal with the responsibilities of the various roles and the protection of patient information.

At lower levels of abstraction the system models would be more detailed and precise. In the IBM Autonomic Management Architecture, a system is described by its sensors and effectors. The sensors are variables that can be read, and the effectors are methods that can be called to change the state of the system. Policies would then have conditions based upon the sensors that are available, and they would affect the state of the system by invoking its effectors. Conditions that could not be stated in terms of the available sensors would thus be invalid, as would actions for which the system has no effectors.

The state descriptions necessary to support the refinement of a set of high level policies may thus be extensive and difficult to obtain and maintain. This is a general problem with model driven

management. It manifests itself in some particular ways in the realm of policy based management, since policies are meant to be declarative and contextually interpreted.

It is not clear how best to capture and structure the contextual information necessary for supporting policy based management in a convenient fashion. Context is often taken as just additional information in the system model, but is typically associated with separate, related system models. For example, two systems may have the same access control policy, but this policy will have to be implemented in different fashions because the two systems have different operating systems with different sensors and effectors. This can clearly be seen as an extension of the system models to include operating system characteristics. However, what if the two systems are identical but happen to reside in separate nations. The same high level policy may have to be interpreted in different ways because of local laws and regulations. This is clearly an effect of the legal system on the IT system. Aggregating the system models in this case may not be desirable.

## **5.2 Trust & Risk**

The above discussion assumes that the system models can be known, and the only problems are in determining them and accurately maintaining the information. Another viewpoint is that of making decisions in an uncertain environment in which everything cannot be known and elements of trust and risk need to be considered. Several trust models have been proposed<sup>59</sup>, but whether adequate models of risk and trust can be designed remains an open question. Trust is the extent to which one party is willing to depend upon somebody or something in a given situation even though negative consequences are possible. Risk is the anticipated hazard following from a fault in the system and can be measured according to the consequences of the negative event.

As far as policies are concerned, elements of trust and risk can be taken into account in the conditions affecting their applicability. This assumes that there is a trust and risk model that can provide adequate guidance given the sensors available in the system. An interesting aspect of such a formulation is that it allows actions to change based upon additional contextual variables. Dispositional trust, for example, could be high within a secure office environment, but low in a public environment, and the same policies would support different decisions based upon this location information.

Risk aversion could be taken into account in the stated preferences at the policy specification layer. Certain users might be less averse to sharing private information than others, for example. Risk models could be further elaborated in the Abstract Policy Models layer, and instantiated as part of the Concrete Policy Sets. Risk and trust factors could then be taken into account as part of the context for enforcement. As mentioned before, these risk and trust models can be considered as part of the system model or as related contextual models that provide additional guidance.

An interesting issue in terms of policy based management is the granularity of the trust and risk information. Should there be a single risk factor calculated for each policy decision, or should the sets of policies have access to all aspects of the trust and risk models and use specialized algorithms for assessing each individual situation.

## 6.0 Conclusions and Research Directions

Organizations are increasingly aware of the need to enforce security and privacy policies to protect sensitive organizational and personal data which they store and utilize to provide high quality services to their customers, constituents, and patients. The growing complexity of their systems and networks and the need to share data across organizational boundaries is making this need more and more difficult to meet. To help satisfy this need, we have identified challenges in policy management that must be addressed in order to provide an end-to-end comprehensive policy management system for security and privacy. The primary challenge is seen as managing policy consistency through a series of transformations that takes policies from high level abstractions through enforcement in IT systems.

With these challenges in mind, we have proposed a three-level framework for discussing policy within which security and privacy policy management research can be conducted. These levels include: the Policy Specification Layer in which policies can be specified using highly useable methods for policy authoring such as natural language processing, the Abstract Policy Model layer which allows the goals and high level objectives of policy sets to be identified by capturing the policy semantics, and the Executable Policies layer which defines how policies are enforced by security mechanisms within the system. We have illustrated how the framework would work through the context of a medical scenario and have provided examples of the policy transformation and synchronization between the four layers. While we believe that the research presented in this paper provide a strong foundation for policy management, we think that it is even more important that the proposed framework can support the exploration of many remaining research issues surrounding security and privacy policy management.

As we have outlined in this paper, there has been considerable research on many aspects of the overall problem of developing a usable policy management framework. While the work we summarize here represents advances in many of these areas, much of it still requires exploration with the real world challenges of developing policy-based systems. The challenge is to create the technologies necessary to provide an end-to-end policy specification, management and deployment system that requires as little human intervention as possible, while also allowing the people involved with the system to understand and change policies as needed. This is not an easy task, and we do not believe that any single research or development effort in this area would meet all of the requirements for the range of privacy and security capabilities that we have encountered in our research. Our objective in developing the architecture has been to provide a framework that many people can contribute to and to provide a base from which we can advance.

We believe this work can benefit from the development of appropriate standards, so that the problem will not continue to grow in complexity with the development of new platforms. We also believe that our open architecture will support incremental advances as research provides new ideas for various aspects of the framework.

What are some of the key challenges? While we do not think we have a complete list, key research questions are provided in Figure 6 below. Our team will continue to work on these research questions with the goal of providing both publications and open source code components to advance the science of policy management.

### **Policy Authoring**

- What other methods of specifying policy in a usable and effective manner can be created for organizations and individuals?
- How are ontological models applied to policy elements?
- How can the management of large-scale sets of dynamically varying policies be simplified?
- Can policies be standardized in ways which enable effective collaboration across organizational boundaries?

### **Policy Analysis**

- What types of analysis might be valuable to perform at the different levels of policy abstraction?
- What types of transformations and synchronizations are required across policy levels?
- Can adaptation of policy specification, editing, analysis and viewing methods work successfully across policy domains?
- How can the results of policy analytics at the lower levels of policy abstraction be effectively communicated to users at the highest level (the business level)?

### **Policy Deployment**

- How can policy be implemented most effectively across heterogeneous configurations?
- How can a policy management system effectively handle mobility and dynamic contexts?
- What types of compliance audits can policy management provide?
- How can policy be managed across organizational boundaries?

**Figure 6 Research questions in end-to-end policy management.**

## **7.0 Acknowledgements**

The authors would like to acknowledge the support of an IBM Open Collaborative Research Award to Elisa Bertino at Purdue and Lorrie Cranor at CMU which has partly supported the work reported here and supported the collaboration between teams at IBM Research, CMU and Purdue.

## **8.0 References**

- 1) D. Norman, *The Design of Everyday Things*. New York, DoubleDay, (1988).
- 2) Organization for Economic Co-operation and Development, *OECD Guidelines on the Protection of Privacy and Transborder Flow of Personal Data*. Paris, France (1980). [www.oecd.org/home/](http://www.oecd.org/home/).

- 3) A. Kobsa, Personalized Hypermedia and International Privacy. *Communications of the ACM*, 45 (5), 64-67, (2002).
- 4) A. Adams, and A. Sasse, Privacy in Multimedia Communications: Protecting Users, not Just Data. In Blandford, A., Vanderdonk, J., and Gray, P. (Eds.), *People and Computers XV – Interaction without Frontiers. Joint Proceedings of HCI 2001 and ICM 2001*, Lille, Springer, Berlin, 49-64, (2001).
- 5) M. Culnan, Protecting Privacy Online: Is Self-regulation Working? *Journal of Public Policy and Marketing*, 20-26, (2000).
- 6) National Research Council, *Who Goes There? Authentication Through the Lens of Privacy*. National Academies Press, Washington, DC, (2003).
- 7) U.S. Fair Credit Reporting Act of 1978. U.S. Code 1681.  
[www.ftc.gov/os/statutes/0312224fcra.pdf](http://www.ftc.gov/os/statutes/0312224fcra.pdf)
- 8) J. Karat, C. Karat, and C. Brodie, Human-Computer Interaction Viewed from the Intersection of Privacy, Security, and Trust. In *The Human-Computer Interaction Handbook*. Lawrence Erlbaum Associates, 639-658, (2008).
- 9) L. Palen, and P. Dourish. Unpacking ‘Privacy’ for a Networked World. *Proceedings of the Human Factors in Computing Systems Conference (CHI 2002)*, 129-136, ACM Press, (2002).
- 10) M. Ackerman and S. Mainwaring. Privacy Issues in Human-Computer Interaction. In L. Cranor and S. Garfinkel (Eds.), *Security and Usability: Designing Secure Systems that People Can Use*, 381-400, Sebastopol, CA, O’Reilly, (2005).
- 11) C. H. Manny, European and American Privacy: Commerce, Rights, and Justice. *Computer Law and Security Report*, 19 (1), 4-10, (2003).
- 12) P. Kumaraguru, and L. Cranor, Privacy in India: Attitudes and awareness. In *Proceedings of the 2005 Workshop on Privacy Enhancing Technologies (PET2005)*, 30 May - 1 June 2005, Dubrovnik, Croatia, (2005).
- 13) E. Ball, Patient Privacy in Electronic Prescription Transfer. *IEEE Security and Privacy*, 1 (2), 77-80, (2003).
- 14) D. Baumer, J.B. Earp, and F.C. Payton, Privacy in Medical Records: IT Implications of HIPAA. *Computers and Society*, 40-47, December, (2000).
- 15) J. Karat, C. Karat, C. Brodie, and J. Feng, Privacy in Information Technology: Designing to Enable Privacy Policy Management in Organizations. *International Journal of Human-Computer Studies*. 63, 1-2, 153-174, (2005).
- 16) I. Altman, *The Environment and Social Behavior: Privacy, Personal Space, Territory, and Crowding*. Monterey, CA, Brooks Cole, (1975).
- 17) J. Smith, Privacy Policies and Practices: Inside the Organizational Maze. *Communications of the ACM*, 36(12), 105-122, (1993).
- 18) L. Cranor, M. Langheinrich, M. Marchiori, M. Presler-Marshall, and J. Reagle, The Platform for Privacy Preferences 1.0 (P3P1.0) Specification, (2002). <http://www.w3.org/TR/P3P/>.
- 19) S.A. Warren, and L.D. Brandeis, The Right to Privacy. *Harvard Business Review*, 4, 195, (1890, December).

- 20) A. Senior, S. Pankanti, S. Hampapur, L. Brown, L. Tian, A. Ekan, J. Connell, C. Shu, and M. Lu, Enabling Video Privacy Through Computer Vision. *IEEE Security and Privacy*, 3 (3), 50-57.
- 21) CRA Conference on Grand Challenges in Information Security and Assurance. [www.cra.org/Activities/grand.challenges/security](http://www.cra.org/Activities/grand.challenges/security). (2003)
- 22) J. Salzter, and M. Schroeder, The protection of information in computer systems. In *Proceedings of IEEE*, 63, 9, 1278-1308, (1975).
- 23) D. Kahn, *The Codebreakers: The Story of Secret Writing*. New York, MacMillan, (1967).
- 24) R. Anderson, Why Cryptosystems Fail. *Communications of the ACM*, 37 (11), 32-40, (1994).
- 25) B. Schneier, *Secrets and Lies: Digital Security in a Networked World*. New York, Wiley, (2000).
- 26) R. de Paula, X. Ding, Pl Dourish, K. Nies, B. Piller, D. Redmiles, J. Ren, J. Rode, and R. Filbo, In the Eye of the Beholder: A Visualization-based Approach to Information Systems Security. *International Journal of Human-Computer Studies*, 63 (1/2), 5-24, (2005).
- 27) J. Kelsey, B. Schneier, D. Wagner, and C. Hall, Cryptanalytic Attacks on Pseudorandom Number Generators. *Proceedings of the Fifth International Workshop on Fast Software Encryption*, 168-188. Berlin, Springer, (1998).
- 28) D. Wagner, J. Foster, E. Brewer, and A. Aiken, A First Step Toward Automated Detection of Buffer Overrun Vulnerabilities. In *Network and Distributed Systems Security Symposium*, 1-15, (2000).
- 29) U. Shankar, K. Talvar, J. Foster, and D. Wagner, Detecting Format string Vulnerabilities with Type Qualifiers. In *Proceedings of the 10<sup>th</sup> USENIX Security Symposium*, 201-220, (2002).
- 30) R. Kemmerer, C. Meadows, and J. Millen, Three Systems of Cryptographic Protocol Analysis. *Journal of Cryptography*, 7 (2), 79-130, (1994).
- 31) B. Schneier, and D. Mudge, Cryptanalysis of Microsoft's point-to-point Tunneling Protocol (PPTP). *Proceedings of the 5<sup>th</sup> ACM Conference on Computer and Information Security*, 132-141, San Francisco, ACM Press, (1998).
- 32) S. Ames, M. Gasser, and R. Schell, Security Kernel Design and Implementation: An Introduction. *IEEE Computer*, 16(7), 14-22, (1983).
- 33) D. Ferraiolo, R. Sandhu, Gavrilu, D. Kuhn, and R. Chandramouli, Proposed NIST Standard for Role Based Access Control. NIST, 2001.
- 34) R. Sandhu , E. Coyne , H. Feinstein , C. Youman, Role-Based Access Control Models, *Computer*, v.29 n.2, p.38-47, February 1996.
- 35) S. Fischer-Hubner, *IT Security and Privacy*, Springer-Verlag, 2001.
- 36) Children's Online Privacy Protection Act of 1998 (COPPA), Title XIII – Children's Online Privacy Protection. [www.ftc.gov/ogc/coppa1.htm](http://www.ftc.gov/ogc/coppa1.htm), (1998).

- 37) P. Ashley, C. Powers, M. Schunter, From privacy promises to privacy management: a new approach for enforcing privacy throughout an enterprise. Proceedings of the 2002 workshop on new security paradigms, ACM, 2002.
- 38) Anderson 2006 Need citation\*\*\*\*\*
- 39) Q. Ni, A. Trombetta, E. Bertino, J. Lobo, *Privacy-aware role based access control*, SACMAT 2007, pgs 41-50.
- 40) Q. Ni, D. Lin, E. Bertino, J. Lobo, An Obligation Model Bridging Access Control Policies and Privacy Policies, SACMAT 2008.
- 41) C. Karat, J. Karat, C. Brodie, and J. Feng, Evaluating Interfaces for Privacy Policy Rule Authoring. *Proceedings of the Conference on Human Factors in Computing Systems – CHI 2006*, ACM Press, 83-92, (2006).
- 42) R. Reeder, L. Bauer, L. Cranor, M. Reiter, K. Bacon, K. How, and H. Strong, Expandable Grids for Visualizing and Authoring Computer Security Policies. In Proceedings of the Conference on Human Factors in Computing Systems – CHI 2008, ACM Press, in press, (2008).
- 43) Q. Ni, D. Lin, E. Bertino, J. Lobo, *Conditional Privacy-Aware Role Based Access Control*, ESORICS 2007, pgs 72-89.
- 44) Q. Ni, E. Bertino, C. Brodie, C. Karat, J. Karat, J. Lobo, and A. Trombetta, Privacy-aware Role Based Access Control. Submitted to TIISEC 2008.
- 45) A. Bandara, “A Formal Approach to Analysis and Refinement of Policies”, PhD Dissertation, Department of Computing, Imperial College London, July 2005.
- 46) D. Agrawal, J. Giles, K-W. Lee, K-W. and J. Lobo, “Policy Ratification”. In *Sixth IEEE International Workshop on Policies for Distributed Systems and Networks (POLICY)*, pages 223–232, (2005).
- 47) J. Lobo, An Approach to Evaluate Policy Similarity, *SACMAT 2007*, (2007).
- 48) A. Kapadia, G. Sampemane, and R.H. Campbell, KNOW Why Your Access was Denied: Regulating Feedback for Usable Security. In *Proceedings of the 11th ACM Conference on Computer and Communications Security (CCS 2004)*. New York, ACM Press, (2004).
- 49) S. Bellman, E.J. Johnson, S.J. Kobrin, and G.L. Lohse, International Differences in Information Privacy Concerns: A Global Survey of Consumers. *The Information Society* 20, 313 – 324, (2004).
- 50) M. Schunter, P. Ashley, S. Hada, G. Karjoth, and C. Powers, Enterprise Privacy Authorization Language (EPAL 1.1). International Business Machines Corporation (IBM). <http://www.zurich.ibm.com/security/enterprise-privacy/epal/Specification/index.html>, (2003).
- 51) T. Moses, eXtensible Access Control Markup Language (XACML) Version 2.0. September 2004. Oasis. <http://xml.coverpages.org/XACMLv20CD-CoreSpec.pdf>, (2004).
- 52) The Center for Information Policy Leadership. Multi-Layered Notices Explained. Available at [http://www.hunton.com/files/tbl\\_s47Details/FileUpload265/1303/CIPL-APEC\\_Notices\\_White\\_Paper.pdf](http://www.hunton.com/files/tbl_s47Details/FileUpload265/1303/CIPL-APEC_Notices_White_Paper.pdf). Accessed January 31, 2007.
- 53) J. Tsai, S. Egelman, L. Cranor, and A. Acquisti. [The Effect of Online Privacy Information on](#)

[Purchasing Behavior: An Experimental Study](#). Paper presented at the Workshop on the Economics of Information Security, Pittsburgh, PA, June 7-8, 2007.

- 54) R. Reeder, C. Karat, J. Karat, and C. Brodie, Usability Challenges in Security and Privacy Policy-Authoring Interfaces. In *Proceedings of INTERACT 2007*, 141-155, (2007).
- 55) W.J. Clancey, *Situated Cognition: On Human Knowledge and Computer Representations*. Cambridge University Press, Cambridge, UK, (1997).
- 56) L. Cranor, M. Langheinrich, and M. Marchiori, A P3P Preference Exchange Language 1.0 (APPEL 1.0). Tech. rep., World Wide Web Consortium, <http://www.w3.org/TR/P3P-preferences/>, (2005).
- 57) OASIS (2005). Privacy Policy Profile of XACML v2.0. [http://docs.oasis-open.org/xacml/2.0/PRIVACY-PROFILE/access\\_control-xacml-2.0-privacy\\_profile-spec-os.pdf](http://docs.oasis-open.org/xacml/2.0/PRIVACY-PROFILE/access_control-xacml-2.0-privacy_profile-spec-os.pdf)
- 58) eXtensible Access Control Markup Language 2 (XACML) Version 2.0 OASIS Standard, ([http://docs.oasis-open.org/xacml/2.0/access\\_control-xacml-2.0-core-spec-os.pdf](http://docs.oasis-open.org/xacml/2.0/access_control-xacml-2.0-core-spec-os.pdf)), (2005).
- 59) A. Alfarez, The PGP Trust Model. *EDI Forum*, <http://www.cs.ucl.ac.uk/staff/F.AbdulRahman/docs/>, (1997)