Escape From the Matrix: Lessons from a Case-Study in Access-Control Requirements

Kathi Fisler, WPI and Shiram Krishnamurthi, Brown University

Department of Computer Science
Brown University
Providence, Rhode Island 02912

CS-09-05
May 2009

Escape From the Matrix: Lessons from a Case-Study in Access-Control Requirements

Kathi Fisler
WPI
Worcester, MA, USA
kfisler@cs.wpi.edu

Shriram Krishnamurthi
Brown University
Providence, RI, USA
sk@cs.brown.edu

ABSTRACT

The freedom to share information through the Web has made the ability to restrict that sharing critical. Access-control is thus a central and growing part of contemporary Web-based system security. While much research has focused on languages and analyses for access-control policies, relatively little has studied the actual utterances of people defining policies and how these map to formal policy languages. We study this question using a case-study and report several observations. We identify the several consequences these observations have for the design of policy languages, policy analysis tools, and policy authoring environments. They further suggest directions for future social-science research to help bridge the human-computer gap.

1. INTRODUCTION

The literature on software- and usability-engineering for privacy and security covers systems with widely varying security needs. Non-technical end-users trying to protect against spyware have very different concerns from designers building large software systems that protect substantial financial assets. Security concerns for information sharing in collaborative groups—of the form that the contemporary Web is especially well adapted to serve—lie somewhere in the middle. In these applications, the costs are as much in terms of intangible assets like reputation as they are in theft of goods or the embedding of viruses, and the security decisions are at least as much about encoding the values and culture of the organization as about preventing malice. The increasing number of Web applications supporting such tasks demand methodologies, and ideally software tools, for developing security policies in such settings.

As researchers in policy language design and analysis, we set out to investigate a narrow question about policy language design. Specifically, we had noticed that though many policy languages—both industrial ones such as XACML [24] and EPAL [28] and academic proposals (just a few representative examples are [6, 7, 9, 15])—are essentially defined in terms of role-action-resource triples, in our own experience writing non-trivial policies [16], we often preferred to describe rules in terms of information flows between end-points without having to specify the intermediate operations by which the information may be transmitted. We set out to understand whether this phenomenon held for other users also.

Our work had another, larger, motivation. While there are numerous tools for policy analysis (representative examples are [1, 3, 16, 17, 19, 20, 21, 22, 30]), it is much harder to make these *productive*, both in terms of creating usable interfaces, and making them provide information that users would care about. Yet we had also noted the difficulty of stating concrete verification properties as distinct from the rules themselves [16], and had thus explored change-impact as an analysis modality. We were thus interested in whether we would find common errors that users make, which in turn could inform the design of better tools for authoring, analyzing, and maintaining policies.

To this end, we conducted interviews to gather requirements for a Web-based application to manage faculty job applications for a computer science department.[1] Faculty hiring follows established processes in which the interviewees had participated for years. We quickly learned a great deal more than differences in stating policies. The variations in concerns, approaches, principles and coverage of the security space were striking, both in contrast to the simple and rigid forms of expression in modern policy languages, and in terms of their lessons for policy tool authors. These form the heart of our findings.

We also proceed past merely enumerating findings. Ultimately, these findings are only valuable inasmuch as they lead to better access control policies. Therefore, our description of each finding is accompanied by a concrete explanation of the impact of the finding on the development of *languages* and *tools*. (We do not discriminate heavily between the two categories, because they often represent two different perspectives on the same issue: languages can prevent problems but require their adoption, while tools can identify problems in existing policies.)

[1]This software is available for free: contact the authors for a copy.

Our findings consist of two observations and four lessons. The observations concern how people approached the authoring task—the presence of personality styles, and the absence of a dominant format for stating requirements—while the lessons can be summarized as follows:

- Analogy and relationship are fundamental idioms
- Org. charts should be checked against policies, not drive their creation
- Participants fail to note temporal changes that lead to data leakage
- Social contracts identify and protect the real assets

Before we delve into our findings (Section 4), we first explain the domain (Section 2) and outline the research methodology (Section 3).

2. THE DOMAIN: FACULTY APPLICATION SOFTWARE

Academic computer science departments in the USA increasingly manage the faculty application process online. Applicants submit their materials (vita, statements, etc.) via the Web. The software emails a letter-submission URL to each reference letter writer. Faculty in the department can view and comment on the applications through the Web; certain graduate students may have similar privileges, depending on the department. Administrative staff, such as faculty secretaries and hiring support staff, handle various requests from members of the department. Technical (computing support) staff maintain the infrastructure.

Access control is critical in this domain. Table 1 shows an example access-control matrix for the portion of such a system that end-users utilize.[2] For each class of end-user (row) and type of information (column), the matrix indicates permitted forms of access. For example, this matrix indicates that applicants are permitted to create, read and upload their vitae, whereas faculty may only read the vitae; applicants are not permitted to view department members' comments on the applications; etc. Even in this setup, there can be significant disagreement about some access decisions (as our interviews confirmed): Should applicants be allowed to check which of their reference letters have arrived (and when)? Should students in the department know who has applied? Should administrative staff be able to read the reference letters?

This example already illustrates some of the subtasks involved in articulating an access-control policy: defining roles, resources, and nuances of actions (such as read versus check-for-existence); determining permissions, which sometimes requires careful consideration (as in the question-mark cases); and setting a default decision (such as permit or deny) for any cases that the policy does not cover explicitly. The simple checkbox-style authoring tools in current applications suggests that the matrix-form given by the table presents a useful metaphor for policy authoring: simply ask the user for the roles, actions, and resources, then let them choose actions to complete the cells. Our interviews demonstrate there are numerous limitations, and even perils, to using such an approach.

[2]Other actions, such as opening and closing the search or enabling faculty access to the software, are relevant only to the software administrators and did not arise in our study.

3. RESEARCH METHODOLOGY

The department we interviewed had already used faculty search software for several years, but the application was being rewritten. We used the rewrite as an opportunity to solicit requirements. While past use had clarified workflow and interface issues, an incident the previous year had prompted queries about the system's access-control policy, so we expressly focused on this in our interviews.

For this study we interviewed only professors, at all ranks (tenured, tenure-track, and otherwise), whom we collectively refer to as "faculty". Fifteen faculty were asked (by email) to volunteer time for an interview about the software. Two replied electronically that they had nothing to contribute. One of the remaining thirteen agreed to meet but stated that he had nothing to contribute as others were better suited to developing these requirements.[3] The remaining twelve consented to recorded interviews, each lasting 20–30 minutes.

We started each interview by mentioning the incident that had led to concerns the previous year. Beginning our interviews in this way was helpful, because all participants immediately acknowledged the importance of the requirements process. All participants were then asked for their "security" requirements. One group got no additional prompt; the second was asked for security requirements "such as faculty are allowed to view applications" (to evoke the format of standard access-control rules); the third, "such as submitted applications never become visible to other applicants" (to evoke information flow policies). The three prompts were intended to discern whether prompting would have any effect on the language in which people stated policies. Given that all participants had extensive experience with the problem domain, we did not describe it to them. Once participants had run through the cases that had occurred to them, we asked follow-up questions about roles and resources not yet covered.

Even though we asked about *security*, the vast majority in all three categories focused of their own accord on *access-control*. Only one respondent even mentioned network security (such as SSL). There are many possible reasons for this but, given that some were prompted with an access-control example, and some may have been aware of our own interest in access-control, we only point out this pattern but draw no conclusions from it. Nevertheless, that is why this paper focuses on access-control.

Coding. We transcribed each interview verbatim into a separate file and used open coding [32] on each transcript to identify terms or phrases related to security or access-control. We marked off all references to subjects (including roles),

[3]We will refer to all participants in the masculine, as the sample sizes were too small to suggest conclusions based on gender. Likewise, the sample sizes at each rank were too small to study.

	Applicant Names	Vita	Statements	Letters	Faculty Comments	Student Comments
Applicants	c,r	c,r,u	c,r,u	e?		
Letter writers				c		
Faculty	r	r	r	r	c,r,u	r
Admin staff	r	r	r	e, r?	r?	r?
Tech staff	r	r	r	r?	r?	r?
Graduate Students	r?	r?	r?			c,r,u
Undergraduates						
Postdocs	r?	r?	r?			

Table 1: Hypothetical access-control matrix for portion of system covered in user study. Roles label rows and resources label columns. Letters in each cell indicate classes of permitted actions: c(reate), r(ead), u(pdate) and e(xistence check). The ability to read implies the existence check. A question mark indicates that domain experts do not immediately agree whether that action should be permitted. Empty cells denote denial of all access.

actions, and resources. We marked any statement that described an access condition, whether stated as a rule, a flow of information, or a desired property of the resulting policy. We marked all justifications for rules and all phrases describing security scenarios. We did not mark phrases that were unrelated to security, such as those talking about the general challenges of conducting faculty searches.

Each marked phrase was classified into one of 102 detailed categories within eight major categories. The major categories were verbal transitions between concrete and abstract details, form of stating access-requirements (rules, workflows, etc.), what sorts of additional information were expressed with a requirement (rationales, exceptions, concerns), what form of information was used to raise a new requirement (a role, a resource, etc.), what sorts of meta- or reflective information was raised (such as opinions of others or self-doubt about a particular requirement), disclaimers (about cases not considered), comments about the policy implementation, and the default decision (permit or deny) that should apply to the policy.

4. FINDINGS

Our initial investigation was to determine how our prompts affected the nature of requirements that participants provided. This question was easy to settle: participants employed role-action-resource triples, information-flow rules, and several other kinds of utterances to state their desired policy. We identified no connection between our prompts and the pattern of their responses.

In the process of conducting the interviews, however, we encountered many other patterns. In the rest of this document, we focus on these other findings. For each finding, we describe potential impact on the design of languages and on tool support (though some findings impact only one of these, not both).

4.1 Analogy and Relationship are Fundamental Idioms

Participants' reliance on analogy was striking, particularly given the variety of forms in which it arose. They used analogies both to express rules and to justify them, often combining the two. They routinely phrased rules or rationales in terms of relationships between roles and resources. Nine of the twelve participants stated at least some rules using the form "treat X the same as Y" (possibly "with the exception of Z").

For instance, participants often drew analogies between faculty hiring and the tenure process to decide on the accesses that junior faculty and graduate students should have to application materials:

> This is sort of like the tenure situation. [Graduate students] don't have the professor job yet, so they shouldn't see the letters that are for getting that professor job.

They sometimes concretized their analogies with metrics:

> The distance between grad student and junior faculty member is greater than the distance between assistant professor, however new, and full professor, however senior.

They justified analogies by role overlaps:

> Default should be [the technical staff] get the same access as grad [students] because some of them are grads.

Role hierarchies are common in access-control contexts and were often identified through such scenarios; Section 4.2 discusses role hierarchy in more detail.

Sometimes the analogies were used to express a negative, rather than positive, expectation:

> The person might come here and student might be their advisee [...] but the level of interaction is not ... the same, I guess, right, I mean how you interact with [senior faculty] is not equal to the way advisee/advisor ...

Finally, some participants used analogy to express rules, rather than to justify them:

> Whatever we decide to do for discussions in the faculty meeting ought to hold for the software.

Participants often spoke of relationships between the sets of permissions accorded particular roles. In one of the more interesting expressions of rules, one participant wanted permissions of one role to lie between those of two other roles, but didn't know what set of restrictions might achieve that: "we'll give that person a certain level of privilege so they can do whatever that faculty member needs" while making it less than full faculty privilege. Other participants stated similar requirements of wanting a rule in a space that would satisfy a goal without being too restrictive or too permissive. One participant justified a rule on the grounds that "we have to draw the line somewhere"; in this case, making sure two roles were *not* analogous seemed necessary, even though the participant couldn't articulate a difference that should distinguish them.

Languages and Tools. This suggests two concrete possibilities for developers of languages and tools:

1. Current access-control languages usually provide no idioms for defining rules via analogy or relationships. These idioms are particularly important for maintaining the policy over time. On the other hand, *a blind reliance on analogy is dangerous*! A connection between two roles now may not persist into the future, and must therefore be repeatedly re-evaluated as the rights and responsibilities of either role changes.
2. When users are trying to study the effect of a policy, it would be helpful to report apparent similarities in the policy (especially if two distinct roles have exactly the same privileges). Policy authors would find it useful to determine whether such similarities are incidental and, indeed, even true (e.g., thinking about how the roles differ in real life may help identify missing policy cases).

4.2 The Org Chart is Dead, Long Live the Org Chart

Security-requirements processes often start from whatever documentation an organization has available about the system to be secured and the staff who will use it [2, 18, 34]. Organizational charts are potentially useful starting points for specifying access control, as they suggest preliminary roles and relative levels of responsibility within the organization. Hierarchies between roles with respect to levels of privilege emerged naturally during our interviews. However, in two cases, *the privilege hierarchy that emerged from the interviews reversed or contradicted relationships that would have been in the org chart.* (Though it is well-known in organization theory that organizations function thanks to an "informal organization" that is quite different from that of the formal list of connections [4], this disconnect has sometimes been neglected in prior security-requirements work.)

- The department has two levels of administrative staff: secretaries who support faculty members, and executive staff who support the department (office manager, grants administrator, etc.). The secretaries report to a member of the executive staff and generally have less access to sensitive information. Interview discussions of staff privileges tended to center around what access secretaries need to assist the faculty. Many participants noted that the executive staff have no need for access to application materials as part of their jobs (unless they were covering for an unavailable secretary), but granting them access was fine because they have access to so much other confidential information (such as faculty salaries) anyway. This consensus aside, however, at least one participant remarked that the executive staff have no need for access to the materials according to the principle of least privilege, whereas secretaries have a strong case for access.

- Most department processes regard post-docs as lying between graduate students and faculty. Giving postdocs access to faculty applications is problematic, however, as they are often on the market themselves and shouldn't see information about their competitors. As postdocs have no official role in the overall hiring process, participants generally felt they should have no access to the system. One participant raised a related, but more delicate, scenario of current graduate students on the market:

  > It's hard, though, because [graduate students] can access [materials] until they apply. Ethically, if they think of applying [to our department] they should remove themselves from they process. Actually, if they are looking for a job they should remove themselves from the process, ethically, because any decisions made here might affect where they are applying.

  This raised a new sub-category of students that would not have been reflected in an org chart designed for general departmental operating procedures.

These observations suggest that the org chart is actually dangerous as a starting point for forming role-hierarchies within policies. The chart is a document vested with authority, so people are more likely to grant privileges in accordance with the chart than to (even think to) contradict or expand it. This can result in leakage of sensitive data.

Languages and Tools. Does this mean the org chart is dead as an authoring aid? Not at all! By virtue of being a formal document, the chart can serve as an input to policy analysis tools to serve as a specification against which to look for potential flaws in a policy. For example, if two roles' privileges are contravariant to their positions in the org chart, the authoring process should ensure this was intentional. Documenting why this is so is bound to be instructive to future maintainers of the policy and software.

4.3 Policy Authors are Poor at Tracking Roles in Space and Time

An individual's access privileges can change over time even as the policy remains fixed: an individual's role might change, or the access guards might depend on the status of a resource (such as whether an application is complete). Role overlaps and changes, in particular, can result in information leaks.

Only four participants raised the possibility of overlaps and changes; interestingly, these were in different contexts and at different granularities of time. One cited graduate students who become applicants later in the same year (at which point they should lose access to information on other applicants). Two cited applicants who become faculty members in later years (who should not be able to see their letters and comments from when they were applicants). One considered former students who applied years after leaving but who had meanwhile retained their computer accounts. One of these four also pointed out that people who have written recommendations may have themselves applied. Less directly, a fifth participant considered relationships over time in the context of conflict of interest, where an applicant's advisor from another institution might now be on the faculty in this department.

We hypothesize that the nature of faculty hiring masks thinking about dynamic changes. The application process is so cyclic—a few months of intense activity, followed by many months or years of none—and while a search is underway the focus is so heavily on the current year, that interviewees seem to forget to consider time outside the narrow window of one search.

Languages and Tools. Role overlaps are a common source of errors in policies [16], and permitting them by default should perhaps be considered a mis-feature of policy languages. The importance of handling temporal changes in policies has been recognized by many authors (representative examples are [6, 8, 13]). In some cases, a richer policy language might indeed help avoid problems. Our study demonstrates, however, that simply adding such features to the language is not enough. Authoring environments must explicitly account for the possibility that authors may forget about the consequences of the passage of time and, for instance, explicitly query authors about the nature of role changes over time and their potential to leak information.

4.4 Social Contracts Identify and Protect the Real Assets

Thinking about access-control policies in terms of concrete resources, as tabular- or rule-based authoring does, can sometimes entirely miss the point. In our case, the interviews revealed that the single most important resource was one that shows up nowhere in tables and rules: the department's *reputation.* Often, this was the resource that people were really trying to protect, even though they were stating concrete rules about other (tangible) resources based on how they thought those decisions would impact this resource.

To identify such hidden resources, it helps to understand the social contracts at work, because they often frame assumptions that people make on how their information will be handled. These assumptions should influence policy decisions, but are often overlooked. One participant raised assumptions within the department about how faculty comments would be shared. Only one participant spoke of such pacts from the perspectives of either applicants ("When you write something to a department you assume that the faculty is going to see it. I don't think you assume much at all as far as the students are concerned.") or letter writers ("I think non-tenured faculty should not be able to see that. Again, it's part of the pact you make with a letter writer."), even though several participants spoke about potential problems with letters comparing applicants to junior faculty or graduate students in the department.

Only one participant explicitly raised the department's reputation: he felt that students should not know who applied, because the students might publicly criticize the department for not interviewing a particular person (without having had access to the data and discussion that led to that decision). One thought "it doesn't look good for the department" to give applicants too much flexibility in marking portions of their application as highly confidential. Another cited standard practice among schools in telling applicants whether their letters had been received. One person justified a decision based on loyalty, which could be viewed as indirectly protecting reputation:

> Students, we know, have a very finite lifetime and loyalty to the department, whereas the assumption is that every assistant professor who's here is here for life.

The importance of reputation as an asset may have motivated comments from a few participants (most of whom made the comments referenced in this section) about the tension between the educational value and (unstated) consequences of letting students have too much access to application materials.

As important as identifying the real assets is, it surprisingly arose in only a few interviews. We had several hypotheses for why this was the case. Since letter writers and applicants are (generally) outside the department, participants could have been taking an internal rather than external view of the process. All participants had been either an applicant or a reference-letter writer in recent years, so lack of experience with those roles is not a plausible factor. Several participants mentioned applicants and letter writers, but either in the context of the user interface or saying that they had very few privileges. These reflect a "use-case" approach, which focuses on the tasks that a user tries to perform with a system. Trust assumptions aren't affiliated with particular actions; it might be harder for participants to recall them in absence of a hook to the vocabulary of users and tasks common in software engineering.

Languages and Tools. Identifying the assets to protect is part of every security requirements process. The trap illustrated here is that the assets to protect are not necessarily the resources under access-control, despite what the very

idea of access-control might suggest. Explicating such "hidden" resources is therefore invaluable, because it helps shift the discussion in a useful direction articulated by the requirements literature (from the "solution space" to the "problem space"). Lightweight security-requirements processes that focused simply on tangible resources could easily fall into a trap of failing to do this.

4.5 Participants Exhibit Personality Styles

In listening to the interviews and re-reading the transcripts, we were struck by the different approaches—bordering on "personalities" or "styles"—that participants adopted during the interviews. We were curious whether these correlated to any patterns in the forms of rules used during the interviews. Four strong styles emerged from our high-level reading of the interviews: **social thinkers** were conscious of the values that policies encoded regarding department culture, collegiality, and social contracts with applicants and letter writers; **problem avoiders** saw policy as protecting against undesirable situations; **pragmatists** focused on realities such as making sure policies wouldn't interfere with workflow or on granting access based on similar data available to users from outside the system; **protectionists** framed most comments around what principles of confidentiality or least privilege would demand.

We were able to assign each participant to one of the four styles based on our high-level reading of their interview and the kinds of justifications or concerns they raised first when discussing a new aspect of the policy. The assignments are obviously somewhat subjective, as the nuances that put a participant in one style or the other were beyond the level of detail captured in our coding of the transcripts. We identified two social thinkers, two problem avoiders, and four each of pragmatists and protectionists. One each of the pragmatists and protectionists fell into their category based more on the justifications they used rather than our high-level reading (their high-level reading suggested no particular style). We will use the modifiers *weak* and *firm* to distinguish these participants and the rest of their groups, when appropriate.

The pragmatists were fairly diverse in their chief concerns. One was most concerned about not having the policy interfere with workflow; one wanted to make sure people had the data they needed to participate in discussions; one believed that the information was available through other (possibly human) channels anyway so security wouldn't achieve much; and one felt that electronic information had low chance of staying secure anyway. This group was more likely than the others to state rules fairly abstractly ("give [staff] a certain level of privilege so they can do whatever that faculty member needs", or "I want to work on paper"). However, most did have clear boundary cases for which they felt strict access controls were essential.

The pragmatists and the protectionists showed strong similarities on certain issues. The participants who needed the most prompting for concrete policy statements came from these two groups. All rules stated as "whatever is needed to achieve X" came from these groups. These groups frequently stated broad rules from which the interview process identified special cases: seven of eight participants in these categories followed this pattern whereas only one of four participants in the other two categories did so. These two groups were also more likely to state rules in the form "treat X like Y", though all four groups used analogies in formulating their policies. Although half of the participants asked whether they were missing any important cases, only one was outside these two groups (in the social group). All of the comments that weighed the risk or benefit of taking a particular decision came from these two groups. All but one comment about temporal changes arose from these two groups.

The protectionists differed most dramatically from the pragmatists in never referencing the perspectives of other faculty or users of the system. All but one participant outside of the firm protectionists cited a colleague with a different opinion or commented that an access decision might benefit from department-wide discussion. Not a single firm protectionist made such a comment. Similarly, only one comment about social-contract assumptions of users outside the department came from this group, as opposed to three from the pragmatists and problem-avoiders and many from the social thinkers. The protectionists were far less likely to have discussed the interface, implementation, or other security issues involved in the system: three comments of this sort came from protectionists (two of those from the weak protectionist), in contrast to five from the social thinkers and a dozen or more from each of the problem-avoiders and pragmatists.

The social thinkers contributed only one comment about the processes surrounding the hiring software, such as the tenure processes required for senior applicants or broad process problems that the new software might help address. The problem-avoiders and protectionists each made roughly ten comments in this space; the pragmatists raised only a couple. Perhaps not surprisingly, both of the problem-avoiders had administrative experience. They almost always stated rules very concretely, rather than through or relative to general principles.

Another "personality"-like issue, but one that was less dominant than those cited here, was whether the participant expressed discomfort with or preference to not be articulating policy. One participant outside of the twelve was very uncomfortable with the responsibility of suggesting policy. One participant clearly stated that he would rather someone else be tasked with doing this (despite his having some strong opinions).

Languages and Tools. We suspect the protectionists lacked a strong mental model of the hiring process or of previous software versions to guide their comments, so they fell back by rote on the principle of least privilege.[4] Seven of the eight comments asking us what cases had been overlooked came from the pragmatists or protectionists. Fortunately, once given specific requests or scenarios to consider (as some traditional requirements processes suggest), both groups gave requirements at similar level of detail to the others. Under-

[4]The pat reduction of "security" to "least privilege" is reminiscent of the Father Sarducci "five-minute university" comedy sketch from Saturday Night Live, which, for instance, summarized all of economics as "supply and demand".

standing the user's "personality" is essential to reconciling their requirements with those of others.

4.6 Participants Lack a Dominant Structure or Consistent Format

Given the technical expertise of the participants, we expected interviews to reveal a systematic process for exploring the policy space. Only one of the twelve interviews had such a structure; in that case, the participant worked through the roles, citing principles and scenarios to present the decisions relevant to each one, or to refine a general role into sub-roles. No others exhibited such a clear organizing principle. Each of roles, resources, scenarios, and the existing process was used by a third to a half of the participants to initiate a new thread of conversation ("So let's now consider ..."). While this does not imply that participants would have had difficulty articulating policy against a single organizing structure, it does raise questions of whether a single organizing principle fits all, and whether using a variety of prompts will better cover the state space.

The different prompts we gave at the start of the interview did not correlate with particular structuring techniques or forms of rules. Each participant used multiple forms to state access-control rules. The standard rule format of "role R is permitted/denied to do action A on resource S" was common but not ubiquitous. Most participants stated four to six such rules, with some rules covering multiple resources (i.e., "faculty should be allowed to see everything"). Half of the participants stated rules of the form "whatever access role R needs to do task T". In these cases, role R was always a service role (such as administrative or systems staff); task T was described vaguely as "fix something".

Notwithstanding the use of least-privilege as discussed in Section 4.5, we were surprised at how infrequently participants invoked other standard security metaphors. Almost nobody raised conflicts of interest without prompting; even when prompted, many had trouble identifying them. Nobody raised (even if only to dismiss) separation-of-duty. Only one person tried to classify resources by levels of privilege:

> Let's distinguish between the more privileged and the less privileged. The letters or recommendation are privileged and everything else is, I think ... the privileged is the letters and the faculty remarks and everything else is nonprivileged.

Languages and Tools. The difficulty that participants had in methodically articulating a policy was telling, and indicates the need for policy authoring processes and tools that embody them. In particular, it is useful to recognize that the user's mental model of security may be very limited (e.g., the persistent reliance on least privilege). Furthermore, it may be useful to inform users about other paradigms (e.g., separation-of-duty) to determine whether they are manifest in the domain. Of course, these processes must take into account the constraints of the domain and not burden users with processes that demand too much material, time or cost (since such processes will, ultimately, be ignored).

We also found that five participants stated rules in terms of the source of some information, such as "grads see materials that candidates provide". While these are close to the standard form, these have two implications for languages and tools. First, the abstraction of who created a resource is not part of the standard matrix representation, though it can be captured in models that reflect provenance. Second, these expectations are closer to information flow than access-control rules.

5. CONTEXT AND FUTURE WORK

The interviews in this case study raise several research questions and issues for designers of policy languages and authoring and analysis tools. Our findings indicate that significant research remains to be done into cognitive aspects of policy authoring and their impact on the form of policy languages. Much policy language research focuses on expressive power from a purely logical perspective. While such work is clearly valuable, it ignores fundamental questions of how well the languages lend themselves to capturing what people are trying to say in a form that makes policies easy to understand and maintain.

The literature on security and privacy is increasingly focusing on such human conditions as trust. Our interviews confirm that these issues are arising as people are thinking about what policies they want to write: informal workflows, reputation as an asset, the extent to which students should be involved, etc., are all examples of trust decisions masquerading as access-control decisions. Determining how to account for these concerns in formal languages is an important and interesting area for future work.

We conjecture that these studies can inform a human process of interviewing users for their requirements. In many systems, however—especially those where end-users are setting policies in isolation, such as in collaborative Web-based applications—it is not cost-effective to deploy an army of requirements-gatherers. In such scenarios, we believe there is considerable room for what we dub an *inquisitive environment* that takes the place of today's passive policy-entry interfaces, asking users questions ("Does this role have subroles?", "Does this datum have parts?", and so on) to tease out the actual requirements, instead of blindly trusting the user's original selections.

Threats to Validity. The most obvious threat is, of course, that the interviewers were anthropological "natives", which is a general problem that arises when people related to organizations study them.

The participants knew that they were not solely responsible for determining the security policy. Although they did not know who else had agreed to participate, they knew that the final policy would be set by a combination of the developers (whom they trusted) and department-wide discussion, if necessary, on controversial issues. Despite this, we saw ample evidence that people took these interviews very seriously. All but one participant made a comment such as "what cases am I missing?", "I'm trying to think this through", or "others may feel differently". Many also expressed thanks that the development team was taking the time to gather the

requirements and to build the software properly.

Limits to Generalizability. Several aspects of this study limit the applicability of our findings:

1. The participants had extensive experience with the domain and the process, on paper and/or as software. They had all reviewed applications, and had either written reference letters or applied recently, or both.

2. All the participants were computer science faculty, so fundamental concepts of security and requirements were presumably familiar to them.

3. The set of users was well-understood, and with the exception of the applicants and letter writers, most participants personally knew most users.

We also did not try to account for political oddities of an academic department, which may be different from those of other organizations.

Working with these participants did have a disadvantage: they often pre-judged that a scenario would be "too hard to implement", and thus avoided distinctions between roles and scenarios of their own accord. We frequently had to remind them to stick to requirements rather than filtering their views through their (often inaccurate!) perceptions of how the system would be implemented.

To maintain the anonymity of our subjects, we cannot discuss details of their expertise in areas of security. Nevertheless, in our qualitative opinion, we did not notice any significant correlation between security experts (as defined by reputation or publication areas/titles) and the quality of their responses.

6. RELATED WORK

Existing works on requirements engineering for security [2, 18, 34] propose processes to get stakeholders to articulate concerns such as threats, vulnerabilities, trust assumptions, and security goals, as a key step toward developing security requirements. These processes are valuable, but seem to depend on a significant allocation of human resources. It isn't immediately obvious how to scale these processes down to small groups or end-users, nor to collaborative authoring contexts in which access-control concerns dominate broader security questions.

Nissenbaum [25] distinguishes between security and trust, where the former is a technical feature and the latter a social one. She views security as neither necessary nor sufficient for trust. Our interviews reinforce this. Trust featured in several comments, sometimes justifying a security decision, but often as a social construct with which the security policy should be consistent. Some comments asked for rules that gave one role a little less privilege than another in order to reflect differing levels of trust, even though the participants could not articulate a usage scenario that should distinguish the two.

Palen and Dourish [27] view privacy as "the continual management of boundaries between different spheres of actions and degrees of disclosure within those spheres". Their work defines three broad boundaries: disclosure, identity, and temporality. Although our interviews asked about security rather than privacy, forms of these boundaries arose throughout our interviews. Determining how to incorporate a boundary-based perspective into policy languages and authoring tools is an interesting open challenge in this area. Dourish and Anderson [14] describe three models of privacy and security that account for their social dimensions. Our interviews reflected their economic and practical-action models; we did not look for patterns in the interviews that would reflect their discursive model. Psychological factors associated with risk assessment, such as Tversky and Kahneman's availability hypothesis [33], did seem to be a factor in our interviews, as illustrated by the extensive prompting some participants required.

Our work raises basic questions about the structure of rules and idioms supported in formal policy languages. Most languages use role-based access-control (RBAC) [15] as a foundation. Trust management [9] enhances RBAC with support for delegation and credentials. These concepts *implement* rather than *specify* trust relationships, and are thus of limited help in maintaining them over time.

Existing research on usability issues in policy authoring tend to focus on specific interface issues rather than the broad questipns about policy specification that we have raised here. Zurko, et al.'s ADAGE authoring framework [36] was explicitly designed for usability and to study expressiveness questions such as the use of various grouping mechanisms (such as roles and labels) for expressing policies, but focuses more on interface issues than on the underlying language (with the exception of strong support for separation-of-duty, which did not arise in our case study). Brostoff et al. [11] developed a tool to help scientist end-users author access-policies; their work focuses more on how to help end-users specify policy components. Both used fairly standard rule-based structures for capturing policies. Karat et al.'s SPARCLE authoring tool [10] uses natural-language processing to improve the interface between humans and RBAC-style languages. Focusing on interfaces to RBAC misses the bigger (and well-documented) problem of stakeholders not articulating what they actually mean. The rules as originally uttered by our participants would have resulted in a dozen underspecified, self-contradictory, and mutually inconsistent policies. Some of these problems persisted even after we prompted participants. Better interfaces are clearly critical for policy authoring tools, but getting a policy that is both correct and maintainable requires more cognitive research on how people articulate their concerns None of these papers report on cognitive studies justifying the linguistic structures through which people describe policies.

Reeder, et al. identify five usability challenges for policy authoring tools [29], based on a study in which novice policy authors tried to write formal versions of existing organizational privacy policies. Three of these challenges could apply in an interview setting and all arose in our interviews. Two of these three (conflicting rules and statements of denied access rather than permissions) were, however, not a problem

in our context, as we allowed negative rules and handled conflicting rules through additional questioning. The third challenge, called "group ambiguity", occurred when one term in the existing policy referred to others collectively. Our participants often had the opposite problem: they failed to decompose an aggregate term (e.g., "application") into components that required different access rules (e.g., the vita versus the list of letter writers). That study was more structured than ours by virtue of providing the policies, so the other issues we identified would not have applied in their work.

Prior studies have identified different personalities among users in security-related contexts. Both Cranor et al. [12] and Olson et al. [26] identify privacy personalities in studies about factors influencing people's privacy preferences. These personalities cluster around the different levels of privacy that people sought, rather than the styles in which they expressed their privacy concerns. While our pragmatists appear to have proposed less restrictive policies than our protectionists, we did not formally analyze our results against this metric. Miller and Edwards [23] identify similar sharing personalities in the context of photo-sharing sites. While their interviews extended far beyond privacy, comments similar to those raised by our social thinkers were common when discussing sharing of photos showing family members. The only other work we have found that asks users to articulate their policies in a free-form context is by Bauer et al. [5], which is part of a general comparison of two policy mechanisms. These papers do not report details about how users expressed their desired policies.

Classical research on how people reason, such as the Wason Selection Task [35], shows that people perform poorly from a logical perspective. Newer work by authors such as Stenning and Van Lambalgen [31] shows that people often reason under assumptions that parameterize the logical system in question (such as assuming the law of excluded middle, or presuming that exceptions are possible for every logical assertion). This body of work has serious implications for policy authoring, as software systems interpret policy rules under a fixed logic that may differ from what the policy author had in mind. Our case study was partly inspired by this body of work. With a better understanding of the assumptions people make when stating policies, we can better align this potential mismatch of logics.

7. CONCLUSION

This paper describes a case-study in interviewing end-users on their desired security policy. By using computer science professors, we were able to eliminate several variables relating to the education and technical sophistication of the users. Despite this filtering, which we expected might produce significant conformity, we still found several (potentially) surprising differences in their views as well as weaknesses in their ability to state an accurate policy.

Our observations have implications not only for the design of policy languages but also for the construction of tools for the authoring and analysis of policies. In particular, our findings suggest that, no matter how much effort we expend on designing policy languages, a passive authoring process is potentially dangerous, as many mistakes originate in policy authors' inability to consider the consequences (across role overlaps, time, etc.) of their choices. We have therefore suggested concrete actions that an authoring environment can undertake to mitigate these mistakes.

8. REFERENCES

[1] Tanvir Ahmed and Anand R. Tripathi. Static verification of security requirements in role based CSCW systems. In *Symposium on Access Control Models and Technologies*, pages 196–203, 2003.

[2] Annie I. Antón and Julia B. Earp. Strategies for developing policies and requirements for secure e-commerce systems. In A. K. Ghosh, editor, *Recent Advances in E-Commerce Security and Privacy*, pages 29–46. Kluwer Academic Publishers, 2001.

[3] Michael Backes, Günter Karjoth, Walid Bagga, and Matthias Schunter. Efficient comparison of enterprise privacy policies. In *Symposium on Applied Computing*, pages 375–382, 2004.

[4] Chester I. Barnard. *The Functions of the Executive*. Harvard University Press, 1938.

[5] Lujo Bauer, Lorrie Faith Cranor Robert W. Reeder, Michael K. Reiter, and K. A. Vaniea. User study of policy creation in a flexible access-control system. In *SIGCHI Conference on Human Factors in Computing Systems*, 2008.

[6] Moritz Y. Becker and Sebastian Nanz. A logic for state-modifying authorization policies. In *European Symposium on Research in Computer Security*, 2007.

[7] Moritz Y. Becker and Peter Sewell. Cassandra: Flexible trust management, applied to electronic health records. In *IEEE Computer Security Foundations Workshop*, 2004.

[8] Elisa Bertino, Piero A. Bonatti, and Elena Ferrari. TRBAC: A temporal role-based access control model. *ACM Transactions on Information and Systems Security*, 4(3):191–233, 2001.

[9] Matt Blaze, Joan Feigenbaum, and Jack Lacy. Decentralized trust management. In *IEEE Symposium on Security and Privacy*, 1996.

[10] Carolyn A. Brodie, Clare-Marie N. Karat, and John Karat. An empirical study of natrual language parsing of privacy rules using the SPARCLE policy workbench. In *Symposium on Usable Privacy and Security*, 2006.

[11] S. Brostoff, M.A. Sasse, D. Chadwick, J. Cunningham, U. Mbanaso, and S. Otenko. R-what? development of a role-based access control (RBAC) policy-writing tool for e-scientists. *Software: Practice and Experience*, 35(9):835–856, 2005.

[12] L. Cranor, J. Reagle, and M. Ackerman. Beyond concern: Understanding net users attitudes about online privacy. Technical Report TR 99.4.3, AT&T Research Labs, April 1999.

[13] Daniel J. Dougherty, Kathi Fisler, and Shriram Krishnamurthi. Specifying and reasoning about dynamic access-control policies. In *International Joint Conference on Automated Reasoning*, pages 632–646, August 2006.

[14] Paul Dourish and Ken Anderson. Privacy, security... and risk and danger and secrecy and trust and identity and morality and power: Understanding collective

information practices. Technical Report UCI-ISR-05-1, UCI Institute for Software Research, Irvine, Ca., 2005.

[15] David F. Ferraiolo and D. Richard Kuhn. Role-based access controls. In *NIST-NSA National Computer Security Conference*, 1992.

[16] Kathi Fisler, Shriram Krishnamurthi, Leo A. Meyerovich, and Michael Carl Tschantz. Verification and change-impact analysis of access-control policies. In *International Conference on Software Engineering*, pages 196–205, May 2005.

[17] Dimitar P. Guelev, Mark D. Ryan, and Pierre-Yves Schobbens. Model-checking access control policies. In *Information Security Conference*, Lecture Notes in Computer Science. Springer-Verlag, September 2004.

[18] C. Haley, R. Laney, J. Moffett, and B. Nuseibeh. Security requirements engineering: A framework for representation and analysis. *IEEE Transactions on Software Engineering*, (to appear) 2007.

[19] Graham Hughes and Tevfik Bultan. Automated verification of XACML policies using a SAT solver. In *International Conference on Web Engineering, Workshop on Web Quality, Verification and Validation*, pages 378–392, 2007.

[20] M. Koch, L. V. Mancini, and F. Parisi-Presicce. On the specification and evolution of access control policies. In *Symposium on Access Control Models and Technologies*, pages 121–130, 2001.

[21] Grzegorz Kolaczek. Specification and verification of constraints in role based access control for enterprise security system. In *International Workshop on Enabling Technologies: Infrastructure for Collaborative Enterprises*, pages 190–195, 2003.

[22] Vladimir Kolovski, James Hendler, and Bijan Parsia. Analyzing web access control policies. In *International World Wide Web Conference*, 2007.

[23] Andrew D. Miller and W. Keith Edwards. Give and take: a study of consumer photo-sharing culture and practice. In *ACM SIGCHI Conference on Human Factors in Computing Systems*, pages 347–356, 2007.

[24] T. Moses. eXtensible Access Control Markup Language (XACML) version 1.0. Technical report, OASIS, February 2003.

[25] Helen Nissenbaum. Will security enhance trust online, or supplant it? In Roderick M. Kramer and Karen S. Cook, editors, *Trust and Distrust in Organizations: Dilemmas and Approaches*, chapter 7, pages 155–188. Russell Sage Foundation, 2004.

[26] Judith S. Olson, Jonathan Grudin, and Eric Horvitz. A study of preferences for sharing and privacy. In *ACM SIGCHI Conference on Human Factors in Computing Systems*, pages 1985–1988, 2005.

[27] Leysia Palen and Paul Dourish. Unpacking "Privacy" for a networked world. In *ACM SIGCHI Conference on Human Factors in Computing Systems*, 2003.

[28] Calvin Powers and Matthias Schunter. Enterprise privacy authorization language (EPAL 1.2). W3C Member Submission, November 2003.

[29] Robert W. Reeder, Clare-Marie Karat, John Karat, and Carolyn Brodie. Usability challenges in security and privacy policy-authoring interfaces. In *INTERACT (2)*, pages 141–155, 2007.

[30] Andreas Schaad and Jonathan D. Moffett. A lightweight approach to specification and analysis of role-based access control extensions. In *Symposium on Access Control Models and Technologies*, pages 13–22, 2002.

[31] Keith Stenning and Michiel van Lambalgen. *Human Reasoning and Cognitive Science*. MIT Press, 2008.

[32] A. Strauss and J. Corbin. *Basics of qualitative research: Grounded theory procedures and techniques*. Sage Publishers, 1990.

[33] Amos Tversky and Daniel Kahneman. Judgement under uncertainty: heuristics and biases. *Science*, 185:1124–1130, 1974.

[34] Axel van Lamsweerde. Elaborating security requirements by construction of intentional anti-models. In *International Conference on Software Engineering*, pages 148–157, 2004.

[35] Peter Cathcart Wason. Reasoning. In B. M. Foss, editor, *New Horizons in Psychology I*. Penguin, 1966.

[36] Mary Ellen Zurko, Rich Simon, and Tom Sanfilippo. A user-centered, modular authorization service built on an RBAC foundation. In *IEEE Symposium on Security and Privacy*, pages 57–71, 1999.