User-centered Design of Identity and Access Management Systems

by

Pooya Jaferian

MSc, Amirkabir University of Technology, 2006

BSc, Amirkabir University of Technology, 2004

A THESIS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF

Doctor of Philosophy

in

THE FACULTY OF GRADUATE AND POSTDOCTORAL STUDIES
(Electrical and Computer Engineering)

The University of British Columbia
(Vancouver)

November 2014

© Pooya Jaferian, 2014

Abstract

IT security management (ITSM) technologies are important components of IT security in organizations. But there has been little research on how ITSM technologies should incorporate human and social issues into their design. Identity and Access Management (IAM) systems, as an important category of ITSM, share such a gap with other ITSM technologies. The overreaching goal of this research is to narrow the gap between IAM technologies and social context.

In the first phase, we developed a set of usability guidelines, and heuristics for design and usability evaluation of ITSM tools. We gathered recommendations related to ITSM tools from the literature, and categorized them into a set of 19 high-level guidelines that can be used by ITSM tool designers. We then used a methodical approach to create seven heuristics for usability evaluation of ITSM tools and named them ITSM heuristics. With a between-subjects study, we compared the usage of the ITSM and Nielsen's heuristics for evaluation of a commercial IAM system. The results confirmed the effectiveness of ITSM heuristics, as participants who used the ITSM heuristics found more problems categorized as severe than those who used Nielsen's.

In the second phase, we conducted a field-study of 19 security practitioners to understand how they do IAM and identify the challenges they face. We used a grounded theory approach to collect and analyze data and developed a model of IAM activities and challenges. Built on the model, we proposed a list of recommendations for improving technology or practice.

In the third phase, we narrowed down our focus to a specific IAM related activity, access re-

view. We expanded our understanding of access review by further analysis of the interviews, and by conducting a survey of 49 security practitioners. Then, we used a usability engineering process to design AuthzMap, a novel user-interface for reviewing access policies in organizations. We conducted a user study with 430 participants to compare the use of AuthzMap with two existing access review systems. The results show AuthzMap improved the efficiency in five of the seven tested tasks, and improved accuracy in one of them.

Preface

The materials in chapters 3 to 6 of this dissertation have each been either published or accepted for publication. The author of this dissertation conceived of the research idea, performed all the design and evaluation, except in Chapter 6 where the design and execution of the study #3 were shared by other co-authors. He also wrote all the papers resulting from this research, under the supervision of the co-authors who provided feedback and guidance throughout the research process. Below are the ethics information, and publication details for each chapter.

- Chapter 3: The materials of this chapter has been published in the CHIMIT (Computer Human Interaction for Management of Information Technology) conference. The interviews in this chapter were approved by the UBC's Behavioral Research Ethics Board (Certification number: H06-80413, Project title: HOT Admin: Human, Organization, and Technology Centred Improvement of IT Security Administration).

  Pooya Jaferian, David Botta, Fahimeh Raja, Kirstie Hawkey, and Konstantin Beznosov. 2008. Guidelines for designing IT security management tools. In Proceedings of the 2nd ACM Symposium on Computer Human Interaction for Management of Information Technology (CHiMiT '08). ACM, San Diego, CA, USA, Article 7 , 10 pages. (acceptance rate: 29%)

- Chapter 4: The materials of this chapter has been published in CHI Extended Abstracts, Symposium On Usable Privacy and Security (SOUPS), and Human-Computer Interaction Journal. The user study conducted in this chapter was approved by the UBC's Be-

havioral Research Ethics Board (Certification number: H09-03371, Project title: ITSM Heuristics).

Pooya Jaferian, Kirstie Hawkey, Andreas Sotirakopoulos, Maria Velez-Rojas, and Konstantin Beznosov. Heuristics for evaluating IT security management tools. Human-Computer Interaction, Vol. 29, Iss. 4, 2014. (impact factor: 3.039)

Pooya Jaferian, Kirstie Hawkey, Andreas Sotirakopoulos, Maria Velez-Rojas, and Konstantin Beznosov. 2011. Heuristics for evaluating IT security management tools. In Proceedings of the Seventh Symposium on Usable Privacy and Security (SOUPS '11). Pittsburgh, PA, USA, Article 7 , 20 pages. (acceptance rate: 33%, best paper award)

Pooya Jaferian, Kirstie Hawkey, Andreas Sotirakopoulos, and Konstantin Beznosov. 2011. Heuristics for evaluating IT security management tools. In CHI'11 Extended Abstracts on Human Factors in Computing Systems (CHI EA '11). ACM, Vancouver, BC, Canada, 1633-1638. (acceptance rate: 45%)

- Chapter 5: A subset of materials of this chapter has been published in the the CHIMIT (Computer Human Interaction for Management of Information Technology) conference. The interviews in this chapter was approved by the UBC's Behavioral Research Ethics Board (Certification number: H08-02527, Project title: IdM CA).

  Pooya Jaferian, David Botta, Kirstie Hawkey, and Konstantin Beznosov. 2009. A case study of enterprise identity management system adoption in an insurance organization. In Proceedings of the Symposium on Computer Human Interaction for the Management of Information Technology (CHiMiT '09). Baltimore, MD, USA, Article 7 , 10 pages. (acceptance rate: 33%)

- Chapter 6: The initial versions of this chapter has been published as a CHI extended abstract. Most of the material in the chapter has been accepted for publication in SOUPS conference. The studies presented in this chapter has been approved by the UBC's Behavioral Research Ethics Board: (1) Access Review Survey (Certification number: H08-02527, Project title: IdM CA) (2) Comparative Heuristic Evaluation (Certification number: H12-01411, Project title: Access certification interfaces) (3) User study of AuthzMap (Certification number: H12-03717 , Project title: AuthzMap Study)

Pooya Jaferian, Hootan Rashtian, and Konstantin Beznosov. 2014. To authorize or not authorize: helping users review access policies in organizations. In Proceedings of the Symposium on Usable Privacy and Security (SOUPS 2014). Pages 301-320. Menlo Park, CA. (acceptance rate: 26%)

Pooya Jaferian, Hootan Rashtian, and Konstantin Beznosov. 2014. Helping users review and make sense of access policies in organizations. In CHI'14 Extended Abstracts on Human Factors in Computing Systems (CHI EA '14). ACM, Toronto, ON, Canada, 2017-2022. (acceptance rate: 49%)

Note that without my supervisor's guidance and support, this dissertation would not have been possible. I therefore have opted to use the term *we* throughout this dissertation.

Table of Contents

List of Tables

List of Figures

Glossary

ACL Access Control List

Amazon MTurk Amazon Mechanical Turk, a crowd-sourcing platform that can be used for usability studies.

ANOVA Analysis of Variance

Between-subjects Study A study design in which each participant is only assigned to one of the conditions.

Discount Usability Evaluation A usability evaluation method that offers a fast, cheap, and early focus on usability. Examples would be heuristic evaluation, and cognitive walk through.

Grounded Theory A data collection and analysis methodology that is used to develop abstract theories from qualitative data.

HCI Human Computer Interaction

HOT Human, Organizational, and Technological

HR Human Resources

Heuristic Evaluation A discount usability evaluation technique. Multiple evaluators inspect

the usability of an interface using a set of usability heuristics, and identify usability problems.

IAM Identity and Access Management

IdM Identity Management

IT Information Technology

ITSM IT Security Management

RBAC Role Based Access Control

Separation of Duties A security control that require different people be responsible for different parts of a task.

SoD Separation of Duties

SOX Sarbanes-Oxley Act, A United States act that requires organizations to adopt certain security controls.

SP Security Practitioners

Within-subjects Study A study design in which all participants are exposed to all conditions.

Acknowledgments

I would like to sincerely thank many people who have supported and helped me during my PhD studies at the University of British Columbia. This dissertation would not have been possible without them.

My deepest gratitude is to my research supervisor, Professor Konstantin (Kosta) Beznosov, for his support during my study. He has contributed intellectually to almost every part of this dissertation. He has been very supportive, and patient during my learning process as a PhD student, and cared about my growth both as a student and as a human being.

I would like to acknowledge Professor Philippe Kruchten, Sathish Gopalakrishnan, and Purang Abolmaesumi, who served on my supervisory committee and have provided insightful feedback and constructive criticism to improve this dissertation. I would be grateful to Professor Sidney Fels and Jennifer Shapka for being my university examination committee, and Professor Heather Richter Lipford, who served as the external examiner of my doctoral examination. They provided many helpful constructive suggestions to strengthen this dissertation.

Thanks to all my co-authors, including Dr. Kirstie Hawkey, Dr. David Botta, Fahimeh Raja, Andreas Sotirakopoulos, and Hootan Rashtian, who helped me in different parts of this thesis with their effort, feedback, and guidance.

My thanks go to all my colleagues from the Laboratory for Education and Research in Secure

Systems Engineering (LERSSE). I am thankful for their valuable feedback and insightful discussions on many parts of my research. It was an honour working beside such an intelligent and insightful people.

Special thanks to my wife, Dr. Azadeh Goudarzi, for her unconditional love and support throughout my years as a PhD student. She accompanied me on this journey, sharing my happiness and stress, and providing positive energy.

Last, and not the least, I want to thank my parents (Yahya Jaferian and Soheyla Atighi) who offered me love and support. I hope that this dissertation makes them proud. Thanks to my brother, Koosha Jaferian, for his encouragement during these years. I am lucky to have him as my brother. Thank you all.

This dissertation is dedicated to

My parents

for their endless encouragement and support

My wife

for her unconditional love

Chapter 1

Introduction

Security of Information Technology (IT) is an important aspect of any organization's IT that helps protecting organization's information assets. The 2009 IT security market overview by Forrester (Penn, 2009) shows a steady increase in IT security spending over the past years (e.g., 10% increase from 2008 to 2009). Data theft issues, compliance requirements, business continuity planning, and a constantly moving security baseline drive the increasing need for IT security. Prior studies show that IT Security Management (ITSM) is challenging (Werlinger et al., 2009a); that is, there are technological, organizational and human challenges in managing IT security.

Computer security research has contributed to addressing technological challenges in ITSM by improving existing security mechanisms, proposing new ones, and identifying and eliminating security vulnerabilities. However, according to Botta et al. (2007), there has been little attention by the research community to the use of these technologies in a social context. Werlinger et al. (2009b) show that ITSM involves collaboration between different stakeholders, and the interaction between stakeholders is particularly complex as they have different levels of security knowledge and awareness, the information is spread across the organization, and the

stakeholders need to pay attention to the core business requirements and organizational constraints such as tight schedules and budgets. Furthermore, Werlinger et al. (2009a) show that IT security involves technological complexity, and Gagné et al. (2008) argue that such complexity is even more than regular IT.

Designing usable interfaces for IT security tools is an important aspect in addressing the social, and technological challenges in ITSM. Chiasson et al. (2007) argue that the usable interfaces is a matter of security since humans are more likely to make mistakes in cases where they deal with complexity, and when interfaces are too cumbersome, present too little or misleading information, or overwhelm the administrator with too much information. However, field studies of security practitioners by Botta et al. (2007) show that the existing ITSM tools do not address the needs of security practitioners, and there are numerous opportunities to improve ITSM technologies.

The overreaching problem that this research seeks to address is the gap between ITSM technologies and the social and organizational context in which they are deployed. Ackerman (2000) defines such problems as a *social-technical gap*, a fundamental mismatch between what is required socially and what we can do technically. More usable tools can narrow the gap, and better support the ITSM activities. While this problem is considered a grand challenge, in this thesis we address three smaller problems:

Problem 1: A lack of comprehensive guiding principles for designing usable ITSM tools: A sub-problem that prevents bridging the gap in ITSM domain is the difficulty in design and evaluation of ITSM tools. First, there is the absence of a comprehensive set of usability guidelines for building ITSM tools. While there are guidelines in the literature, they target specific ITSM tools such as intrusion detection systems (Werlinger et al., 2008c), focus on specific aspect of ITSM activities such as collaboration (Werlinger et al., 2009b), or apply to a domain similar to, but different from ITSM such as IT management (Haber and Bailey, 2007).

Besides lack of guidelines for design, the usability evaluation of ITSM tools is difficult. Laboratory experiments may have little validity due to the complexity of real-world security problems, difficulties in recruiting security practitioners, and the spontaneous (e.g., security incident response), longitudinal (e.g., deploying an identity management system), and collaborative (e.g., diagnostic work) nature of ITSM activities. Discount usability evaluation techniques (Nielsen, 1995) such as heuristic evaluation could be viable substitutes for lab studies. While generic usability heuristics such as Nielsen's (Nielsen and Molich, 1990) could be useful for evaluation, prior research (Baker et al., 2002; Mankoff et al., 2003; Somervell and McCrickard, 2005) shows that domain-specific heuristics are more effective for evaluation of specific classes of systems than generic heuristics. However, we could not find a validated set of usability heuristics for ITSM domain.

To address this problem, our goal is to identify the relationship between existing design guidelines for ITSM tools, and develop a comprehensive framework of ITSM usability guidelines. Furthermore, we aim to develop and evaluate a set of domain-specific usability heuristics for evaluation of ITSM tools.

Problem 2 A lack of understanding how organizations manage users' identities and access: Bridging the social-technical gap in ITSM requires a fundamental understanding of the dynamics of a specific ITSM activity. Therefore, we narrow down our focus to one of ITSM activities, Identity and Access Management (IAM), due to the challenging nature of the activity, and the wide reach of it across the organization. There has been very few empirical studies that provide a real understanding of the IAM activities and challenges in actual organizations. Prior research touched upon certain key areas in IAM, such as challenges of managing file system access (Bauer et al., 2009), authoring access policies (Beckerle and Martucci, 2013; Brodie et al., 2005; Inglesant et al., 2008), understanding and resolving conflicts in file system access policies (Reeder et al., 2008),

managing access in document sharing systems (Smetters and Good, 2009), and case studies of implementing access control in organizations (Kern et al., 2002; Schaad et al., 2001; Stevens and Wulf, 2009). Nonetheless, there is no integrated model that shows how various IAM related activities such as identity management, access management, and audit relate to each other, and why they are challenging. Such a model can help organizations better understand the limitations of their existing approach to IAM, and identify ways to improve their practice. Furthermore, it offers new ways of improving technology to overcome IAM challenges.

To address this problem, our goal is to study how different organizations manage users' identities and accesses, what approach do they use, and how the organizational context impacts IAM activities. Furthermore, we will identify challenges in conducting IAM, and propose an abstract model of the IAM and its challenges. Finally, our goal is to suggest improvements to the technology or practice.

Problem 3: A lack of usable interfaces for understanding and reviewing access policies in organizations: After our initial analysis of interviews, we chose access review as one of the IAM related activities we found particularly challenging. We found that existing access review systems only show the immediate access policy (i.e., users, access privileges, and user-to-access privilege assignments) and hide other relevant contextual information such as other users' access, job information, the history of the policy, and policy violations. For example, during access review comparing one user's access to other similar users would help detecting and removing unnecessary access, or seeing policy violations in the interface can help resolving those violations. Nevertheless, these pieces information were not immediately available in the existing systems, and the user had to go through multiple steps or use multiple information sources to find them. As a result, we found that the process of access review is time-consuming, error-prone, and uncertain.

To address this problem, our goal is to obtain a realistic understanding of access review activity, review existing technologies, and design and evaluate a new user interface that can support access review activity, and address the shortcomings of the existing systems.

1.1 Contributions

The main contribution of this research is addressing the three mentioned problems. In the following, we give an overview of the three parts of our research and contributions of each:

- **Guidelines and heuristics for design and evaluation of ITSM tools:** In Phase One, we collected different usability guidelines, good design practices, and useful features of ITSM tools from the literature on social aspects of ITSM. We categorized and combined data from the literature into a set of 19 high level usability guidelines and identified the relationships between the guidelines and challenges in IT security management. We also illustrated the need for the guidelines, where possible, with quotes from five interviews we performed with security practitioners.

  Then we analyzed the collected guidelines through a theoretical lens of activity theory, and generated an abstract, and generalizable set of principles. We refined these principles into a set of seven usability heuristics, and named the set *ITSM heuristics*. We then conducted a between-subjects user study with 28 participants to compare the use of ITSM heuristics to Nielsen's heuristics for usability evaluation of an IAM system. Our results show those evaluators who used ITSM heuristics reported problems with higher severity, and fewer false positives compared to those who used Nielsen's heuristics. We also studied different aspects of conducting heuristic evaluation using ITSM heuristics such as the number of evaluators needed for evaluation, and their required background.

  The unique contributions of this phase to the overall goal of the thesis are three-fold: First, developing a set of usability guidelines for ITSM tools that can guide the design

of such tools, and eventually help in bridging the social-technical gap. Second, developing a set of validated usability heuristics for identifying usability problems in ITSM tools. Third, a detailed, step by step process for the creation of domain-specific usability heuristics that can be used by other researchers to develop heuristics for their domain of choice. The overall outcome of this phase can be invoked as a vehicle to facilitate bridging social-technical gap between IAM activities and technologies. We use the guidelines and ITSM heuristics in the phase 3 of this thesis to design and evaluate our proposed solutions.

- **Understanding the socio-technical context of IAM:** In Phase Two, we narrow down our focus to one specific ITSM activity, identity and access management. To understand IAM and its challenges, we conducted 19 interviews with security practitioners engaged in IAM in large organizations. We followed a grounded theory approach to collect and analyze qualitative interview data. We provided a thick description of how the IAM is performed in organizations, and what are the challenges faced by stakeholders. We further identified the relationship between the challenges and activities, and proposed an abstract explanatory model of IAM. The unique contribution of this phase is a profound understanding of the socio-technical context of IAM. Such an understanding is essential before attempting to bridge the social-technical gap.

- **Develop and validate solutions for improving IAM technologies and practices:** In Phase Three, we narrow down our focus again to one specific IAM activity, access review. We extended our understanding of access review activity by further analyzing interview data and conducting a survey of 49 security practitioners. Furthermore, we studied the usability problems with existing technologies for access review using heuristic evaluation. Based on the interview, survey, and heuristic evaluation data, we identified a set of design goals for access review interfaces. To show that our goals are feasible and effective, we developed a novel access review tool named AuthzMap. We improved the

tool through multiple rounds of informal feedback and one round of heuristic evaluation by 12 usability experts. We then compared the use of AuthzMap to two of the existing access review systems. We measured the effectiveness of each system in a lab study with 430 participants. The unique contribution of this phase is a profound understanding of access review activity, and its challenges. We also highlighted the importance of context in understanding and interpreting access policies. Furthermore, we proposed a novel user interface for access review, and showed its effectiveness through a lab study.

1.2 Organization of the Thesis

This thesis include three phases and six different studies. The relationship of these studies are shown in Figure 1.1. In Chapter 2, we provide a necessary background on the study topic, and review some of the related work. In Chapter 3, we present our method for creation of guidelines for ITSM tools, and our guidelines framework. Chapter 4 includes a review of prior heuristic creation literature, our method for heuristic creation, the set of ITSM heuristics, and the user study for validation of ITSM heuristics. Chapter 5 presents the methodology, and results of our interview study. In Chapter 6, we present various studies towards bridging the social technical gap for access review including: (1) interview and survey study of access review, (2) heuristic evaluation of an existing access review tool, (3) design of a novel user interface, AuthzMap, for access review, (4) comparative heuristic evaluation of AuthzMap and two of the existing access review systems, (5) a comparative user study of AuthzMap and two of the existing access review systems. In Chapter 7, we discuss how we addressed social-technical gap, present implications of our findings beyond IAM, and discuss the validity of the research. Chapter 8 summarizes the results of individual studies, discusses the relationship between them, lists the contributions of the dissertation, and suggests areas for future work.

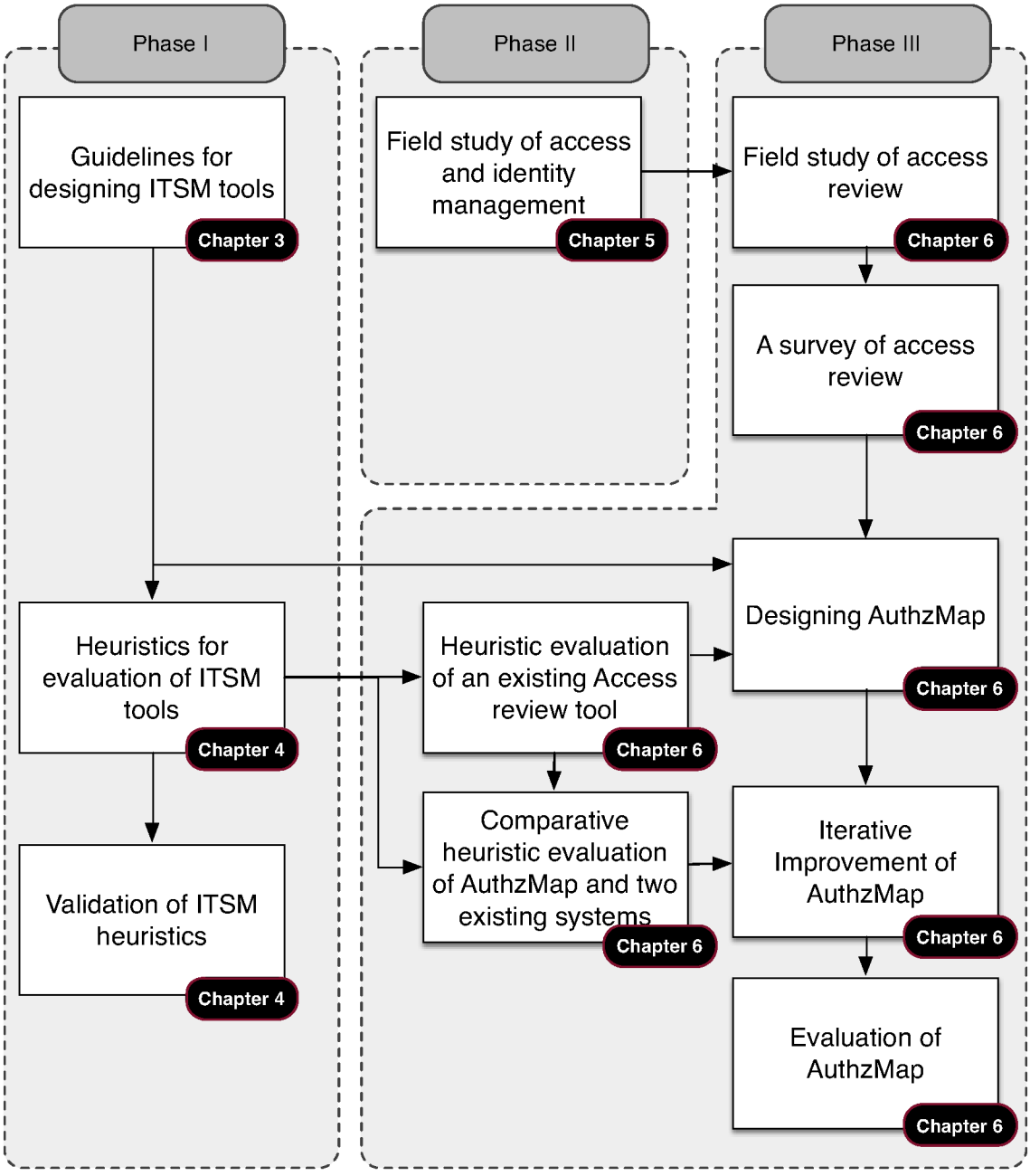


Figure 1.1: Overview of thesis components and the relationship between them

Chapter 2

Background and Related Work

In this section, we provide a brief overview on the definition and scope of ITSM technologies and Identity and Access Management (IAM) systems. Then, we review the prior research on socio-technical aspects of ITSM and IAM. Finally, as we extensively use activity theory, and grounded theory throughout this thesis, we provide background on these topics. We include the chapter specific related work inside their the approprate chapters.

2.1 Background

National Institute of Standards and Technology (NIST) special publication 800-36 defines ITSM products as "components in the design, development, and maintenance of the secure information technology infrastructure" (Grance et al., 2003). Forrester Research Inc. (Penn, 2009) provides a taxonomy that classifies ITSM products into eight categories: (1) content security (e.g., email or web security), (2) endpoint security (e.g., personal firewalls or antiviruses), (3) identity and access management, (4) application security (e.g., code testing or web application firewalls), (5) network security (e.g., firewalls), (6) data security (e.g., file encryption), (7) security operations (e.g., log management, forensics, or security configuration

management), and (8) risk and compliance management (e.g., security assessment, training). These technologies are used directly by multiple stakeholders. For example, the primary users of content security, network security, data security, and security operations tools are security practitioners (SPs); application security tools are used by developers; risk and compliance management tools are used by auditors; and end-point security and identity and access management tools are used by variety of end-users in their day-to-day activities. A tool can affect stakeholders indirectly as well. For example, a firewall impacts how employees access network, but they do not directly use the firewall.

Identity and Access Management (IAM) comprises the processes and infrastructure for the creation and maintenance of user's digital identities and the designation of who has access to resources, who grants that access, and how accountability and compliance are maintained (Blum, 2005). International Organization for Standardization (ISO) proposes a framework for identity management (ISO, 2009a) and a framework for access management (ISO, 2009b). In the ISO framework, the access management framework includes identity management, which provides guarantee of the authenticity and legitimacy of the users. The access management framework also has other components including privilege management; user authentication management; and control, traceability, monitoring, and review of user access.

In the scope of this thesis, we refer to a system that provides both identity management and access management as an *identity and access management system* or *IAM system*. Other sources might call such a system identity management system (IdM), access management system, or access and identity management system (AIM). Forester's review of IAM vendors published in 4th quarter 2009 (Cser, 2009) shows that Oracle, CA, and IBM are the current leaders of the market followed by Sun Microsystems, and Novell as strong performers. Courion, Hitachi ID system, SAP, and Microsoft are contenders. We show an overview on the market presence of the three market leaders in Table 2.1.

Table 2.1: Market presence of the leading IdM vendors (Cser, 2009)

Vendor	Production customers	New customers	IdM revenue	Overall revenue
Oracle	2200-2600	400-600	$160-$180 million	$22,430 million
CA	2000-2500	400-500	$180-$200 million	$4,271 million
IBM	2700	200	$220-$240 million	$103.6 billion

2.2 Socio-technical Aspects of ITSM

Social issues in IT management and subsequently ITSM have been studied before. According to Botta et al. (2007), ITSM is one aspect of IT management. Therefore, the result of the prior studies on socio-technical aspects of IT is also applicable to ITSM. Researchers from IBM have done extensive research on the nature of IT administration work and its challenges (Barrett et al., 2004) and identified recommendations for tool design (Barrett et al., 2005; Haber and Bailey, 2007). As a part of their research, Kandogan and Haber (2005) focused on ITSM tools and practices. Their findings show that security practitioners work in a collaborative environment, need to communicate with people with different backgrounds, work with large data sets, and interact with complex systems. In a different study, Siegel et al. (2006) shows the importance of organizational and human factors in ITSM. Kraemer and Carayon (2007) study the factors that result in human errors in IT security. Goodall et al. (2004) study intrusion detection task in ITSM and show its complex and collaborative nature.

2.2.1 The HOT Admin Project

The Human, Organization, and Technology Centred Improvement of IT Security Administration (HOT Admin) research project aims to investigate human, organizational, and technological factors of IT security from the perspective of SPs. According to Hawkey et al. (2008a), the project goals are: (1) to devise a methodology for evaluating the usability of ITSM tools; and (2) to design effective technological solutions and guidelines to aid SPs. The HOT Admin researchers have performed a participatory observation in one academic workplace and

conducted interviews with 36 SPs. Data from the interviews and observation were analyzed according to several themes including: tools and practices (Botta et al., 2007), security management model (Hawkey et al., 2008b), differences between ITSM and IT (Gagné et al., 2008), collaboration in ITSM (Werlinger et al., 2009b), challenges in ITSM (Werlinger et al., 2009a), and suboptimal situations in ITSM (Botta et al., 2011). The results of HOT admin project show that IT security practitioners (SPs) work in a complex and fast paced environment (Botta et al., 2007). They also show that ITSM is challenging (Werlinger et al., 2009a) and it involves communication and collaboration between different stakeholders (Werlinger et al., 2009b). Werlinger et al. (2009c) studied the complexity and collaboration aspects of ITSM in detail by narrowing down their focus to diagnostic work in ITSM. Additionally, Gagné et al. (2008) identified differences between IT management and ITSM. Finally, Werlinger et al. (2008c) observed the process of deployment and on-going usage of an intrusion detection system (IDS) and identified the challenges in this process.

2.3 Identity and Access Management

Identity and Access management in enterprise context has been studies by both academic and non-academic entities.

The socio-technical issues in IAM have been studied before by non-academic organizations. The Burton Group (2005)[1] commissioned several studies about various aspects of IdM. The Burton Group sources include their client organizations, non-client organizations who use identity management, presentations by vendors of identity management products, and discussion with consultants from other advisory organizations. Their findings can be classified into three main categories: (1) Business drivers for IdM systems (2) Prerequisites for success (3) Requirements for IdM systems. Burton Group reports are targeted towards the business sector

[1]The Burton Group is a firm that provides in-depth, IT research and advisory services to executives and technologists at Global 2000 organizations. The company has been recently aquired by Gartner Inc.

and there is no focus on solutions for improving technology. In addition, they are more focused on the functionality of IdM systems rather than the usability.

The academic studies of Identity and Access Management has been usually focused on technical and theoretical aspects. Particularly, access control which is a subset of IAM has been a mainstream area in Computer Security research since nineties, but the research community has not paid enough attention to non-technological aspects of it. For example, a large scale survey by Fuchs et al. (2011) on the state of role-based access control research confirms this fact. They show that 52% of the recent publications in the area of role-based access control had a theoretical focus. 41% of the publications focused on integrating RBAC into various technologies (e.g., cloud, web, middleware). Only about 7% of the publications focused on practical aspects of role-based access control in organizations. Our review of literature has led to few studies in this area that we particularly review in Chapter 5.

The Identity Project (Wright, 2007) is a study of IdM practices in UK higher education institutions. The results of Identity Project are based on a broad survey (Brown and Smith, 2007), which were validated and refined by 161 semi-structured interviews in the participating institutions (McLeish, 2007). The Identity Project's results are broad and significant, and are tuned to the improvement of business processes in the UK higher education sector. The findings of their research can be classified into challenges and best practices in IdM. Being broad, the Identity Project findings can be complemented by more detailed study of the IdM with the focus on user centered design of the tools. Additionally, the findings of the Identity Project are limited to academic institutions. As academic freedom has been found to lead to certain challenges (Werlinger et al., 2009a), it remains to be seen whether their findings are generalizable.

Most recently, Bauer et al. (2009) describe real life challenges in access control management as gleaned through interviews with policy professionals. The main focus of their study is on the ongoing management of accesses to file systems and physical environments, and they did not

address the full process of IdM including management of identities, auditing, etc. In addition, they only focus on file system and physical security which involve simple policies.

Although not explicitly about IdM, Heckle et al. (2008) discuss organizational challenges in implementing a single sign-on system without previously assisting end-users to develop an accurate mental model. Also, Post and Kagan (2007) identify security controls as an interfering factor to end-users' work and propose recommendations for alleviating this problem.

2.4 Activity Theory

According to Halverson (2002), HCI theories can be useful for four different purposes: first, they have descriptive power and can help in making sense of and describing the world. Second, they have rhetorical power and can "help us talk about the world by naming important aspects of the conceptual structure and how it maps to the real world." Third, they have inferential power and they can help in making inferences about phenomena that have not yet sufficiently understood. Fourth, they can be applied to inform system design.

According to Rogers (2012), several theories have been developed or imported from other fields such as cognitive psychology, sociology, and cultural studies to be used as a theory in HCI to serve any of the four purposes mentioned above. According to Halverson (2002), "theories are more like a pair of dark glasses. We put them on and the world is tinted. The change brings some objects into sharper contrast, while others fade into obscurity." Therefore, one can choose the theory that best fits the problem at hand. In this thesis, we use *Activity Theory* extensively for its descriptive power to develop ITSM heuristics (in Chapter 4), inferential power to interpret the results of the field study (in Chapter 5), and it pragmatic power to inform system design (in Chapter 6). Therefore, here we give a brief summary of activity theory.

Activity theory was developed by Leontiev (1974) as a general psychological theory, and was later proposed as a potential framework for HCI research by Kuutti (1995). Activity theory

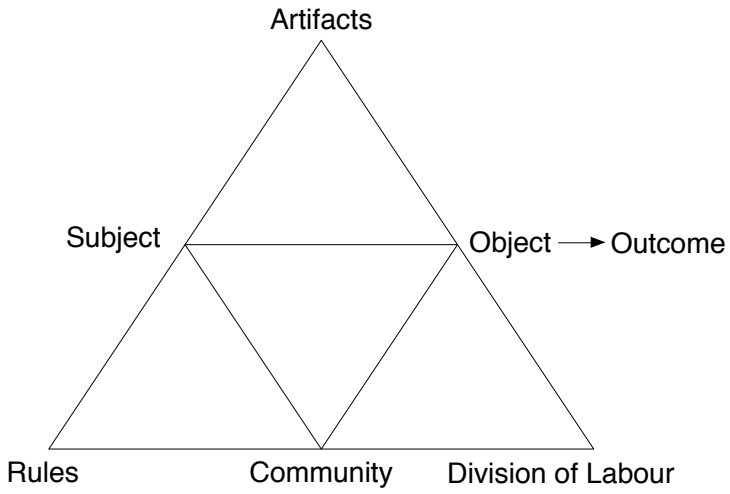


Figure 2.1: Components and internal relations of an activity system

moves the unit of analysis beyond user actions, with *"Human Activity"* as the unit of analysis. Kaptelinin and Nardi (2006) suggested five basic principles of activity theory: (1) *Consciousness and object-orientedness:* all human activities are performed by a conscious actor towards an object. (2) *Hierarchical structure:* activities have three levels: activity, actions, and operations. (3) Activities involve *internalization* and *externalization*. (4) *Mediation:* activities are performed by using and transforming artifacts. (5) *Development:* activities evolve and develop over time. Engeström (1999) proposed a formulation of activity theory to explicate the components and internal relations of an activity system (Figure 2.1). According to Engeström (2001), the components of activity can be classified into subject, object, mediating artifacts, rules, community, and division of labour. He also suggested five activity theory principles including: (1) activity as a unit of analysis, (2) multi-voicedness, (3) historicity, (4) contradictions, and (5) transformation. The sets of principles suggested by Kaptelinin and Nardi (2006) and Engeström (2001) are not mutually exclusive or contradictory; but they do provide different perspectives on activity theory.

2.5 Grounded Theory

Grounded theory is a methodology that can be used to develop theories and models from qualitative data. It has been widely used in the HCI domain to facilitate insight into peoples values, understanding and experience with technology (Furniss et al., 2011).

Grounded theory is considered a non-intrusive approach, and it usually starts with iterative collection of qualitative data including interviews, videos, images, etc. Then the data is broken down to pieces by performing *coding*, and then the codes will be related to each other to identify patterns and themes in the data. Then a theory or model is built based on the themes and their interrelationship. Grounded theory was first proposed by Glaser and Strauss (1967), and later developed in two main directions: Glaser and Holton (2004) advocated the classic grounded theory approach, and insisted that researcher should perform the data collection and analysis without any bias, preconceived theory, or use of literature. Furthermore, Glaser and Holton (2004) suggested a more open approach to coding data, and a positivist perspective. On the other hand, Corbin and Strauss (1990) argued for more structured approach to grounded theory by providing a set of tools and methods that should be used during data analysis. Additionally, researcher can use literature, or prior theories to design the study, and guide the analysis of the data. Furthermore, Corbin and Strauss (1990) highlighted the importance of paying attention to the contextual factors that may be impacting the study phenomena, and offered a version of grounded theory that fits a constructivist paradigm. A survey by Matavire and Brown (2008) of 126 grounded theory studies from Information Systems (IS) literature suggests that grounded theory was mainly used in four different forms: (1) Glaser's version of grounded theory, (2) Strauss's version of grounded theory, (3) Mixed-methodology that combines grounded theory with other approaches such as case studies, (4) application of grounded theory techniques, without adhering to strict grounded theory process suggested by either of Glaser or Strauss. In this dissertation, we use Strauss's version of grounded theory as our approach for the following reasons. First, our goal was to provide an understanding of IAM rather than just suggesting

a theory, and Strauss's version, with its reliance on thick description was better suited for our purpose. Second, Strauss's version acknowledges that researcher's interpretation is a part of the research results. We find this perspective more realistic. Third, we found explicit guidelines and methods suggested by Corbin and Strauss (1990) helpful rather than strict during data analysis.

Chapter 3

Guidelines for Designing ITSM Tools[1]

IT security is an important issue for organizations that want to protect their information assets from threats inside or outside the organization. Previous research such as an interview study of security practitioners by Botta et al. (2007) and observation study of three security practitioners by Rayford B. Vaughn Jr. and Fox (2001) show that, beside technological factors, human and organizational factors also impact IT security management. Security practitioners (SPs) face challenges (discussed in detail in Section 3.1.2) related to each of those factors according to Werlinger et al. (2008a). In order to improve the effectiveness of IT security in an organization, these challenges must be addressed.

One way to address the challenges in ITSM is to develop effective technological solutions and tools to aid IT practitioners in managing security. A key factor that impacts the effectiveness of ITSM tools is their usability according to Chiasson et al. (2007). In this chapter, we present a set of guidelines for ITSM tools based on the available literature and results of the HOT Admin project (see Chapter 2 for an overview of the project). Developing a set of guidelines

[1]This chapter is based on the following publication:
P. Jaferian, D. Botta, F. Raja, K. Hawkey, and K. Beznosov. 2008. Guidelines for designing IT security management tools. In Proceedings of the 2nd ACM Symposium on Computer Human Interaction for Management of Information Technology (CHiMiT '08). ACM, San Diego, CA, USA, Article 7 , 10 pages. (acceptance rate: 29%)

specific to such tools is necessary, due to the importance of IT security in organizations and the evolving and competitive market of tools for managing it (Beal, 2005). For each guideline, we identify the challenges that it can alleviate. We support the need for each guideline through literature and illustrate it through quotes from five SPs interviewed in the HOT Admin project. In addition, we propose a framework for classification of the guidelines. This framework can be used by tool developers to select appropriate guidelines when developing ITSM tools, as well as by SPs and their managers for evaluating such tools.

The rest of this chapter is organized as follows. Next section presents background and related work. Section 3.2 describes the methodology we used to obtain and classify guidelines. Section 3.3 presents our framework of guidelines for ITSM tools, discussing each guideline in turn. Section 3.4 describes how to apply the guidelines and discusses limitations of our work and our plans for future research. Section 3.5 concludes.

3.1 Background and Related Work

3.1.1 User Interface Guidelines

According to Smith and Mosier (1986), user interface design entails a considerable investment by various stakeholders and design guidelines can help the stakeholders in the design process. For example, from the guidelines, a system analyst can derive design requirements, a software designer can derive application-specific design rules, and a manager can make the interface design process more efficient. According to Smith and Mosier (1986), there are challenges and considerations when guidelines are applied and used. First, it should be noted that not all guidelines are applicable to all tools. Therefore, developers should select a subset of guidelines that are applicable to the specific tool they are developing. Second, guidelines must be generally worded so that they might apply to many tools. Therefore, specific *design rules* should be derived from more general guidelines.

Smith and Mosier (1986) suggest that every guideline development effort should begin and end by acknowledging the significant contributions of other people. Therefore, reviewing available literature on the subject under study is an essential part of developing guidelines. Also, guidelines can be based on experience—either practical or derived through research. For example, a literature review has led to the development of guidelines for designing multi-media learning tools by Grunwald and Corsbie-Massay (2006) or designing systems to support co-located collaborative work on a tabletop display by Scott et al. (2003). In contrast, Theng et al. (1999) propose guidelines based on case studies of three digital libraries, while Baldonado et al. (2000) propose design guidelines for systems that use multiple views based on their experience. One large set of guidelines is the *Research-Based Web Design & Usability Guidelines* by Koyani et al. (2006). A survey of literature and other sources resulted in an initial set of guidelines. These were reviewed to eliminate duplicate and conflicting guidelines. Consequently, the relative importance and strength of evidence for each guideline were determined by external reviewers. Finally, they were grouped and organized through a card sorting exercise with a group of web designers. Guidelines related to IT and ITSM tools have been discussed in the literature. Haber and Bailey (2007) developed a set of design guidelines for IT administration tools. They compiled their list of guidelines based on field studies. While their extensive data collection and rigorous analysis support the validity of their results, their guidelines do not address specifics of ITSM which might be different from IT as shown by Gagné et al. (2008). Chiasson et al. (2007) discuss the possibility of using four approaches (fostering an effective mental model of the system, ecological interface design (see Vicente and Rasmussen, 1992), social navigation (see DiGioia and Dourish, 2005), and persuasive technology (see Fogg, 2002)) in designing tools for security administrators. Based on these approaches, they compiled a list of ten guidelines for ITSM tools. While they note that their list is preliminary, the use of persuasive technology might not be suitable for ITSM context. The goal of persuasive technology is to address the problem of security being the secondary task which is not the case in ITSM.

The guidelines for ITSM tools that we present in this chapter were derived through a combina-

tion of our own research results from the HOT Admin project and a thorough survey of related work.

3.1.2 Challenges in IT Security Management

In Chapter 2, we summarized the results of the HOT-Admin project. The HOT-Admin findings by Werlinger et al. (2008a) concerning challenges to ITSM are important to our development of guidelines, as our goal is to address the challenges by proposing improvements for security tools.

One set of challenges in IT security arises from fairly ubiquitous human and cultural traits that become an issue, in particular when SPs need to interact with other stakeholders (Werlinger et al., 2008a). To begin with, a *lack of security culture* can challenge the modification of existing practices (e.g., multiple employees using the same account to access a system). *Lack of training* makes implementation of security controls difficult, as people are not well educated about best IT security practices. Further, *communication of security issues* can suffer from communication break-downs, usually because of different stakeholders having *different perceptions of risk*.

A second set of challenges in IT security are related to the characteristics of organizations (Werlinger et al., 2008a). Besides people having different perceptions of risk, establishing the organizational process of *risk estimation* is a challenge. The trade-off between security and business processes often results in *low priority of security*, which, combined with the costliness of IT security, leads to insufficient budgets. SPs are typically over-worked and *tight schedules* can lead to human errors or suboptimal security controls. Mergers, acquisitions of other organizations, and business partnerships all involve the challenge of *interaction with other organizations* that have different IT security needs, cultures, and practices. *Distribution of IT security* across the organization is commonplace; large organizations may have different IT departments, each of

which is responsible for its own security. *Controlling access to data* is challenging as sensitive data is often distributed in organizations and accessed by many stakeholders. Finally, with an *open academic environment*, solutions need to allow for academic freedom in educational organizations.

A third group of challenges is related to technological issues (Werlinger et al., 2008a). The complex structure of computer networks (e.g., many nodes and users), and the need for different solutions (e.g., firewall, intrusion detection system, anti-virus) for managing IT security create a challenge of *technological complexity*. The frequent revelation of new *vulnerabilities* is a challenge, because SPs must deal with them or risk their security being compromised. The *mobility of access* to organizations' IT poses yet another security challenge.

3.1.3 Guidelines for IT Security Tools

Next, we briefly overview some of the main research efforts in developing design guidelines for ITSM tools. Sources in support of specific guidelines are given in Section 3.3.

As discussed above, the results of the HOT Admin project (see Chapter 2) comprise one source for guidelines. While some of the research themes offer guidelines (e.g., Botta et al. (2007); Gagné et al. (2008); Werlinger et al. (2008a,b)), the provision of guidelines is not the main goal of these papers and they do not provide an integration of all guidelines.

Based on data collected from ethnographic field studies of system administrators, IBM researchers propose guidelines (Barrett et al., 2004, 2005; Haber and Bailey, 2007). As there are many similarities between general IT practitioners and security practitioners (Gagné et al., 2008), guidelines for general IT tools are often applicable to ITSM tools as well. Kandogan and Haber (2005) also propose a small set of guidelines specifically for ITSM tools.

Chiasson et al. (2007) combine results from usable security, ecological interface design, social

navigation, and persuasive technology to propose an initial set of design principles for security management systems. How their principles might address the breadth of challenges has yet to be articulated.

3.2 Methodology

Our main research questions in this work are: (1) What are the characteristics of a good ITSM tool? (2) How can these characteristics be implemented in ITSM tools? (3) Which challenges in IT security are addressed by these characteristics? (4) How can we express these characteristics in the form of guidelines and organize them in a way that can be useful for developers? To answer these questions, we collected data from two different sources: related literature and the HOT Admin corpus of semi-structured interviews with SPs who use IT security tools.

To develop the core guidelines, we first selected a set of primary publications to analyze. This set contained publications from the HOT Admin project (4 papers) and publications about ITSM tools that we found important (14 papers, including those mentioned in Section 3.1.3). Using these sources, we started compiling the guidelines for ITSM tools. We included explicit guidelines as well as recommendations for improving security tools, good practices followed in a specific ITSM tool, and wish lists about tools. In this process we identified 164 guidelines.

After identifying the guidelines, we categorized them using a Grounded Theory approach (Charmaz, 2006). First, we performed *open coding* using codes that emerged from the data itself. Then we employed *axial coding* to combine those open codes that are conceptually the same. As we had a large number of guidelines, we wanted to combine the guidelines in the way that would be more useful for tool developers. We therefore performed a card sorting exercise and grouped the guidelines according to the challenges that they address. This resulted in an early version of the framework, similar to that shown in Figure 3.1. To validate and refine the guidelines, we both broadened our survey and analyzed additional interviews.

To broaden the survey, we performed a more comprehensive literature search. We reviewed the papers published in top conferences related to the topic, performed keyword searches, and mined the references from our original set of 18 papers. The result of this search was a list of 56 papers. We then reviewed the papers and found another 22 papers that could contribute to our guidelines.

We analyzed five semi-structured interviews with SPs to find support for our guidelines and illustrative examples. The interviews are part of the HOT Admin corpus, but had not been analyzed when the HOT Admin papers cited in our survey were written. In this chapter, we use P1 to P5 for referring to participants. Participants included two security managers at a technology company (P1, P2), a security analyst at a telecommunications company (P3), a security consultant (P4), and a security analyst/manager at a second telecommunications company (P5). Each interview was 1-2 hours long, audio-recorded, and transcribed. In the interviews, SPs were asked about their tasks, their organizational model, the tools they used, and the ITSM-related challenges. It is worth mentioning that the interviews were not performed solely to gain knowledge about design guidelines for security tools; however, they did contain considerable information about ITSM tools. To analyze the interviews, we used the guidelines initially identified as codes (i.e., pre-defined codes constructed from prior materials as suggested by Coffey and Atkinson (1996)).

3.3 ITSM Design Guidelines

We have developed a framework (Figure 3.1) for classifying the design guidelines for ITSM tools. Its main purpose is to aid developers in selecting the guidelines. Each layer of the framework addresses a different set of challenges. The lower layers contain the guidelines that are applicable to a larger set of tools, while the upper layers show guidelines that are more specific to a certain set of tools. For example, the lowest layer in the framework comprises general usability guidelines for ITSM tools. These guidelines are applicable to all ITSM tools,

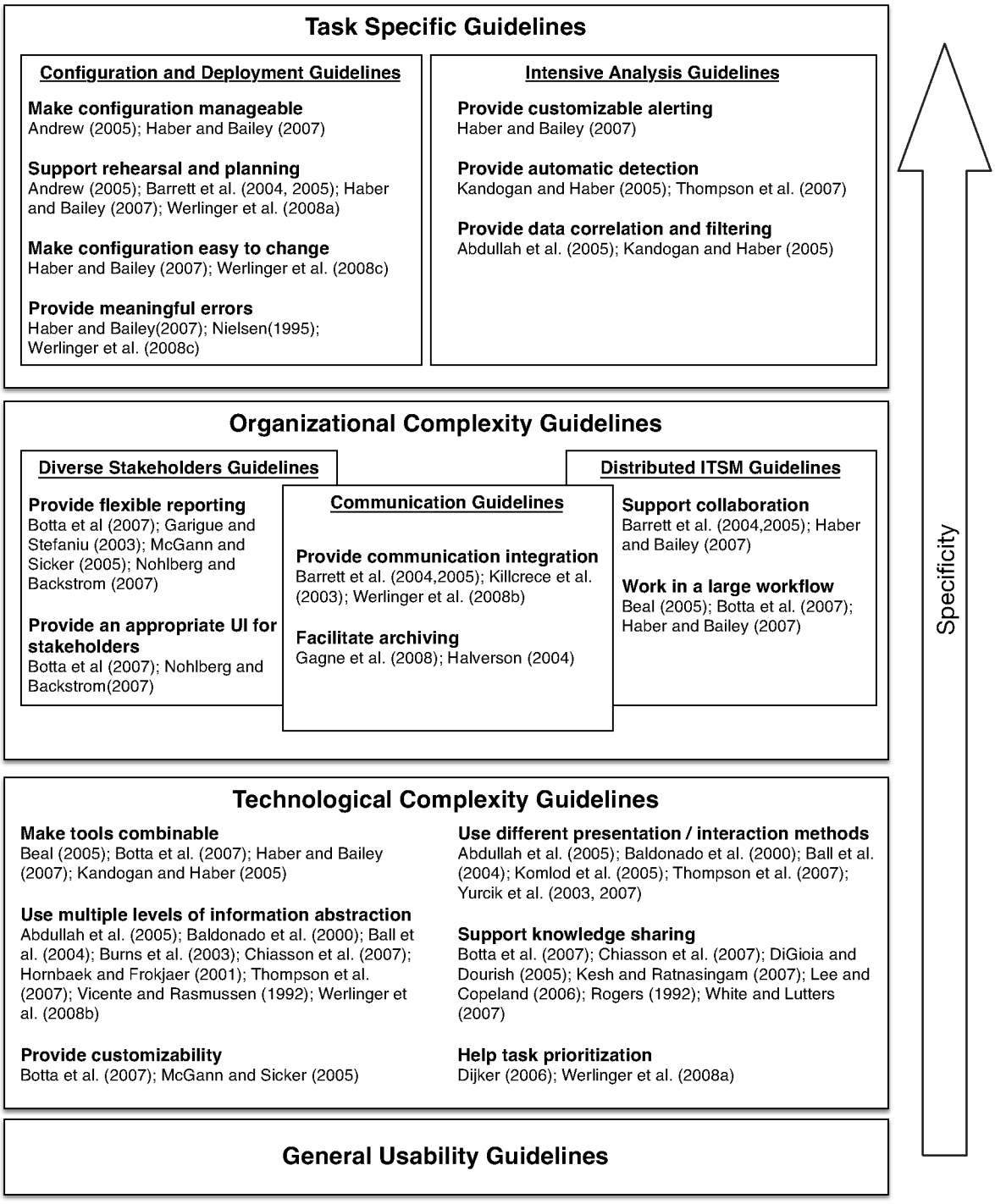


Figure 3.1: Framework of design guidelines for IT security management tools. The references listed under each guideline point to the supporting literature for it.

as well as other tools. The next two layers contain guidelines that are necessary due to the work environment of SPs, which is characterized by technological and organizational complexity. As most of the ITSM tools should work in complex technological environments, the guidelines in the technological complexity layer are applicable to most ITSM tools (but not security tools for end-users). The guidelines in the next layer deal with the organizational complexity of ITSM. These are subdivided into three groups: guidelines to address general communication challenges, guidelines applicable to tools used in a process that involves other stakeholders, and guidelines applicable to tools used by distributed SPs. The upper layer of the framework contains guidelines that are grouped based on task properties of the tool: guidelines for tools that require intensive configuration and deployment, and guidelines for tools used in a process that requires intensive analysis.

We next discuss the guidelines contained within each layer. For each guideline, we discuss the ITSM challenges addressed and cite the related work that supports its inclusion in our framework. When possible, we provide illustrative examples from participants and give alternatives of the guideline.

3.3.1 General Usability Guidelines

The first layer includes general usability guidelines and recommendations that are applicable to tools for SPs. When performing the card sorting exercise, we realized that many of the guidelines for security tools were based on general usability principles such as those proposed by Nielsen (1995) and Smith and Mosier (1986). Because these guidelines were originally developed for more general tools and interfaces and are available in many different sources, we do not list all of them here. However, we give an example of a general usability guideline that is particularly important for ITSM tools: providing help and documentation to users.

In the literature, there are sets of guidelines about help and documentation features for of IT

and ITSM tools. For example, tool documentation should be available on the Internet and searchable using search engines (Haber and Bailey, 2007). Several help features have been suggested for security tools (Herzog and Shahmehri, 2007); although directed at tools for end-users, most of them are applicable to ITSM tools as well. These include providing context sensitive help, online help, wizards, light-weight help features, and social navigation. One technique, safe staging, may not be as useful if the tool will only be used by expert users.

3.3.2 Technological Complexity Guidelines

According to Werlinger et al. (2008a), there are multiple challenges related to technical complexity, including *mobile access* and *vulnerabilities*. We next present guidelines that can address these challenges.

Make Tools Combinable

Botta et al. (2007) discovered SPs must often use multiple tools to perform a single task, but the process of combining tools to perform a task is not well supported by available tools (Beal, 2005; Kandogan and Haber, 2005). As one of our participants illustrated: *"So the vendors themselves are looking at things in isolation instead of looking at it as a whole thing that needs to be addressed"* (P4). Another described challenges when using multiple tools: *"We are really, really having a problem at correlating output from all these tools. At the beginning they were using three or four, it was easy to manually correlate, but when they started hitting six, seven, eight, plus, it was very difficult to correlate because the outputs are all different"* (P5). This participant also mentioned that development of a console to configure and execute 17 vulnerability analysis tools resulted a significant decrease in the time needed to perform an analysis task (from 10-15 days, to two days).

Combining tools in an *ad hoc* fashion is a kind of *bricolage*; it is recommended that tools

should survive in an arena of bricolage (Botta et al., 2007). Vendors should standardize event formats to permit integration of tools (Kandogan and Haber, 2005). Standardized configuration and logging formats will allow files from different tools to be searched and correlated together (Haber and Bailey, 2007). Another option is to provide APIs/plug-ins to facilitate integration of tools into system-wide monitoring or management meta-tools (Haber and Bailey, 2007).

Support knowledge sharing

As SPs perform their tasks within complex technological environments, a great deal of knowledge is created. This knowledge is either kept in the mind of the security practitioner, or written in notes or documents, or kept in the form of executable scripts (Botta et al., 2007). This knowledge is a valuable asset and can be used in the future by the same or other SPs; it therefore should be kept and managed (Kesh and Ratnasingam, 2007). This knowledge can be managed at two levels: among SPs in the organization or among all the users. Therefore, security tools should facilitate knowledge management at different levels. To support knowledge sharing, SPs can use databases (Lee and Copeland, 2006), Microsoft SharePoint sites, document management systems, or Wikis (White and Lutters, 2007), or synchronous communication channels (Rogers, 1992). This practice is illustrated by one participant who mentioned: *"We have an IT manual which is kept up to date electronically and hard copy. We have our SharePoint site where they can go and everything is at their fingertips. It links them to every single place they need to know how to go to"* (P4). Another form of sharing is *social navigation*, which is suggested as a model for usable security by DiGioia and Dourish (2005), and shown to be important in the context of ITSM by Chiasson et al. (2007). Although arising from technological complexity, the need to support knowledge sharing is closely related to organizational complexity as described in Section 3.3.3.

Use different presentation/interaction methods

According to Baldonado et al. (2000), presenting information in multiple views or presentation formats can facilitate investigation of a single conceptual entity. Textual and speech data is sequentially processed through auditory cognitive functions, while graphical data has the advantage of using parallel visual and spatial functions (Ball et al., 2004). Therefore, graphical data results in faster situational awareness and effective identification of patterns and vulnerabilities in network. Furthermore, using different presentations of the same data can help situation awareness through a reduction in the high cognitive load characteristic of ITSM (Yurcik et al., 2003). In the related literature, this guideline often accompanies a proposal for different visualization methods for a large set of data. For example, one proposed visualization method for intrusion detection system (IDS) alarms is to show alarms in a two-dimensional space (y-axis: IP address, x-axis: time) (Abdullah et al., 2005). Three-dimensional space has also been proposed for IDS data (color, opacity, shape) (Komlod et al., 2005). Different levels of detail can allow SPs to zoom in for more details (Komlod et al., 2005). Visualization of the network data can reduce the time and training required for network traffic analysis (Ball et al., 2004). The combination of concurrent textual and visual interfaces has been advocated for security tools by Yurcik et al. (2007) and for IDSs by Thompson et al. (2006) as each interface is shown by Thompson et al. (2007) to has its own strengths and weaknesses. This guideline may be particularly important for those tools that involve intensive analysis, as described in section 3.3.4.

Use multiple levels of information abstraction

Vicente and Rasmussen (1992) suggest using an abstraction hierarchy in order to support operators of complex systems during unanticipated events. As SPs need to deal with vulnerabilities and unanticipated scenarios in complex network environments, tools can follow Vicente's method of showing the system at different levels of abstractions, with the Ecological Interface

Design (EID) framework. Other researchers suggest using EID in the design of ITSM tools in general (Chiasson et al., 2007) and for network monitoring tools in particular (Burns et al., 2003). According to Werlinger et al. (2008b), presenting information at different levels of abstraction to different stakeholders can help prevent disclosure of confidential information by presenting it to each stakeholder at an appropriate level. It can also prevent miscommunication by providing information appropriate to the stakeholder's level of security knowledge.

In many tools, presenting information at different levels of abstraction is realized by providing an interface with two views: overview and details. A study about reading electrical documents by Hornbaek and Frokjaer (2001) suggests that presenting information at different levels of detail can reduce user errors. Similarly, Ball et al. (2004) show that using multiple levels of abstraction, as well as visualization, reduce the time and training required for network traffic analysis.

According to Baldonado et al. (2000), when there is diversity in levels of abstraction, different presentation formats (Section 3.3.2) can be used. For example, both textual and graphical interfaces for IDSs have been used by Thompson et al. (2006) to present information at different levels of abstraction. Similarly, a visualization technique for IDS by Abdullah et al. (2005) provides an overview of its alarms with details on demand.

Provide Customizability

As SPs frequently deal with unpredictable situations (e.g., new vulnerabilities), an essential feature for ITSM tools is to be customizable (Botta et al., 2007). This need is illustrated by one of our participants (P1): *"For the reasons why I have built some stuff from hand, we'll say that no, they don't do everything that I need them to do. So sometimes I do need to custom craft something or I need to automate something. Or I need to do something maybe that the tool doesn't do."*

In a comparison of security analysis tools for SIP-based VoIP systems by McGann and Sicker (2005), one important criterion was the ability to define new and customized test-cases.

Help Task Prioritization

SPs frequently must deal with many competing priorities. A survey by Dijker (2006) found that one of the main factors that frustrates SPs is wasting time and that they need better planning and organization. Therefore, it is important for tools to facilitate the process of planning and prioritization. Planning facilities can be implemented in different ways. At the most basic, a tool could afford note-keeping functionality so that SPs can write down their priorities with regards to the tool. With some intelligence, a tool could help to prioritize vulnerabilities based on their criticality (Werlinger et al., 2008a).

3.3.3 Organizational Complexity Guidelines

Several aspects of organizational complexity must be addressed by the guidelines. SPs need to communicate with many stakeholders, including both other SPs and diverse stakeholders within the organization (Werlinger et al., 2008b). We first present guidelines that address general communication challenges and then present the guidelines that specifically address the challenge of dealing with diverse stakeholders. Finally, we present guidelines that address the challenges arising from the distribution of security tasks across multiple SPs.

Communication Guidelines

Kraemer and Carayon (2007) show that ineffective communication is a contributing factor to human errors. According to Werlinger et al. (2008b), SPs need to communicate with other stakeholders during many activities, and the current tools do not provide sufficient communication support for SPs. Tools that facilitate communication can address challenges of *commu-*

nication of security issues, *distribution of IT management*, *interaction with other organizations*, and *different perceptions of risk*.

Provide Communication Integration One way for tools to facilitate communication is to allow for integration with communication media. Tools should have communication facilities to allow collaboration between different users (Barrett et al., 2004, 2005). Tools can reduce communication overhead between different stakeholders by showing relevant security configuration information to different stakeholders (Werlinger et al., 2008b). One important feature, whether communication is between a tool and a user or between users, is to support a secure method of communication (Killcrece et al., 2003).

The need to support communication between tools and users is illustrated by one of our participants when discussing a network monitoring tool (P4): *"You hate to find out you have a problem when you are actually working; rather get paged at night and be able to fix it before they show up."* Tools should be integrated with different channels (e.g., email, text messages, web site) (Werlinger et al., 2008b). Mobile communication modalities (e.g., pagers, Blackberry email) should be integrated into the solutions (Barrett et al., 2005). Furthermore, tools should be configurable as to the destination and stakeholder to which these messages (e.g., alarms, logs) should be sent (Werlinger et al., 2008b).

Facilitate Archiving Tools should facilitate keeping track of communication and information related to tools. Practicing this guideline has two benefits. First, keeping a record of communication between different stakeholders is already practiced by SPs; this may be due to the need for SPs to adhere to legislation (Gagné et al., 2008). It is also illustrated by one participant (P4): *"So we have archives with backup tapes—we have the Cadillac of backup tapes for our kind of organization because we have a thing you can walk into practically—so we keep everything."* If tools provide support for this need, they can help remove the burden of archiving and

managing communication. A second benefit of archiving is to keep the information and knowledge that is generated during one project or incident (Halverson, 2004). This information can be used in future incidents to analyze the trends in network, or it can be used as a knowledge base (previously discussed in Section 3.3.2).

Diverse Stakeholders Guidelines

According to Werlinger et al. (2008a), one important organizational challenge of IT security is the involvement of various stakeholders within the organization, and some aspects of security management may involve non-experts (Botta et al., 2007). Furthermore, effective communication of security issues to different stakeholders is an important factor to be considered. According to von Solms and von Solms (2004), *"Not realizing the core importance of information security awareness amongst users"* is considered one of the "deadly sins" of ITSM. We next present guidelines that are mainly aimed at addressing communication challenges (*communication of security issues*, *different perceptions of risk*) that result from diverse stakeholders.

Provide an Appropriate UI for Stakeholders According to Botta et al. (2007), ITSM tools often have many types of end-users including experts such as SPs and less technical administrators and managers. Each category of users may have its own preferences and needs in terms of the user interfaces. As illustrated by one of our participants (P5): *"We actually use the command line interface route and we try to keep it as simple as possible because we were putting another layer on top of it we couldn't go into the graphical one. But sometimes clients want graphical stuff. Especially if they are not 100% techie, it's easier."* Therefore tools should provide appropriate user interfaces based on the user's expertise and needs. One suggestion by Nohlberg and Backstrom (2007), when developing a UI for ITSM tools that will be used by managers, is to provide an overview early with as little information as possible and provide further details on demand. This guideline relates to Sections 3.3.2 and 3.3.2, as different

stakeholders require different presentations of data or different levels of detail in their user interface.

Provide Flexible Reporting One aspect of flexible reporting is generating reports that are customized to contain information for a specific stakeholder. For example, reports that are aimed at managers should be concise and mainly focus on business objectives and the effectiveness of the organization in reaching them (Garigue and Stefaniu, 2003; Nohlberg and Backstrom, 2007). This is illustrated by one of our participants (P5): *"And the CEO is comfortable talking to me because I am talking his language. I am talking your return on investment, . . . and those are the terms that I use quite often. I do it deliberately. It's a technique that I've learned and I've used shamelessly."* Also, the report may be packaged differently depending on the type of stakeholder. For example, one participant talked about packaging a report for managers (P4): *"It's got to have color and it's got to be flashy so that they'll pay attention. They don't want to read a ten page document on anything. They want a quick learn."* From a different angle, two of our participants (P5, P1) discussed that reports should contain constructive recommendations that are simply represented (e.g., in a table). According to Garigue and Stefaniu (2003), reports can be packaged based on predefined templates or standard frameworks like Sarbanes Oxley bill or IS 17799. Furthermore, they classify reports into four categories (governance and policies, application and systems development and deployment, active security posture, infosec operational services), and provide examples of the important reports that can be generated in each category.

Another aspect of the flexible reporting is making reports accessible by different stakeholders. To realize this, reports should be easily distributable and accessible across the organization. According to Botta et al. (2007), reports can be generated in standard formats like HTML, PDF, and spreadsheets; generating reports in the web format is considered an important feature for security analysis tools (McGann and Sicker, 2005).

The flexible reporting guideline relates to Section 3.3.2 because a well sorted report can help prioritization. Also the guideline relates to Section 3.3.2, as providing different presentations of data will make reports more understandable. Finally, flexible reporting relates to Section 3.3.2, as the reports generated by a tool should provide an appropriate level of detail based on the intended audience.

Distributed ITSM Guidelines

The guidelines presented next address the challenge of *distribution of IT management*. In many organizations ITSM is distributed across multiple SPs (Botta et al., 2007), either informally or through an official distributed organizational model for ITSM (Hawkey et al., 2008b). In these organizations, SPs need to collaborate with each other, as well as other stakeholders, to perform tasks (Werlinger et al., 2008b). Tools should function as part of a larger workflow and provide support for collaborating and sharing.

Work in a Large Workflow One of the important needs of SPs while working under distributed ITSM is to be able to automatically distribute tasks. According to Beal (2005), security tools should follow the way corporate networks have evolved and become integrated together. To allow collaboration among stakeholders, Botta et al. (2007) point to a need for workflow support for the varying roles of different individuals. One of our participants (P3) desired an access control platform that supported the workflow of granting access to a user: from the end user request, to the person in charge of authorization, to the administrator making changes to the security controls. According to Haber and Bailey (2007), shifts in responsibilities could be encoded in scripts with a new sysadmin automatically notified when it is their turn and the pertinent interface displayed.

Support Collaboration One important feature of tools that can help collaboration is to provide a shared view of the system state (Barrett et al., 2004, 2005; Haber and Bailey, 2007). According to Haber and Bailey (2007), tools should formally support sharing by showing which users are currently working with system and what they are doing. In addition, Barrett et al. (2004, 2005) suggest sharing can be supported with proper approval and authentication. Another important aspect of collaboration is to provide support for grounding new participants as quickly as possible when they join the activity (Haber and Bailey, 2007). This guideline is an extension of the knowledge sharing guideline (Section 3.3.2).

3.3.4 Task Specific Guidelines

This layer contains guidelines that may or may not be applicable, depending on the nature of the application. The first set of guidelines is specific to applications that require intensive configuration, particularly during deployment. The second set is specific to applications that require SPs to perform intensive analysis.

Configuration and Deployment Guidelines

SPs must often perform complex configuration of tools, particularly during deployment. Due to the *technological complexity* of ITSM and its tools, the task of configuration can require a great deal of effort (Werlinger et al., 2008a). A second challenge that impacts configuration and deployment is *vulnerabilities* (Werlinger et al., 2008a). To deal with frequent vulnerabilities, SPs need to patch systems often; however, patching a network of thousands of nodes is tedious work that can be very costly. For example, manually deploying a patch on a 1000-node network can cost as much as $1M (Andrew, 2005). Because security has a *low priority* within many organizations (Werlinger et al., 2008a), SPs may be urged to complete configuration and deployment as quickly as possible, and without compromising availability and performance. We next present several guidelines aimed at dealing with these challenges.

Make Configuration Manageable As described above, SPs frequently need to apply configuration changes to hundreds of nodes in a network or deploy nodes at a similar scale. This complex process should be done very quickly and accurately. Therefore, tools should enable SPs to automate and manage this process, as well as control its details. To realize this, Haber and Bailey (2007) suggest that tools should provide progress indicators, forecast the deployment process, perform operations in an asynchronous non-blocking manner, and provide history and detailed steps of the executed operation. According to Andrew (2005), tools should also support change roll-back.

Support Rehearsal and Planning As SPs work with complex and critical systems, changes in configuration may have unanticipated outcomes that cannot be tolerated by other stakeholders. For example, security patches are not usually tested for all environments (Andrew, 2005). Also, Werlinger et al. (2008a) illustrate how a security patch that decreased the performance of an application triggered conflict between SPs and internal users. Therefore, SPs should be very careful in deploying new solutions, changing configuration, or applying patches. According to Haber and Bailey (2007), system administrators practice rehearsal and planning to avoid unanticipated events on production systems. They first rehearse the operation on a test system and then apply it to the production system. Security tools that require extensive configuration should support rehearsal and planning practices. According to Barrett et al. (2005), it should be easy to build a test system with various degrees of fidelity to the production system, and it should be easy to validate the results of the test system.

Also, tools should support migration of scripts/operations from test to production environments (Haber and Bailey, 2007). Logging each step of the procedure and providing facilities to compare the outputs from test and production environments would facilitate rehearsal and planning (Barrett et al., 2004). According to Andrew (2005), virtual environments (e.g., using tools like VMware) can assist with testing.

The rehearsal and planning guideline is related to making the configuration process manageable (Section 3.3.4). The rehearsal process can be completed more easily and with less overhead if tools support features like undo. Furthermore, providing forecasting of the deployment can be a useful indicator for comparing the rehearsal with actual execution.

Make Configuration Easy to Change One of the tasks of SPs is to change the configuration of the system. Configuration frequently requires dealing with many parameters, some of which are unknown to the security practitioner. Therefore, tools should provide facilities that help SPs change configuration of the system easily. To realize this, Haber and Bailey (2007) suggest that tools should provide commented configuration files and/or group related parameters together in high-level profiles, so that a change in the profile would change all related parameters automatically. Also, Werlinger et al. (2008c) recommend that tools should provide a quick tuning option that allows batch configuration of parameters.

Provide Meaningful Errors Although providing meaningful errors is a standard usability practice (Nielsen, 1995), we re-iterate the guideline here as configuration and deployment of complex systems is particularly error prone. To ease the process, particular care should be taken for any error messages generated by tools during configuration and deployment so that it is presented to the user in a meaningful way. For example, (Werlinger et al., 2008c) show that insufficiently meaningful error messages caused delays during installation of an IDS in one academic organization. Haber and Bailey (2007) suggest that tools should provide help in case of errors or alerts, instead of presenting cryptic messages.

Intensive Analysis Guidelines

According to Werlinger et al. (2008a), investigation of attacks and vulnerabilities is one of the most important and challenging tasks for SPs and requires periods of intensive analysis. To deal

with the challenge of *vulnerabilities*, tools should support SPs in the investigation tasks. As SPs must conduct analysis within the constraints of *tight schedules* (Werlinger et al., 2008a), tools should provide mechanisms to reduce the number of false positives (FPs) because FPs have to be investigated. The next three guidelines are applicable to tools which require SPs to perform intensive analysis.

Provide Customizable Alerting Many security tools that monitor systems, generate alarms communicated to SPs. Haber and Bailey (2007) recommend that tools should provide customizable thresholds for generating alarms and selectable destinations for sending alarms (e.g. pager, email, console). Furthermore, SPs should be able to suppress alarms with lower priority. As mentioned by one of our participants (P3): *"Given that I had knowledge of the perimeter security systems that were protecting these systems internally in the organization, I modified that critical level [of the tool]."*

Provide Automatic Detection According to Kandogan and Haber (2005), SPs need to find attacks or unusual behavior patterns in large amount of logs and data that are linked together. To help SPs perform their tasks more effectively, Kandogan and Haber (2005) recommend providing automation in detecting problems. Application of data mining and other analytic methods in activity classification, analysis, and noise reduction can help; automatic detection could be implemented as software agents or *bots* that handle obvious cases and notify SPs about critical ones (Kandogan and Haber, 2005). Thompson et al. (2006) suggest using intelligent pattern recognition techniques to find salient patterns.

Provide Data Correlation and Filtering According to Kandogan and Haber (2005), during analysis, SPs frequently need to collect data from several sources and then correlate it. For example, Abdullah et al. (2005) show that correlating alarms of different IDSs can reduce

the number of false alarms. As discussed by one participant (P3): *"These tools generate general or global reports based on what they are analyzing, right? And, I took those reports, with other tools I complemented like NMAP to do the same analysis, and I was checking and corroborating that they effectively correspond."* Security tools can improve the process of data correlation by providing required filtering to reduce the large quantity of data, providing output in standard formats that can be shared between different tools, providing facilities to deal with the problem of out-of-sync clocks in correlating time-stamped data from different sources, and providing facilities to automate the process of data correlation (Kandogan and Haber, 2005).

3.4 Discussion

3.4.1 Applying the Guidelines

These guidelines can be used for multiple purposes. They can be used by developers as they compile requirements for applications with SPs as end-users. Consideration of the guidelines may also be helpful when developing rich use case scenarios to ensure that the scenarios address the technological and organizational challenges of the intended operational environment. The guidelines could also be used by SPs and managers as they evaluate tools in the context of the challenges inherent within their organization. In this case, the guidelines should be treated as high-level criteria. If a criterion is not met in some way, then the application may have room for improvement.

Whether designing an application, deciding which one to acquire, or evaluating an application, the sets of guidelines that are relevant to the application and to the situation should be considered. The guidelines are grouped by whether they are (1) generally applicable, (2) relevant to technological or organizational characteristics of the operational environment, or (3) relevant to task-specific challenges. *General usability guidelines* apply to all ITSM-relevant applications. We argue that the guidelines to address *technological complexity* are also applicable to

all ITSM-relevant applications. Not only is the technological environment of ITSM changing with the advent of new technologies, but the rate of change in ITSM is faster than in general IT (Gagné et al., 2008). That is, technological complexity is characteristic of ITSM. As discussed next, the applicability of the remaining guidelines will depend on the organizational environment and the specifics of the tasks the application will perform. Not all the guidelines will be applicable in all situations.

The organizational environment in which the application is deployed may be such that the guidelines to help *organizational complexity* are relevant. An organization may be large with many cooperating SPs, or small with an IT department consisting of a "one-man shop." Depending on the intended use of the tool in that environment, guidelines to help *multiple stakeholders*, or guidelines to help *distributed ITSM*, or both may apply. A developer of a tool that will be used in a "one-man shop" may not need to consider guidelines that address organizational complexity, particularly those guidelines that address *distributed ITSM*. However, a tool—such as an IDS—installed in a large organization would likely require ongoing distributed cooperation between SPs (Werlinger et al., 2008c) and therefore benefit from the guidelines on *distributed ITSM*. In contrast, if an access control system were to be implemented in the same large organization, it would require a great deal of initial cooperative consultation to establish job roles and their corresponding privileges. For such an application, the organizational complexity would likely focus more on multiple stakeholders than on distributed ITSM, so it would benefit from guidelines to help *multiple stakeholders*.

Task specific challenges may be addressed by guidelines to help *intensive analysis* or guidelines to help *configuration and deployment*. For example, an IDS is typically difficult to configure and also requires intensive ongoing analysis (Werlinger et al., 2008c), while a network scanning application (e.g., Nessus) requires intensive analysis, but typically does not require much configuration. The network scanning application would therefore only need the guidelines to help *configuration and deployment* while the IDS could benefit from consideration of all the

guidelines in this layer.

3.5 Conclusion

In this section, we provided the result of our survey on design guidelines for ITSM tools. The primary sources for the guidelines are recommendations about ITSM tools available in the literature; we have augmented these from our own experiences interviewing SPs in the HOT Admin project. We have gathered the different recommendations and combined them into a framework of high-level design guidelines for ITSM tools. The guidelines in our framework are high level and blur the boundaries between usability, organizational usability (Elliott and Kling, 1997), and utility. This framework can be used by tool developers, as well as by SPs and managers evaluating security tools. To justify the guidelines, we provided empirical evidence of their need. In addition, we identified relationships between the guidelines and known challenges in ITSM. These relationships can help users of the framework determine the importance of each guideline for their tools.

Chapter 4

Heuristics for Usability Evaluation of ITSM Tools[1]

Information technology security management (ITSM) tools serve several purposes including protection of network and data, detection of threats and vulnerabilities, and management of users and their accesses (Beal, 2005). Previous research (see Chapter 2) has highlighted the importance of collaboration and information sharing support between various stakeholders in IT security tools. Werlinger et al. (2009b) identifies nine security activities that require collaborative interactions and developed a model of the complexity of their interactions. This complexity arises from organizational attributes (e.g., distribution of IT management); the need for security practitioners (SPs) to interact with multiple stakeholders; and their need to engage in multiple security related activities. Each of these activities may require different explicit or tacit knowledge and kinds of information to be conveyed.

[1]This chapter is based on the following publications:
P. Jaferian, K. Hawkey, A. Sotirakopoulos, M. Velez-Rojas, and K. Beznosov. Heuristics for evaluating IT security management tools. Human-Computer Interaction, Vol. 29, Iss. 4, 2014. (impact factor: 3.039)
P. Jaferian, K. Hawkey, A. Sotirakopoulos, M. Velez-Rojas, and K. Beznosov. 2011. Heuristics for evaluating IT security management tools. In Proceedings of the Seventh Symposium on Usable Privacy and Security (SOUPS '11). Pittsburgh, PA, USA, Article 7 , 20 pages. (acceptance rate: 33%, best paper award)
P. Jaferian, K. Hawkey, A. Sotirakopoulos, and K. Beznosov. 2011. Heuristics for evaluating IT security management tools. In CHI'11 Extended Abstracts on Human Factors in Computing Systems (CHI EA '11). ACM, Vancouver, BC, Canada, 1633-1638. (acceptance rate: 45%)

According to Chiasson et al. (2007), usability is an important quality for ITSM tools, but evaluating the usability of specific ITSM tools is challenging. Laboratory experiments may have little validity due to the complexity of real-world security problems and the need to situate a specific tool within a larger context (Neale et al., 2004). However, it is difficult to recruit SPs for simple interviews, let alone field observations (Botta et al., 2007; Kotulic and Clark, 2004). Direct observation of tool use can be time consuming as much security work is spontaneous (e.g., security incident response), or occurs over many months (e.g., deploying an identity management system). As ITSM tool use is intrinsically cooperative, its study inherits the difficulties of studying cooperation listed by Neale et al. (2004). Therefore, heuristic evaluation of ITSM tools could be a viable and low cost component of tool usability evaluation.

The existing sets of heuristics did not capture many of the challenges specific to the ITSM domain. As we report in this chapter, we needed to explore how domain specific heuristics have been created in the past, develop a methodological approach for creating them, and apply the method to the creation and evaluation of a new set of heuristics for usability evaluation of ITSM tools. Our results suggest that using a combination of a bottom-up approach (by deriving guidelines from literature and interview data), and a top-down approach (by abstracting the guidelines using activity theory (see Kaptelinin and Nardi, 2006)) can lead to a set of heuristics that can find problems in IT security tools. In this chapter, after presenting the set of heuristics we created, we report on the empirical evaluation of our heuristics in which we compared their usage to Nielsen's (Nielsen and Molich, 1990). We conducted a between-subjects study with 28 participants and examined different aspects of evaluation when deploying the two sets of heuristics. Our results suggest that the number of major problems that are found using the ITSM heuristics is higher than the number of problems that are found using Nielsen's. Furthermore, while Nielsen showed that about five evaluators are able to find about two thirds of the problems, in our evaluation of the IdM system, five evaluators only found about half of the problems found by 14 evaluators; we observed few overlaps between problems identified by individual participants using either Nielsen's or the ITSM heuristics. Based on the result of

evaluation and participants' feedback, we discuss how ITSM and Nielsen's heuristics can be employed for usability evaluation of ITSM tools.

4.1 Background and Related Work

Heuristic evaluation (HE) is a usability evaluation method based on a set of usability principles called heuristics. According to Nielsen (2005a), they are called "heuristics" because they are more in the nature of rules of thumb than specific usability guidelines. Usability heuristics are more general than guidelines, and require more expertise of the evaluator. An evaluator inspects a user interface and identifies usability problems and their severity based on heuristics and his judgment of the interface. According to a survey by Vredenburg et al. (2002), heuristic evaluation is the most popular informal usability evaluation technique among user-centered design (UCD) practitioners. Furthermore, Jeffries et al. (1991) show that heuristic evaluation can lead to finding more serious usability problems compared to usability testing, guidelines and cognitive walkthrough (see (Polson et al., 1992) for details of cognitive walkthrough). Nielsen's theoretically grounded and extensively tested heuristics (Nielsen and Molich, 1990) are the most widely accepted heuristics. They were developed based on existing HCI guidelines, are consistent with Norman's theory of action (Norman and Draper, 1986), and focus on the dialogue between a single user and the physical world. Nielsen's heuristics have been modified or extended to create domain specific heuristics (e.g., ambient displays (Mankoff et al., 2003), virtual reality (Sutcliffe and Gault, 2004), medical devices (Zhang et al., 2003), intelligent tutoring systems (Muller and McClard, 1995), and intrusion detection systems (Zhou et al., 2004)). Furthermore, heuristics have been created without using Nielsen's heuristics (e.g. computer games (Pinelle et al., 2008), groupware (Greenberg et al., 2000), shared visual work surfaces for distance-separated groups (Baker et al., 2002), Large Screen Information Exhibits (LSIE) (Somervell, 2004), Ubiquitous systems (Scholtz and Consolvo, 2004)). We only have found one instance of applying HE on an ITSM tool. Zhou et al. (2004) developed six

heuristics based on Nielsen's heuristics for intrusion detection systems. The authors described that they developed the heuristics based on surveys and interviews, but they did not provide any detail of their methodology. Four of their six heuristics are identical to Nielsen's heuristics and the rest are extensions to Nielsen's.

4.2 Proposed ITSM Heuristics

In this section, we describe how usability heuristics can be created, classify prior heuristic creation literature according to its methodology, and discuss the benefits and drawback of each approach. Then we describe the method of heuristic creation we employed, followed by the list of proposed heuristics.

4.2.1 Methods for Creating Usability Heuristics

According to Shneiderman and Plaisant (2010, chap. 2), guidance to designers can emerge in three forms: "(1) high level theories and models, (2) middle-level principles, and (3) specific and practical guidelines". Te'eni et al. (2007, chap. 8) suggest that principles represent the theory with an eye to what should be practiced, and the guidelines take the principles one step further toward their application. Nielsen (2005a) defines heuristics as "general rules that seem to describe common properties of usable interfaces." Considering these definitions, we classify heuristics as middle-level principles. In addition, Nielsen (1994) suggests that a list of heuristics should be short (about seven to ten) and easy to teach. Heuristics should also be open to interpretation, so that multiple evaluators can use them to find diverse problems.

Two approaches can be used to develop domain specific principles: (1) Bottom-up: qualitative data is collected and analyzed to understand the characteristics of the domain, and principles are created using real-world data. (2) Top-down: high-level expert knowledge, and/or theory is used to derive specific recommendations for the target domain. Table 4.1 provides comparison

of the major literature on heuristic creation.

In a bottom-up approach, two types of qualitative data were used in the literature to synthesize heuristics. First, researchers studied positive and negative aspects of specific systems in the target domain. For example, Somervell (2004) employed claims analysis[2] or Pinelle et al. (2008) used content of product reviews for analyzing positives and negatives aspects of systems. Second, guidelines from prior literature were used to synthesize heuristics (Nielsen, 1994). The advantage of the bottom-up approach is that the heuristics are grounded in real-world data, and reflect real problems with the tools in the target domain. The disadvantage is that the produced heuristics are limited by the scope and richness of the qualitative data and the interpretation of that data by the researchers.

In a top-down approach, expert knowledge is used to derive heuristics from high-level theories or principles. Heuristics can be derived from a substantive theory,[3] a formal theory,[4], or existing heuristics. For example, Baker et al. (2002) used the mechanics of collaboration framework (a substantive groupware theory), or Scholtz and Consolvo (2004) used the general HCI literature (formal theories) to derive heuristics. Also expert knowledge can be used to modify Nielsen's heuristics for the target domain (Mankoff et al., 2003). The top-down approach relies on expert knowledge to modify a theory or an existing heuristics set, and customize it for usability evaluation of the domain specific systems. Therefore, the process of heuristic derivation is not systematic, and is prone to researcher bias.

To address the above limitations, a more rigorous process can be used by combining both bottom-up and top-down heuristic creation. The process can be started bottom-up from empirical data by using a qualitative data analysis method such as Grounded Theory (Glaser and Strauss, 1967). This process will result in design guidance grounded in empirical data. Then a

[2] See (Carroll and Rosson, 1992) for details of the claims analysis method

[3] "a theoretical interpretation or explanation of a delimited problem in a particular area." (Charmaz, 2006, pg. 189)

[4] "a theoretical rendering of a generic issue or process that cuts across several substantive areas of study." (Charmaz, 2006, pg. 187)

top-down approach can be used to justify, support, and combine the identified design guidance into heuristics. We advocate that the complementary top-down approach be rooted in a theory rather than expert knowledge, to leverage a more formal and less ad-hoc process. The use of theory reduces researcher bias in interpreting qualitative data, and abstracting and refining the findings into heuristics. In addition, a link between theory and heuristics will provide insight into the theory behind the heuristics, and help in communicating them to others. In the literature, Somervell (2004) adopted this approach by combining both bottom-up and top-down approaches systematically. We used a similar approach to create ITSM heuristics that suits best to the ITSM context. Unlike large screen information exhibits systems (the domain of interest in (Somervell, 2004)), which are limited in number, there are a vast number of ITSM tools. Because analyzing those tools was not a viable approach, we used literature and interviews as a data source, and grounded theory as the analysis method. For our top-down approach, we used formal theory as there was no substantive theory for ITSM. We further discuss our creation process in the next section.

4.2.2 Our Methodology for Creating ITSM Heuristics

We used a combination of top-down and bottom-up approaches to develop ITSM heuristics. These approaches consisted creating guidelines from the literature[5], and interpreting and explaining guidelines using the theoretical lens of activity theory.

Guideline creation: We started with a bottom-up approach by understanding the characteristics of IT security management (ITSM) tools that help SPs perform activities more effectively and efficiently. We collected data from two sources: related work and interviews performed in the HOT-Admin project (see Section 2.2.1). We first analyzed a set of primary publications that included HOT-Admin publications (4 papers) and other publications about ITSM tools

[5]We elaborated the guideline creation in Chapter 3. In this chapter, we dedicate one paragraph to reiterate and summarize the guideline creation process, as it is an important part of the entire heuristic creation methodology.

Table 4.1: Comparison of the major heuristic creation literature. The "T", "B", and "?" indicate top-down, bottom-up, and unknown method of heuristic creation.

Author	Domain	Creation method	Creation method details
Nielsen (1994)	General	BT	First, a bottom-up approach was used to gather 101 usability guidelines and principles from different sources. Then a top-down approach was used by performing factor analysis and expert review to combine and narrow the guidelines to heuristics.
Mankoff et al. (2003)	Ambient displays	T	The heuristics were developed by changing Nielsen's heuristics based on prior experience of authors and by getting feedback from experts in designing ambient displays.
Baker et al. (2002)	Shared visual workspaces	T	Mechanics of collaboration theoretical framework was used to derive heuristics. The exact process of derivation was not described.
Greenberg et al. (2000)	Groupware	T	Locales framework concepts were re-cast as heuristics.
Pinelle et al. (2008)	Computer games	B	108 game reviews from Gamespot.com were analyzed using qualitative analysis techniques, problems were extracted from reviews, categorized, and then heuristics were derived from the categories.
Scholtz and Consolvo (2004)	Ubiquitous computing	T	Expert knowledge and general HCI literature were used to derive a framework for evaluation of ubiquitous systems.
Somervell (2004)	Large Screen Information Exhibits (LSIE)	TB	A bottom-up approach was used by performing claims analysis Carroll and Rosson (1992) to analyze design decisions of five major LSIE systems. A top-down approach was used to combine similar claims and derive high-level heuristics. Each claim was classified according to its impact on three critical parameters for design of notification systems. In addition, scenario-based design categories Carroll and Rosson (1992) were used to further classifying the claims. The heuristics were created using the categories.

(14 papers). We identified 164 explicit guidelines for building ITSM tools, recommendations for improvement, design decisions in a particular tool that have positive impact on usability, and pros and cons of various tools. We categorized these using Grounded Theory. First, we performed open coding using codes that emerged from the data, followed by axial coding to combine conceptually similar open codes. Meanwhile, following the theoretical sampling technique, we broadened our sources of data by reviewing the papers published in well-known conferences related to the topic, performing keyword searches, and mining the references from our original set of 18 papers. Our goal in this stage was to saturate the identified themes in the first round of analysis, refine the identified guidelines and find a better relationship between them. The result of this search was a list of 56 papers. We then reviewed the papers and found another 22 papers that could contribute to our guidelines. We also analyzed five semi-structured interviews with SPs to find support for our guidelines and illustrative examples. This process resulted in 19 guidelines for ITSM tools (see Chapter 3 for details of the guidelines).

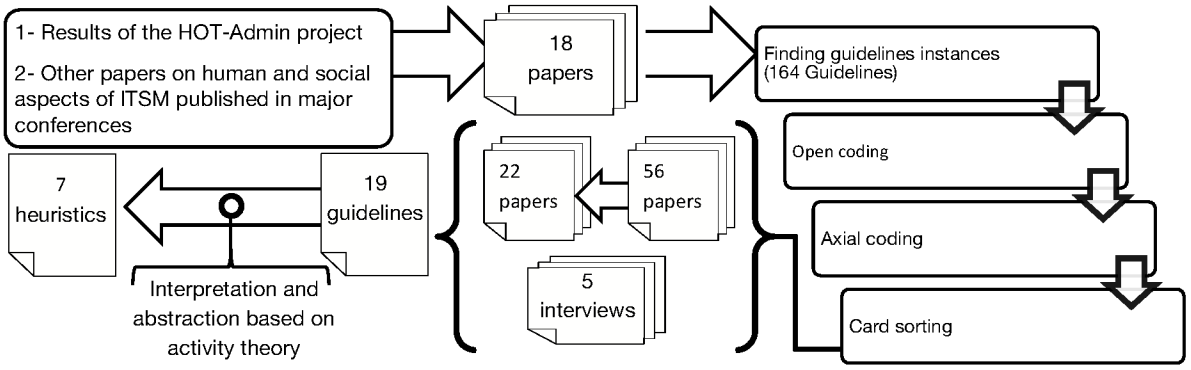


Figure 4.1: Overview of the process of developing ITSM heuristics

The identified guidelines were based on collected data and were specific and limited to the data we analyzed. Therefore, we used a top-down approach to look at the guidelines through a theoretical lens provided by a theory that can describe the characteristics of ITSM domain. Using theory leveraged our interpretation of data and added another level of validation to our findings.

Choosing a theory: To choose a theory that can be applied to the analyzed data, we searched

for specific IT security management theories, but to our knowledge, no such theory existed. We then sought general HCI theories that have been applied to contexts with technological, social, and organizational complexity. Information processing psychology, which has been extensively used as the dominant theoretical foundations for HCI according to Kaptelinin et al. (2003), was the first candidate; but it was rejected quickly as it doesn't take into account the context in which users' actions are situated. Consequently, we reviewed three widely used post-cognitivist theories: Activity Theory (Kaptelinin and Nardi, 2006), Distributed Cognition (DCog) (Hollan et al., 2000), and Phenomenology (Dourish, 2001). According to Kaptelinin et al. (2003), all of these theories can be used as foundations for understanding the use of technology in a social and organizational context. As we describe next, we found the activity theory perspective to be the best fit for the ITSM domain. From the phenomenology perspective, each instance of human activity is unique and different from other instances. Phenomenology argues against abstracting human activities and finding commonalities between various instances. Such a perspective is advantageous in describing a specific human activity, but according to Nardi (1995), it has limited ability in "higher-order scientific tasks where some abstraction is necessary" (e.g., developing abstract heuristics). From the DCog perspective, a social system can be modeled as a network of people, and artifacts, all of which are capable of cognition and transformation of information. Two main assumptions of DCog are the symmetry of human and tools, and smooth functioning of the system. While such viewpoints can be advantageous in contexts where smooth functioning and limited creativity is expected (e.g., the call center in an organization), it is limited in the ITSM domain, which involves unknown situations, breakdowns, creative use of artifacts, judgment and reflection, contradictory goals, and learning. Activity theory principles fit well with ITSM characteristics. For example, principles such as contradictions can describe breakdowns, mediation can describe creative use of artifacts, and internalization and externalization can describe judgment and reflection. Additionally, the prior use of activity theory by Zager (2002) for modeling certain aspects of IT security shows the fit between the theory and the domain.

Applying the theory: To use activity theory to abstract and combine the guidelines, we analyzed the guidelines using the theoretical lens provided by activity theory. We used ten activity theory principles from two well-known sources by Engeström (2001), and Kaptelinin and Nardi (2006) (see Section 2.4 for the list of principles), and cross-tabulated them with the guidelines in a matrix. The matrix allowed us to summarize how theory explains each guideline. We then chose one of the principles as the main explaining principle and the rest as supporting principles before combining guidelines explained by the same main principle. This led to 13 guidelines combined under six categories. The remaining guidelines could not be classified under a single category as the guidelines had different components explained by different main principles. These guidelines were broken down and classified under four of the previous categories and a new category. We then tried to convert each category into a heuristic. When categories are crafted as heuristics, they should be concise, easy to understand, and open to interpretation. We used an iterative approach of multiple piloting sessions and getting feedback from peers. We illustrate an example of our heuristic synthesis process in Figure 4.2. In this example, we generated the guideline "Make tools combinable" (Fig. 4.2b) using six sources (Fig. 4.2a), and "make tools customizable" using four sources. Then activity theory could explain "Make tools combinable" by the creation (*externalization*) of mediating artifacts (*mediation*) to address unexpected conditions (*contradictions*), and "make tools customizable" by the customization (*externalization*) of the mediating artifacts (*mediation*) as users' knowledge or activity evolves over time (*development*) and the tool is no longer best suited to the activity (*contradictions*). We then chose "mediation" as the main principle. We then combined these two guidelines into "flexible mediation" (Fig. 4.2d). We later reworded the heuristic to "flexible representation of information" based on the feedback from our pilot testing participants (Fig. 4.2e).

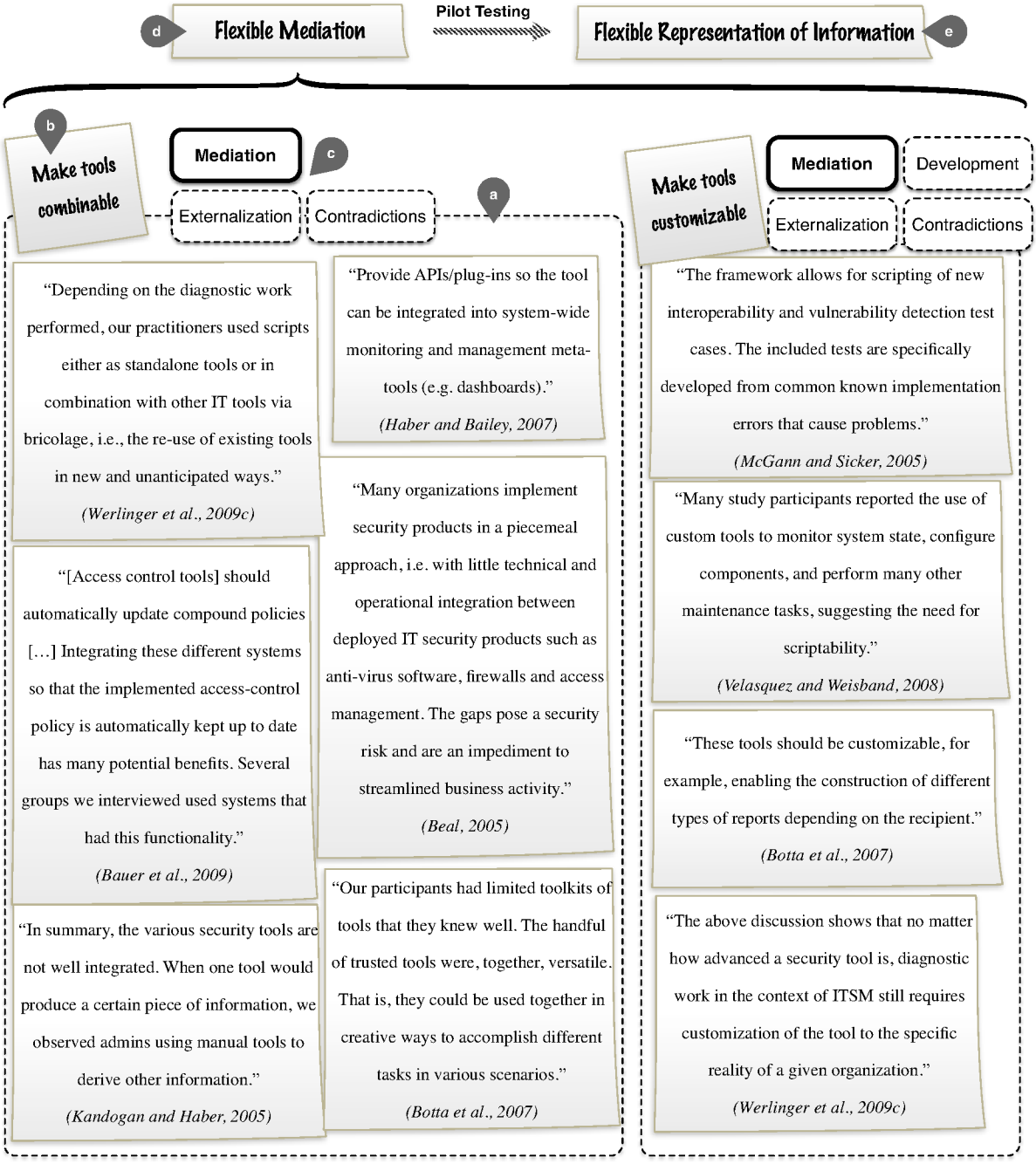


Figure 4.2: An example of heuristic synthesis process: we used a bottom-up approach by analyzing literature on ITSM tools (a) to create guidelines (b), and a top-down approach (c) using activity theory to extract preliminary heuristics (d) which later were reworded to final heuristics (e).

4.2.3 Proposed ITSM Heuristics

In this section, we present seven heuristics for the usability evaluation of ITSM tools. We provide the title and the description of each heuristic and then empirical support for it from the literature. To illustrate the importance of the heuristic with real-world examples, we include interview snippets from seven interviews with SPs conducted as a part of our ongoing research projects (participants are identified by codes from SP1 to SP7). We then provide theoretical support for the heuristics.

Heuristic #1 - Visibility of activity status: *"Provide users with awareness of the status of the activity distributed over time and space. The status may include the other users involved in the activity, their actions, and distribution of work between them; rules that govern the activity; tools, information, and materials used in the activity; and progress toward the activity objective. Provide communication channels for transferring the status of the activity. While providing awareness is crucial, limit the awareness to only what the user needs to know to complete his actions."*

Discussion: In ITSM, the actions that form an activity are distributed across time and space. These actions are performed in an organizational context with certain norms and rules. Plans are created and modified by different stakeholders, and roles are assigned dynamically to address unknown conditions. Prior ITSM research shows the importance of providing awareness of organizational constraints (Zager, 2002), communication channels (Werlinger et al., 2009b), methods for sending cues to different stakeholders to inform them about when and how to act (Botta et al., 2011), awareness of what other stakeholders perform in the system, sharing the system state between different SPs and grounding new participants in ITSM activities (Haber and Bailey, 2007). To illustrate this, SP2 said his team receives daily reports on employees' malicious actions and described the importance of awareness in preventing insiders' malicious behavior: *"We can lock down - we use policies and things like this to keep people from doing*

what they shouldn't be doing. We lock down firewalls so that they cannot do what they shouldn't be doing. Because we are running reports and you know who is doing stuff."

Looking at the problem through the lens of activity theory, tools can provide awareness about the components of activity including artifacts, community, and rules. Carroll et al. (2003) identify three types of awareness: (1) social awareness, the understanding of current social context in an activity (e.g., rules, artifacts); (2) action awareness, the understanding of actions of collaborators on shared resources; and (3) activity awareness, the understanding of how shared plans are created and modified, how things are evaluated, and how roles are assigned. As ITSM tools deal with sensitive information, they should follow the recommendation by Erickson and Kellogg (2000) and provide visibility in the form of social translucence rather than social transparency.

This heuristic is different from Nielsen's *"Visibility of System Status"*, which focuses on immediate status of the system to help users select appropriate actions and evaluate the outcome of their actions. The ITSM heuristic includes aspects of the system status that might not be available locally and immediately.

Heuristic #2 - History of actions and changes on artifacts: *"Allow capturing of the history of actions and changes on tools or other artifacts such as policies, logs, and communications between users. Provide a means for searching and analyzing historical information."*

Discussion: Accountability and reflecting on work are important aspects of ITSM (Gagné et al., 2008; Velasquez and Durcikova, 2008). As ITSM involves creative work to address unknown conditions, providing usage histories supports creativity, learning, and quality improvement (Shneiderman, 2000). Audits, which aid in reflecting on work, are mandated in IT security as a part of regulatory legislations such as the Sarbanes-Oxley Act (see Sarbanes, 2002). Prior ITSM research by Gagné et al. (2008) shows SPs archive logs and keep a history of communications for audit and accountability. To illustrate, SP7 described that health care

organizations allow physicians to openly access patient data but they archive and audit every access attempt: *"I let you do it, but I audit a crap out of the system. So if somebody complains or someone reports that I saw somebody accessed something and I don't think it is appropriate then you've got a really robust audit records."* Histories can also be used to understand other stakeholders' actions. For example, Bauer et al. (2009) found that sometimes access control policies are changed by multiple SPs; keeping track of changes will help other SPs maintain a working knowledge of the implemented policy. Finally, Velasquez and Durcikova (2008) show that historical information can be used for trend analysis, learning about the network, and evaluating the outcome of actions that span time and space.

From the theoretical perspective, artifacts in an activity carry a history with them. Awareness of this history influences the way those artifacts are used. Hollan et al. (2000) studied experts working in complex environments and found that usage histories are incorporated in cognitively important processes. Historical information could be in the form of the usage histories of the user himself or of other users of the system. According to Shneiderman (2000), usage histories can be employed to reflect on work, and to get feedback from peers.

Heuristic #3 - Flexible representation of information: *"Allow changing the representation of information to suit the target audience and their current task. Support flexible reports. Allow tools to change the representation of their input/output for flexible combination with other tools."*

Discussion: Botta et al. (2007) show that SPs often use inferential analysis and pattern recognition to develop policies, audit security, or troubleshoot security incidents. For example, they need to look for certain patterns in network traffic to detect an anomaly; or they need to analyze users' access to different resources in order to build an effective set of role-based access control (RBAC) roles. To perform these activities, SPs often use their tools in creative ways that were not anticipated by tool developers; or alternatively, they combine their tools. For example, SP2 described their reason for building custom tools: *"Sometimes, I do need to custom craft*

something or I need to automate something. Or I need to do something maybe that the tool doesn't do." Botta et al. (2007) identify SP's practice of bricolage (i.e., combining different tools in new ways) to address complex problems and argue that ITSM tools should survive in the arena of bricolage. Haber and Bailey (2007) and Beal (2005) also highlight the need for better integration between ITSM tools. SP3 described the problem with correlating data from 17 vulnerability analysis tools: *"We are really, really having a problem at correlating output from all these tools. At the beginning they were using three or four, it was easy to manually correlate, but when they started hitting six, seven, eight, plus, it was very difficult to correlate because the outputs are all different"*. Therefore, they wrote a homegrown solution to convert and import all the data into a database, and cross-reference the findings of different tools.

Tools should also be flexible in representing information. This allows users to use a representation that best suites the task, and the background and expertise of the user. SP3 described why they preferred vulnerability analysis tools with command line interface when they built the homegrown solution: *"We actually use the command line interface route and we try to keep it as simple as possible because we were putting another layer on top of it we couldn't go into the graphical one."* SP3 also clarified why some users prefer GUI based tools: *"... sometimes clients want graphical stuff. Especially if they are not 100% techie, it's easier."* In prior ITSM research, the need for flexible interaction methods (e.g., Command Line Interface and Graphical User Interface) (Botta et al., 2007; Thompson et al., 2007), flexible reporting (Botta et al., 2007; Velasquez and Weisband, 2008; Werlinger et al., 2009b), visualization techniques (Dourish and Redmiles, 2002), and multiple views (Haber and Bailey, 2007) are highlighted.

From the activity theory perspective, ITSM tools are mediating artifacts. Their mediation role can be between users (e.g., wiki, or other communication channels), between users and other tools (e.g., visualization of network traffic), or between two other tools (e.g., a script) (Botta et al., 2011; Maglio et al., 2003). Therefore, a tool should be able to process an input from a user or another tool, and to provide an output that is understandable to the user or the tool that

receives the output. This concept was further explained by Norman (1991). He described that artifacts have two types of representation: the internal representation that is not accessible by the outside world, and the surface representation that is their interface to the world. Providing a flexible surface representation is particularly important in ITSM as Botta et al. (2011) show that tools are creatively combined by SPs and their output is used by users with different knowledge and background. Previous activity theory research by Rabardel and Bourmaud (2003) also argued in favor of highly customizable and open tools, when users combine and adapt different tools to build instruments for unexpected and unknown conditions.

Heuristic #4 - Rules and constraints: *"Promote rules and constraints on ITSM activities, but provide freedom to choose different paths that respect the constraints. Constraints can be enforced in multiple layers. For example, a tool could constrain the possible actions based on the task, the chosen strategy for performing the task (e.g., the order of performing actions), the social and organizational structure (e.g., number of stakeholders involved in the task, policies, standards), and the competency of the user."*

Discussion: ITSM tools are used in organizational context with rules, norms, and constraints. Violating these constraints will result in sub-optimal situations; therefore, tools can help enforce such constraints. Botta et al. (2011) show that enforcing norms by ITSM tools in the form of procedures for notification and support for particular templates and standards can prevent communication and collaboration breakdowns. Werlinger et al. (2009a) argue that ITSM tools can promote security culture in organizations and address the lack of training by enforcing policies. SP4 further clarified the importance of leveraging policies using tools: *"So really IDS [Intrusion Detection System] should really be something that supports you in your ability to leverage policy so really security I think is 90% policy and the rest of it is tools. [...] You've got to have the policy structure behind you and the tools to find out if your policies are being respected."*

From the activity theory perspective, there are rules and norms that govern every activity.

According to Kaptelinin and Nardi (2006), promoting rules and norms by tools can lead to awareness and internalization of those norms by stakeholders. Vicente (2000) points out the importance of enforcing rules and constraints by tools, while allowing users to flexibly explore the possible action space. This helps users be aware of constraints, and gives them flexibility to adapt to unexpected situations. Vicente argues that constraints can be expressed at five different levels: work domain, control tasks, strategies, social-organizational, and worker competencies. Rules in ITSM can include security and privacy policies or standards, organizational constraints, and organizational culture.

Heuristic #5 - Planning and dividing work between users: *"Facilitate dividing work between the users involved in an activity. For routine and pre-determined tasks, allow incorporation of a workflow. For unknown conditions, allow generation of new work plans and incorporation of new users."*

Discussion: Botta et al. (2007) show that the ITSM context requires quick responses to unknown conditions by stakeholders, who work with tight schedules in which ITSM has a low priority. Therefore, Werlinger et al. (2009a) show that planning and dividing work between stakeholders is important. SPs often need to coordinate activities with multiple stakeholders involving other SPs, IT admins, managers, end-users, and external stakeholders. For example, to address a security incident, SPs often need to collect data from end-users; analyze the incident; coordinate and collaborate with IT specialists, who own the impacted sub-systems (e.g., database admins); communicate with managers to warn them about the risks associated with the incident and possible disruptions in service; and even collaborate with external SPs to solve the problem. In all of these cases, proper planning tools should be available to quickly involve stakeholders and divide work between them. SP1 described the importance of dividing work on the efficiency of the security group: *"First and foremost, explicit definition of what you do and do not do. Everyone is capable of doing more than they can do on a day-to-day basis. However, if you haven't established clear lines of demarcation from what is your responsibility*

and what is not, particularly security people have the need to save the world and so they tend to do everything and therefore they burn out."

Activity theory points to the division of labour as an important aspect of activity. Furthermore, division of labour should take into account constraints at the social organizational level, as well as possible methods for generating plans and collaborating considering those constraints.

Heuristic #6 - Capturing, sharing, and discovery of knowledge: *"Allow users to capture and store their knowledge. This could be explicit by means of generating documents, web-pages, scripts, and notes, or implicit by providing access to a history of their previous actions. Tools should then facilitate sharing such knowledge with other users. Furthermore, tools should facilitate discovery of the required knowledge source. The knowledge source can be an artifact (e.g., document, web-page, script) or a person who possesses the knowledge. Provide means of communicating with the person who possesses the knowledge."*

Discussion: According to Botta et al. (2011), SPs rely heavily on knowledge to perform their tasks. For example, to implement security access controls, a SP needs to know about the activities that a stakeholder performs, and the resources required for performing those activities. When asked how one can know the people that should be contacted in order to grant access to any of the 600 available applications in the organization, SP6 responded: *"So our access procedures state that every application that has any level of criticality is supposed to have a published knowledge-base document in our service desk. [It] defines what the application is, who owns it, who is the technical owner."* Therefore, SPs need to discover and use the knowledge of other stakeholders whether they are inside or outside of the organization. Prior research shows the importance of managing knowledge (Botta et al., 2011; Kesh and Ratnasingam, 2007) and suggests policy specification as a method to transfer knowledge (Werlinger et al., 2009a). Rogers (1992) shows the importance of transmitting knowledge at the "window of opportunity" during troubleshooting in a network environment that involves multiple stakeholders and describes it as a challenging task.

From the theoretical perspective, the relationship between different actors in the activity is mediated by artifacts. According to Engeström (1999), in order to transfer knowledge, users should be able to externalize their knowledge as artifacts. Facilities for identification and access to the required knowledge must then be provided. If externalization of knowledge is not feasible, a method for finding and starting collaboration with the person who possesses the knowledge should be provided. In this case, the communication channel is considered the mediating artifact.

Heuristic #7 - Verification of knowledge: *"For critical ITSM activities, tools should help SPs validate their knowledge about the actions required for performing the activity. Allow users to perform actions on a test environment and validate the results of these actions before applying them to the real system. Allow users to document the required actions in the form of a note or a script. This helps the users or their colleagues to review the required actions before applying them to the system."*

Discussion: Many actions in ITSM are responses to new, unseen, and complex situations (Botta et al., 2011, 2007), and they are performed on artifacts critical to the organization. Moreover, the actions are distributed in time and space and the result of an action cannot be evaluated in real time. Therefore, errors in ITSM activities could lead to a security breach or disrupt services, which might impose high costs on the organization. For example, Botta et al. (2011) show that an error during deployment of a security patch might disrupt service and conflict with an organization's business activities. It can be hard to predict, or instantly determine, the outcome of the patching process, as other stakeholders need to confirm that the service is not impacted. To mitigate this, Haber and Bailey (2007) show that SPs employ "rehearsal and planning", by rehearsing the actions on a test system before performing it on a production system. SP5 described this activity during installation of an IdM system update: *"[We have] multiple environments where we can rehearse different [changes to the system]. Because the customer releases are so complex to do, you definitely want to try it a couple of times before*

you do it in production."

This practice can be clarified from a theoretical perspective. To find a solution to a problem, a SP consults several information sources and combines them into a single artifact (e.g., a plan, a guide document, a check list). This artifact acts as an external memory to the SP. The SP also internalizes knowledge from different sources, which might not be completely correct or applicable to the situation at hand. Therefore, this knowledge should be verified before applying it to the system. Engeström (1999) explains that the process of revising knowledge involves externalization, revision, and internalization of the revisions. In the context of ITSM, SPs perform externalization when they employ rehearsal. If something goes wrong in the rehearsal, SPs re-examine their interpretation of the external knowledge sources and go through the rehearsal and revision cycle again. After successful rehearsal, SPs can perform the rehearsed actions on the critical artifact.

4.3 Evaluation Methodology

Background: While the ITSM heuristics are grounded in empirical data and supported by theory, the effectiveness of them must be evaluated. Heuristic creation literature has tackled the problem of evaluation in four ways: (1) no evaluation or informal evaluation (Greenberg et al., 2000; Scholtz and Consolvo, 2004), where the effectiveness of heuristics have not been formally evaluated; (2) long-term evaluation by using and refining the heuristics in real-world projects (Nielsen, 1994); (3) controlled study of the effectiveness without using a control group (Baker et al., 2002; Pinelle et al., 2008); and (4) controlled comparative evaluation, where the effectiveness of heuristics is compared to existing heuristics (Mankoff et al., 2003; Somervell, 2004).

We chose the last method to evaluate the effectiveness of the ITSM heuristics. Similar to other domain specific heuristics, we did not use a long term evaluation approach as it requires

longitudinal studies, and access to real-world usability projects. The controlled study without a control group does not allow recommending the use of the new heuristics over Nielsen's. A controlled comparative evaluation can show us if the new heuristics are more effective than Nielsen's for the ITSM domain, and if using them adds value to the heuristic evaluation.

According to Hartson et al. (2001), the ultimate criteria for effectiveness of a set of heuristics (or a usability evaluation method in general) is finding real problems that user will encounter in real work contexts, which will have an impact on the usability (e.g., user performance, productivity, and/or satisfaction). However, Olson and Moran (1998) argue that it is almost impossible to determine if each usability problem is real or not. The best we can do is to estimate the impact of the potential problem on the users who will use the system. Therefore, we evaluated our approach based on the following criteria proposed by Hartson et al. (2001): (1) thoroughness, the ability of the method to find most of the known problems (see Section 4.3.1 for the definition of known problems); (2) reliability, the ability of the method to find severe problems; and (3) validity, the ability of the method to find valid problems (4) effectiveness, ability of the method to find most of the known problems while it leads to few invalid problems, (5) cost-effectiveness, the cost of using method. While Hartson proposed six criteria for evaluation, we excluded downstream utility, which refers to the quality of the reported problems and how well they lead to effective redesign of the technology. According to Hartson et al. (2001), while evaluating the downstream utility of usability evaluation methods is commendable, it requires long-term studies of the impact of identified problems and it is out of the scope of this thesis.

We also investigated the characteristics of heuristics evaluation using the ITSM heuristics including (1) the impact of the number of evaluators on the results; (2) performance of individual heuristics; (3) similarity between ITSM and Nielsen's heuristics; (4) the impact of participants' background on the reported problems; and (5) the usefulness, learnability, and applicability of heuristics.

To achieve the aforementioned goals, we performed a comparative study of the ITSM heuristics

with Nielsen's heuristics. This between-subjects study divided participants into two groups: those that used Nielsen's heuristics (Nielsen condition, 14 participants) and those that used the ITSM heuristics (ITSM condition, 14 participants). For the Nielsen condition, we performed four in person (three, two, two, and one participants per session) and six remote evaluation sessions (one participant per session). For the ITSM condition, we performed three in person (three, three, and one participants per session), and seven remote evaluation sessions (one participant per session).

Recruitment: The main inclusion criteria were a HCI background, and familiarity with heuristic evaluation. We sent emails to all graduate students in the CS and ECE departments of UBC. We also sent emails to the user experience mailing lists in Vancouver, to online HCI communities, and the CHI-Announcements mailing list, in order to reach participants with HCI experience; and to the HCI-Sec mailing list[6] to reach participants with a background in both security and usability. All participants were given a $50 honorarium for their participation.

Participants: In an attempt to balance the expertise of participants in each group, we screened them to assess their HCI and computer security background. In Table 4.2, we present the participants' demographics. All but one participant had received formal HCI training, with the majority (17) receiving formal training on heuristic evaluation. The majority (19) had performed at least one heuristic evaluation in the past. The participants' average years of professional computer security experience in ITSM condition was about 3 times more than that of Nielsen condition. This difference was mainly due to the high variance in computer security background[7]. We further examine if the difference in computer security background could have an impact on the outcome of the evaluation in Section 4.4.2.

As we described, the majority of participants had performed heuristic evaluation before. Ac-

[6]HCI-Sec is a mailing list for those who do research on usability of security technologies.

[7]There was one outlier with 8 years of professional computer security experience in the ITSM condition. Removing the outlier changes the average years of professional computer security experience as follows: $\bar{x} = 0.46$, $\tilde{x} = 0$, $\sigma^2 = 0.44$.

Table 4.2: Participants' demographics for each condition.

Condition		**ITSM**	**Nielsen**	**Total**
Group Size (N)		14	14	28
Age	19-24	2	2	4
	25-30	6	7	13
	31-35	4	1	5
	36-45	2	4	6
Gender	Female	6	6	12
	Male	8	8	16
Education	Diploma	1	0	1
	Undergraduate	6	8	14
	Graduate	7	6	13
Years of experience ($\bar{x}$, $\tilde{x}$, σ^2)	HCI research and professional	3.57, 2.5, 13.03	3.29, 2.0, 8.49	3.43, 2.0, 10.70
	Computer security research	0.64, 0, 1.93	0.50, 0, 2.57	0.57, 0, 2.18
	Computer security professional	1.0, 0, 4.46	0.32, 0, 0.52	0.66, 0, 2.52

cording to Nielsen (1994), it would be impossible to wipe the mind of evaluators of the additional usability knowledge they have, and in reality each evaluator would apply certain heuristics from the sets he or she was not suppose to use. Therefore, familiarity with Nielsen's heuristics would be an advantage for participants in ITSM group. We deliberately recruited participants with a heuristic evaluation background, and made the trade-off between controlling differences in the heuristic evaluation background and ecological validity of the study. Rather than controlling the knowledge of evaluators in the ITSM group by recruiting participants without prior exposure to Nielsen's heuristics, we recruited participants who were representative of those who will use ITSM heuristics in the real world.

Target System: We chose an Identity Management (IdM) system for performing the heuristic evaluation. An IdM system is used to manage the digital identities of users in an enterprise and control the accesses of those identities to resources. Furthermore, the system allows request, review, approval, certification, and removal of access. An IdM system is used by various stakeholders in an organization. End-users use the system for creating accounts, requesting access, or changing their passwords. Managers use the system for approving employees' requests for access, reviewing and verifying the validity of their employees' access, and checking who have

access to the resources they own. Security administrators use the system to implement the requests for access, perform large scale provisioning and de-provisioning of access, and create roles.

We chose IdM system because of its significance. We showed (Section 4.1) that IdM systems have wider reach across the organization, and are used in day-to-day activities by various stakeholders. This increases the importance of usability in such tools. Additionally, IdM systems impact the functioning of other applications, because they integrate with and manage the access to those applications.

We installed CA Identity Manager 12.0 CR3 in a laboratory environment on a virtual machine using VMWare Server. Access to the system was through its web interface.

Study protocol: An overview of the study protocol is provided in Figure 4.3; we now describe the details of each step.

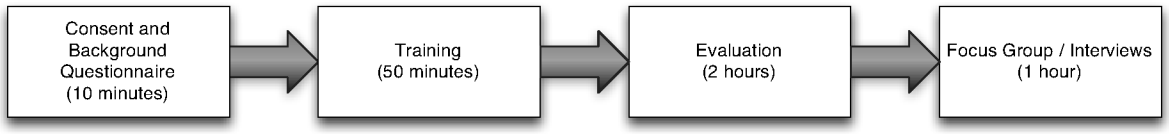


Figure 4.3: Study protocol overview

We began by obtaining the participants' consent, and then asked them to complete a background questionnaire to obtain demographic information and data to assess their HCI and computer security background.

We then provided training on heuristic evaluation, and described the specific heuristic set to be used. We demonstrated the application of the heuristics in a running example of evaluating a network firewall system. The examples were designed to reflect problems with real network firewalls. For example, we described the application of ITSM heuristic #2 using a problem where two security administrators make changes in the firewall rules, but there was no history of who made the changes. Or we described Nielsen's heuristic #4 using a problem where the

firewall rules file contained the following rule: *"eth0 inbound block"*, but in the UI, the same rule is shown as *"block all incoming connections on eth0"* (i.e., there is a lack of consistency between inbound, and incoming). We concluded the training session with an introduction to the IdM system. In all cases, training material was presented through online slides with vocal narratives. That allowed us to provide exactly the same training to all participants regardless of their location.

After the training, participants inspected the interface individually. They had access to the list of ITSM (first paragraph about each heuristic in Section 4.2.3 including the bold title and italicized description) or Nielsen's heuristics (the version available in (Nielsen, 2005a)), an instance of the IdM system, and an evaluation guide. In the evaluation guide, we limited the evaluation to a few typical usage scenarios (see (Rosson and Carroll, 2002) for description of scenarios in the context of HCI) to manage the scope of the evaluation and guide evaluators during the evaluation. The participants could then login to the system as the various stakeholders while they performed the steps of the scenarios. An overview of the four scenarios used in the study is presented in Table 4.3. We asked participants (1) to identify usability problems; and (2) for each problem, to specify the scenario and the heuristic (using an online form). Participants had two hours to perform the evaluation. We limited the evaluation time to control the time variable, avoid participant fatigue, and emphasize the discount and time-limited nature of heuristic evaluation.

After the evaluation, participants were provided with a post-evaluation questionnaire to rate their experience in using heuristics. We then conducted either a focus group (for sessions with multiple participants) or an interview (for sessions with one participant) to collect qualitative data on participants' experience.

We piloted and refined our study protocol and materials through several iterations. We performed two complete pilot study sessions (six and two participants); and we held several pilot tests as we iterated upon the individual study components, including the background question-

Table 4.3: Details of the four scenarios used during the comparative study.

Scenario	Description
Self-serve user creation	A *contractor* arrives at a company and wants to create a user account. He uses the self-service feature in the IdM system to create an account. Then *a SP* reviews and approves his request.
Bulk user creation	A *SP* receives a file containing all the users' job status changes in HR system, uploads the file to the IdM system, and troubleshoots errors.
Request privileges	An *employee* initiates an access request. The request is approved by a *manager*, and then reviewed, and implemented by a *SP*.
Certification process	The security team initiates a certification campaign and sets a deadline. Managers receive requests to review and certify the privileges of employees. At the deadline, the security team closes the certification process by revoking all of the non-certified privileges.

naire (six participants), the description of the heuristics (six), the training materials (two) and the evaluation guide (seven).

4.3.1 Data Analysis

The following steps were performed by two researchers to aggregate the identified problems and determine their severity (any inconsistencies were resolved by consensus):

Aggregating problems: To aggregate problems found in each condition and generate a list of known problems we performed two steps:

1. Problem Synthesis: We first decomposed problems into their finest level of granularity. Compound problems should be decomposed as each component of the problem might have a certain severity, and therefore a priority for fixing. Compound problems include those that refer to different actions, different artifacts, or different mechanisms in the interface. Then, if an evaluator reported duplicate problems, we removed the duplicate. We then eliminated unknown

problems, which we could not reproduce (e.g., it happened due to a sudden breakdown or crash in the system during the study, or the description of the problem was not understandable). We removed false positives, which had any of the following characteristics: (1) the problem was caused by the constraints or requirements of the underlying operating system or hardware/software infrastructure, (2) the problem was caused by the business constraints or requirements of the program, or (3) the reasoning of the participants in describing the problem was fallacious.

2. Combining problems: We began with an empty list of aggregated problems. Each identified problem was compared with the problems in the aggregated list and if it was not present, it was added to the list. Otherwise, the description of the problem and its associated heuristics in the aggregated list were updated.

Assigning severity ratings: Similar to the study conducted by Nielsen (1992), we asked four researchers with knowledge of usability and security, who had training in heuristic evaluation, to independently judge the severity of the problems. The participants used the following protocol to determine the severity of the problems: First, we asked them to answer the following questions: (1) Will the problem happen frequently to the users when they perform the activity? (2) Will it be easy for users to achieve their goal when they face the problem? (3) Is it a one-time problem that users can overcome once they know about it or will users repeatedly be bothered by the problem? Then, we asked them to use subjective judgment to categorize the problem into one of the five levels of severity proposed by Nielsen (2005a): 0-not a usability problem (I don't agree that this is a usability problem at all), 1-cosmetic (need not be fixed unless extra time is available on project), 2-minor (fixing this should be given low priority), 3-major (important to fix, so should be given high priority), and 4-catastrophe (imperative to fix this before product can be released). We gave the list of all problems to each expert without any information about the evaluators or heuristics with which the problems were found. Based on the mean rating, we categorized problems into major (mean severity > 2) and minor (mean severity ≤ 2). We demonstrate examples of problems and their severity in Table 4.4. From 131

identified problems, we chose two high severity (3.00 and 2.75), and two low severity (1.25 and 1.25) problems that were found mainly by ITSM and Nielsen participants.

4.4 Evaluation Results

Overview: Table 6.2 shows the classification of the problems in each condition. "Problem Reports" shows the initial number of problems reported by the participants. "Valid" shows the number of valid reported problems after the synthesis stage. "Known" shows the number of problems after the combining step in which we combined similar valid problems into one known problem. Table 6.2 also shows the classification of known problems as either major or minor severity. We provide a summary of participants' individual performance in Table 4.6. We calculated the performance of the strongest and weakest participants, the proportion of problems found by the first and third quartile, and the ratio between these values. These proportions are calculated based on the total number of problems (131).

Performance of heuristics: We compared the performance of the heuristics used in each condition according to their thoroughness, reliability, validity, and effectiveness. We will discuss the cost-effectiveness in Section 4.4.3. We compared the results from two different perspectives: (1) a per condition basis: we compare the output of evaluation as a whole. (2) a per evaluator basis: we compare the performance of individual participants.

Thoroughness: We calculate *thoroughness* as the proportion of the problems identified in each condition. Our results show that the evaluation with the ITSM heuristics resulted in finding 71% of total known problems (93 out of 131) while the evaluation with Nielsen's heuristics resulted in finding 66% of them (86 out of 131). In some cases, finding fewer, but more severe, problems might be more important than finding many minor problems. To examine this, as suggested by Hartson et al. (2001), we used the notion of Weighted Thoroughness (WT) by increasing the weight of the problems based on their severity. Using Equation 4.1 (we used an

Table 4.4: Examples of the problems identified by the participants. "Context" describes the context in which the problem was identified. "Problem" describes the problem. "Freq." shows the number of times the problem is reported in the ITSM(I), and Nielsen(N) conditions. "Avg. Sev." shows the average severity of the problem. "Heuristics" shows the heuristics with which the problems were identified (e.g., I4 means ITSM heuristic #4). "IC" indicates that the problem could not be associated to a heuristic by an ITSM participant.

Context	Problem	Freq.		Avg. Sev	Heuristics
		I	N		
As a part of Scenario #2, participants should upload a file that performed a bulk create, update, and remove. Out of 8 actions scripted in the file, 1 always failed, and the system showed a message that 7 records have been updated and 1 has failed.	There is no way for the user to know if the file causes an error. If the file is large, there is no way for the user to determine which record caused an error.	9	2	3.00	I1, I7, N1, N5
As a part of scenario #3, the employees could write their access request in a free text submission form, and then submit the request for processing.	During writing the request, if the user pressed enter, the request was submitted, instead of creating a new line. There was no way to edit the request again, or revert the action.	1	4	2.75	IC, N5
As a part of scenario #1, the employee could perform self-registration from the login page by filling a form. After self-registration, the user was presented with a screen saying you are successfully logged out of the system.	There is no link back to the login, or any other pages from the login page.	5	8	1.25	N7, IC
As a part of scenario #3, the security admin should review the employee's access request and grant the required access.	There is no knowledge base for the security admin to see the consequences of such access or a way to get a second opinion on giving user the access.	3	0	1.25	I6

Table 4.5: Overview of the number and classification of identified problems in each condition.

Condition	Reports	Valid	Known	Major	Minor	False Positive	Unknown
ITSM	239	201	93	38	55	18	16
Nielsen	233	187	86	20	66	45	17
All	472	388	131	43	88	62	33

Table 4.6: Individual differences in participants' ability to find problems.

Condition	$Max(\%)$	$Min(\%)$	$Q_1(\%)$	$Q_3(\%)$	Max/Min	Q_3/Q_1
ITSM	23.7	3.82	7.1	13.9	6.2	2.0
Nielsen	18.3	3.1	5.9	11.5	6.0	1.9

equivalent equation for Nielsen condition), the weighted thoroughness of ITSM and Nielsen's heuristics are 77% and 60% respectively.

$$WT = \frac{\sum_{p \in KnownITSM} Severity(p)}{\sum_{p \in Known} Severity(p)} \times 100 \tag{4.1}$$

To compare two conditions on a per evaluator basis, we tested the following hypothesis: (1) H_1: Participants will report more problems if they use ITSM heuristics than Nielsen's. H_0: There is no difference in the number of reported problems. The result of a Mann-Whitney U test did not reject H_0.

Reliability: It is important for a set of heuristics to be able to identify major usability issues as they may seriously hinder the ability of the user to operate the system effectively and efficiently. The results (Table 6.2) show that participants using the set of ITSM heuristics found almost twice as many major usability problems than the participants using Nielsen's set.

We tested the following hypothesis to show the difference in severity on a per-evaluator basis: H_1: The average severity of the problems reported by participants will be higher if those individuals use ITSM heuristics than if they use Nielsen's. H_0: There is no difference in the average severity. The result of a Mann-Whitney U test rejected H_0 in favor of H_1 (U=26,

Z=-3.309, p=0.001).

Validity: We examined whether the evaluation with the ITSM heuristics generated fewer false positives than Nielsen's. Participants using the ITSM heuristics reported 201 valid problems and 18 false positives, whereas participants using Nielsen's heuristics reported 187 valid problems and 45 false positives. The ITSM heuristics yielded fewer false positives (Table 6.2) than Nielsen's heuristics. Comparing the number of unknown problems identified in each condition revealed a very small difference between conditions.

We tested the following hypothesis about difference in the number of false positives on a per-evaluator basis: H_1: participants will report fewer false positives if they use ITSM heuristics than if they use Nielsen's. H_0: There is no difference in the number of false positives. The result of a Mann-Whitney U test rejected H_0 in favor of H_1 (U=38, Z=-2.823, p=0.005).

Effectiveness: We calculated the effectiveness using Equation 4.2 suggested by Hartson et al. (2001). We used the same weight (α) for validity and thoroughness. Our results showed that the effectiveness of ITSM heuristics was 0.80 and the effectiveness of Nielsen's heuristics was 0.72.

$$Effectiveness = \frac{1}{\alpha\left(\frac{1}{validity}\right) + (1-\alpha)\left(\frac{1}{thoroughness}\right)} \tag{4.2}$$

The number of evaluators required to perform the evaluation: To replicate Nielsen's original analysis (Nielsen and Molich, 1990), we formed aggregates of participants and found the proportion of usability problems identified by each size of aggregate. Following Nielsen's methodology, we calculated the proportion of found problems based on the total number of problems found in each condition. The result is depicted in Figure 4.4. The graph shows that increasing the number of evaluators will increase the proportion of the identified problems, but the rate of the increase diminishes as we increase the number of evaluators. The two plots from our experiment are very similar and they show a similar trend as compared to the results from

the Mantel, Groove, and GroupDraw experiments.[8] Yet, Nielsen's experiment shows faster diminishment compared to our results.

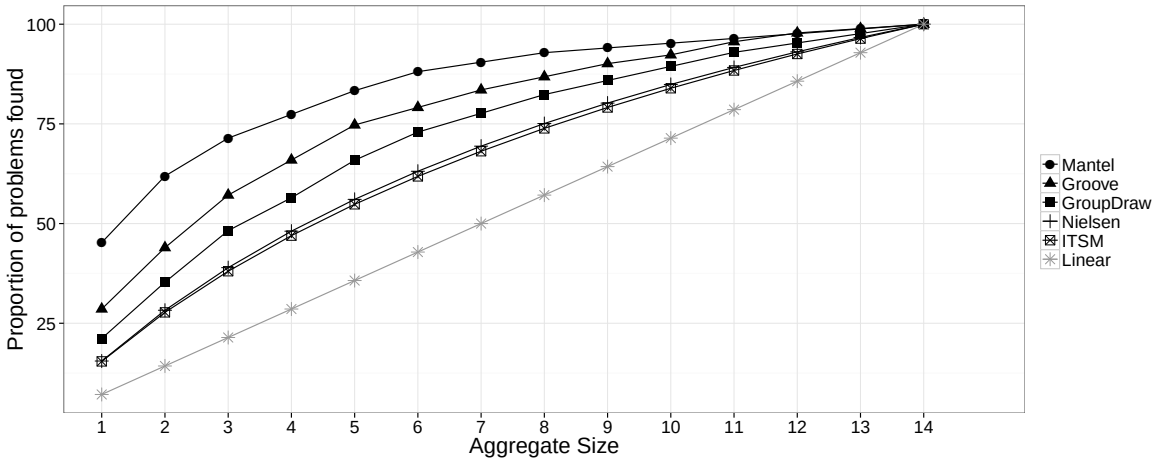


Figure 4.4: Average proportion of problems found by aggregate of participants in ITSM and Nielsen conditions. We also overlaid the results from Nielsen's Mantel experiment (Nielsen and Molich, 1990), and Baker's Groove and GroupDraw (Baker et al., 2002) experiments to allow comparisons.

We illustrate the distribution of the known problems that are found by participants in the ITSM or Nielsen condition in Figure 4.5. To generate the diagram, we grouped participants based on their condition and then sorted them from weak to strong (participant A is stronger than participant B, if A found more problems than B). We also sorted the problems from easy to hard (problem A is easier to find than problem B, if A was found by more participants than B). We highlighted the severity of the problems by color. The diagram shows that, similar to Nielsen's original experiment (Nielsen and Molich, 1990), there are easy problems that are overlooked by strong participants while there are hard problems that are only found by weak participants. Also, there were major problems that were only found by weak participants and there were minor problems that were only found by strong participants. This confirms Nielsen's argument that heuristic evaluation is a method that should be done collectively (i.e., no strong evaluator can uncover all of the major problems). Figure 4.5 also shows that there was relatively little duplication between participants in each condition. We further discuss the lack duplication in

[8]To allow comparison, and since the mentioned experiments employed more evaluators, we assumed that the total number of problems in each experiment was equal to the problems found by aggregate size of 14.

Section 4.5.

4.4.1 Performance of Individual Heuristics

To see which heuristics contributed the most in finding usability problems, we visualized the number and mean severity of known problems associated with each heuristic in Figure 4.6. The graph shows that the severity of the problems found with ITSM heuristics is higher than the ones found with Nielsen's heuristics. The graph also indicates that ITSM heuristics #6 and #7 were associated with the fewest problems and ITSM heuristic #1 was associated with the most problems. The average severity of the problems associated with ITSM heuristic #2 was the highest and Nielsen's heuristic #8 was the lowest.

As our results indicated a large overlap between the problems found using ITSM and Nielsen's heuristics (i.e., there were known problems that were found in both conditions), we further investigated the similarity between the two sets. For this we calculated a similarity metric between the two heuristics (A and B) as follows:

$$WT = \frac{|A \cap B|}{|A \cup B|} \times 100 \tag{4.3}$$

The result of the similarity analysis is presented in Table 4.7. For each ITSM heuristic, we highlighted the most similar Nielsen heuristic. These similarities are not surprising. For instance, the ITSM #1 and Nielsen's #1 can both lead to finding a subset of visibility problems; both ITSM #3 and Nielsen's #7 can lead to problems related to interacting with users with different levels of expertise, ITSM #4 and Nielsen's #5 will both lead to preventing errors by applying constraints. ITSM #6 and Nielsen's #9 are also shown to be similar, as providing users with the required knowledge will help them understand errors and recover from them. We also show how each of the Nielsen's heuristics performed in finding problems unique to their con-

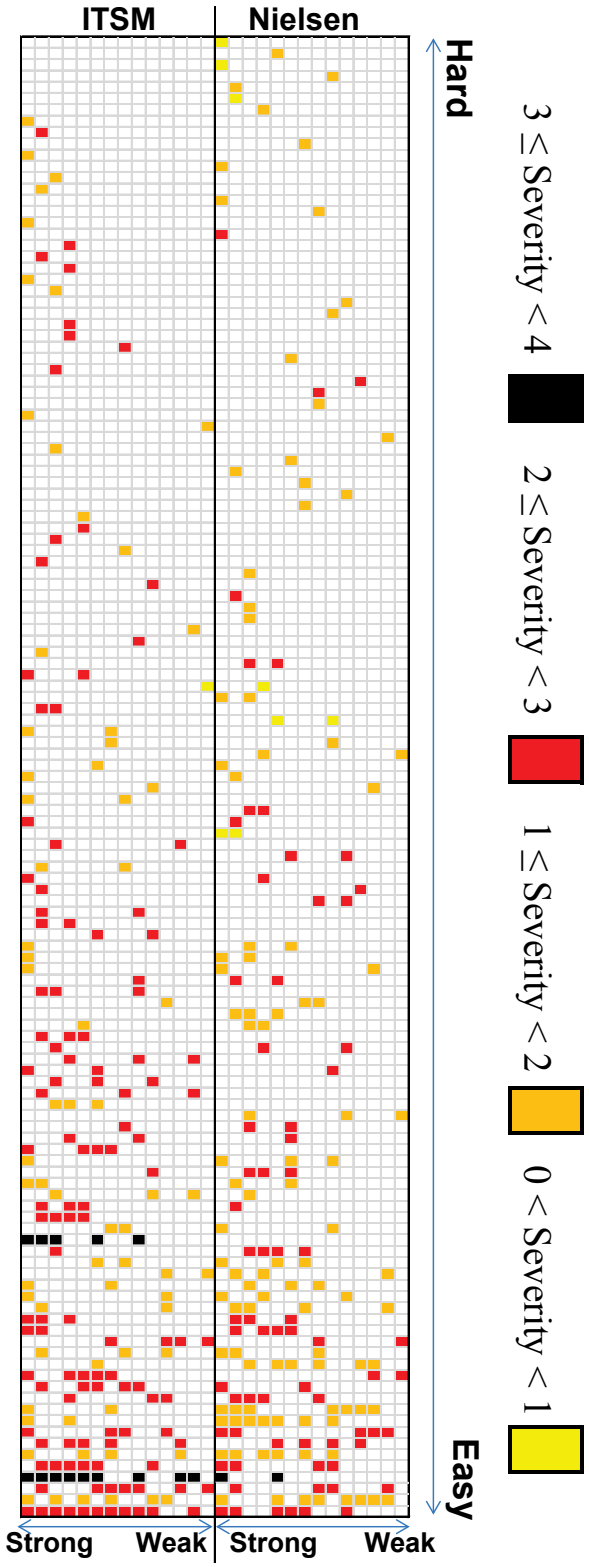


Figure 4.5: Problems identified by each participant in each condition. Each row corresponds to a participant and each column corresponds to a problem. Participants in each condition are sorted from top (weak) to bottom (strong) and problems are sorted from right (easy) to left (hard).

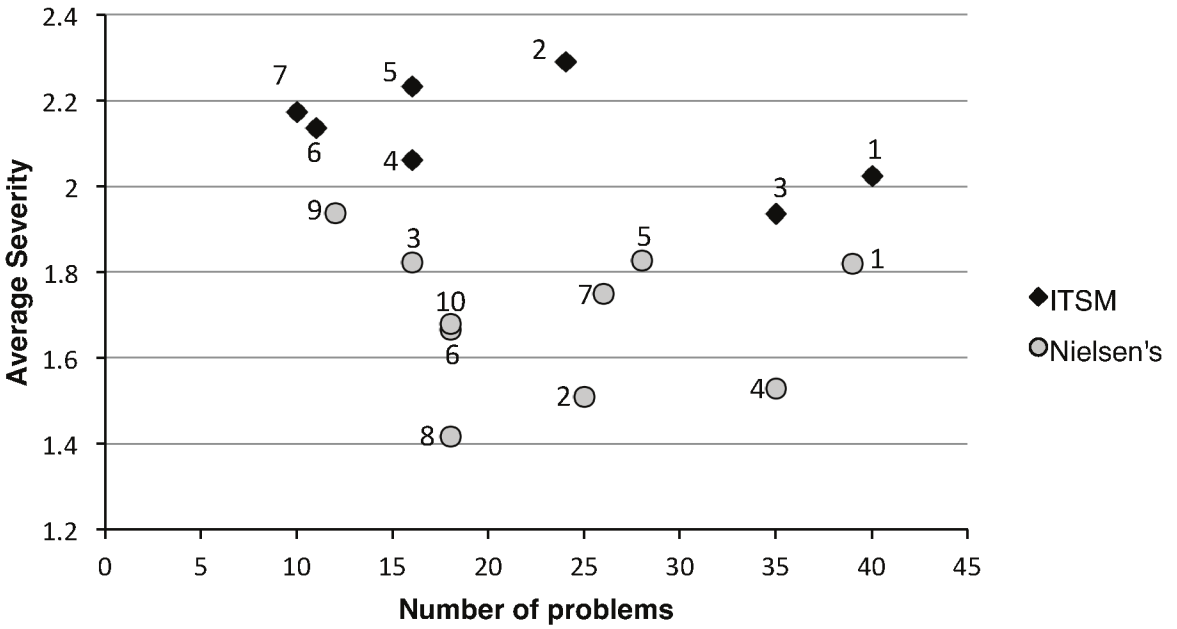


Figure 4.6: The number and mean severity of problems identified by each heuristic. The ITSM heuristics are shown using black diamonds and Nielsen's heuristics are shown using gray circles. Each heuristic is labeled with its number.

Table 4.7: Similarity between individual ITSM and Nielsen's heuristics. Each cell shows the value of similarity metric for the heuristics denoted by row and column indexes. For each ITSM heuristic, the cell with the highest number (i.e., the most similar Nielsen heuristic) is highlighted.

		Nielsen Heuristics									
		1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
ITSM Heuristics	**1**	29.5	16.1	14.3	17.2	15.3	16	17.9	11.5	10.6	11.5
	2	14.5	6.5	8.1	7.3	15.6	2.4	8.7	0.0	12.5	7.7
	3	25.4	11.1	13.3	18.6	18.9	15.2	27.1	12.8	9.3	10.4
	4	14.6	7.9	6.7	8.5	18.9	9.7	13.5	6.3	3.7	17.2
	5	5.8	7.9	0.0	4.1	4.8	6.3	10.5	3.0	0.0	0.0
	6	6.4	2.9	3.8	0.0	5.4	3.6	2.8	3.6	15.0	3.6
	7	6.5	2.9	4.0	4.7	8.6	3.7	9.1	3.7	10.0	0.0

dition (i.e., problems that were not reported in ITSM condition), and the average severity of those unique problems in Table 4.8. We will corroborate this data with participants' feedback to determine which of Nielsen's heuristics complement the ITSM heuristics in Section 4.5.

Table 4.8: Ability of each of Nielsen's and the ITSM heuristics to find problems unique to their condition. The "Proportion of unique" row shows the proportion of problems uniquely found in the Nielsen or ITSM conditions using the corresponding heuristic. The "Average severity" row shows the average severity of those unique problems.

	Nielsen Heuristics									
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
Proportion of unique	0.41	0.52	0.19	0.49	0.32	0.39	0.35	0.61	0.25	0.56
Average severity	1.47	1.31	2.06	1.10	1.49	1.57	1.42	1.18	1.42	1.33

	ITSM Heuristics						
	1	**2**	**3**	**4**	**5**	**6**	**7**
Proportion of unique	0.43	0.50	0.37	0.25	0.63	0.55	0.50
Average severity	2.12	2.31	1.87	2.38	2.35	2.29	2.10

4.4.2 Impact of Participants' Background on Their Performance

Nielsen (1992) suggests that the evaluator's HCI and domain expertise are two factors that influence the quality of heuristic evaluation. We analyzed the HCI and computer security background of the participants to find the correlation between expertise (years of HCI and computer security experience, number of previously performed heuristic evaluations), and performance (number of raw, known, false positive problems, and average severity). We first used a factor reduction technique to find the possible medium or strong correlations and then investigated correlations and their statistical significance with either Pearson's product-moment coefficient (for normally distributed data) or Kendall tau rank correlation coefficient (for non-normal data). In the Nielsen condition, we found a strong negative correlation between the number of previously performed heuristic evaluations and the average severity of reported problems ($r = -0.70, p < 0.05, N = 14$). In the ITSM condition, we found a strong positive correlation between the number of previously performed heuristic evaluations and the number of false positives ($\tau = 0.55, p < 0.05, N = 14$). For the overall study data, we identified a medium to strong correlation between the years of HCI experience and the number of reported problems ($r = 0.47, p < 0.05, N = 28$). We did not find any correlation between the severity of the reported problems and the background of the participants (i.e., between severity of the reported

problems and years of HCI ($\tau = -0.17, p = 0.21, N = 28$), professional computer security ($\tau = 0.18, p = 0.28, N = 28$), or academic computer security ($\tau = 0.19, p = 0.22, N = 28$) experience.) In Section 4.3, we reported that the average length of computer security experience in ITSM condition was more than three times higher than in Nielsen condition before removing the outlier. Yet, as we showed above, there is no correlation between the amount of computer security experience and the severity of the reported problems. This suggest the differences are due to the condition rather than participants' security experience.

4.4.3 Participants' Feedback in Post-evaluation Questionnaire

We asked our participants to evaluate with a 5-point Likert scale (5=strongly agree, 1-strongly disagree) how useful the set of heuristics was in identifying usability problems (usefulness), how easy it was to understand and learn the heuristics (learnability), and how easy it was to apply the heuristics to the IdM system (applicability). The mean usefulness, learnability, and applicability ratings for ITSM condition was 3.14, 3.36, and 2.86 respectively, and for Nielsen condition was 3.36, 3.57, and 3.5. We conducted a Mann-Whitney U test to evaluate whether the set of heuristics used impacted the usefulness, learnability, and applicability, as reported by our participants. Although, the ITSM heuristics were new to our participants there was no significant difference between the ratings for the two sets of heuristics. As we highlighted before, one measure of cost-effectiveness of a usability evaluation method is the effort required to learn it. Therefore, our results suggest that there is no statistically significant difference between the cost-effectiveness of the two sets of heuristics.

We also asked participants to evaluate with a 5-point Likert scale the usefulness, learnability, and applicability of each individual heuristics. The mean scores of the ITSM and Nielsen's heuristics are shown in Figure 4.7. Repeated measures ANOVA calculations between the mean scores of individual Nielsen's heuristics revealed only a significant difference in terms of usefulness ($F(9,117) = 2.40, p < 0.05$). Post hoc tests using Bonferroni correction showed a

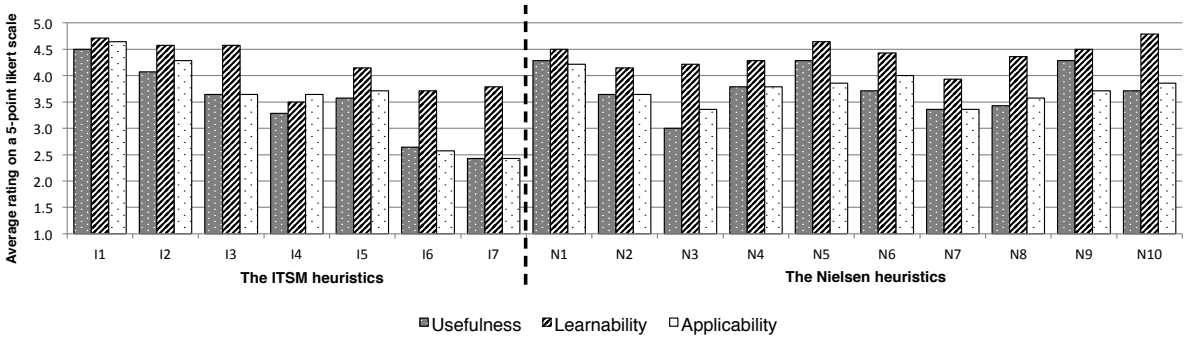


Figure 4.7: Mean scores of participants' reported usefulness, learnability, and ease of application for the different heuristics (5=strongly agree, 1=strongly disagree).

marginally significant difference[9] between heuristics N#1 and N#3 (p=0.138), N#5 and N#3 (p=0.138). Repeated measures ANOVA tests for the ITSM heuristics determined a statistically significant difference in heuristics usefulness ($F(6,78) = 10.18, p < 0.005$), learnability ($F(6,78) = 6.92, p < 0.005$) and applicability ($F(6,78) = 12.45, p < 0.005$). Post hoc tests using Bonferroni correction shows that the significant difference in usefulness was mainly caused by heuristics I#6 and I#7, the significant difference in learnability was caused by the difference between heuristic I#4 and heuristics I#1 and I#3, and the difference in applicability was mainly caused by heuristic I#7.

4.4.4 Qualitative Feedback During Focus Group/interview Session

In this section, we provide a summary of participants' feedback during interviews and focus groups. We identify participants in the ITSM condition by PI1 to PI14, and the Nielsen condition by PN1 to PN14.

We asked open-ended questions about the usefulness, ease of understanding, and applicability of the heuristics. Furthermore, we asked participants if they noticed problems that could not be found with or associated to the heuristics, and to improve the current heuristics set or add a new heuristic to it.

[9]The use of Bonferroni with large number of pair-wise comparisons leads to a very conservative result; therefore we considered the difference significant.

In the Nielsen condition, all participants confirmed the heuristics' usefulness, e.g., PN4 explained: *"They give me a standard way to review each of the screens. [...] At least be able to evaluate based on a common set of methods or processes"*. But it was challenging for some of the participants to apply heuristics to the system, without understanding the background of the real users of the system, e.g., PN8 explained: *"I found them useful for some of the actors [in the scenarios]. When it gets to [a manager], it becomes harder to get into the user's mindset. And when we get to the [security admin], it is not useful at all because he is an expert"*. This point was also confirmed by PN10, and PN13 who indicated that the heuristics such as "Flexibility and efficiency of use", and "Match between system and real world" require understanding of flexibility for a security admin and his mental model of the real world. PN8 also indicated that many of the problems that might be important for end-users, might not be as important for security admins who will be trained to use the tool.

Similarly, the ITSM participants found the heuristics useful, but not without problems. Many of the participants (PI7, PI5, PI6, PI1, PI14, PI10) indicated that while they understood heuristics #6 or #7, they were not applicable to the four scenarios in the study. Yet, PI13 indicated that it took some time to grasp the last two heuristics: *"at the beginning, I was very focused on the first heuristics. But it was towards the end that I was starting to think about the problems related to [ITSM #6, #7]. But when you start thinking about them, it becomes intuitive to see the problems related to those."* On the contrary, PI8 describe ITSM heuristic #4 as hard to apply: *"I had a hard time to apply [ITSM #4] because it can be applied at different levels. You can say the system should limit what user can enter in his request as much as possible, or user can enter whatever he wants and then it is up to manager to review and decide whether user entered a valid request."* PI8 then expressed disagreement that tools always "should" constrain possible actions, and suggested that tools "could" constrain possible actions depending on the situation.

We asked participants in the Nielsen condition about the aspects of the system not covered by

the heuristics. Four participants indicated that problems with the workflow cannot be classified in Nielsen's heuristic and that they classified them as lack of showing different steps of workflow (PN4, PN10), lack of ability to revert back to one of the previous steps of the workflow (PN6), and lack of a coherent workflow (PN1). PN12 believed that N#1 should be changed to "Visibility" because visibility can go beyond the system status. PN7 described that the interface offered too many options for performing tasks, and no heuristic covered that. Then, PN7 suggested a heuristic for changing the presentation based on the role of the user in the organization. Similarly, PN8 suggested dividing N#8 to two heuristics: (1) aesthetics and, (2) the level of detail for expert and non-expert users.

In the ITSM condition, PI6, PI14, and PI12 asked for Nielsen's heuristics set to use in addition to the ITSM heuristics, and other participants suggested individual Nielsen's heuristics. For example, PI7 indicated the need for an error prevention heuristics, as well as better error messages. PI1 suggested an error recovery like undo or redo. The need for a consistency heuristic was indicated by PI2, PI14, and PI6. A heuristic for organization of the screen was suggested by PI2, PI12, and PI6. Understandable language was indicated by PI10, PI12, PI14.

We mapped the heuristics that participants in each condition indicated as missing to one of the heuristics in the other condition. As a result, we saw participants in the Nielsen condition found ITSM heuristics #1, #3, and #4 necessary. Participants in the ITSM condition indicated the need for Nielsen's heuristics #2, #3, #4, #5, #8, and #9.

Almost all participants explained that they first identified problems and then tried to "assign" each problem to one of the heuristics. Yet, they still found the heuristics helpful in finding problems: e.g., PN2 explained: *"[The heuristics] remind you of existence of possible problems. I might forget to look at help and documentation if I don't have the heuristics."* PN9 had heuristics in mind when looking at the interface: *"For me, I found the problem and then I matched it. But I had heuristics in my mind, and when I looked at the interface I was thinking if it breaks anything."* PN8 explained the role of heuristics in disambiguating problems: *"I*

look for a submit button, I don't see it. I think, it might be a problem. But then heuristics help me find exactly what the problem might be." PI12 described the role of heuristics in predicting problems and designing test cases to uncover them: *"As soon as I read the description for scenario one, I thought 'oh I bet that is gonna break heuristic number five'. And then I figured out a little test case and tested it."* After this point was brought up by PI12, it was confirmed by the two other participants in the same focus group.

When commenting on the study procedures, participants indicated that if they had more time, they would have found more usability problems.

4.5 Discussion

The evaluation results suggest that our heuristics performed well overall in finding usability problems in ITSM tools. In this section, we interpret the results and discuss their implications.

Few overlaps between individual evaluators: We observed fewer overlaps between problems identified by evaluators in our experiment than problems identified by evaluators in Nielsen's original experiment (Nielsen and Molich, 1990). In both conditions of our experiment, only three problems were identified by the majority of participants, and more than half of the problems were identified by only one. In contrast, in Nielsen's evaluations of the Mantel and Savings systems (Nielsen and Molich, 1990), only one and two problems respectively were identified by just one participant. In Baker et al. (2002) evaluation of GroupDraw and Groove, 14 out of 64 and 5 out of 43 problems were found only by one participant. Our results show fewer overlaps between problems identified by different participants, compared to Nielsen's and Baker's results. Four factors might have contributed to this outcome. First, the evaluated IdM system was fairly large; the participants had to visit 20 different web pages in order to successfully complete all scenarios. This multitude of web interfaces provided an opportunity for finding more diverse problems than in the cases of systems used in Nielsen's or Baker's

evaluations (e.g., Mantel only had a single screen and a few system messages, GroupDraw had two screens). Second, we used fewer participants (14 per condition) compared to 77, 34, 25, and 27 participants in Mantel, Savings, GroupDraw, and Groove systems, respectively. Third, the evaluated system was a commercial product rather than a prototype and, it did not contain many obvious usability problems. Fourth, participants thought they could find more problems if they had more time. For reference, Nielsen and Molich (1990) do not mention any time constraints during their study; and Baker et al. (2002) allowed participants to use as much time as they needed to evaluate the interfaces. Our results illustrate how hard and time consuming the heuristic evaluation of ITSM tools in the size and complexity of the IdM systems is; we discuss why we limited the evaluation time to two hours in Section 4.3.

Large overlaps between known problems reported in the ITSM and Nielsen conditions: The ITSM heuristics were consistent with activity theory and Nielsen's heuristics were consistent with action theory. As a result, we expected to have very few problems that overlap the two conditions. But our results show that 48 problems (37%) were found in both conditions. Three factors might have resulted in this observation. First, participants in the ITSM condition could remember Nielsen's heuristics, which helped them see problems at the action level. Second, most of the participants found a problem first, and then fit it into one of the heuristics. Third, the similarity between heuristics (Table 4.7) could also result in overlaps.

Usefulness of ITSM heuristics: It is encouraging that despite the novelty of the ITSM heuristics, participants found them to be no less effective, easy to use, or easy to learn than Nielsen's heuristics. Yet, when we looked at the individual heuristics, we saw that ITSM heuristics #6 and #7 were not as useful and easy to apply. Looking at the number of problems reported using those heuristics confirms this observation (Figure 4.6). We can provide several explanations for this observation. First, the study scenarios did not include extensive deployment or configuration tasks that involve verification to a great extent, or tasks that deal with unforeseen conditions or troubleshooting, which require extensive knowledge sharing. Second, one

participant indicated that the last two heuristics were ignored because of focusing on the first heuristics in the list. Therefore, the order of the heuristics might have influenced their use in our time-limited evaluation sessions. Our judgment here is based on the participants' feedback and it needs further study. Third, we believe that the ITSM heuristics #6 and #7 were less open to interpretation than heuristics #1 or #3, which are applicable over a broad range of tasks.

Specificity of heuristics to the ITSM domain: While our goal was to develop specific ITSM heuristics, some of the heuristics seem to be rather general and applicable to other domains as well. This generality was the result of finding general guidelines that eventually led to creation of heuristics. Looking at the data that supports those general guidelines shows that ITSM shares characteristics with other domains. For example, it shares complexity with IT, creativity with software development, and uncertainty with military. As a result, some of the recommendations for designing better ITSM systems might be similar to the recommendations for designing other systems with similar characteristics.

An ideal set of heuristics for evaluation: Our results suggest that using the ITSM heuristics leads to finding more severe problems. However, using Nielsen's heuristics led to finding a unique set of problems that couldn't be found using ITSM; while those problems might not be as severe, addressing them can improve the interaction between user and the system. Therefore, we believe that the ITSM and Nielsen's heuristics can offer different perspectives for evaluation and they can be combined and used in three different ways all of which have trade-offs that should be considered according to the ITSM system being evaluated and the resources available for the evaluation: First, both sets can be used together in one evaluation session. This approach gives participants a holistic view of possible problems at both the action and activity levels. On the contrary, Nielsen argued that the use of more than 10 heuristics is not effective, and evaluators cannot remember all of the heuristics. Furthermore, evaluators who have previous experience with Nielsen's heuristics might tend to focus more on those heuristics and might ignore the ITSM heuristics. The second approach would be to use a

subset of Nielsen's heuristics (at most three to be consistent with Nielsen's recommendation of using at most 10 heuristics) in addition to the ITSM heuristics. Our participants suggested the need for six of Nielsen's heuristics. Based on the data from Table 4.8, we can suggest the use of Nielsen's heuristics #2, #4, and #5. The benefit of this approach is reducing the evaluators' mental overload, and allowing them to find action level problems that are critical to the target application. The drawback is that participants might focus on those three Nielsen's heuristics that they know, rather than the ITSM ones. The suggestion of specific Nielsen's heuristics is solely based on the data of this study, and the evaluated IdM system. The choice of the heuristics should be changed depending on the goal of the evaluation. For example, if the main design goal of a project is aesthetics, one can replace one of Nielsen's other heuristics with #8. The third approach would be to use Nielsen's and ITSM heuristics in separate evaluation sessions by the same or different evaluators. We expect this approach to have the highest thoroughness and yet the highest cost.

The impact of participants' background on their performance: Our results suggest that the average years of HCI experience is positively correlated with the number of reported problems, but not with their severity. This result supports Nielsen's finding that regular specialists will find more usability problems than novice evaluators. On the other hand, the results suggest that the number of previously performed heuristic evaluations negatively impacted the severity of the problems in the Nielsen condition and false positives in the ITSM condition. This observation was surprising. We hypothesize that participants with prior heuristic evaluation experience tend to evaluate the systems for end-users, with a focus on aesthetics of the interface. This resulted in minor problems in Nielsen condition, and false positives in the ITSM condition. Further study is needed to validate the reasons behind this observation.

Generalizability of evaluation results: If we were to replicate the comparative evaluation study for a different ITSM tool (e.g., one of those listed in Section 4.1), we would expect the ITSM heuristics to be still applicable and to find more severe problems than with Nielsen's. The scope

of the empirical data which the heuristics were created based on was ITSM tools in general, rather than a specific IdM system. Also, the heuristics were supported by a general HCI theory. This leads us to believe the ITSM heuristics are general enough for evaluating most ITSM tools.

At the same time, the performance of individual heuristics may vary for different categories of ITSM tools. As we discussed in Section 4.1, IdM systems have a wide reach across the organization, and are used by many users. Therefore, the study participants reported a large number of usability problems for the visibility of activity status. In contrast, a security operations tool such as network traffic analyzer, has a narrow reach across the organization and is mainly used by SPs. Such a tool should help SPs deal with complex and large scale network traffic logs, and detecting malicious, unknown content in network traffic. Therefore, the evaluators of the tool may focus on flexible representation, and knowledge sharing rather than visibility of activity. The evaluation results could vary between individual systems of the same type. For example, the evaluated IdM system offered rather meaningful error messages to users. Therefore, heuristic N#9 (error recovery) was the least used Nielsen heuristic. Another IdM system may present errors as, say alphanumeric codes, or without indicating exact problems. In evaluating such a system, we might see more use of N#9.

4.6 Limitations and Future Work

We used participants with HCI and heuristic evaluation background, and as we discussed in Section 4.3, we made a trade-off to compare the two sets of heuristics in an ecologically valid setting. As a result, we cannot make arguments about the performance of the two sets of heuristics when they are used by participants who were not previously exposed to heuristic evaluation.

While increasing the number of participants in either of the conditions did not saturate the

list of identified problems, we observed that the rate of finding problems decreased. Continuing the experiment with more participants would certainly allow us to find the point of diminishing returns in the number of problems. But comparing our results with the Group-Draw evaluation (Baker et al., 2002) suggests that even doubling the number of participants would not allow us to observe saturation. Furthermore, our study required a four-hour time commitment from participants with an HCI background, which made recruitment challenging. Because determining such a saturation point was not the main goal of our study, we leave such investigation for future work.

There are several opportunities for improvement and future work. First, during the problem synthesis stage, the severity of problems was estimated by four severity raters with a background in both usability and security. While this is a standard approach for determining the severity of problems in heuristic evaluation, it is only an approximation of severity. Asking the opinion of real system users to determine the severity of the problems is another method. Neither of these approximations might be precise, but combining the ratings would increase confidence.

4.7 Conclusion

In this chapter, we reviewed the prior research on heuristic creation. We then described our methodical way of creating domain specific usability heuristics, which we applied to create a set of usability heuristics for evaluation of ITSM tools. To examine the applicability of the heuristics, we compared their use with Nielsen's heuristics for the evaluation of an IdM system. We tried to maximize the ecological validity of the study by using a real ITSM tool, and recruiting participants with an HCI background and familiarity with heuristic evaluation.

Our results show that a combination of a top-down and bottom-up approach resulted in a set of heuristics that were applicable to the target domain (as they were based on the domain-specific

data), and yet were general enough to help evaluators find diverse set of problems. Comparing the new heuristics to Nielsen's heuristics revealed that the severity of the problems found by participants in the ITSM condition was higher than those found in Nielsen condition. Furthermore, our participants found the ITSM heuristics to be as relevant, easy to apply, and easy to learn as Nielsen's. The results of our evaluation also shed light on the use of the heuristic evaluation for evaluating a complex domain-specific system. While Nielsen found that five evaluators are able to find about two thirds of the problems, in our evaluation of the IdM system, five evaluators only found about half of the problems found by 14 evaluators. Additionally, the complexity and scale of the system can result in a lack of overlapping problems between evaluators. Finally, our results show that Nielsen's heuristics can also be effective in finding a class of problems in ITSM tools that cannot be found by the ITSM heuristics. Therefore, we discussed three approaches for using a combination of the two sets of heuristics.

The ITSM heuristics are a component of tool usability evaluation and can be used as a discount method to find usability problems in prototypes or actual tools. These problems can be further investigated by a user study or a contextual inquiry session. Design guidelines (e.g., Chapter 3) can then be used to address the problems.

Chapter 5

Field Study of Identity and Access Management[1]

Identity and Access Management (IAM) comprises the processes and infrastructure for the creation and maintenance of user's digital identities and the designation of who has access to resources, who grants that access, and how accountability and compliance are maintained (Blum, 2005). While IAM can be studied in different contexts (e.g., the Internet), the scope of IAM in this dissertation is enterprise identity and access management.

IAM has become an important aspect of IT security infrastructure in organizations, and some consider it to be the most important solution for enabling compliance with legislative requirements (Wright, 2007). According to Microsoft Corporation (2006), further drivers of IAM adoption include cost reduction, better security, better access to information, and better agility during mergers and acquisitions. However, the practice of IAM is challenging, both organizationally and technologically (Microsoft Corporation, 2006; Wright, 2007). Identifying these

[1]The initial version of the field study (analysis of 4 interviews) was published as a conference paper:
P. Jaferian, D. Botta, K. Hawkey, and K. Beznosov. 2009. A case study of enterprise identity management system adoption in an insurance organization. In Proceedings of the Symposium on Computer Human Interaction for the Management of Information Technology (CHiMiT '09). Baltimore, MD, USA, Article 7 , 10 pages. (acceptance rate: 33%)

challenges and studying how they can be addressed are important steps toward improving IAM systems and practices in organizations.

Despite the widespread and increasing adoption of IAM solutions, there are few available empirical studies that examine the practice of IAM in organizations. In order to improve the usability of IAM systems, or change the IAM practices, more empirical studies using real-world data are needed to illuminate nuances of the issues that are already indicated by prior research, and to reveal topics for further research.

In this chapter, we present the results of a field study of IAM in organizations. We started our research with three broad questions:

1. How do organizations manage identities and accesses of employees ?

2. Why is IAM challenging, and what are the challenges in managing users' identities and accesses ?

3. How can technology or practice be improved to address the challenges ?

Answering the above research questions allows a deep understanding of the IAM and its challenges, and could guide improvement of technology or practice to address the challenges. To answer the above questions, we chose grounded theory to collect and analyze real-world data about IAM. Grounded theory is an appropriate methodology to perform an exploratory study of a socio-technical phenomena, and to provide a rich understanding of the studied phenomena using qualitative data. As the outcome of our data analysis, we provide a thick description of IAM activities, and their challenges. We further develop two models that explain the relationship between IAM activities, and challenges. We further compare our findings with prior theoretical and empirical access control literature, and discuss similarities and differences.

5.1 Methodology

To answer the research questions listed in the beginning of this chapter, we used grounded theory methodology. We conducted 19 semi-structured interviews with security practitioners involved in identity and access management in large organizations. The list of interview participants, their role, and their organization sector are shown in 5.1. The scope of the interviews was various activities related to IAM in the participants' organizations. The interviews were audio-recorded, transcribed, and analyzed.

Table 5.1: Interview Participants' Demographics

Code	Job Title	Organization	Gender
P1	Security Manager	Insurance A	Male
P2	Security Analyst	Insurance A	Male
P3, P7	Security Manager	Software A	Male
P4	Security Administrator	Software A	Female
P5	Security Administrator	Software A	Male
P6	Compliance Manager	Software A	Male
P8	Security Analyst	Insurance A	Female
P9	Consultant	Health care	Male
P10	Consultant	Financial	Female
P11	Consultant	Financial	Male
P12	VP Sales	Software B	Male
P13	Business Executive	Software C	Male
P14	IT Manager	Insurance B	Male
P15	Chief Technology Officer	Software D	Male
P16	Consultant	Consulting	Male
P17	Consultant	Consulting	Male
P18	Security Administrator	Insurance A	Male
P19	Security Analyst	Insurance A	Male

5.1.1 Recruitment

According to Charmaz (2006), grounded theory requires two types of sampling: purposive, and theoretical. We began recruitment in December 2010 with purposive sampling. Our inclusion criteria for participants was experience with deployment, or ongoing management of identity and access management systems. While our initial impression was that any security

administrators could be a potential participant, we soon realized that few administrators have experience in this area, and finding target participants would be extremely challenging. We used multiple techniques to reach to potential participants:

Professional contacts: We previously had experience of recruiting security professionals as a part of the HOT-Admin project (see Chapter 2 for an overview of the project). We identified some of those contacts that might fall into our inclusion criteria, and invited them for interviews. The following participants were recruited through professional contacts: P1, P2, P10, P11

Industry partner: We asked our industry partner to identify potential participants and introduce them to us. We recruited the following participants through our industry partner: P3, P4, P5, P6, P7

Local IT security communities : We participated in local IT security networking events such as Vancouver Security Special Interest Group (Infosec BC) meetings, and Vancouver Information Systems Security Association (ISSA) meetings to identify potential participants, and connect with them. We also presented the results of our studies on two occasions to promote our research, create awareness about our project, and get feedback on the findings. We recruited the following participants through local communities: P8, P9

Online: At the later stages of the project, and as a part of our theoretical sampling, we recruited participants through a survey (see Chapter 6 for survey results, and Appendix C for survey questions). We designed a survey with the focus on access review activity, and publicized the survey with the help of Forrester Research company. At the end of the survey, we invited participants to have an interview with us. We recruited the following participants through our online survey: P12, P13, P14, P15, P16, P17

We offered the participants who were recruited online a $50 Amazon gift card as an honorarium. We did not offer the rest of the participants any incentives.

We recruited participants P1 to P11 as a part of our purposive sampling. Then we turned our focus on a particular set of IAM activities, such as provisioning and access review. We recruited P12 to P17 as a part of our theoretical sampling, by focusing on participants with experience of ongoing IAM management rather than IAM deployment and configuration. We also used data from two of the previous interviews that were done by other researchers in the HOT-Admin project. Those two interviews (P18 and P19) were not performed as a part of this project (i.e., done using a different interview guide, and by researchers other than the author). But the participants in those interviews talked extensively about their experience with identity and access management.

5.1.2 Interview Process

The interviews were conducted by one or two interviewers in-person (9 interviews) or over the phone (11 interviews). Each interview was started by explaining the research project to the participants and then asking them to sign a consent form (participants P1 to P9, and P18 to P19) or verbally consent to participate in the study (participants P10 to P17). Then we started audio-recording the interview. During the in-person interviews, we collected pictures of the artifacts participants showed to us. We also gathered and analyzed the documents provided by the participants. The length of the interviews was between one and three hours.

We designed a comprehensive interview guide (Appendix B) that we used during the interviews. The interview guide gave us the list of all possible topics to cover, but in almost all of the interviews we chose to use a subset of questions based on our analysis of previous interviews, and the participant's experience and expertise. We also modified and re-phrased some of the questions based on the participant's responses to previous questions.

5.1.3 Data Analysis

We used grounded theory coding methods suggested by Charmaz (2006) for data analysis. We imported the transcripts of the interviews to a qualitative data analysis software (initially Qualrus v2.1 and then MAXQDA v11) for coding and memoing purpose. The four initial interviews were analyzed by two researchers, and the rest of the interviews were analyzed by one researcher.

Data analysis was started with an initial listening to the interview to make sense of the entire interview before getting into the analysis. Then we started the formal analysis by performing open-coding. According to Glaser and Strauss (1967), open coding uses codes emerged from the data itself, and they could be *constructed* codes, or be *in-vivo* codes (participants' own words in the interviews). We used a line-by-line coding approach recommended by Glaser (1978) in analyzing the initial set of interviews (P1 to P11, and P18-P19). We performed constant comparison suggested by Glaser and Strauss (1967) by constantly comparing the similar incidents from the same or different interviews with each other, and created categories for similar incidents. The constant comparison technique led to identifying certain emergent themes in the data, which we chose as a set of focused codes. Interviews (P12 to P17) were coded using focused coding technique, and were done on incident-to-incident basis (i.e., the codes could span over multiple paragraphs).

After the initial open coding stage, we performed axial coding suggested by Corbin and Strauss (1990). We identified the relationship between the emerged focused codes, and categorized the similar codes as sub-categories of more abstract codes.

Towards the end of the analysis, we performed selective coding. We chose one category as the core category around which the theory is constructed. We chose *access provisioning* as our core category for multiple reasons. First, we saw that the category appeared frequently in all of the interviews, and it was one of the main concerns for our participants. Second, the

category was very well connected to other categories. Third, we saw that the ultimate goal of IAM is achieved through the access provisioning activity, and other observed activities play a supporting role in the entire IAM big picture.

We used two types of memos extensively during the analysis. First, during the coding practice, we wrote our interpretation of the coded excerpt in the qualitative analysis software. Second, during the initial listening (for the last six interviews), we wrote memos while listening to the interview using an audio bookmarking mobile application.

We stopped data collection and analysis, as we reached theoretical saturation. While analyzing the last six interviews, we observed that the identified categories were saturated, and the new data fitted in those categories.

5.2 Results

In this section, we present the findings from interview data. We first define identity and access management to provide a foundation for the rest of the results. Then we organize the findings according to the following themes emerged from the data: (1) motives for IAM, (2) stakeholder diversity, (3) IAM life-cycle and its sub-themes, (4) audit in IAM, (5) transition from manual to automatic provisioning, (6) coping with manual provisioning challenges. Each of the emerged themes provide an answer to one or multiple research questions presented in the beginning of this chapter. Particularly, we list and discuss the challenges related to each theme.

5.2.1 Definition of Identity and Access Management (IAM)

Our participants provided a consistent definition for identity and access management. They usually divided IAM into three parts: (1) Identity management, which involves creation of an identifier through an approval process, and: *"making sure the identifier is created for the right*

person, sponsored correctly" (— P1). Furthermore, it involves updating that identifier as the user changes job in the company, and removing the identifier as the user leaves the company. (2) Access management, which *"is building on the top of identity management and basically granting access to necessary systems and limiting the access to only what that person needs and not giving anything more."* (— P11). Furthermore, participants mentioned that IAM should be auditable, and allow answering the question of *who has access to what ?*: *"[IAM] provides full auditing of any kind of events associated with this identity lifecycle."* (— P10).

Participants explained that the IAM is an enterprise-wide system: *"I define it as an enterprise wide system. It could cater to both internal or external users"* (— P8). Most of our participants preferred to exclude customers from the external users of IAM. When we asked (P1) if they include customers in IAM he replied:

> "We keep them completely separate, and the reason is that we don't want external identifiers and internal identifiers to mix. and we try to keep a very clear separation between externally facing systems and the access to them and the internally facing systems and the access to them."
>
> — P1

A common theme among participants was the three motives for identity and access management. Participants considered cost reduction, efficiency, and security as the main motives:

> "Broadly theres three motives. But what somebody will ask – whats their priority depends on where they are at. Big one is cost. They want to squeeze cost out of their help desk, out of their security admin team. Another one is customer service or SLA, right. They want to turn the requests around more quickly. They want the request you want to be easier. They want approvals to happen faster. They want fulfillment to be, you know, same day, same hour, not next week. And the last one and this is the one that people market the most but its not necessarily the most prevalent is security, which is sometimes labelled regulatory compliance or risk management or controls but they are just synonyms for security. And thats usually driven by an audit failure or regulatory pressure. People dont usually care about security unless there is some external pressure to do so. And when there is external pressure they care a lot."
>
> — P15

Next, we elaborate the definitions of "identity", "access privilege", and "enterprise context", which are the basis for understanding IAM.

Identity

Our participants defined identity as an *identifier*. An identifier is created based on different attributes of a person, such as name, employee number, location, and the relationship with the company (i.e., internal or external). For example, (P1) described their policy for identity creation: *"identity is basically a user ID and a user ID is based upon the employee number plus their initials."* Participants also used the terms *identity* and *user* interchangeably.

While identity is usually associated with a person, some of our participants provided examples of identities associated with an organization, or a group of people. For example, (P10) described that the external users of their IT system are not specific persons: *"We have financial organizations represented as users. These are users within the organizations who have the knowledge of this particular user account and act on behalf of this organization."* (— P10). When we asked (P10) why they do not create identities for individual users in the external or-

ganization, she described that it is for the simplicity of implementation, and they plan to create individual identities for the users at next phases of their IAM implementation.

Access Privileges

Access privileges *"are assignable security rights"* (P15) and they guarantee a user's access to certain IT related artifacts. The artifacts can be an application, part of an application, an action (read/write/delete) on a certain object in an application, ability of execute transactions, etc. (P12) explained that access privileges can also include access to assets: *"mobile device is a resource. A pass card is a resource. Your office is a resource that is assigned to you."*. (P15) also explained that groups and roles can be considered access privileges. During the interviews, we saw that participants used the terms *access privilege*, *entitlement*, *permission*, or *resource* interchangeably.

Enterprise Context

Almost all of our participants explained that identity and access management is particularly important in large enterprises with a complex IT infrastructure. For example, (P1, P2, P8)'s company had about 2,500 employees at the time of the study. Approximately 2,000 of them worked in the head office, and the rest in branch offices. Their processing environment included a single IBM mainframe (z/OS), more than 200 Intel-based Microsoft Windows 2003 servers, and several UNIX (AIX) servers. The personal computers that were in use at both the head office and branch offices numbered around 3,000. Participants in other organizations also dealt with such a complex IT infrastructure. For example, (P3) described: *"we have about 600 applications. So each of those counts as a system and each has its own owner."*

Participants described the organizational context as extremely dynamic, in which people change job, change location, and go through re-organizations, mergers, and acquisitions. Even during the course of our interviews, we saw that the lead of IAM changed three time (from P18 to P2

to P8) in one of our visited sites. We saw similar dynamics in other sites as well. For example, (P6) explained:

> "I can tell you I've been in this role for 2-1/2 years and I've seen five department consolidations in finance alone. And that's not the count - finance is a different department for us."
>
> — P6

5.2.2 Stakeholders

A common theme in the data was stakeholder diversity in identity and access management. Participants identified the following stakeholders:

User: Users were those for whom an identity was created, and access was provisioned. Users were usually performed actions such as requesting access, and changing (or recovering) passwords.

Manager: Managers were accountable for requesting, approving, reviewing, and revoking access from employees under their authority. For example, (P6) mentioned:

> "So the process for getting access at [Company Name] is you can request it, your manager must approve it."
>
> — P6

Application owner: Application owners were responsible for deciding who should access the applications they owned. (P2) clarified the role of application owners in the IAM activities:

> " Any business related data is managed by a single person in that area. So for example all our financial information; an executive in the finance is responsible for the usage of that data and what appropriate usage of that data should be."
>
> — P2

He also added that application owners can delegate their responsibility to another person with a technical knowledge of the application:

> "Obviously that's not a job one single person at an executive level would be able to handle, so they have the ability to delegate some of that responsibility to what we call, data stewards; and those usually are the system analysts in those areas that have a better understanding of that data, what it should be used for, what it shouldn't be used for."
>
> — P2

Security Team: For most organization, security team was responsible for setting the access policy, and overseeing other stakeholders:

> "So anything policy related actually lives right now with information security [...] Anything to do with access administration in the context of a, I guess centrally managed information system be it RACF or active directory goes through our security administration group; any access to LAN resources - files, folders on various servers throughout the organization - is actually decentralized to the various departments."
>
> — P2

5.2.3 The Basic IAM Life Cycle

While the details of IAM life cycle was different from organization to organization, we observed the following theme for IAM life-cycle (Figure 5.1):

1. An identity is created for a new employee
2. A basic set of access privileges, usually common to a large number of employees is given to the user (e.g., email, Internet, LAN) automatically
3. User is given a set of job specific access privileges manually
4. The employee changes job in the organization

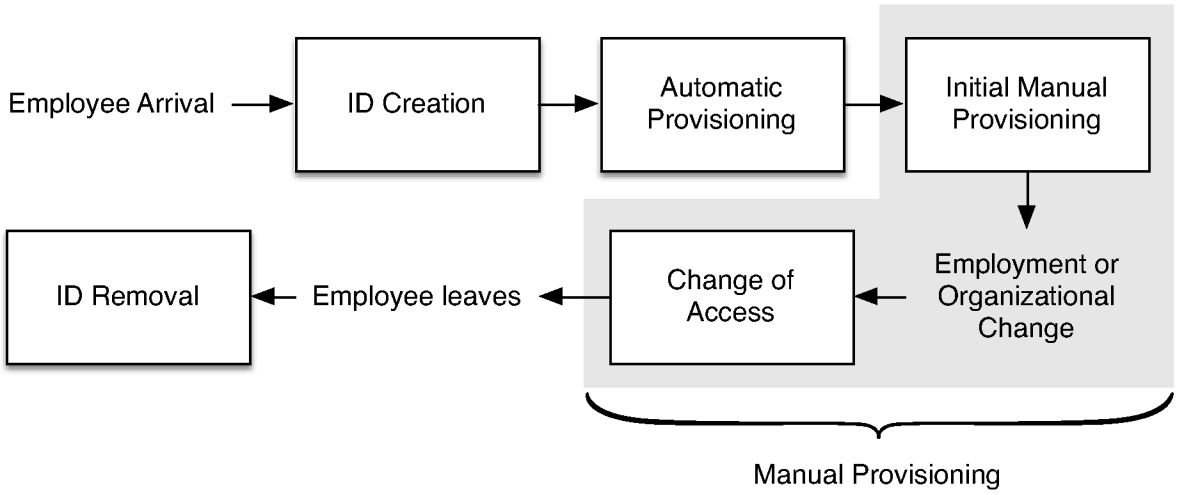


Figure 5.1: The overall IAM life-cycle

5. User's access privileges changes as a result of the job change

6. When employee leaves, the access is revoked, and the identity is discarded

This theme suggests that IAM involves four major phases: ID creation, automatic provisioning, manual provisioning, and ID deletion. ID creation and deletion fall under the identity management definition. Automatic and manual provisioning are part of the access management. In the subsequent sections, we provide a detailed description of each phase, the challenges our participants faced, and the practices they adopted to deal with the challenges.

5.2.4 ID Creation

ID creation was one of the themes emerged from the data. Participants identified ID creation as an essential but challenging IAM activity. Most of our participants described that the identity creation started from the Human Resources (HR) system. When a new employee was hired by the company, its information was populated in the HR database. The IAM system waited until the HR information was complete, and then it received a feed from the HR system containing the employee's information:

> "So human resources will get a new hire - whether it's an employee or a non-employee contract. They put all that information into the HR system. It goes through that HR process. The end result - and whatever they do, it's sort of black boxed to me. Right? Because the person may never start, they may, you know - they may not - the background checks may not come out - or it's just a black box. The end result is they will come up with a finalized start date. There is a feed from that HR system nightly that comes over and says here are the people that we have hired and they are confirmed starts. That's feed to our system"
>
> — P1

(P7) explained that it is important to use only one system as the source for creation of identities. In his company such a source was the HR system:

> "Unsuccessful shops will flounder, and the reason why they are unsuccessful is that unless you are getting your human resources from one source - in our case it's our human resources ERP, you will have people going around the process all the time."
>
> — P7

But using a single source of identity was not always possible. For example, in health-care domain, users such as clinician were not considered an employee, and therefore, their information was not kept in an HR system:

> " There is a whole class of individuals like clinicians who are regulated providers come into the hospital and they have a different process you know they are not hospital employees typically they have their own practice somewhere. But they have hospital privileges in terms of coming in and using operating rooms and everything else."
>
> — P9

ID Creation Goals

As we show in Section 5.2.1, the interview data suggests three main goals for identity creation activity: the validity of the created identities (i.e., security), efficiency, and cost reduction.

Participants explained that efficiency is particularly important for two main reasons:

Cost reduction: Those new employees who are not productive due to lack of access are considered a cost for the business. Therefore, timely access provisioning would be financially beneficial for the company:

> "So in this case we do hear feedback. One of the costs that we are trying to optimize is we might have consultants who start right away. We don't want to pay them to sit around and wait for them to be provisioned."
>
> — P7

Employee retention: Timely provisioning also shows employees that the company values their work, and wants them to be productive as soon as possible:

> "How many times do you show up the first day of work they don't have a phone for you, they don't have a computer for you; they don't have an identity for you. What are you supposed to do? Are you supposed to sit around, be the new guy, gill in some forms with pen and paper and take a coffee mug and what else? Take your pads and your office supplies and wait for everything else to happen. That's just ridiculous and embarrassing - it doesn't say hey, I'm a valued employee, I'm expected to achieve results here."
>
> — P7

ID Creation Challenges

Using an HR system removed the burden of identity creation from the security team, and provided a single source of authority for employee data. But our interviews suggest that using HR introduces a set of challenges:

Incomplete Meta-data: The data from the HR system was not specifically collected to create identities, and provide those identities with access. This led to lack of required meta-data for making access related decisions:

> "At the same while we are sending out SAP and other technologies that are role based, we understood that the HR systems could not have all the meta data necessary to make automated decisions about who should be an extension in what roles."
>
> — P7

Data sensitivity: HR database contained sensitive employee information that should not be exposed to other systems (e.g., social security number, salary, benefits). For example, (P3) described HR data as *"purely gold master"*, and explained that they built a system to extract non-sensitive data from HR for IAM purpose:

> "It's an extract from the HR data base which is all the non-sensitized employee data. Anything that is not really sensitive; like the employee name, their manager maybe; their phone number. But nothing about the person themselves - no social security number, no home address, no salary."
>
> — P3

Efficiency: Participants highlighted that the employee information was entered in the HR system after it was fully collected, all the legal requirements (such as background checks) were satisfied, and all the necessary paper-work was completed. This process was time consuming, and usually went beyond the employee's first day at work. Consequently, the IAM system did not receive employee information in a timely fashion, and the employee could not be provisioned until his HR data is complete:

> "we move forward and there's so much shift now in decentralized work force and not everybody shows up day one at this nice tidy little office where everything is done together, right? It's – we've got an oil and gas company where they are talking about people in the field that they never even meet ! They never even meet these people ! So imagine if you could accomplish this through this process in that regard because these people are mailing papers in or they are forced to come somewhere just to be in person to accomplish some of this set-up."
>
> —P12

(P12) then explained one of their customers started identity creation from an applicant

tracking and recruiting software, and the applicant itself provided the initial identity data. (P12) further clarified that the process of ID creation should happen before HR data entry:

> "you step back and ask the question, in an organization, where does your identity get created? And most identity implementations today would say, oh, it's after – after you're created in HR. But that's a gross fallacy because how did all your data get there? The business process happens way in advance of that."
>
> — P12

(P2) reported a similar issue in his organization. The unionized environment of the company led to delays in processing employee information in the HR system, and eventually identity creation:

> "we're a unionized environment, when they actually get assigned an employee number is a critical time because it feeds into their union seniority. So it ties into when the job's actually accepted by the individual and there's some rules and regulations around that and it sometimes actually delay process for that person. So they can actually be here and have accepted the job, but they haven't actually been processed per-se by HR"
>
> — P2

External employees: (P1) described that the company's workforce was not only limited to internal employees, but also people from external organizations who need access to company's resources. When the HR system was used for identity creation, the external employee's information was entered in the HR system. But those external employees were not paid by the company, and were not belonged to the company's HR system:

> "they're going to have to use our HR system, at least for registering some superficial information about their employees. so we don't want to know about the pay scale and all that kind of stuff but they'll have to be using our HR system because our HR system provides the basic identifier so they're going to have to go through our HR system to have that identifier generated. [...] they don't follow our HR policies and they don't follow our security policies. it's a weird relationship."
>
> — P1

This issue became more complicated when an identity should be created for a group of people that represent a whole organization (as we discussed in Section 5.2.1). In this case, an entry should be created in the HR system that does not represent a real person.

Data quality: The HR data directly impacted how the identities were created, and what access privileges were assigned to the identities. Therefore, the quality of HR data was critical for the success of the IAM. But the HR employees did not possess computer security knowledge, and they were not aware of the consequences of their errors in data entry. For example, (P7) talked about *"a real pivotal step in maturity"*, as they generated awareness in their HR department about the importance of HR data quality:

> "Now they understand the ramifications of setting certain key pieces of data into the human resources system is tied directly to types of access people have in the IT systems. So that, building that level of syncopation between us and that organization."
>
> — P7

5.2.5 Automatic Provisioning

After the identity was created for the user, a set of access privileges should be linked to the identity. Our participants named this activity provisioning. To create the association, *accounts* are created on the end-points (applications). One participant used the key ring metaphor to describe the relationship between identity and accounts: *"[Identity would be] sort of like the*

key ring and all of the little keys upon it would be the accounts." (P4). After account creation, the fine-grained permissions could be added to the account. Therefore, access provisioning is the process of associating a user's identity with accounts, and accounts with permissions. A major theme in the data was that provisioning can be done *automatically* or *manually*.

We define automatic provisioning as the automated provisioning of access privileges based on a user's attributes. When we asked participants what constitute user attributes, (P4) described:

> "It's very simple, it's based on the type of employee they are. Their are regions and in a couple of cases there's more to it, it might be based on a department code, or a title. Right now it's really pretty simple."
>
> — P4

We observed that the type of access users got as a result of automatic provisioning was usually high-level, and common to many employees:

> "But we have an idea of a primary type of account so the whole infrastructure of [company name] - you have an Active Directory domain and it is one domain so there is one account. That gives you what they call basic access. That's access to the network, access to remote in to the network, remote access and access to the intranet. So essentially you could log in to your workstation on day one that you are hired and you will have this basic level of access. Internet - remote access - VPN, you get an exchange account mailbox so you could transfer mail. Pretty much, that's about it."
>
> — P1

To achieve automatic provisioning, two types of mechanisms were used: *identity policies* that defined how users were assigned to roles, and *roles* that determined the access of users who were assigned to them.

Identity Policies

Some organizations used the notion of *identity policies* to describe the rules using which automatic provisioning was done. Identity policies were usually in the form of conditional statements. If a user's attributes matches the antecedent of the conditional statement, the user will be assigned to one or a set of roles:

> "So when a user is created by [the IAM software], [the IAM software] has certain custom logic built in - well built in - I put it there. But it will look as it creates the user at the attributes and if it matches a certain policy such as [employee type] A, C, or S and region equals North America, then give them this role."
>
> — P4

One of the participants (P4) showed us the set of identity policies used in their company. The policies were designed to assign users to a subset of 20 to 25 roles, based on two types of attributes: the employee type (e.g., full-time, part-time, contractor) and the employee work location (e.g., North America, Europe). While they used very few attributes to assign roles, their identity policies were quite complex, and hard to understand. For example, even the participant, who wrote the identity policy herself, could not interpret the policy fully, as it contained cryptic codes. We asked the participant to describe the meaning of different attribute values in the policy description, but she could only recall one out of five values in the policy. Furthermore, the policy contained two different attribute values for the same purpose. For example, when we asked the participant what the employee status "AC" means, she said: *"Active - or A - A also stands for Active."* (— P4) Then we asked about the difference between "A" and "AC", she replied: *"Nothing it's just that we used to use AC and now we use A but I've left it in there just on the odd chance that there's somebody still around with an AC."* (— P4)

Roles

We asked our participants to define role-based access control. The high level definition was common between participants. For example, (P4) defined role-based access control as:

> "Based upon certain user attributes and this is in general for [IAM software], you would assign users roles and then these roles are used to grant them some kind of access."
>
> — P4

According to this definition, users are assigned to roles based on a set of *user attributes*. In other words, *"You do not pick a role. A role picks you."* (P12). Consequently, these roles were used to assign one or a set of *access privileges* to users. We usually observed that roles have a conceptual meaning, i.e., they represent a concept outside of the IAM system. This conceptual meaning usually matched certain user attributes such as a location, a job, or an application. For example, (P8) described that one role was used to create accounts on one large application (i.e., abstracted an application concept). She also added that they use roles that abstract the concept of departments in the company:

> "we have about 200 roles and that consists of base roles, and one application role, well our one application is like a huge application that I would say most of our users in this organization will touch so it covers a lot of user base and also three or four division roles or department roles."
>
> — P8

(P12) further clarified that roles can have various conceptual meanings, and they can map to a user type, a job function, or a location:

> "We use different role types as well. So for example, we use something called a 'user type role' which maps to more of the entity type that I am. So for example, I am a full time employee. I'm a part time employee. I'm a contractor. I'm a consultant. A significant amount of access will map to those correlated attributes. Okay? And then we have what we call 'BU roles' which align to your job function. So whether it be – again a cost center job code, whatever the case may be, it aligns to some job function. So if I'm an accounts payable clerk with job code 1234567 or 8 I get this access. And then we have what we call 'IT roles' that are more siloed based on some of their attributes like a location. Everybody at the [City Name] location gets in the [City Name] distribution group. Everybody – you know, whatever those attributes may be."
>
> — P12

(P6) talked about a set of *simple roles* that are not necessarily have any conceptual meaning, but they group a set of access privileges. He called these simple roles *ingredients* of building access for users based on their job:

> "So what we do is we have an ingredient list of roles, what I mentioned before - let's say about a 115 - we call these simple roles. These are the task based roles. They are broken - some of them have as little as one T-code. Some have as much as let's say 15 volume."
>
> — P6

In contrast to simple roles, (P12) talked about the concept of *enterprise roles*, which are a collection of simple roles, and assets:

> "But we also use the term enterprise role or profile which is a broader aggregation of access and assets that users with common requirements share. So, for example, I am a branch teller and I may get the teller role in application A and the teller admin role in application B and a pass card and a branch key and a whole set of things that are granted to me because of my enterprise role."
>
> — P12

As a summary, we saw that roles can have different meanings. They can be as narrow as a

single access privilege, or as large as multiple roles and assets. We saw that users were mainly assigned to roles automatically using identity policies, but we also saw that administrators can manually assign users to roles. In the context of this thesis, we define roles as a group of access privileges that are used for automatic provisioning. If a role is used for manual provisioning, then we call it an access privilege.

5.2.6 Manual Provisioning

After the initial automatic provisioning, users will be provisioned with job specific access using the *manual provisioning* process. While the automatic provisioning assigned users to roles without human involvement, manual provisioning was a labor intensive activity that involved multiple stakeholders:

> "Anything that's non-basic access, which is anything else - that's what we are focusing on now; because that is terribly manual today - still. It involves somebody figuring out what they need access to, creating a ticket in Service Desk, and waiting for someone else to figure out which systems they are referring to and then going in and provisioning that access. And there is some document ultimately about what - who is the owner of the system, who is the access approver of the system. Here is the method that you give access to."
>
> — P3

Figure 5.2 shows the most common interaction pattern we saw between different stakeholders involved in the manual provisioning activity. The activity was initiated by a manager who identified a set of systems and assets that his new employee needed. In some cases, the employee itself requested access, and the request was reviewed and approved by the manager. We saw that in different organizations, different communication channels were used to send access requests, such as an electronic form, email, a ticketing system, or an IAM system. The request was usually made to either the service desk, or the security group. After receiving the request, a member of the security group (or service desk) determined the desired systems and

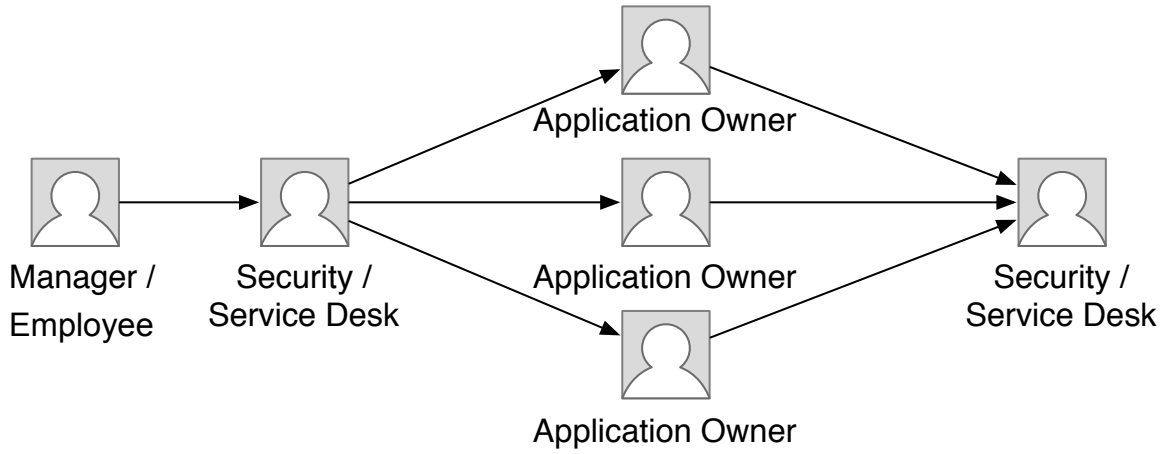


Figure 5.2: The interactions between stakeholders involved in the manual provisioning activity

deployed the request to the application owners of those systems. The data might be distributed, and thereby owned by several application owners. In this case, the security administrator (or service desk) made multiple requests. After receiving the request, the application owner (or a delegate with technical knowledge of the application) decided whether the requested access is appropriate or not, and notified the security administrator (or service desk) about the decision. The security administrator (or service desk) implemented the request upon application owner's approval. If the security administrator (or service desk) did not receive a response, he performed a follow-up cycle, to handle non-response or lag from application owners.

When we discussed the details of manual provisioning with participants, we identified that the activity was *inefficient*, and *error prone*, and it *involved communication and collaboration complexity*. We found the following reasons for the observed challenges:

Managers' lack of knowledge of what to request: As we discussed before, managers used generalized communication channels to specify their request, but they usually did not know what to ask. In some companies, the security group provided a detailed multi-page form for managers to identify resources and request access. The managers were overwhelmed by the amount of information that they were expected to fill in (P18). This

led to *inefficiencies*, and *errors* in the process.

Mirroring: Frequently, managers would ask the security group to make a new employee's access the same as some another current employee's access:

> "A manager who hired a new employee who knew that you had the access that you needed to do the job for him or her would say, 'Oh, make this new employee's access just like yours.' And so then an employee would then inherit very high levels of privileges and access based on the success of a previous employee in terms of doing that job."
>
> — P1

One of our participants (P6) called this practice *mirroring*, and explained it generally leads to *errors*. Some of our participants such as (P1) and (P6) disagreed with mirroring practice, and even (P1) suggested that they disallowed this kind of generalized request as it could provide more access than required for the employee. Conversely, (P14) explained that they use this practice in their organization frequently. But (P14) also suggested that they do audits regularly, and make sure the mirrored user does not have unnecessary access privileges.

In cases where mirroring was disallowed, those managers who did not know what to ask for then tried to work around the security requirement by asking for the list of accesses possessed by a current employee, and requesting access for every item on that list (P1).

Lack of common terminology: In some cases, there was no common terminology throughout the organizations for requesting access, resulting in *communication and collaboration complexity*. When an access request was received, it was difficult for the security group (or service desk) to identify which systems in which divisions were requested:

> "[...] but it's terminology we don't know. We say what area is this in, is this in the assessment part, is this prevention, is this the claims? - sometimes the user has no idea, they just say I don't know I just have to get access to it. So we end up going to the security coordinators for all those divisions saying are you familiar with this app? Or whatever they are talking about. Oh yeah, this is actually this. Aha. Then we know, and then we can tell if the request has come to us in a form or an e-mail and it's approved, we can set up the access or send it to the area responsible for setting up the access."
>
> — P18

Also when managers were requesting access using the access profile of a similar user, they did not necessarily understand the esoteric access rules and system names – these had to be translated into language that the managers could easily understand:

> "[...] and we get this huge profile - here's all the access the user has. We then have to translate that into more of an English format for the individual, saying this means this, this means this, and this means this. And then, they have to put that into a form of what they want."
>
> — P18

Communication with application owners: When the security team (or service desk) processed the request and deployed it to various application owners, it was often difficult to identify the correct owner. Also, as mentioned, some owners would not respond in a timely fashion, or even not respond at all, making provisioning impossible. The security administrator could end up implementing the requested access in a piecemeal fashion, thereby decreasing employee productivity as the employee would then have to wait for access. This led to *communication and collaboration complexity*, and *inefficiency*:

> "[...] and so the security administrators would then send notes, e-mail, to the data guardians saying, 'Are you OK if we grant Bob or Jill access to the system?' And then there would be that sort of follow-up cycle where the data guardian would ignore them or not respond or say I'm not the data guardian or that kind of crap. So there would often be a delay in terms of getting approval... so systems access to the employee would then build over time as individual data guardians would respond"
>
> — P1

Lack of context in requests: The communication between stakeholders happened using generalized request management tools, such as email or a ticketing system. Therefore, it was impossible to enforce rules such as separation of duties during request process, which eventually led to *inefficiencies* and *errors*:

> "A ticketing system like, you know, BMC Remedy or HPO Service Desk, they'll do request management but they are not compliance or identity tools. They are just dumb request forms for the most part where it doesn't know what you already have access to or what any of the compliance policies are"
>
> — P12

Additionally, generalized request management tools were providing too many choices for users without really thinking about what the user can theoretically have access to, further contributing to the *inefficiency*, and *errors* in the process:

> "Traditionally [companies] just put together a big pick list of everything under the sun that you can ask for and then hope that it's going through some approval process and then relaying on some periodic attestation to catch things that are inappropriate."
>
> — P12

Understanding Access Privileges

Our participants adopted a set of practices to alleviate some of the aforementioned challenges. (P3) suggested that they documented the process of requesting access to each application, to

create a common terminology, and make the identification of application owners easier:

> "So our access procedures state that every application that has any level of criticality is supposed to have a published knowledge-base document in our service desk that defines what the application is, who owns it, who is the technical owner of the words — a business person may own it but someone in a tech area, you know IT has to actually do stuff for the business. There is a tech owner - right — who is the approver or reviewer in some cases they call them BAR, business access reviewer. Sometimes the BAR is the approver, sometimes there is a separate approver like the BAR's VP or something. And then it contains something like what service desk group you should open a ticket in and other types of identification information."
>
> — P3

Similarly, (P12) talked about a resource catalog that includes all the policies related to the access privileges of an application:

> "Whether they be SoD policies that say you can't have A if you have B or what we call 'restricted access' policies that say you can't have entitlement X if you are not cost center Y or division X or whatever the rule is, the ability to define that rule it lives with the entitlement in the resource catalog. So as the resource owner we have the resource catalog of all requestable resources and entitlements. And the rules about them belong with those in the resource catalog."
>
> — P3

(P7) talked about their plans to eliminate the generalized access request forms, and to provide users with an *"access catalog"*. Such a catalog will show all the requestable access privileges for an employee, and then automatically triggers the approval process, when the user requests an access privilege:

> "[The new technology we are planning to deploy, will be] allowing people to say hey, I need special limited access for twitter - I am in marketing and I need to twit, and I need to consume twits. I am going to go ahead - here it is in the catalog - here's where I request it - the request goes in, the work flow now says oh, for this type of access I need SUP this person reports to approve it. I need to check their - I need someone in operations to check their educational requirements to make sure they have done their security training so they know how to use these tools effectively and safely. Then we can go ahead and wrap this because people actually grant entitlements. Or, route it to the system that automatically grants the entitlement once those conditions are satisfied. What we are doing here is we are combining human work flow with automated work flow."
>
> — P7

Similarly, (P12) suggested that a set of proactive control should be put in place to prevent users from initiating requests that violate existing policies or rules:

> "we also applied what we call proactive controls. So our whole kind of philosophy is that before I even request something we're going to apply compliance policies like SoD or other restrictive policies like if I'm not in Division X, I can't even request access to System Y. So we take all of those policy controls and put them right in part of the request process"
>
> — P12

5.2.7 Change Process

As we discussed in Section 5.2.1, the enterprise context is extremely dynamic. When users change responsibilities, jobs, or departments, their access needs to change as well. A common theme was that the change process involved both automatic and manual provisioning. When an employee changed job, its user id was assigned to appropriate roles through automatic provisioning, and revoked from the roles related to the previous job. Additionally, the employee's

new manager requested additional access through manual provisioning. Therefore, the change process faced similar challenges to manual provisioning. Despite this similarity, the removal of access from the last job made change more difficult than the initial provisioning:

> "So the critical difference and you know we always talk about change is the hardest identity process because on-boarding, you're just adding everything you can think of and at off-boarding you are taking away everything you can figure out. Change is the hardest because you've not only got to figure out delta – so in any given job change because of the enterprise roles, there's lots I'm probably keeping. Like I'm probably not going to whack my LAN ID because I'm transferring departments but here is some access that will change. So there's the delta but there's also the timing of this change because job changes take effect on effective dates and there's overlaps required with pre-existing access so change is by far the hardest business process."
>
> — P12

(P1) defined *privilege accumulation* as keeping the access privileges from the past job during change process. In P1's company, people frequently accumulated access privileges as they changed jobs:

> "Historically here if you were an individual who started at [The Company Name] in 1930 (I'm exaggerating) by the time you retired and had 40 years you would have access to every single system that you had ever used in your entire lifetime with [The Company Name]."
>
> — P1

Participants explained why privilege accumulation is a hard problem to address. First, users usually went through the manual provisioning during their previous job. Revoking those privileges later was a task similar to manual provisioning, and shared similar challenges. (P2) discussed that the automated de-provisioning during change process is their ultimate goal, as they currently use human process, and it is error-prone:

> "So having a system in place that provided that compliance and that checkpoint that when somebody moved jobs it would remove the old entitlements and add the new ones with a guarantee. Any time there's human involvement there's ... it's prone to error."
>
> — P2

Second, there was a lack of accountability from the managers to ask for removal of access from the employees: *"if a manager fails to advise sec admin to cancel access, there were no consequences to the manager."* (— P1)

Third, (P19) talked about the sense of entitlement that some employees had about their past access privileges. He explained that lack of security awareness led to employees demanding to keep access from their previous jobs:

> "So some people view that as an infringement upon their union rights. You can't take things away from me. I have seniority. You can add to me, but you cannot take away from me. They don't understand like the security concept of you're doing this job now, you're not doing this job, you don't need that access anymore"
>
> — P19

To cope with these challenges, some of the companies had an access governance team in place that monitored the changes, and made sure users have the correct access:

> "Now what happens is that we have a report that runs every single day and it tells me people transport or change. So if she - what happens is the trigger would be one of two things: she gets a promotion. She went from warehouse manager to public relations manager. She will request something. I need a public relations manager role. My team goes automatically - why? That's not what you are. You are warehouse. No, I got a promotion, I'm this. Okay, we'll give you these three but you are losing those three."
>
> — P6

While using automated provisioning could address the challenges with errors, and accountability, it made handling *exceptional cases* difficult. For example, (P6) explained that in certain cases, employees requested a transitioning period, and asked for keeping access from their

previous job for a period of time, making automated change process ineffective:

> "We send a request to the manager that says [Employee Name] has changed from position A to position B. They are requesting position B roles. We are going to remove his position A roles. Do you agree with that? And if the manager says no, he is training this person as replacement for three months. Okay fine, we will allow them to have it; we send a notice to security and security will send - will set those three roles to expire in three months."
>
> — P6

5.2.8 Identity Removal

Another identified theme in our data was ID removal. ID removal or leaver process happened when an employee left the organization:

> "As soon as your employment gets inactive in your company's IT system, your identity should be automatically transferred in a locked state. All your user accounts and entitlement should be locked that you would not be able to use it."
>
> — P17

Similar to identity creation, identity removal was usually triggered by the HR system. The IAM system received a notification from HR about the termination of the employee, and it suspended all of the user's accounts:

> "On the flip side, if somebody leaves - is terminated - a reverse process happens. HR enters the term and they send us the action saying this is a confirmed term - we back process and it goes and it will then remove those accounts. Well it removes the active directory account and it will disable the actual mailbox."
>
> — P3

(P1) described that they used to do the identity removal manually, and based on notifications from managers. But they found this process to be ineffective, as managers were not accountable

for reporting the terminated employees:

> "It used to be that managers are responsible for filling out a form and advising sec-admin that employees were terminated. and this was almost universally never done and the reason it was never done was that although it was the manager's responsibility, back into that operational thing again right? which says, 'I know that that's the rules but I'm not going to do it because I don't need to.' there's no accountability, if a manager fails to advise sec admin to cancel access, there were no consequences to the manager. so we were finding that many employees would depart [The Company Name] and yet their IDs would still exist. and the access those IDs still was active."
>
> — P1

(P1) explained that they changed the identity removal procedure by getting notifications from the HR system, and removing identities automatically. But he also found the automated method ineffective in exceptional cases:

> "[Status changes from HR are] not universally 100% effective because what can happen is employees can not quite depart. Meaning that you can go on severance, which means that you have a year, 2 years worth of severance and you haven't departed. [...] You aren't in the building. It fails sometimes. Or you go on a leave of absence and then, again, you haven't quit departed. So you could have somebody who, say, goes on a leave of absence for a year, two years, or whatever. Sick leave, LTD; that could go one forever and then you never know officially whether that employee's still an employee of [the name of the organization]. So status changes weren't 100% reliable"
>
> —P1

To address the issue with those exceptional cases, (P1) explained that they

> "periodically run one query job and say, 'identify the IDs that have not been logged into for the last six months, some period of time.' and then we follow up saying, 'what's going on?"'
>
> —P1

As we described above, the ID removal faced challenges similar to ID creation phase, such as efficiency, data quality, etc. But the lack of efficiency was particularly a critical issue for ID removal. (P17) explained that the lack of timely ID creation causes inefficiencies in productivity, but lack of timely ID removal has security consequences:

> "From the efficiency perspective it's the joiner process which is critical. From the compliance or from the audit stance, from the control standpoint it's typically the leaver process which is critical."
>
> — P17

(P7) further explained that terminating an employee in the HR system might not seem to be urgent, but revoking access from the terminated employees should be done as soon as possible. This led to some companies do ID removal manually:

> "It's a 24/7 business and right now there is emergency - if we have a situation where an employee is terminated for whatever reason and we want to revoke their access right away, there are some manual processes that we use today to make sure we get that stuff out of the system in a timely fashion. And then the automate system goes in and does the control level function of removing their identities and unhooking their access and such."
>
> — P7

5.2.9 Audit and Accountability

In the preceding sections, we show the basic IAM process and elaborated its activities. Furthermore, we show that the process involves human, and it is prone to human errors. As a common theme, IAM stakeholders perform audit activities on the access policies to detect errors. We saw three types of audit during our interviews:

Policy Review: In Section 5.2.5, we talked about the use of identity policies, and roles in automatic provisioning. The policies and roles were usually created by the security team in collaboration with the business. But those policies could become outdated, or contain

errors. In some cases, companies reviewed the policies and roles, and checked their validity:

> "So again in our world we look at it and say – we use this terminology called a membership criteria that governs the pre-approved access that you get. So again if I'm in cost centre A, B or C and my job code is 1, 2 or 3, I get the teller enterprise role. And the teller enterprise role gives me a LAN ID, an SAP account, a mainframe account and a pass card, right? So I'm just as interested from a compliance and auditing perspective to have an auditor sign off on the membership criteria by which that pre-approved access is granted. So that's another piece of the certification puzzle that's, I think, quite important."
>
> — P12

Since the scale of roles and identity policies was not large, we did not see participants' emphasis on any challenges in policy review.

Access Review: To check the validity of the access given through the manual provisioning, companies reviewed the access privileges of employees. They called this activity *access review*, *access certification*, *access recertification*, or *access certification and attestation*. In the context of this thesis, we call the activity *access review*. The access review was usually done by a manager reviewing employees under his authority. We found that access review involves challenges such as the large scale of review, lack of technical knowledge by managers, high frequency of reviews, human errors, and difficulty in handling exceptions in users' access. We discuss the details of access review in Chapter 6.

Access Log Review: One of our participants, who worked in health care domain, explained that the access policy in health care is not as strict as enterprise domain. They allowed users to have access to a wide range of resources, but they collected and reviewed extensive audit logs of who *accessed* the data:

> "So the whole access model in health care tends to be, you let people do what they need to do to get the job done. They may not be like why did you look at so and so's record it's not because I'm not trying to predict whether or not you have a relationship and should have access to that . Because that is a really tough problem. I let you do it but I audit a crap out of the system so if somebody complains or someone reports that I saw somebody accessed something and I don't think it is appropriate then you've got a really robust audit records, and you can call them to task and you involve professional bodies and all the sudden you know you're disciplined or suspended or whatever."
>
> — P9

Such a model was later confirmed by (P15) and (P16) who had experience in implementing IAM in health care domain. But in an enterprise context, keeping audit logs would be a performance overhead, and was not done regularly in organizations:

> "the reality is that the audit log creation can impact system performance so therefore audit logs are not reliably created. Application developers do not capture the requirement from a security point of view to create audit logs. So not all systems create audit logs."
>
> — P1

5.3 From Manual to Automatic Provisioning

Among all of the activities we discussed before, we identified the manual provisioning as the most challenging. It was a manual, labour intensive process, required communication and collaboration between multiple stakeholders, was prone to human errors, and it required intensive audits.

As a common theme, almost all of the participants expressed their desire to automate the provisioning process and eliminate manual provisioning challenges. Automated provisioning was one of the main drivers for those companies adopting an identity and access management sys-

tem, as it led to efficiency and compliance. For example, (P2) who was managing the adoption of an IAM system explained that they are planning to automate the provisioning by adopting a role-based access control approach:

> "And so part of the identity management project is to go through a role-based access control and role mining exercise to define roles for the different business areas where you've got a defined, concrete, set of privileges somebody needs to do that specific job. So at that point when somebody moves into a new job you know exactly which access they require and what access their old job had and could be removed and that's all done through an automated system rather than a person actually going in there having to do all that stuff."
>
> — P2

But surprisingly, none of our participants had success with automating the entire provisioning process. For example, a year after we talked to (P2), we visited (P8) who replaced (P2) in managing IAM in the company. She explained to us that they still have not used role-based access control throughout their organization.

This led us to choose *access provisioning* as the core category during selective coding process. It was a high level category (theme) that included three sub-categories (sub-themes): *automatic provisioning*, *manual provisioning*, and *transition from manual to automatic provisioning*. We built our theory around the access provisioning concept, and identified the relationship between other categories and our core category. This helped us to answer the following questions:

- Why is provisioning challenging ?
- How do companies deal with the challenges ?
- How can organizations automate the provisioning ?
- How do other IAM activities contribute to provisioning ?

Further data collection and analysis revealed a set of challenges in both automatic and manual provisioning. In the next sections we discuss these challenges.

5.3.1 Coping with Manual Provisioning Challenges

In Section 5.2.6, we identified three main challenges in manual provisioning: (1) inefficiencies, (2) errors, (3) and communication and collaboration complexity. A common theme was to cope with the manual provisioning challenges in three different ways:

- We demonstrated in Section 5.2.6 that companies try to understand, and document access privileges, and improve the way the manual provisioning process operates. This practice positively impacted all of the three challenges in access review, but did not eliminate any of them.

- We shown in Section 5.2.9 that companies perform access reviews to prevent, identify, and fix errors during the manual provisioning. On the other hand, access review did not address inefficiencies or communication and collaboration complexity.

- Companies can transition from manual to automatic provisioning to eliminate all of the mentioned challenges. This approach seems to be ideal, but transitioning required a set of activities including integration of IAM with end-points, role-mining, and role-engineering. These activities acted as barriers to the transition. Furthermore, when companies moved to automatic provisioning, a set of challenges forced them to abandon the approach and go back to the manual provisioning. In the subsequent sections, we will explain these issues in detail.

5.3.2 Integration With and Managing the End-points

Transitioning from manual to automatic provisioning required an integration between an IAM system, and the end-points so that the IAM can manage the assignment of users and access privileges. But the integration was proved to be difficult due to a mix of technological and organizational challenges:

Unmanageable access privileges: In many cases, organizations owned applications that did not expose external APIs for managing their internal access privileges. The problem was not only a technological limitation, but in some cases, was policy related. The organization's policy did not allow managing certain critical applications centrally, as the application should be managed by its own dedicated administrators:

> "But again you know we find new applications, it doesn't mean that I can get to the entitlement system. It still could be black boxed. For security reasons - so a treasury application - maybe they don't want people understanding how the internal security works or the entitlements work. Or, for example the systems that are hosted - for us it's just a web interface to a bank, treasury or whatever. There's no way to get in there and they are not going to let us in."
>
> — P3

Autonomy of applications: Some applications followed a set of universally standardized best practices, and therefore, the company preferred to manage them separately:

> "SAP is a little different - SAP was something that was added on top of all the existing processes and SAP comes to its own best practices and thing."
>
> — P3

Technological complexity: It was difficult to integrate some of the applications with the IAM system. After doing a cost-benefit analysis, organizations decided to manage those applications manually:

> "We have actually quite a number of applications that would allow us to actually create the permissions and set the permissions for most of our applications and we haven't had any luck with them. As a matter of fact we've taken out of the creation simply because they have actually created more of a burden to us on the applications that we have tried that would allow us to actually have sort of like the consolidated account creation and permission settings."
>
> — P14

5.3.3 Engineering and Mining Roles

To enable role based automated provisioning, a complete and correct set of roles needs to be created. The creation of the roles was one of the difficult aspects of setting up the IAM system for most of our participants. Few of the companies that we talked to considered engineering roles. For example, in (P1, P2, and P8, P18, P19) organization, the security group decided to build roles by following both a top-down and a bottom-up role engineering approach (see (Vaidya et al., 2007) for a discussion of role engineering). They started the role engineering process by developing a set of roles from existing user-permission assignments, which was called *role mining*. This was a bottom-up approach that required mining existing user-permission assignments in different access control repositories. (P1) highlighted the importance of role-mining:

> "if you don't know how to do discovery, if your tool can't do discovery you're committing the staff to 2-3 years work of heavy lifting to do discovery. So a tool that did discovery and managed roles potentially can save you years of effort."
>
> — P1

The role mining engine in their IAM system could analyze different repositories in each application and find users with similar accesses (P18). Consequently, the security group collaborated with individuals from each business area to check the differences between those similar accesses and created a single role that corresponded to a single job description (a top-down

approach). Additionally, the security group collaborated with application owners to determine which roles should be authorized to access each resource in the organization. About a year from the time we talked to P1 and P2, we spoke to the new IAM leader in the company. She described that the role engineering project is still incomplete, and they only implemented roles in few departments. She further explained that their plan is to extend role engineering process to other departments.

We further explored why role engineering is so challenging to perform. The participants provided three main challenges: business complexity, uniqueness of access, and lack of accurate data.

Business complexity: (P3) explained that they are reluctant to start a role engineering process as it should start from business, and business involves complexity, and constant changes:

> "Well it starts - it doesn't at IT - if it starts at IT it's going to fail. It starts in the business and the business is a mess - not only our business, any business is a mess, right? You have to figure out what people are doing - whether it is right or wrong - or the best way or not the best way. And then you have to manage the change. Ten years ago in [the company name] I could tell you that we used to reorganize about twice a year and that could mean your office numbers, back changes, your cost centre changes, you know all these different things changed. Your titles - titles used to change very frequently."
>
> — P3

Uniqueness of access: (P12), who had years of consulting experience with various companies, explained that the uniqueness of people's access makes it hard to generate roles automatically, and therefore, part of the provisioning should still be done manually:

> "We do identity consulting which led us down these paths so we've used the tools from VIOU, Birdstream, Eurekify, which have all of course since been acquired by CA, Oracle, Sun, whatever, and they are interesting tools in the sense that you can collect the source data and run it through some algorithms to try to detect patterns, but the catch 22 that seems to get lost on people is if people's access is even in the slightest way able to be correlated, you wouldn't have this mess. So our experience has been that those role mining tools produce fairly predictable outcomes which are useful. Like, for example, everybody gets a LAN ID. Okay. Thanks a lot ! That's really good ! I could have told you that !"
>
> — P12

Lack of accurate data: (P2) Explained that the access control data for role-mining exercise may not be accurate, as users may have more or fewer access privileges than what their job requires. Therefore, the output of role-mining should be evaluated and refined by consulting to people who understand the business:

> "[The output of the role mining might be] appropriate or not appropriate. But a lot of times it's interesting because you'll identify those because if you've got a group of, you know, 15 case managers for example and you bring them into the system it'll say, 'Ok, 12 out the 15 have 80% of access in common and these two people only have 20%.' And at that point we would go with the business analysts and go through those entitlements and say, 'oh this person has access they should not have that has been carried over from somewhere else'."
>
> — P2

We further explored what is the proper way to tackle the role engineering problem. The participants suggested that the right way to engineer roles is to do it gradually, and over-time. (P12) explained that: *"when we engage with a customer, we will embark on some role modeling but it's not the be all, end all. We don't try to bring it to even a – state of ultimate completion."* He further explained that the access management process is composed of role-based automatic provisioning, and ad-hoc manual provisioning. The company should start with a small scale

role-based provisioning, and as they continue to perform manual provisioning, their understanding of the meaning of access privileges, policies, and exceptions will grow. Then using the manual provisioning data, roles can be engineered for some of the areas in the company.

5.3.4 Challenges in Automatic Provisioning

In previous sections, we showed that transitioning from manual to automatic provisioning was challenging. Even if those challenges were overcame, performing and maintaining automated provisioning was difficult as well.

In Section 5.2.5, we explained that the roles are used to automatically provision users. (P15) explained that roles give organizations a leverage in provisioning:

> Defining a role lets you assign a bunch of entitlements at once and then lets you give those entitlements a label that business users will understand. And it lets you in many cases automatically assign a bunch of entitlements based on identity attributes. All of that together is leverage.
>
> — P15

We found that in two situations, the challenges of using roles outweigh their leverage. First, people's access tend to be unique, and it is difficult to assign a large number of users to a role. Second, organizational change causes changes in roles and makes their maintenance difficult.

Uniqueness of Access

While we show in Section 5.3.3 that uniqueness of access makes role-mining difficult, it also negatively impacts the leverage provided by roles. Roles would be effective only if there are a large number of users with the same access requirements:

> That leverage only works if the number of users that have the exact same requirements is large. Otherwise you have this level of a direction between the user and the entitlements and you apply it to one guy or two guys, right. Useless. So RBAC is an efficient model where you have substantial populations of users with identical security access requirements.
>
> — P15

(P12) suggested it is impossible to implement role-based access control fully, as every users' access is unique, and hard to determine a priori:

> "It's not realistic to think that 100% of your access is going to be automated and pre-approved because you'll end up in a scenario where for every user you have an individual role, which completely defeats the whole purpose. We prescribe a model of, if you can even get to 60 — 70% of your access automated, pre-approved through roles that would be tremendous and phenomenal."
>
> — P12

Some of our participants elaborated why people's access is unique. (P6) explained that the job title does not necessarily determine the access of a user (e.g., a business analyst in one department gets a completely different set of accesses than a business analyst in another department):

> "If you have uniform job titles throughout a company you can do [role based access management]. You can build all the roles and you can do this assignment automatically. If you don't, then you wind up having a request - so you get a set of generic roles, you get ESS, MSS, SAP utilities - which are some basic simple things just to get in - like a network password. Then after that your job roles are done by request."
>
> — P6

Similarly, (P11) explained that roles usually define a set of functions for users to perform. In addition to that, the scope of objects that the user can apply those functions on should be limited and be different from one user to another:

> "That is correct. So defining those roles, it can get quite complex. Because, you are not just defining what tasks they are capable of doing, you are defining the scope of what they are capable of doing that on."

(P6) gave an illustrative example of why limiting the scope of the access is important. Their company designed a set of roles for provisioning users in North America. But they soon realized that those roles would not fit in Europe, Middle East, and Africa. When we asked the participant why you cannot use the same roles when you are expanding your provisioning to other parts of the world, he explained:

> "So the difference is at a very high level you can take a role, give it to a user and you could use this logic: if you could make journal entry in New York, you could make a journal entry in Tokyo and you could make one in Berlin. It's the same principle. However: How do I restrict you from making that same journal entry when you are sitting in Berlin to do it in New York's journal?"
>
> — P6

Constant Changes in the Business

Another barrier to use roles is constant changes in the business, which leads to changes in the responsibilities, departments, and applications. (P6) explained that the main concern with functional roles (roles that are mapped to job functions) is the constant changes in the responsibilities of employees, which would lead to constant changes to the structure of the roles:

> "They have something called composite roles. Which is the amalgam of these already built-in. But based on our principle where we do not have this job based level, having these becomes a problem because you start winding up having multiple versions of them. So if you have 115 roles that translates to lets say 50 functional roles. [...] Now one person and this is assigned to five people each per composite role. One of these people wants a change to this composite role. Well that change now affects five people. And if I can't do that and I say well there's only one person wants it, I have to create another composite role. [...] So when you see all those changes happening, those composite roles hurt you. Because then you have to keep generating them over and over again. And there are maintenance problems because they affect large numbers of people. "
>
> — P6

(P6) also informed us they are planning to re-evaluate the use of functional roles in the future, as the rate of change in the company may reduce, and the organization may become more stable. His motive for that was the drawbacks of simple roles, which were assigned manually:

> "Because you have a large number of people doing it. You could have 500 of [simple roles] and every day you are changing them. It's a maintenance nightmare. You have a lot of people that are changing and moving them and looking at them."
>
> — P6

5.4 An Integrated Model of IAM and its Challenges

In this section, we present the abstract model we developed using grounded theory (Figure 5.3). The model explains the relationship between different IAM activities, and the challenges we observed.

The ultimate goal of identity and access management is granting users access to necessary systems and limiting the access to only what that user needs. Manual and advance provisioning

play a pivotal role to achieve this goal. Therefore, the centerpiece of our IAM model is those two activities.

Our results indicate that companies used a combination of manual and advance provisioning. Manual provisioning involved creating accounts on the individual systems, and managing the accesses of users manually. This process involved challenges such as inefficiency, errors, and communication and collaboration complexity (Section 5.2.6 and Section 5.2.7). These challenges motivated companies to move away from manual provisioning to automatic provisioning. On the other hand, transitioning to and staying in the automatic provisioning was difficult, which led to companies chose to perform part of their provisioning manually. We observed that companies frequently cycle between the manual and automatic provisioning, and during the cycle, their knowledge of business and access privileges increases.

Besides the two centerpiece activities, there was a set of supporting activities that contributed to the provisioning.

First, identity management activity orchestrated the automatic provisioning, and partly the manual provisioning. When the identity was created for a user, automatic provisioning assigned the user to roles based on the identity attributes. Furthermore, during manual provisioning, accounts were created by security administrators (or service desk) on those end-points that were not managed automatically, and then those accounts were associated with access privileges. The success of automatic provisioning relied heavily on the accurate and efficient identity management. Therefore, an additional challenge that prevented companies from adopting automatic provisioning was ineffective identity management. Due to the importance of identity management, we focus on modeling its challenges in the next section.

Second, access review was a supporting activity for manual provisioning. We showed that the manual provisioning was error prone, and the goal of access review was to detect and eliminate those errors. In Chapter 6, we will show that access review is challenging, and those challenges

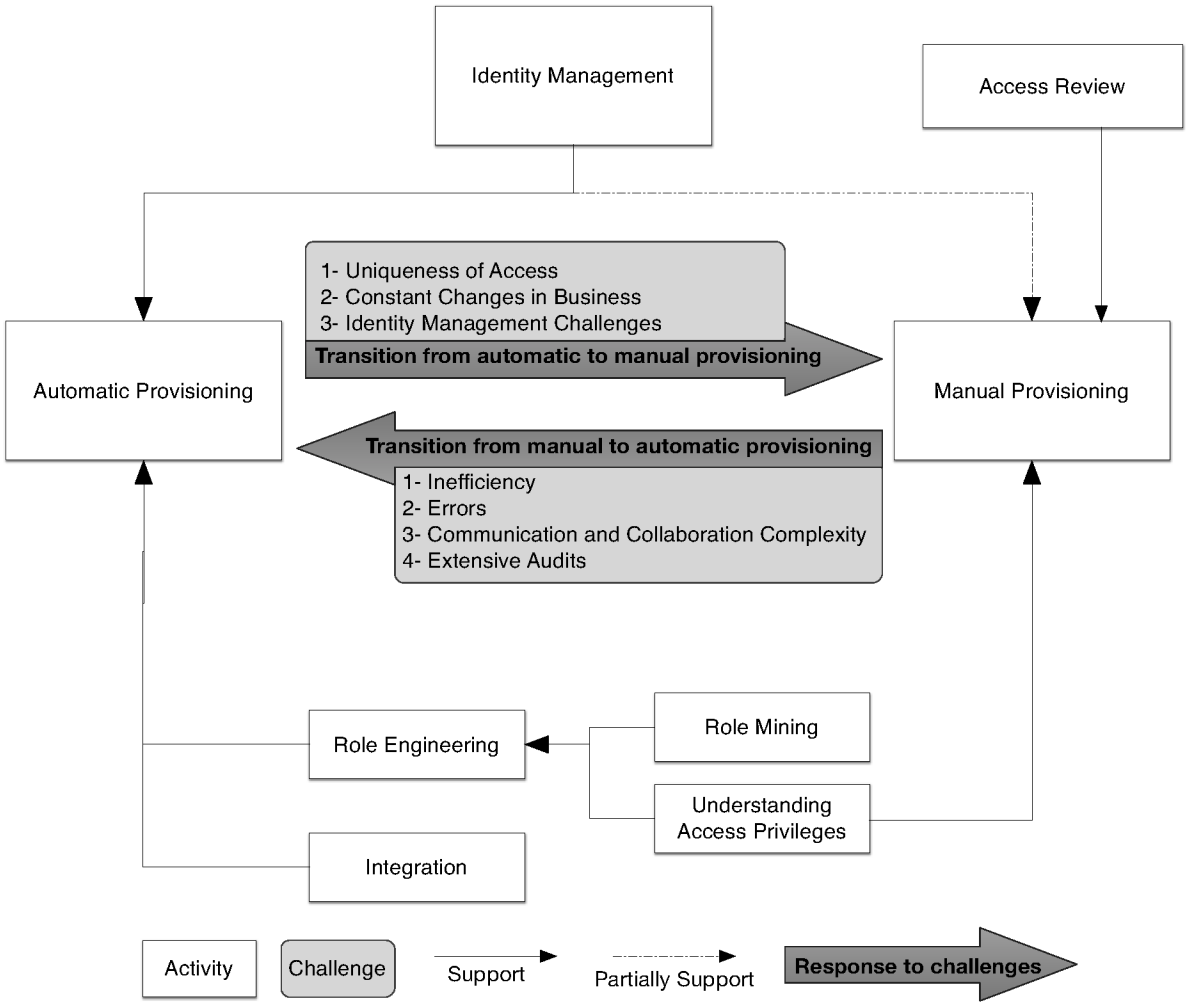


Figure 5.3: The integrated model of IAM and its challenges

contribute to the challenges of manual provisioning.

Towards transition to automatic provisioning, companies perform four activities: understanding and documenting access privileges, role mining, role engineering, and integration with different end-points. Understanding access privileges is a supporting activity that contributes to both manual provisioning, and to the transition from manual to automatic. A well understood set of privileges will reduce all three challenges we found in manual provisioning. Furthermore, such an understanding would help engineering of roles. We also show that the other three activities that support transition from manual to automatic provisioning are challenging, and those challenges are a barrier to the transition.

In summary, the integrated model of the IAM and its challenges explains that why companies use both automatic and manual provisioning. They constantly analyze their manual provisioning, and identify opportunities for transitioning to automatic. Meanwhile, they face exceptional cases, and business changes during their automatic provisioning, and address them by manual provisioning. Cycles between manual and automatic provisioning usually lead to better understanding of business processes and access privileges.

5.4.1 Identity Management Challenges Model

In the previous section, we discussed how identity management (IdM) orchestrated the provisioning, and the challenges in IdM led to sub-optimal operation of access management. The effective functioning of IdM relies on the identity data source. Our interviews showed that the biggest tensions in IdM operation was caused by the relationship between IdM and the source. To study the interaction of these two activities, we modeled both in activity theory framework suggested by Engeström (1999) (see Chapter 2 for an overview of activity theory) and presented the model in Figure 5.4.

The model shows two activities that should work together in order to create, and manage identities effectively. The IdM activity needs to rely on an identity source that is usually managed in a separate activity to keep track of people. Our interviews showed that an HR system was mainly used as the identity source to detect new hires and create identities, to track changes in employee statuses and update their identity, and to remove identities for those employees who left the company. Comparing the interrelationship of the two activities suggests four tensions between various components of them. These tensions are labeled by numbers in Figure 5.4 and listed below:

Tension #1: Mismatch between the notion of employee and identity: The main object of HR is an employee, and the main object of IdM is an identity. Similarly, the HR keeps em-

ployee records as artifacts, and IdM keeps identity records as artifacts. Our interviews showed that there was a mismatch between the notion of identity and employee. While an employee was usually mapped to a unique identity in the IdM system, there were cases where multiple employees were mapped to one identity (i.e., multiple employees shared an account). Furthermore, identities may go beyond the employees, and include contractors, visitors, or even external organizations. This led to companies changing the notion of employee in their HR system to include non-employees.

Tension #2 and #3: IdM and HR motives: The two main motives of identity management activity were security and efficiency. But we observed tensions between IdM motives, and the HR activity. First, the security motive of IdM was not the main concern of HR activity subjects. Furthermore, HR activity subjects did not necessarily possess security awareness, and in many cases were not aware of the consequences of incorrect or incomplete HR data. Second, the efficiency motive contradicted the rules governing HR activity, many of which were irrelevant in the context of IdM. Therefore, we observed that the employee information entry in the HR system was delayed in order to comply with HR related rules and regulations.

Tension #4: HR and IdM collaboration We also observed a tension in collaboration between HR and Security. HR could not provide all the HR data to security, as the subset of the data was considered sensitive (e.g., social security number, salary). Therefore, organizations could not directly have access to the HR database, and many of them had to design an intermediate system between HR and IdM.

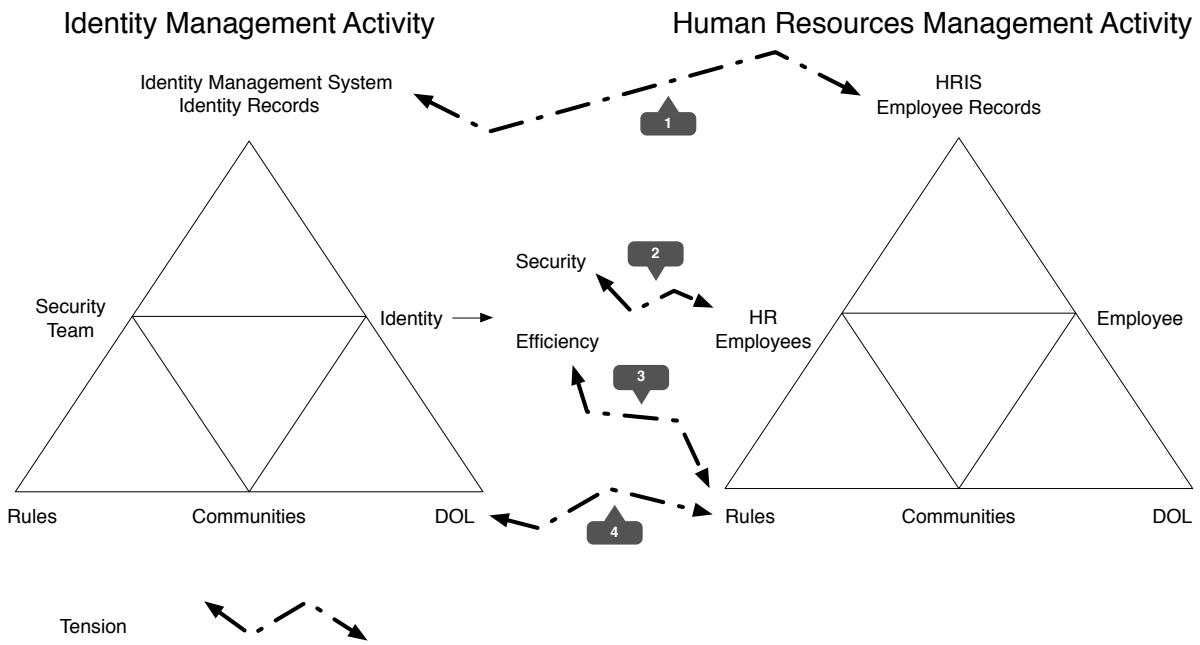


Figure 5.4: Tensions in identity management activity

5.5 Grounding the Work in Access Control Literature

5.5.1 Theoretical Aspects of Access Control

According to Samarati and Vimercati (2001), there are three different abstractions to an access control system:

1. access control policy, which defines the rules that are used to grant or deny access.

2. access control model, which provides a formal representation of the policy and how it works.

3. access control mechanism, which defines the low level functions that implement the access control.

Sandhu and Samarati (1994) classify access control policies into three main categories: DAC, MAC, and RBAC. In the following we provide an overview on each policy.

	File 1	File 2	File 3
User 1	Read Write	Read	
User 2	Read	Read Write	
User 3	Read Write		Own Read Write

Figure 5.5: Access matrix proposed by Lampson

Discretionary access control model (DAC)

The idea behind DAC is to determine the access of a user to an object based on the identity of the user and the authorizations that the user has on the object.

Access control matrix suggested by Lampson (1971) is the simplest DAC model. It consists of three entities: *Users*, *Objects*, and the *Access Rights* that users have on objects. According to Sandhu and Samarati (1994), the access matrix can be implemented using various mechanisms including access control lists (ACLs) or capabilities. In the ACL implementation, each object has an ACL containing the list of users and their access rights on that object. In the capability implementation, each user has a list of objects he has access to. In more advanced forms of ACLs, groups can be defined to assign access rights to users in bulk, or rules can be defined to automatically assign access rights to users based on certain attributes of users (see ABAC in Section 5.5.1). According to Sandhu and Samarati (1994), ACL is known as a flexible method to define the access rights of a user. On the other hand, there are certain disadvantages to this mechanism. First, there is no control over how information is passed between different users, which is important in a military context. Second, if the context changes rapidly, it is hard to manage and change access control lists and capabilities and keep them up to date. Third, understanding the effective policy in this model, which can also include negative authorizations, can be challenging as shown by Reeder et al. (2011).

In our field-study, we saw that companies use ACLs during manual provisioning. As we discussed in Section 5.2.6, account are created for users in an end-point, and then those accounts

are associated with the specific resources in the end-point. Furthermore, our results show that ACLs are hard to maintain in case of rapid changes as they require manual changing of the policy. Conversely, we did not see the challenges with information flow, as our results were limited to enterprise context rather than military. Furthermore, we did not see the problem with negative authorizations, as we only saw instances of negative authorizations in windows file system as discussed by Reeder et al. (2011).

Mandatory access control model (MAC)

MAC model addresses the issue with passing of information between different users by assigning security labels to users and objects. Each object has a security label which determines its sensitivity, and each user has a security label that determines its trustworthiness. There are various MAC policies, but two of the most famous policies are write up and read down (see (Sandhu and Samarati, 1994) for details of each). MAC model is primarily used in military context, in which the flow of the information is important. As expected, during our field study we did not see the use of MAC model in the enterprise context.

Role based access control (RBAC)

Role based access control addresses some of the issues with previous models by introducing the notion of roles. Osborn et al. (2000) show that RBAC is policy neutral and it is possible to implement MAC and DAC policies using RBAC.

The core RBAC model by Ferraiolo et al. (2001) consists of three entities: users, roles, and permissions (also called privileges or entitlements). Users are assigned to roles, and permissions are assigned to roles as well. Roles should resemble the job functions in the organization. A user gets permissions by becoming a member of roles according to his job function.

The core RBAC model also requires the ability to review (or query) user-role assignments as a

mandatory feature, and role-permission assignments as an optional feature.

There are two additions to the core RBAC model that organizations can implement depending on their needs: role hierarchies, and static and dynamic separation of duties.

Kern et al. (2002) suggest that the assignment of users to roles can be done manually, or automatically based on certain attributes of the user (e.g., location).

The RBAC model can be used in conjunction with Attribute Based Access Control (ABAC) (Kuhn et al., 2010) model or variations of it (Thomas and Sandhu, 1997; Thomas, 1997). In ABAC, the assignment of users to roles is determined by certain attributes of the user, the object, and the context. Using ABAC allows for creating dynamic active policies.

As we show in Section 5.2.5, our participants did not solely use the classic RBAC model with roles mapped to organizational structure. They rather used roles with different meanings, such as location, department, and job functions. But there was not a direct mapping between the organizational structure and the roles structure. Furthermore, none of our participants mentioned the use of hierarchical RBAC. Role hierarchies is particularly useful when RBAC model resembles the organizational structure, and therefore, lack of using hierarchical RBAC can be explained by lack of mapping roles to job functions. The access control model used in most of the companies for automatic provisioning was similar to ABAC, were users were provisioned with roles based on their user attributes. On the other hand, manual provisioning followed a DAC policy, implemented with access control list mechanism.

Our findings (Section 5.2.9 and with more details in Chapter 4) explain the need for querying user-to-role assignments. The organizations we talked to, usually required managers to review the access privileges of the employees. In contrast, if the automated role based provisioning was used, the query of user to role assignments was not mandatory for reviews, but prove to be helpful as a contextual information. Also, few companies required the application owners review the roles, and check if they group a correct set of permissions. To satisfy this need, their

IAM system should provided a report of who has access to what.

5.5.2 Practical Aspects of Access Control

In the previous section, we reviewed the three main access control policies, and their associated models, and mechanisms. We also compared our findings to the theoretical and practical aspects of access control. In this section, we discuss the issues with practicing such mechanisms in the organizations and go beyond the theoretical issues by focusing on human and organizational aspects of access control.

Empirical studies on RBAC

Previous empirical studies reported certain benefits and challenges for RBAC. According to Gallaher et al. (2002), RBAC could simplify the administration of access in a dynamic, fast changing organizational context, enhance organizational productivity, reduce new employee downtime, enhance the security, and simplify compliance. On the flip side, Gallaher et al. (2002) report that implementation of RBAC is challenging. Role definition is a time-consuming activity. After defining roles, integrating all of the IT systems in the organization with the RBAC system requires a huge effort. Connor and Loomis (2010) conducted a survey of access and identity management in 2010. More than 150 organizations in various sectors (except military) responded to the survey. The result of the survey shows that RBAC adoption is increasing every year in the IT-intensive industries (Figure 5.6).

Additionally, Connor and Loomis (2010) determined the primary access control mechanisms used to manage different end-points in organizations. Their results (Figure 5.2) show that both roles and ACLs are widely used as the access control mechanism. Many of the NIST's survey respondents reported the use of hybrid approaches. In a hybrid approach, RBAC and ACLs are used at the same time as the primary and secondary mechanisms.

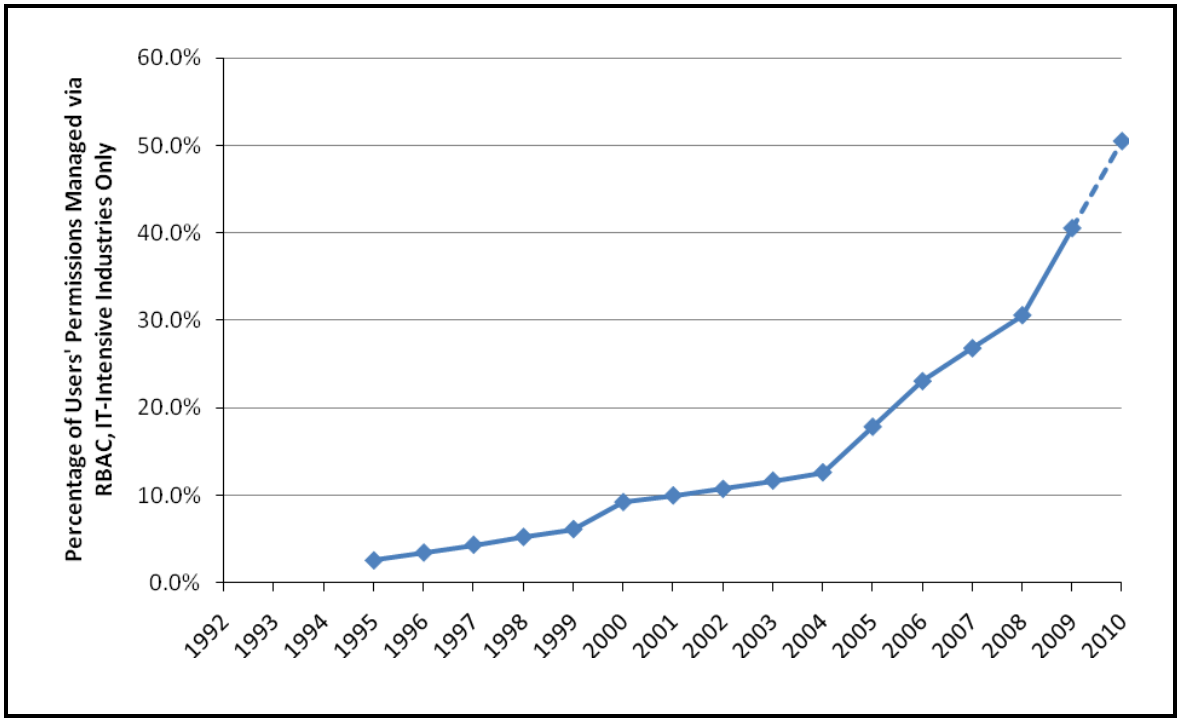


Figure 5.6: Adoption of role based access control in the organizations (figure from (Connor and Loomis, 2010))

Table 5.2: The use of different access control mechanisms in IT systems (figure from (Connor and Loomis, 2010))

System	ACLs	Roles	Rules	Attributes	Other
Human resource information systems	37%	56%	6%	2%	0%
Sales and customer relationship management systems	41%	52%	2%	4%	0%
Accounting and financial management systems	41%	50%	6%	2%	2%
Purchasing, order management, and logistics systems	41%	50%	7%	2%	0%
Business process management systems	42%	44%	7%	4%	2%
Enterprise database systems	43%	41%	10%	6%	0%
Electronic health record and health information systems	48%	34%	10%	7%	0%
Identity management systems	39%	34%	15%	7%	5%
Physical security services	50%	28%	9%	9%	4%
Directory services	49%	27%	10%	6%	8%
Network identity services	53%	22%	14%	6%	4%
Web services	51%	20%	14%	6%	8%

Connor and Loomis (2010) compare the time requirements of four provisioning tasks using RBAC and ACLs. They show that the time to perform four tasks: (1) assigning existing privileges to new users, (2) changing privileges of existing users, (3) establishing new privileges to existing users, and (4) terminating privileges will be 12.4, 7.8, 9.8, and 7.6 minutes respectively using ACLs and will be 6.9, 6.6, 8.0, and 4.7 respectively using roles. On the other hand, the authors do not provide their methodology for calculating the provisioning times.

Our results support most of the NIST findings. First, our integrated model 5.4 shows why the companies use a hybrid approaches in managing access, as completely automating the process is impossible, and using manual provisioning is costly. In our interviews, we saw all of the benefits reported by NIST about using roles. We also show that organizations need to engage in role-engineering and integration activities to achieve automated role-based provisioning, and those two activities are challenging. Conversely, NIST did not report the challenges with maintaining an RBAC system, including the constant changes in the company, uniqueness of access, and identity management challenges. We find these three set of challenges as barriers to adoption of RBAC, and an important decision factor for companies not to use RBAC for automatic provisioning.

Prior research shows that the access control policy changes rapidly in organizations. Kern (2002), in a report on implementation of RBAC, shows that in a bank with 40,000 users, there are about 12,000 changes of user-role assignments per week and 600 changes of permission-role assignments per week.

Managing Exceptions in Access Policies

Managing exceptional cases is one of the main issues with using any of the access control models discussed in section 5.5.1. For example, sometimes an employee changes job and he needs access to his previous job's applications for a short period of time. In this scenario,

an exception should be made to the policy, and the employee should keep the access for the last job. The issue with exceptions has been previously discussed in the literature. In a study by Bauer et al. (2009), security administrators responsible for file system access control and physical security mentioned that managing exceptions are administrative nightmares. Stevens and Wulf (2009) performed a series of field studies in various organizations to understand their access control practices. Their findings show that while organizations use mostly access matrix model, access control decisions can be made at three different times:

Ex-ante control, in which a policy has been implemented before access attempt happens, and the policy is enforced automatically by the system.

Uno-tempore control, in which the permissions are decided at the time that access happens.

Ex-Post control, in which the permission is audited after the access attempt happens.

The authors discuss that most of the technological solutions only support Ex-ante control, but people use the existing technologies to practice the two other types of control manually. For example, they make exceptions to the security policy to grant temporary access to a user that urgently needs access to a resource, or they keep audit logs in order to prevent people from accessing resources that they are not supposed to access. The authors argue that the lack of technological support can lead to human errors and inefficiencies, when people try to align the existing technologies to their access control practices.

While Stevens and Wulf (2009) show that Ex-ante, and Uno-tempore is practiced in organizations, they found limited evidences of Ex-Post practice. The Ex-Post practice has been also discussed in the literature as a new access control paradigm. Povey (2000), in his paper titled *"optimistic security"* points to the gap between what the organizations' need and what mechanisms can implement. He mentions that it is impossible for an access control system to be aware of dynamically changing context. He proposes the concept of *optimistic security*, which can be summarized by the following principle suggested by Blakley (1996): *"Make the*

users ask forgiveness not permissions". In optimistic security, the system allows the access but provides mechanisms for auditing the access attempts and consequently reversing the illegal actions, removing privileges, or punishing users. According to Povey (2000), such systems might not be suitable for contexts with high risk of fraud, but could be useful particularly in health care domain or any context that requires timely responses to the access needs of users.

Our results support findings of Stevens and Wulf (2009) and Bauer et al. (2009). We show that exceptions lead to uniqueness of access and eventually forces companies to adopt manual provisioning. The automatic provisioning in our integrated model (Section 5.4) is similar to Ex-ante control where policies are set before hand and applied automatically, before access attempts are made. The Uno-tempore is similar to manual provisioning, where a user has to manually request access to resources. We also elaborated the Uno-tempore control, and show that such requests can be made before actually attempting access by a manager who expects that the user will attempt access in the future. Furthermore, we identified that in most companies they adopt generalized tools such as email and ticketing systems for practicing Uno-tempore controls. But we also saw some of the companies started adopting new techniques for supporting Uno-tempore controls by offering access catalogs, and access control knowledge bases.

We also saw the instances of Ex-Post control (or optimistic security) in our interviews 5.2.9, but only in medical context, where urgent and immediate access is an important requirements. Similar to the suggestion by Povey (2000), we have not seen the use of optimistic security in organizational context. Our participants suggested that implementing optimistic security in organizations and recording a log of all access attempts is considered a performance overhead, and it is hard to maintain.

Additionally, our results further clarify the previous findings, and show the relationship between Ex-ante, Uno-tempore, and Ex-post controls, and explains the benefits and drawbacks of each approach.

Ferreira et al. (2006) also point to the need for a method to provide users with access to unauthorized resources at the time they attempt the access. The authors call this approach *break the glass*, as the user access the unauthorized object, but there will be cues that the user exercised the access. Like the optimistic security, in this approach users' actions on the unauthorized objects should be monitored. In their proposal, Ferreira et al. (2006) use role based access control model to implement break the glass approach. Whenever users try to access an object, their permissions are determined using the roles they were assigned to. If they do not have permissions, they are presented with a warning message that reminds them all of their actions will be logged. Upon users' consent, the access is granted. This provided non-repudiation to a break the glass method.

In our interviews, we saw a modification of the approach suggested by Ferreira et al. (2006). One of our participants (P15) suggested that users should be able to request access, when they try to attempt access to a resource. But such a request should still go through the regular approval workflow before user can access the resource.

Schaad et al. (2001) describe a case study of adopting RBAC in a financial organization. The authors explain that during implementation of RBAC in a European bank, they designated a well-established method to assign users to roles. But they soon realized that administration of freelancers and consultants broke their control principles. For such users, the access was directly granted by including users' identity in the ACL of applications.

Our model explains why the organization in the case study by Schaad et al. (2001) had to manage freelancers and consultants manually. First, managing them involved identity management challenges of external employees. Second, their access could be unique and not similar to actual employees of the company, which makes the already built roles impractical.

Communication and Collaboration Challenges

While access control mechanisms are capable of implementing various policies, communication and collaboration complexity can result in ineffective access control.

Bauer et al. (2009) performed 11 interviews with security administrators who were responsible for managing access to file system or physical resources in organizations. Their goal is to understand the challenges faced by administrators in their daily activities. The authors do not explicitly mention which access control model was used in the interviewees' organizations. But based on the information provided in their publication, and the fact that their participants administered file systems and physical security, we expect the use of DAC policies with ACL implementation. Using the data collected through the interviews, the authors describe two actors involved in the access control policy generation and management: (1) *Policy Makers* who define the access policy, and (2) *Policy Implementers* who implement the policy. They show that policies are managed by multiple people and miscommunication and lack of shared view of the policy can cause errors in access configuration. Therefore, it is hard for each actor to maintain an understanding of the implemented policy. Usually, the policy maker does not have a direct access to the implemented policy, and cannot verify if it is implemented correctly. Also, the policy implementer is usually blamed for incorrect policy implementation, while the mistake could be from the policy maker.

Our study results support Bauer et al. (2009), and further expand their findings. We saw that the policies were usually set by three different stakeholders: managers, application owners, and security administrators (or service desk). Managers and application owners played a role of policy maker, and security administrators (or service desk) played a role of policy implementer. We also identified similar miscommunication problems, and lack of shared views in Section 5.2.6. Our results also suggest to deal with the errors, companies made the policy makers accountable, and asked them to frequently review the access. In addition, they tried to

understanding and document access privileges to deal with the communication and collaboration complexities.

Role Engineering

In a case study of a European bank by Schaad et al. (2001), the authors provide details of their implementation of RBAC for the financial institute. They mention that the access should be determined based on a combination of official job position and job function of employees. For example, positions can be Clerk, Group Manager, or Regional Manager, and job functions can be financial analyst, share technician or internal software engineer. Therefore, the bank decided to use the concatenation of all combinations of job positions and functions as roles. This approach resulted in 1300 roles in an organization with 40,000 users. The authors argue that this is a normal ratio of roles to users in any organization (3% to 4%). Most of the users in the authors' organization were assigned to one role and at most four roles, and the distribution of users across the roles was not uniform.

Our results suggest that maintaining such a large number of roles would be a huge challenge for organizations. Furthermore, our participants suggested that companies should focus on automating access provisioning for the accesses common between a large number of users, and avoid using roles for provisioning users in very specialized jobs.

Kern et al. (2002) also report on the experience of engineering roles in different companies. The authors argue against using large number of roles and suggested using only one dimension in roles (e.g., job function). Applying a role engineering methodology, Kern et al. (2002) could design 120, 400, and 450 roles to manage 17,000, 30,000, and 31,000 users in three different organizations. In a different publication, Kern (2002) mentions that they parameterized roles by assigning attributes to users, roles, and permissions, and made automatic decisions based on the attributes. The second challenge with implementing RBAC is the number and complexity of

applications in the organizations. Every application has its own method of controlling access. To implement RBAC throughout the organization, an IAM System should be used as a layer on top of all applications. Also Kern et al. (2002) show that implementing all RBAC concepts is not possible with the use of an AIM system. For example, the concept of sessions cannot be implemented in an AIM system, as it is not capable of managing of users' access when access attempt takes place.

We observed similar challenges in our field study as well. Our integrated model (Figure 5.3) showed that integration of IAM with end-points was a necessary step for role based automated provisioning, and it was a big technical and organizational challenge, and it was impossible in some cases. Furthermore, as we discussed before, the tracking of people's access (sessions) was difficult, as it involved an overhead on the system.

5.6 Implications For Design

In this section, we provide a set of recommendations for improving IAM technologies and practices based on the field-study findings.

Use a dedicated identity source: IdM challenges model (Section 5.4.1) showed that the relationship between IdM and HR activities causes a number of tensions. Furthermore, our main model (Section 5.4) showed that such tensions will impact the effectiveness of automated provisioning. To address these tensions, we suggest having a dedicated identity source for creating and managing identities (as suggested by P12). We saw the use of a similar proxy system between HR and IdM in (P1, and P3's organization). But those systems did not allow initiating identity creation. Such identity source should allow on-boarding, and off-boarding of users separately from the HR system. Furthermore, the system can be managed by the security team (or their delegates), who are aware of the consequences of errors in identity related data entry. As our participants insisted on the

importance of having a single source of authority for identity management, we suggest the dedicated ID source should be able to synchronize with the HR system, and allow security administrators to review and resolve the differences between the two system.

Educate the HR team about security: The IdM challenges model (Section 5.4.1) showed the lack of security awareness of HR employees will impact the performance of IdM activity. Our participants emphasized on the impact of mutual understanding between security and HR team. In cases where companies use HR data to create identities, the HR staff should be made aware of the use of HR data for identity creation purpose, and be educated about the consequences of their errors.

Design safeguards against HR errors: Our integrated model of IAM shows that simple errors in HR data entry may lead to significant issues in the IAM system. Therefore, proper safeguards should be put in place against such errors. Rather than automatically modifying identities in high risk situations, the IAM system should ask administrators to manually confirm the sanity of data received from the HR system. This recommendation is also supported by the general *error prevention* usability principle suggested by Nielsen (2005a).

Allow conversion of ad-hoc policies to roles and identity policies: Our proposed model (Section 5.4) suggest that companies continuously try to analyze their existing ad-hoc policies, and convert them to roles and identity policies. To help the transition, tools should allow security administrators to analyze existing ad-hoc policies, finding patterns, and converting that pattern to a combination of identity policies, and roles. Our interview data suggests this process cannot be accomplished by security team alone, therefore, tools should allow seeking help from business people to interpret the meaning of identified patterns in ad-hoc policies, creating roles, and writing identity policies that map users to roles.

Verification of the impact of new roles: To provide any leverage, roles should be designed in a way that a large number of users can be assigned to them. Furthermore, such an assignment should not break any of the existing policies. Therefore, role engineering tools should allow evaluating the impact of the created roles, so that security team can determine if they provide any leverage. For example, tools can show users who will be assigned to a newly created role, and highlight the changes of the access policy due to the changes to the role structure. Therefore, security administrator can verify if the created role will lead to automatic provisioning of a large number of users, and can validate if assignment of the users to roles will violate any of the existing policies such as SoD.

Document the meaning of access privileges and roles: Our integrated model of IAM showed that understanding access privileges can contribute to both automatic and manual provisioning. Therefore, we suggest the information about the access privileges including the system they belong to, their technical information (e.g., their system code), their business meaning, the risk associated with having the access privilege, and the information of a person who can answer questions about the access privilege should be documented. Such information can act as a common vocabulary between managers, application owners, and security team, and will help them in creation, interpretation, and implementation of the requests. Furthermore, the information can be helpful during the role-engineering exercise, and access review.

Show context during access request work-flow: Our participants suggested that they use generalized ticketing systems (or even emails), for making, and tracking the access requests. IAM tools should allow incorporation of context into access request tools. For example, tools should show the employee what he can request, and show the approver the information about the requester, his other access privileges, and the history of his requests.

Allow understanding and making sense of access policy: Our model (Section 5.4) highlighted that the access review activity cannot be completely automated, as companies cannot

eliminate advance provisioning. Therefore, tools should allow reviewers make sense of access policy, and make a judgment about the validity of access. We further expand this recommendation in Chapter 6, and show how this recommendation can be implemented.

5.7 Discussion

We showed that efficiency, cost reduction, and security were three motives behind identity and access management. Ideally, an IAM system should provide users with the least set of privileges required for their job, and not more. But we showed that such a model leads to inefficiencies, as users need to ask for privileges for any ad-hoc task that requires additional privileges, and then the excessive privileges should be revoked from the users as soon as they are no longer needed. While this model aims to ensure the principle of least privilege, it can be a road block for employees doing their job, and overload managers and security administrators with access requests, reviews, and removals. We also show that this model can eventually negatively impact the security of the system as managers or security administrators tend to commit errors when facing large number of requests, or access privileges to review. We also showed that complete automation of this process is impossible. We observed companies try to find a balance between security and business continuity by perform risk assessment, and in cases where business continuity outweighs security risk, they relax the security requirements.

Therefore, we suggest a model where security administrators act as facilitators, rather than getting involved in every access decision. We believe this model can benefit both security and business continuity. In this model, employees can be responsible for requesting access to the resources they need through an access catalog, and access can be automatically granted in the absence or tolerable level of risk. Depending on the risk level, security team can be notified about the request, and monitor the users' access attempts using an optimistic security paradigm (see Section 5.5.2). If the access request involves high degree of risk, it can go through proper approval workflow, and then the policy can be reviewed and updated, so that future cases can

be handled automatically. In this model, users can perform most of their access requests in a self-serve fashion, and efficiency will improve. The managers can focus only on approving and reviewing high-risk access privileges, which eventually reduces human errors, and security team can focus on improving and evolving the high level access policy, rather than dealing with ad-hoc access requests.

5.8 Conclusion

In this chapter, we used grounded theory to collect and analyze qualitative data from 19 security practitioners about their experience with IAM adoption and use. Our results provided a thick description of different IAM activities and their challenges. We then provided an explanatory model to describe how different activities and challenges are linked to each other. We then reviewed related work in this area, and tried to analyze them through the lens of our findings. Finally, we provided suggestions for improving technology or practice, and suggested a user-centric approach for access management.

Chapter 6

Designing Usable Access Review Interfaces[1]

6.1 Introduction

Understanding and authoring access control policies has been known as a challenging problem according to Reeder et al. (2008), and Smetters and Good (2009). But the focus of previous studies were on personal access control, where the data owner, policy maker, and policy implementer are the same person. This problem has not been extensively studied in organizational context. Bauer et al. (2009) found that managing access control policies in organizations faces a unique set of challenges. In large organizations, those who make policies are different from those who implement these policies. Therefore, developing a shared understanding of policy between different stakeholders is challenging. In this chapter, we explore and address this problem by proposing and evaluating AuthzMap, a new user interface for sense making and

[1] A subset of material presented in this chapter has been published or accepted as the following conference publications:
P. Jaferian, H. Rashtian, and K. Beznosov. 2014. To authorize or not authorize: helping users review access policies in organizations. In Symposium On Usable Privacy and Security (SOUPS 2014). Pages 301-320. Menlo Park, CA. (acceptance rate: 26%)
P. Jaferian, H. Rashtian, and K. Beznosov. 2014. Helping users review and make sense of access policies in organizations. In CHI'14 Extended Abstracts on Human Factors in Computing Systems (CHI EA '14). ACM, Toronto, ON, Canada, 2017-2022. (acceptance rate: 49%)

reviewing implemented access policies or, in short *access review*.

In Chapter 5, we showed that access review is an important IT security activity in organizations, and is used to detect and eliminate errors that happen in manual provisioning. The managers are mandated by many security regulations (e.g., SOX (United States Code, 2002), HIPAA (Centers for Medicare & Medicaid Services, 1996)) to regularly review and validate the access privileges of users. However, Cser (2011) suggests that access review for every 2,000 to 3,000 users consumes approximately one full-time-employee equivalent per year, and many organizations cannot even finish one access review process before a new campaign begins.

Recent security incidents that cost governments and organizations billions of dollars show the importance but yet lack the ability in reviewing users' access rights. For example, the US army soldier, Chelsea Manning, who leaked the US embassy cables was cleared to access classified resources when she was on training as an intelligence analyst. She then changed her job and location multiple times before going to Iraq. According to Swensen (2011), if a superior reviewed Mannings' access and requested the revocation of unnecessary privileges, she would not have been able to leak the data.

The overarching goal of this chapter is to investigate improvements to technology support for access review. Towards this goal, we performed four studies. In the first study, we used interviews from Chapter 5 as well as a survey of 49 security practitioners to understand how people make sense, and review access of users, and to identify the challenges in access review. In the second study, we used the results of heuristic evaluation from 24 evaluators (Chapter 4) to understand usability problems with one of the existing access review tools. We then designed a new user interface, guided by activity theory and ITSM guidelines (Chapter 3), to address the identified challenges. We named the proposed interface AuthzMap. In the third study, we improved AuthzMap through multiple rounds of informal evaluation, expert reviews, and a heuristic evaluation with 12 usability experts. In the fourth study, we conducted an online exploratory study with 430 participants to test if AuthzMap improves the usability over two of

the existing interfaces.

6.2 Background

Organizations use many IT applications to run their business. Employees who use an application for their job are provided with a set of access privileges, and other employees should be prohibited from accessing the application. Therefore, applications provide a set of *permissions* that can be assigned to a *user* to control what the user is authorized to do. Sometimes, permissions are grouped into *roles* to simplify and automate the provisioning process. In Chapter 5, we show that organizations use a combination of automated and manual provisioning. Furthermore, we show that the manual provisioning is prone to errors. Therefore, organizations are mandated by many security regulations (e.g., SOX (United States Code, 2002)) to frequently perform access reviews to make sure that users have the least set of privileges required for their job.

Next, we describe how access review is performed through an example. In an organization, security administrator, John, sends a request to manager Bob to review the access privileges of fifty employees who work in Bob's department. Bob is provided with the list of employees and their access privileges. He reviews the list of users one user at a time, looks at their privileges, and verifies if the user-to-privilege assignments are valid. For example, Bob sees that Alice is assigned to 20 privileges (R1, R2, ... R20). Bob needs to understand the meaning of the privileges, what they authorize Alice to do, and if the authorizations are required for Alice to do her job. If an authorization is required, Bob *certifies* the assignment of Alice to the privilege. Otherwise, he *revokes* the assignment. If Bob cannot understand the meaning of an access privilege, he may communicate with John or other managers to ask for a clarification. This example shows that access review requires intensive analysis, communication, and collaboration with other stakeholders.

6.3 Related Work

There have been few studies related to access review in organizational context. Bauer et al. (2009) performed a field study of access control practices in organizations. They suggest that the implemented access policy and the record of changes should be understandable and visible. Our findings confirm this, and our proposed interface improves understandability and visibility of access policy.

As opposed to access review, the problem of policy authoring has been previously studied. Brodie et al. (2005) designed a privacy policy management workbench called SPARCLE to create policies in natural language. Although SPARCLE was successful in facilitating policy definition and management, it was not used or evaluated for the access review. Inglesant et al. (2008) studied personal access control in Grid computing context. They showed that resource owners have difficulty expressing policies in RBAC and they prefer the use of natural language. Reeder et al. (2008) proposed a new UI named "expandable grid" for understanding effective access policy in case of conflicting access rules. Expandable grid improves the understanding of access policy by end-users of commodity OSs, and their main goal is to address the issue with conflicting access rules that happen regularly in the Windows file system. The data from our interviews show that in enterprise environments, standard role-based access control without negative authorization rules is used. We also adopt the idea of expandable grid for use in an organizational context and use it in the design of AuthzMap. Smetters and Good (2009) studied the use of policy authoring for personal documents. They found that users rarely change access policies, and tend to specify complex and error-prone policies. Our findings suggest that unlike access control for documents, the users' accesses change frequently in organizations. Vaniea et al. (2012a,b) examined the effect of proximity of access management interface and the resources. They show that users detect errors better if controls are positioned near resources. Their proposed method was implemented and evaluated in the context of managing photo album privacy policy. In an organizational context, this proposal might not be possible,

as resources do not have direct graphical representation, and the number of resources and permissions could be large. Beckerle and Martucci (2013) identified six guidelines for designing usable access control rule sets, and showed that implementing those guidelines will help understandability of access policies. Their proposed solution can be used before presenting access control rule set in AuthzMap to reduce the complexity of policy.

6.4 Study 1: Understanding the Activity

As we discussed in Chapter 5, the primary goal of the interview study was to understand how organizations perform identity and access management, identify the challenges they face, and understand how technology can be improved to address the identified challenges. After we developed our integrated model of IAM and its challenge, we decided to focus on one specific activity in the model that could be improved by better technological support. Therefore, we focused on access review. Particularly, we tried to answering the following research questions: (1) Why organizations perform access review? (2) Who are the involved stakeholders? (3) Why access review is challenging? (4) How better decisions can be made during access review?

6.4.1 Methodology

We used a combination of qualitative and quantitative methods to understand the access review activity. Initially, we performed 13 semi-structured interviews with security practitioners responsible for access management in large organizations. After the initial analysis of the interviews, we performed a survey to collect descriptive statistics about various aspects of access review found during the interviews. We also used the survey to recruit 6 more interview participants as a part of our theoretical sampling. Our results are mainly based on the qualitative data analysis. We used the survey data to triangulate, complement, and elaborate the qualitative findings. We discussed the methodology of our interview study in Chapter 5. In this section,

we provide the details of our survey methodology.

Survey Goals

We designed a survey with three goals in mind: (1) Clarify certain findings in the interview study, in which we observed contrasting results. (2) Collect quantitative data that can help us in designing a new interface for access review. (3) Collect data that help us design an ecologically valid study to test the AuthzMap interface.

In particular we tried to answer the following questions:

1. How are users provided with access to resources in organizations ?
2. Who is responsible for performing access review in organizations?
3. Which stakeholders are involved in access review, and how do they collaborate with each other?
4. What pieces of information can be useful during access review?
5. What would be an indication of risk when evaluating access privileges of users?

Answering questions 1-3 extends our understanding of access review, and questions 4 and 5 will help us designing better technologies for access review. To answer the aforementioned research questions, we designed a questionnaire with 20 questions. The questions were a combination of closed and open-ended, and participants were allowed to skip answering any of them. Based on our field study experience, the target audience for the survey were security practitioners who are busy, and not willing to spend long period of time on a study. Therefore, we minimized the time required for completing the survey (between 10 and 15 minutes). For each question, we specified a set of possible answers based on the field study results. But we also added an "Other" option, so that participants can enter their answer (i.e., in case their

answer is not in the provided options). The survey questions are presented in Appendix C.

Recruitment

We distributed the survey through Linked-In communities. The link to and the description of the survey was posted twice to three different Linked-In groups related to identity and access management (the three groups had 12,125, 8,792, and 4,598 members at the time of conducting the survey). Also the link to the survey were posted to IAM related discussion forums, including support forums for CA, Oracle, and Novel IAM systems. We also posted the survey to the Forrester Research[2] discussion forum. Unfortunately, these recruitment efforts did not lead to recruitment of any participants. But we received a collaboration request from Forrester Research company. They showed interest in collaborating with us in publicizing the survey, in exchange for survey data. As a result, we sought help from Forrester Research. We used their social media channels (including their weblog and twitter feed) to distribute the survey notice. As a token of appreciation, we promised a raffle for a 128Gb iPad, and a complementary report of the survey results from Forrester Research. We should note that our sampling method was not completely random, as participants may self-select themselves for the study. Furthermore, our participants were among those who read Forrester research Weblog or twitter feed, or those who have heard about the survey through the word of mouth. We also do not expect the raffle to had an impact on the participants' decision of participating in the study. As we show in the next section, most participants were highly experienced security professionals, who are busy, and well paid. Therefore, we believe the raffle did not introduce further self-selection bias. We received 57 responses to the survey, out of which 49 were valid responses (e.g., two participants declined the consent form, and 5 just browsed through the survey).

[2]Forrester Research is an independent technology and market research company that provides advice on existing and potential impact of technology, to its clients and the public.

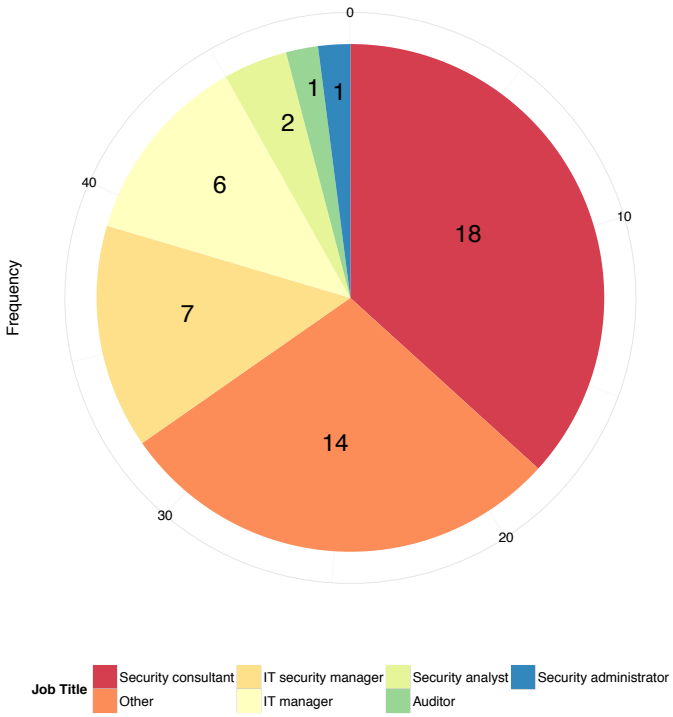


Figure 6.1: The job title of the survey participants

Survey Participants Demographics

The job title of the survey participants is shown in Figure 6.1. Our results show that the participants were mostly security consultants and managers. Also we had very few security administrators responding to the survey. Those participants who chose the "Other" option in the survey, had the following job titles: Analyst, Chief Technology Officer (CTO), Business Executive, Director of Financial Controls, Lead IDM Consultant, I&AM Engineering Manager, Application support including identity systems, Technical architect IdM & Access management, Software, Developer, Human Factors/Design Research, Solution Architect, System Administration, Sales Rep, Business development IAG, Business Systems Analyst, VP Enterprise Architecture.

We then asked participants about their experience in IT security and experience with identity and access management. The results of these two questions are presented in Figure 6.2. There was a strong correlation between the years of IT security experience and IAM experi-

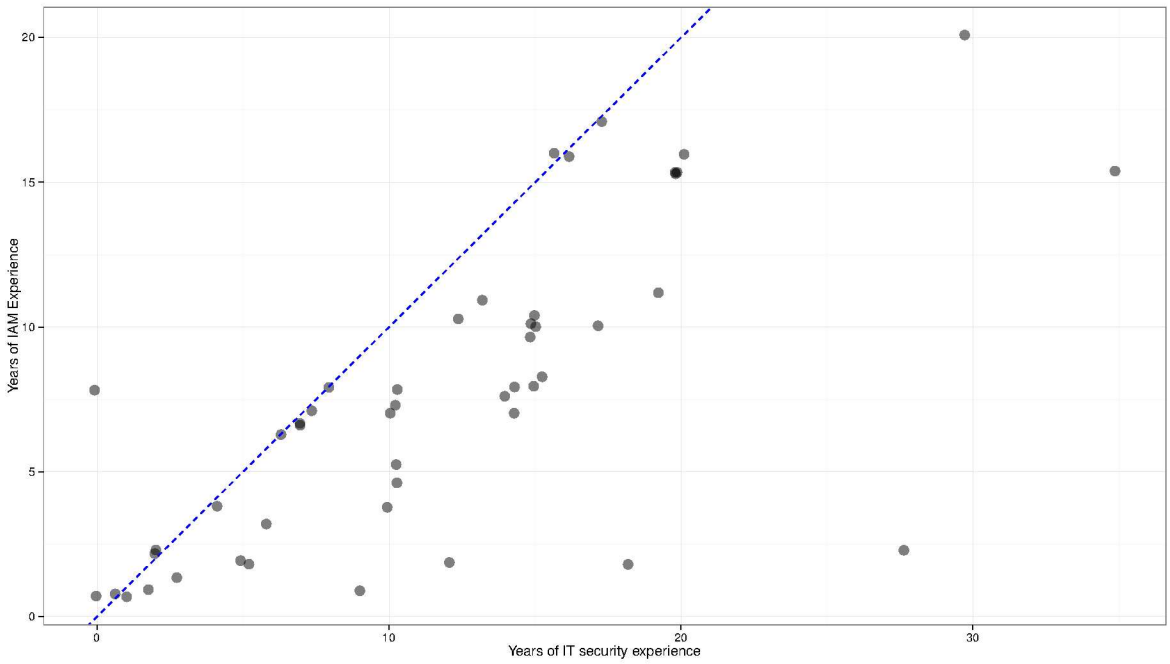


Figure 6.2: The participants years of IT security and IAM experience. Each dot indicates a participant. The blue dashed line is the identity line. The points are jittered to avoid over-plotting.

ence (*Pearson's* $r(47) = 0.70, p < 0.01$). In other words, participants were dealing with IAM problems during their career as a security professionals. For most cases, participants' years of IT security experience was higher or equal to years of IAM experience. There were two cases were participants reported no experience in ITSM, but some experience with IAM. In one case, participant was a director of financial controls and noted that *"I'm in Finance, not IT. I am responsible for IT controls as part of the overall system of controls. I have experience of an indirect nature with several IAM applications."* The other case was a participant who reported his/her job as *"Human Factors/Design Research"*.

6.4.2 Results

In this section, we combine the results of the interview study with the survey to provide a detailed description of access review activity. We provide our description in the activity theory framework, and then discuss the identified challenges. As we discuss each component of the

activity or challenge, we first present our findings from qualitative study, and then elaborate the findings with quantitative data (only in cases where we collected quantitative data through our survey).

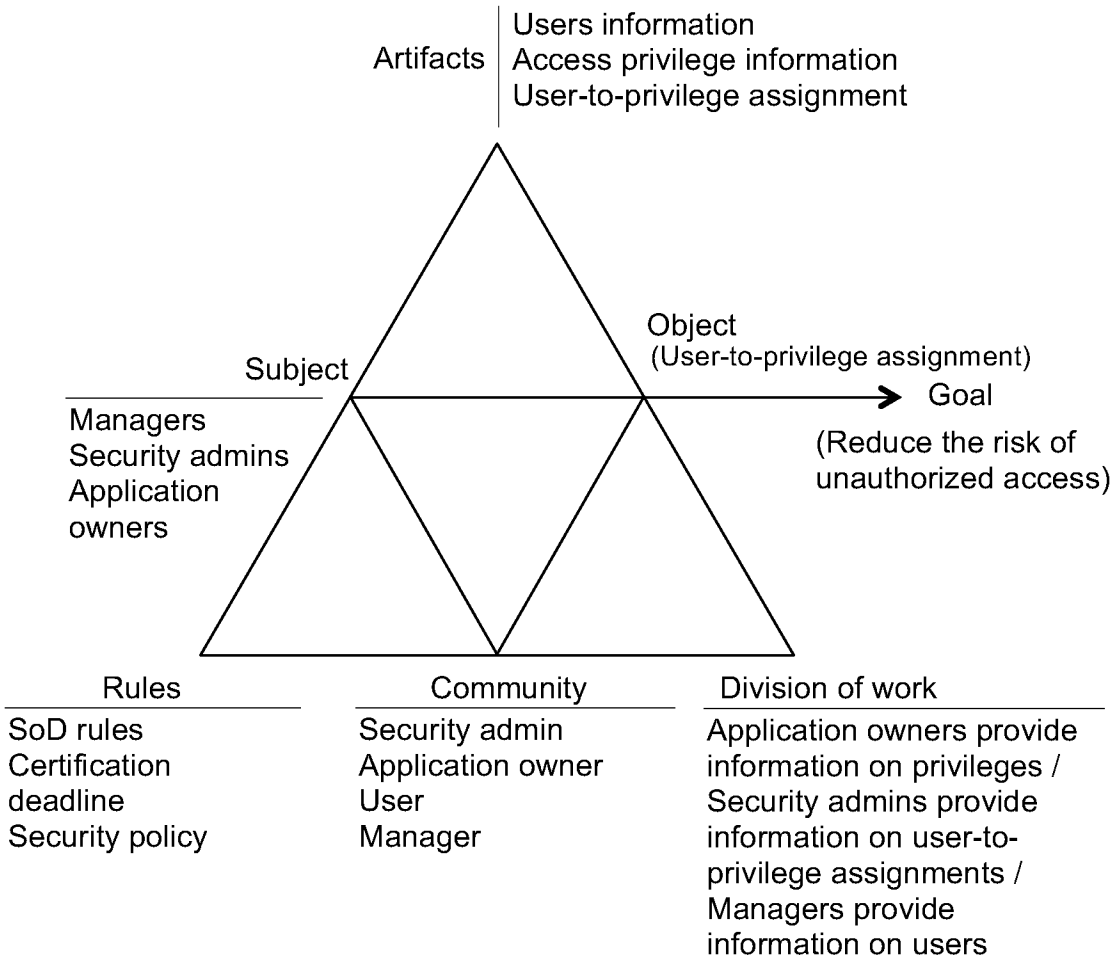


Figure 6.3: Overview of access review activity presented in the triangular model of activity

We use the triangle model of activity proposed by Engeström (1999) to lay out our description of access review (Figure 6.3). We will later refer to this formulation when we justify our design decisions.

The goal of the activity: Access review is an activity with the goal of verifying users' access rights to minimize the risk of unauthorized and unmanaged access and comply with regulatory legislations.

Subject: "Reviewer" is the main actor in the activity who performs access review. Our participants indicated that the following stakeholders act as reviewers:

Managers: Most of the participants indicated that managers review employees under their authority. P1 further described the role of the manager in access review: *"[Manager asks:] what access does Jim have? I'd like to review Jim's access because he's changing roles within my department, there's no official job posting but I'm doing a realignment and I would like to review Jim's access. So you need to do a specific report on Jim, which is to say here is the access profile that Jim has."*

Application owners: Two of our participants indicated that an application owner reviews the users who have access to the application, and certifies or revokes the users access privileges: *"Our team wrote some [scripts]. It goes out and it collects from these 80 or so applications, what the access lists are, what the rights are, it creates a report, we put it in a service desk ticket. Then it goes out to the [application owners] and they review it."* (P3)

Security administrators: P6 explained that his team is responsible for security compliance of a large enterprise application, and therefore he performs access reviews: *"We send a request to the manager that says Bob has changed from position A to position B. They are requesting position B roles. We are going to remove his position A roles. Do you agree with that?"*

We asked survey participants who *currently performs* access review in their company, and who *should perform* it. We included different options based on the interview data, and also added "Other" option to check if the survey participants perform review differently or can think of new approaches. The summary of the responses is shown in Figure 6.4. Two participants chose the *"Other"* option as their current method of performing reviews: one mentioned that they used a mixture of the methods, without providing further details, and one participant mentioned that they review the role structure. For the desired state of access review, four participants chose the *"other"* option. One mentioned that the governance team in the company

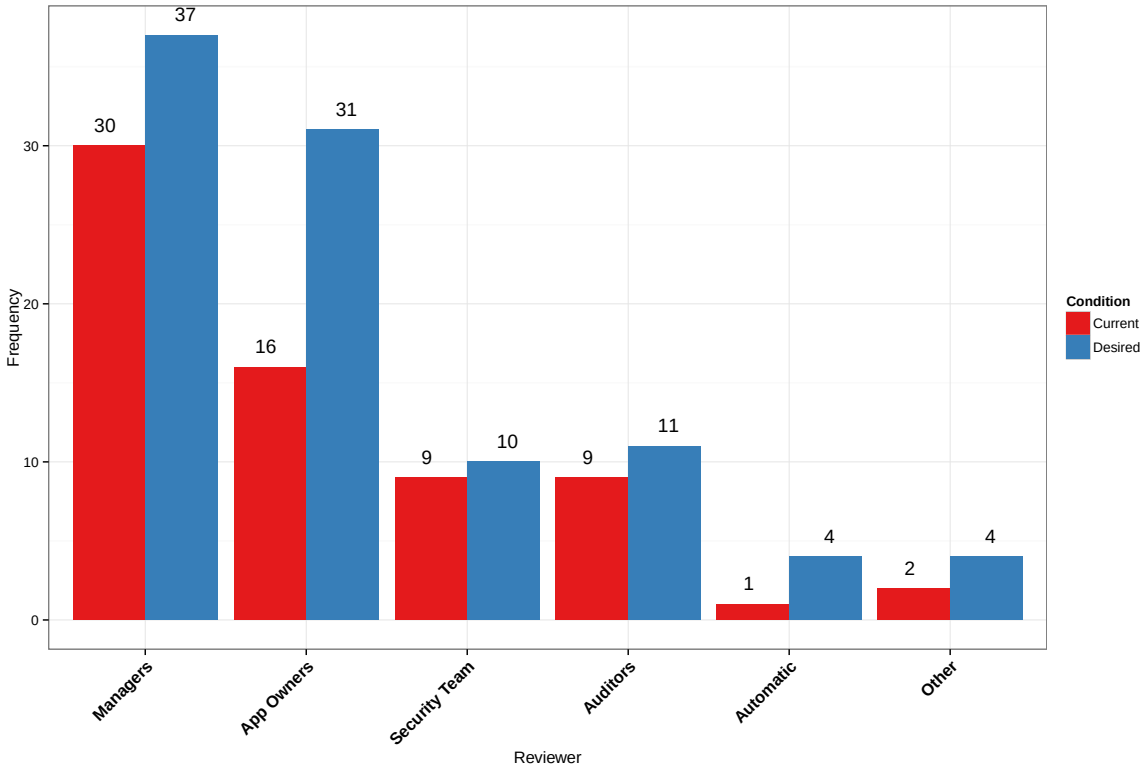


Figure 6.4: Survey participants' responses on who currently performs and who ideally should perform access review in their company

should review those access privileges that are undecided by the managers after the review. Another participant wrote that the type of access privilege should determine who reviews it. One participant described that "role owners" should also review the access, beside the managers. Furthermore, those people who approved the assignment of user to role should also review the access. One other participant mentioned that while managers and application owners both should review the access, the composition of the roles should be reviewed as well.

Object: The object towards which the activity is performed is a user-to-access privilege assignment. When managers or security administrators perform access reviews, they review a set of access privileges assigned to a user (user access review). When application owners perform reviews, they review a set of users assigned to an access privilege (application access review). We limit the scope of the AuthzMap to user access review as our survey suggested that this is the dominant approach. The same design techniques can be applied for building an interface

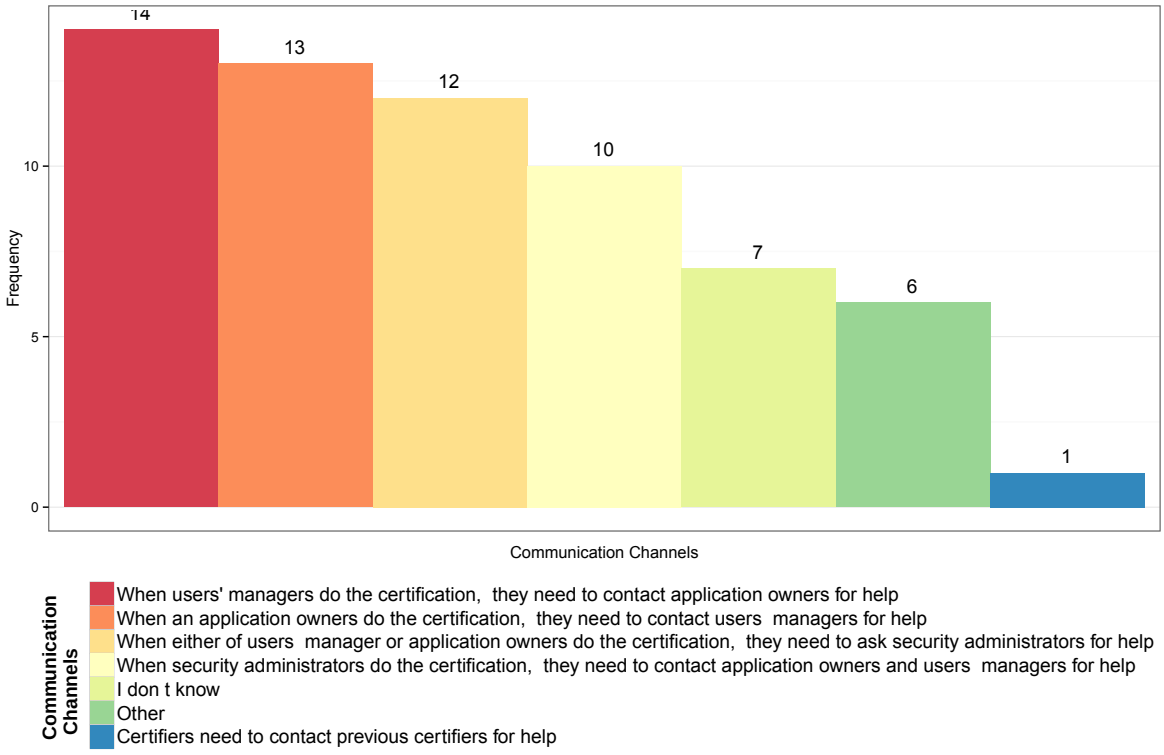


Figure 6.5: Communication channels that should be used during access review from survey participants' perspective

for application access review.

Community and division of work: Access review involves security team members, employees, managers, and application owners. Involved stakeholders divide the work as follows: A member of the security team requests review of users' access rights. The reviewer (a manager in most cases) receives the request. He goes through the list of users, selects a user, and identifies the user's access privileges. For each user-to-access privilege assignment, he chooses to certify or revoke the assignment. The reviewer might contact the application owners, the user, or the security team when he is unable to determine the correct action. To confirm and extend the list of communication channels used during the review, we asked participants about the possible communications between various stakeholders involved in the access review activity. A summary of the participants' responses is shown in Figure 6.5:

Those participants who chose the "Other" option offered the following suggestions: (1) One participant mentioned that there is no need for any communication if the list of users and their access privileges is available to the manager. Another participant noted that a *recertification team* should be in place to monitor and help if needed. One participant explained that manager needs to discuss with users if there is a business need for the access.

Rules and constraints: Different rules and constraints impact access review. (1) The security policy of the organization determines the validity of a user-to-access privilege assignment. For example, P9 explained that in health care, they follow an optimistic security paradigm and allow more access than usual so the physicians can access patients' files in emergency cases: *"So the whole access model in health care tends to be, you let people do what they need to do to get the job done."* (2) Static separation of duties rules determine if a user can be assigned to two or more specific roles at the same time. (3) The review deadline set by security team constrains the time window of the review.

Artifacts: Reviewers use three artifacts during access review: (1) User's information, which include the identity related information, the job title, and other attributes like the phone number, email, department, etc. (2) User-to-access privilege assignment information, which include who requested, who approved, and who implemented the assignment, when and why the user was assigned to the access privilege, and who previously reviewed the assignment. (3) Access privilege information, which include the privilege's name, description, the owner, and the type of access it provides.

6.4.3 Access Review Challenges

Our interviewees indicated that access review is a challenging activity. We classified these challenges into five categories:

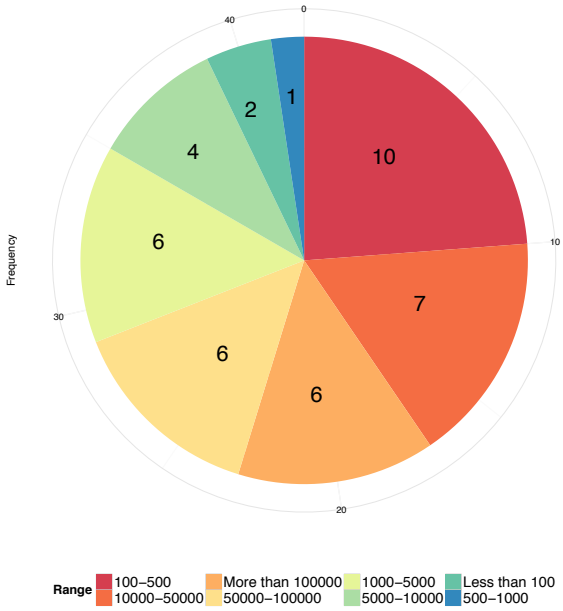
Scale: Access review can involve large number of users, roles, and permissions. P6 explained

that just one of the large applications in his organization has 16,000 users, up to 115 roles per user, and up to 407 permissions per role. He also indicated that reviewers have to review up to 200 users in a review activity. While these numbers vary from application to application, and from organization to organization, they show the magnitude of data that a reviewer needs to deal with.

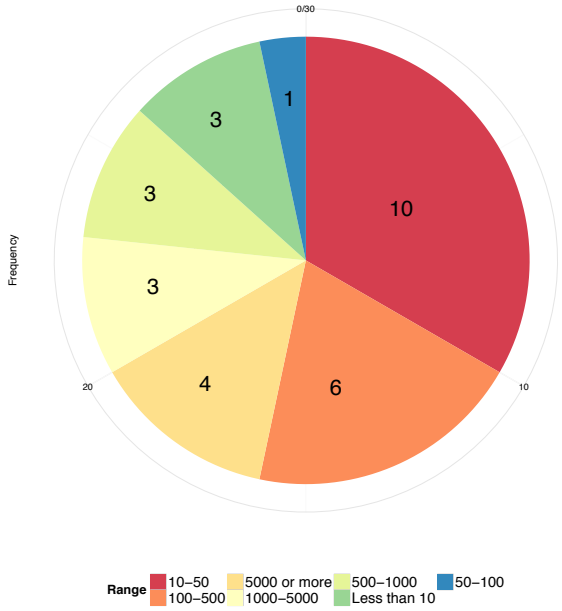
Our survey results also shed light on the scale of access review. We asked survey participants to estimate the number of users, number of applications, number of access privileges per user, and the number of users reviewed by one manager in each review session. The summary of the responses is shown in Figure 6.6.

Lack of knowledge: When managers act as reviewers, they do not have the expertise to understand the meaning of the access privileges. P2 illustrated this problem in detail: *"we send these god-awful long reports to the new manager hiring the employee is going into, saying 'let us know which access this person needs to keep and what they need to remove.' And a lot of it's, you know, cryptic RACF information and stuff they just have no idea what they're even reading so they either take their best guess and say, ok, then maybe this sounds kind of like something they might need. Or they just say they need it all."*

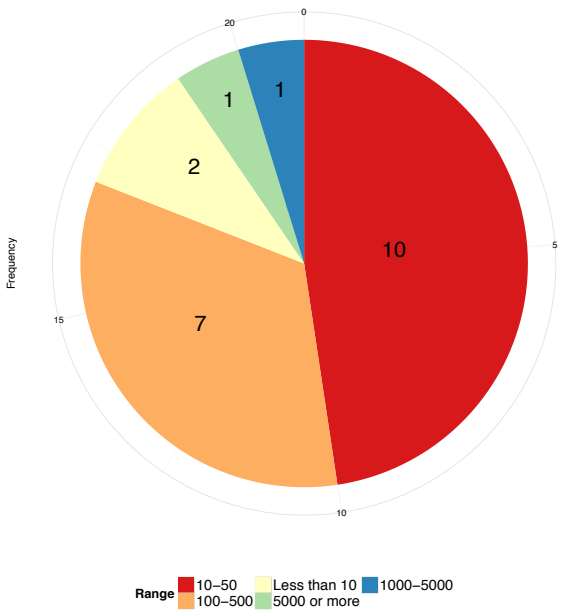
Frequency: While reviewing access is not the main job of managers, they are frequently asked to perform this activity. For example, P3 explained why they perform quarterly access reviews: *"... Once a quarter! We do quarterly access reviews. [...] Once a year is never good for any control because if you fail, you fail; at least twice a year you have a chance to re-mediate."* Additionally, P3 talked about ad-hoc access reviews: *"Every day, [access management software] looks at [every] person who has access and says has the person changed in any way. Did they move departments, did they move to geographical locations - if so it triggers an event which puts a ticket into the service desk system, sends a note to the Access Reviewers and says you need to review this ..."*



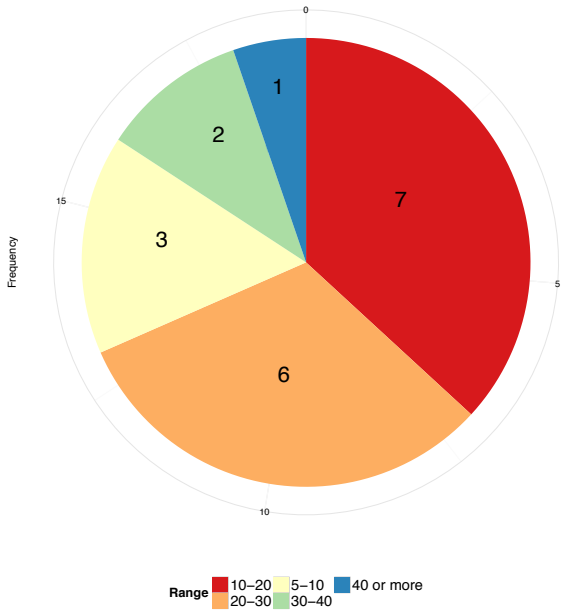
(a) Number of users in the organization



(b) Number of applications in the organization



(c) Number of access privileges per user



(d) Number of users to be certified by one manager

Figure 6.6: Descriptive statistics on the scale of access review in survey participants' organization. Participants were allowed not to answer the questions, and therefore, the number of data points in each graph may not add up to 49.

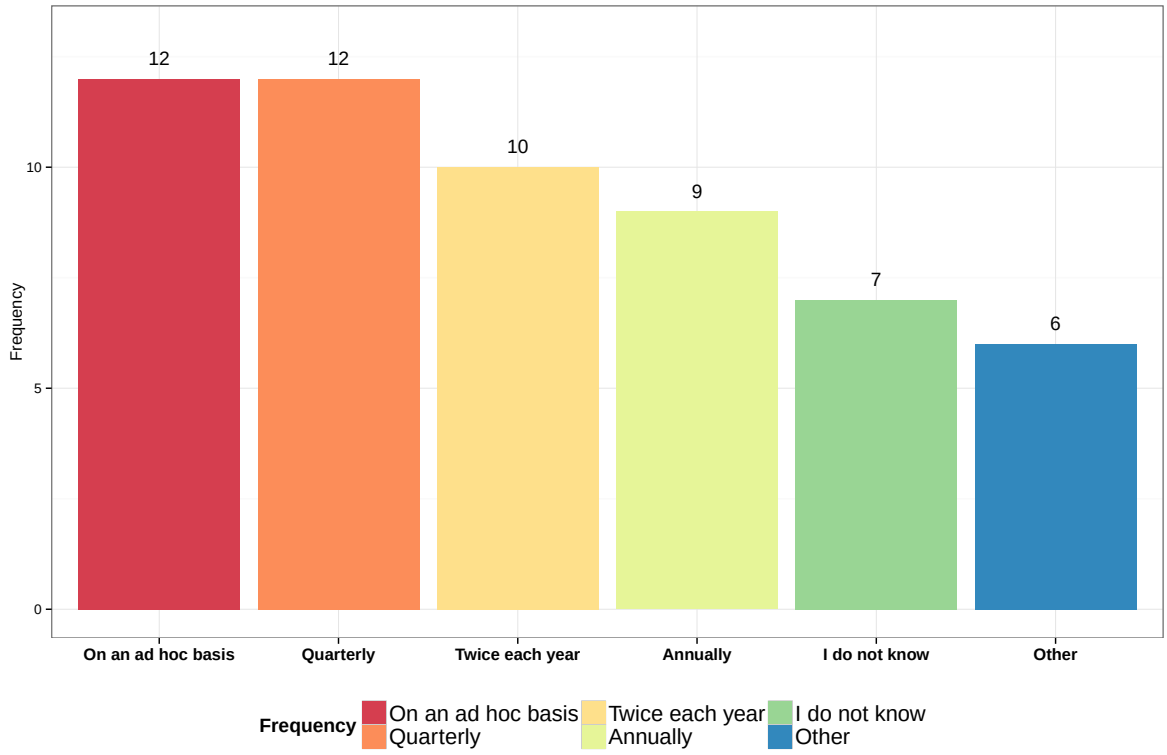


Figure 6.7: Frequency of performing access review in survey participants' organization

Our survey data provided more details on the frequency of access reviews. We asked participants how frequently they do reviews, and they could choose multiple answers ranged from Ad-hoc to once a year. The answers to this question are summarized in Figure 6.7. Participants also could choose the "Other" option, and provide comments. One participant wrote that review should be done *"On every employee job transfer, both by previous and current managers."* Another participant also suggested reviews on job transfers, and added that a review should be done during deployment of new applications as well. Another participant wrote that they determine the review schedule at the beginning of each year, and one other participant said that they do not do it consistently. Finally, one participant stated that they do it automatically, and in real-time. The survey results show that the ad-hoc and quarterly reviews are the most common, followed by twice a year, and annual reviews.

We also asked participants what events in the company can trigger access review. Participants

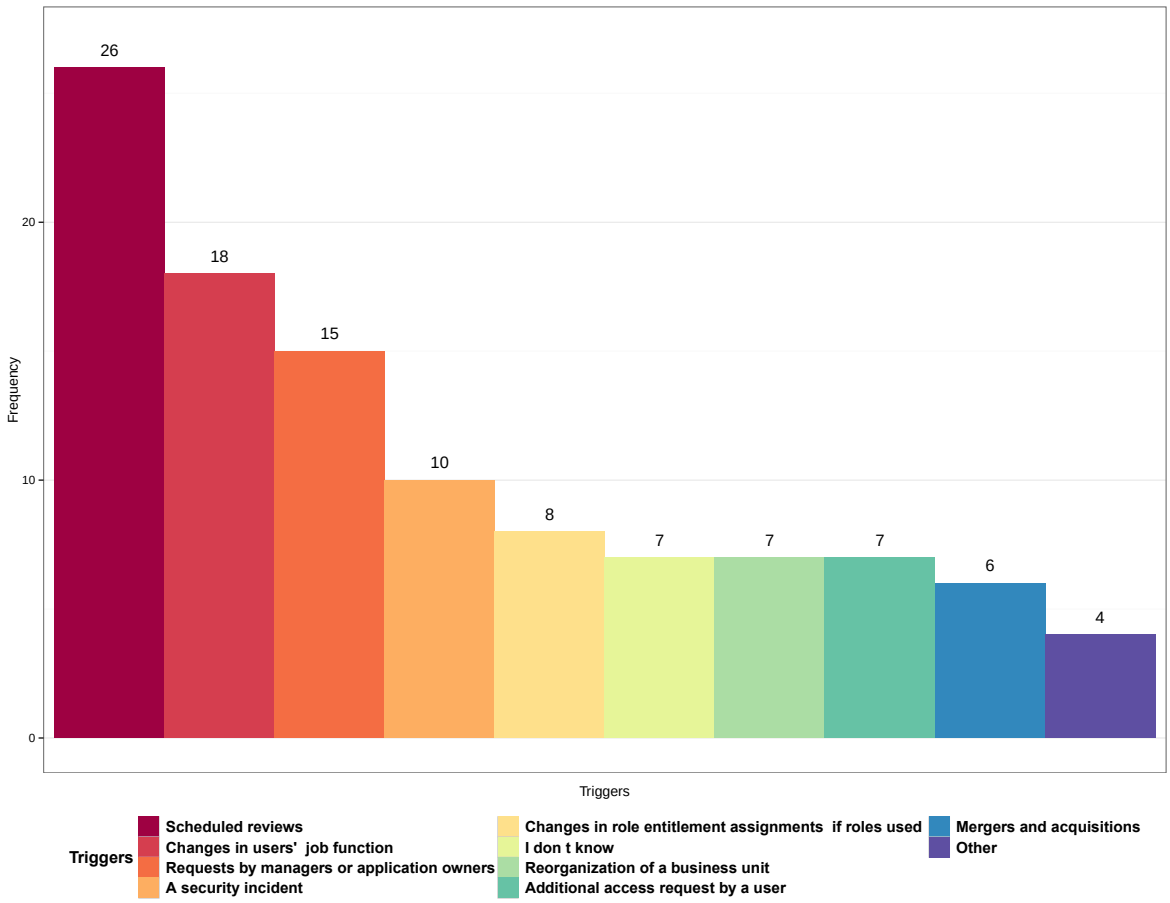


Figure 6.8: The list of events that can trigger access review in survey participants' organization.

could choose multiple options including the "Other" option. The summary of responses is presented in Figure 6.8.

Those participants who provided the other option, suggested the following events can trigger access review: (1) Change of a manager (2) Employee separations (3) Audit findings (4) New projects (5) New applications (6) Hiring new employees.

Human Errors: P3 described why human errors are common during reviews: *"So the policies of the company states that the business is responsible for the access. So the ultimate decision maker is the business. However they failed because it's a human process right? It's eyeballing [and] sometimes the lists are large."* Such errors would be costly for organizations, both in

terms of leading to data breaches, and failing compliance reviews.

Exceptional Cases: In organizations, the validity of user-to-privilege assignments cannot be determined accurately only by knowing the user's job function. Users might need to fill in another employee's role for a period of time, or they might need temporarily access certain resources when they are on training. P6 explained a case where they thought they should remove existing access from a user because he asked for new access. They later realized the user is on training and still has his old job: *"The manager says no, he is training this person, as replacement, for three months."*

To design a tool that addresses the identified challenges, we will first analyze the usability of one of the existing tools, and then propose a set of design goals to address the challenges and usability problems we found.

6.5 Study 2: Evaluating an Existing Technology

Besides the field study and the survey, we analyzed the set of problems reported by the participants in the heuristic evaluation study presented in Chapter 4. The goals of this analysis were to triangulate field study findings with heuristic evaluation findings, and to find concrete set of problems in one of the existing access review interfaces.

Methodology: We provided the full details of the methodology in Chapter 4. Here, we provide a summary of our method. We used two sets of heuristic for the evaluation: Nielsen's heuristics by Nielsen and Molich (1990), and IT security management tools (ITSM) heuristics (see Chapter 4 for the list of ITSM heuristics). We recruited 28 participants with background in HCI and heuristic evaluation and asked them to evaluate the usability of an IAM system for performing 4 tasks, one of which was access review. We divided the participants into two groups, Nielsen and ITSM. Participants in Nielsen group used Nielsen's and participants in ITSM group used ITSM heuristics for evaluation. Each group was given a short recap on heuristic evaluation,

and an introduction to the heuristics that were assigned to them. We also gave them a five minutes introduction to the target IAM system. Then we asked participants to evaluate the system during a two hour period based on four provided scenarios. One of the scenarios was access review, which involved a security administrator requesting and setting a deadline for review, and a manager going through the list of users that need to be reviewed, reviewing their roles, and making certification or revocation decisions. In this section, we only focus on the problems reported in the fourth scenario of the heuristic evaluation study about the access review interface.

Results: The main access review interface evaluated by the participants is shown in Figure 6.9. We name this user interface "Search" throughout this chapter. After the study, the problems reported by individual evaluators were aggregated to form a complete list of problems. For the purpose of this study, we then chose those problems that were related to access review.

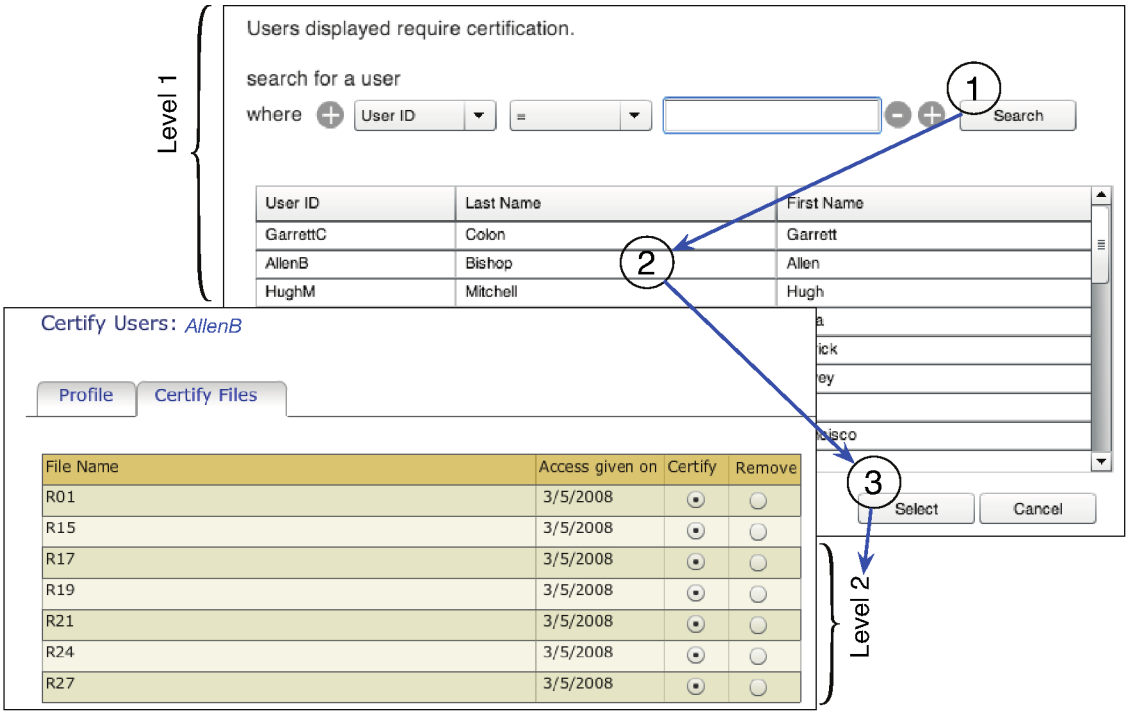


Figure 6.9: A screenshot of the Search interface. (1) Reviewer searches for a user. (2) Selects the users. (3) Clicks on the Select button and certifies or revokes access privileges in Level 2. A more detailed description of the Search interface is available in Appendix D.

Table 6.1: Reported problems with the Search interface during heuristic evaluation. The alphanumeric codes show evaluators who reported the problems. Evaluators with code starting with N used Nielsen's heuristics, and evaluators with codes starting with I used ITSM heuristics to find problems.

No.	Problem	Evaluators
1	It is hard to find users that need review and cycle through them	N10, N13, I8, I10, I12, I1
2	There is no way to certify users in batch	N4, N14, I9, I1
3	The list of each user's roles is not present in the search screen (first screen)	I1
4	There is no information on the history of roles like when and why they are assigned to the user	I13, I1
5	There is no information on which users in the search list is already certified	N1, N6, I10
6	There is no information about overall status of review in one place	I8, I11
7	The default action that is set to certify will result in errors.	I12
8	There is no information about the consequences of revoking a role	I14
9	The deadline for finishing review is not shown	N14, I12

A total number of nine usability problems were reported about the access review interface of the evaluated identity and access management tool. We presented the list of problems, and the evaluators who found them in Table 6.1. In summary, problems #1, and #2 referred to the lack of flexibility of the interface. Problems #3, #5, #6, and #7 was related to the lack of contextual information. Finally, problem #4 referred to the lack of history, and #7 to the lack of error prevention. We tried to address these problems when designing a new user interface for access review.

6.6 AuthzMap Design Goals

To design a new access review tool, we followed a design approach proposed by Kaptelinin et al. (1999), and used activity theory to guide the design process. Furthermore, we particularly asked the survey participants two questions:

- To rate (on a 5-point Likert scale) the usefulness of various pieces of information we found useful based on interview data. Participants' answers to this question are summarized in Figure 6.10, and will be used in clarifying design goals. This information is used to prioritize the organization of different information pieces on the interface.

- To rate (on a 5-point Likert scale) the risk associated with different cues during access review. Participants' answers are summarized in Figure 6.11. This information is used to particularly highlight the cues associated with risk on the interface.

Next, we present four main design goals. For each goal, we first present the theoretical support, and then we use the field study or survey data to describe how we applied theory to the design of an interface for access review.

Flexible support for review actions: The goal of access review is verifying access privileges. This goal can be broken down to lower level subgoals, and actions to satisfy those subgoals. These actions can include: viewing list of users and identifying them, identifying users' job function, checking the list of users' privileges, and certifying or de-certifying user-to-privileges assignments. To address the *Scale* challenge, a tool can help users perform the above actions more efficiently. This can be achieved by more flexible search and filtering mechanisms to view and identify users, and applying decisions in batch. According to Kaptelinin et al. (1999), technology should also support alternative ways to attain an activity goal. To achieve this, we present information at different levels of abstraction. The user can choose the right level of detail, based on his knowledge and

understanding of the access policy. For example, a user with the knowledge of the access policy can use a more abstract view, but a user who needs more information can use a more detailed view. This approach can address the *lack of technical knowledge* challenge.

Visibility of context: Activity theory emphasizes that tools and artifacts used during an activity are part of the context, and the technology should facilitate access to those artifacts, integrate them with each other, and present them in a way that reflects the spatial layout and temporal organization of the context. The context of an access review activity includes users, access privileges, and user-to-access privilege assignments. In addition, our interview and survey results show that the following artifacts are part of the context and can be used for making access review decisions:

(1) Job information: Our interview participants indicated that when users change their job or move between departments, their access changes. For example, P6 explains why job information can be important during access review: *"Now what happens is that we have a report that runs every single day and it tells me [if] people transfer [to another department] or change [their job]. [For example,] she gets a promotion. She went from warehouse manager to public relations manager. She will request something. I need a public relations manager role. My team goes automatically: 'why? That's not what you are. You are warehouse. No, I got a promotion, I'm this. Okay, we'll give you these three but you are losing those three."'* Additionally, 83% of our survey participants agreed with the importance of job information during access review (Figure 6.10). Providing job information help reviewers better understand why a user have certain access privileges, and therefore, address the *lack of knowledge* challenge.

(2) Other users' access: During access review, reviewers may need to review many users instead of one. These users have certain access privileges in common (e.g.,

basic access to the Internet, email, Sharepoint). For example, P1 explained that users who are doing the same job usually have similar access: *"... a manager who hired a new employee [and] who knew that you had the access that you needed to do the job for him or her would say, 'Oh, make this new employee's access just like yours.' And so then an employee would then inherit privileges based on the success of a previous employee in terms of doing that job."* Therefore, comparing access privileges of a user known to reviewer to that of an unknown users will facilitate sense making. In our survey, 52% of the participants agreed with the usefulness of other employees access during review (Figure 6.10). This will address *lack of knowledge*, and *scale* challenges and reduces *human errors*.

(3) Previous reviews: The reviewer can employ the past review decisions and replicate them in his review. Replication is particularly useful if none of the user's attributes has been changed since the last review. 48% of the participants agreed with the usefulness of previous reviews, and only 9% disagreed with it (Figure 6.10). Having access to and using past reviews can address *frequency*, and *scale* challenges, and reduces *human errors*.

(4) Other users involved in the activity: The process of provisioning users with access privileges is a collaborative activity between different stakeholders. Therefore, the interface should show who requested the access, who approved the request, and who executed the provisioning of access. (P12) explained that such information will help reviewer make an informed decision: *"So again, you think of the attestation process or even at any moment in time on a view user, we always talk about helping somebody make informed choice. So if I'm evaluating the correctness of an SAP account and I can look at when it was requested, who reviewed it, who approved it, when your last login time was, I can serve to make a pretty informed choice about why you have this or its level of appropriateness."* 70% of our survey

participants also agreed with the usefulness of the information about other involved stakeholders (Figure 6.10). This can address *lack of knowledge* of why a user has certain access privileges.

(5) Policy violations: Our survey results (Figure 6.11) show that SoD violations are the most important violation to be detected during access review. Therefore, they should be highlighted on the interface. This can address *scale*, and *lack of knowledge*.

Make history visible: According to Kaptelinin and Nardi (2006), analysis of the history of an activity can reveal the main factors influencing the development of the activity. Furthermore, Hollan et al. (2000) studied experts working in complex environments, and suggested historical information can be incorporated in cognitively important processes. For access review activity, historical information can help reviewers understand how the policy has evolved over time, and therefore make better decisions in uncertain scenarios. This would address the challenges of *scale*, and *exceptional cases*.

To incorporate history in the interface, we first identified which of the three access review artifacts (users, privileges, and user-to-privilege assignments) carry historical information. Interview data revealed that users, and user-to-privilege assignments (unlike privileges) change over time, and therefore, have historical information. For example, P6 explained that employees frequently change their job and access: *"[Employees' position] changes a lot when you start going through economic churns. So when you are laying-off 50 people at a time, 100 people another time, or department consolidations. I can tell you I've been in this role for two and a half years and I've seen five department consolidations in finance alone."* Also when we asked P4 about how frequently they make changes to the access privileges, she responded: *"We don't. I wouldn't say never - very rarely. If we were to add a new region, which I don't think there are any left to be added at this point."* Unlike our interpretation from interviews, only 39% of

survey participants agreed with the usefulness of privilege history, and only 36% agreed with the usefulness of job changes history (Figure 6.10). Despite these responses, we decided to include this goal in design of AuthzMap for the following reasons: (1) As survey participants have not seen such a feature in their existing tools, they may not be able to evaluate its usefulness. (2) Our goal was to correlate both pieces of information, which increases the usefulness of them. Our survey only asked about the usefulness of pieces separately. (3) Our survey results still suggested that only 24% and 18% of participants disagreed with the usefulness of privilege and job history. Therefore, this information can be incorporated in the interface, and tested in a user study to clarify the debate between participants.

Therefore, AuthzMap visualizes the history of users' job changes, and the history of user-to-privilege assignments, and correlates them with each other. Showing the history can addresses *frequency* challenge by showing previous decisions, and help with understanding of *exceptional cases*.

Knowledge sharing: According to Kaptelinin and Nardi (2006), technology should help in problem articulation and seeking help from colleagues. The interview participants indicated that reviewers hardly understand the meaning of the roles and access privileges. Therefore, our participants used the following strategies to mitigate the lack of knowledge:

(1) P18 talked about translation of technical terms to business related terms to help reviewers understand the meaning of roles: *"... and we get this huge profile - here's all the access the user has. We then have to translate that into more of an English format for the individual."*

(2) P7 described the use of communication channels to get help with certification decisions: *"The security coordinators take it to the [application owner] and explain what*

the risks are. They're the ones who do a kind of mini risk assessment say: OK, such and such business unit wants access to this data for such and such reason."

Therefore, one of the design goals in the proposed interface was to provide knowledge of each access privilege for the reviewers in the form of a description (74% of participants found it useful), criticality (86% of participants found it useful), and application associated with the privilege (89% participants found it useful). Moreover, communication channels (as previously shown in Figure 6.5) should be available in the interface to get help from other users with the knowledge of access privileges. Knowledge sharing would address the challenges of *lack of knowledge*, and can help with *exceptional cases*.

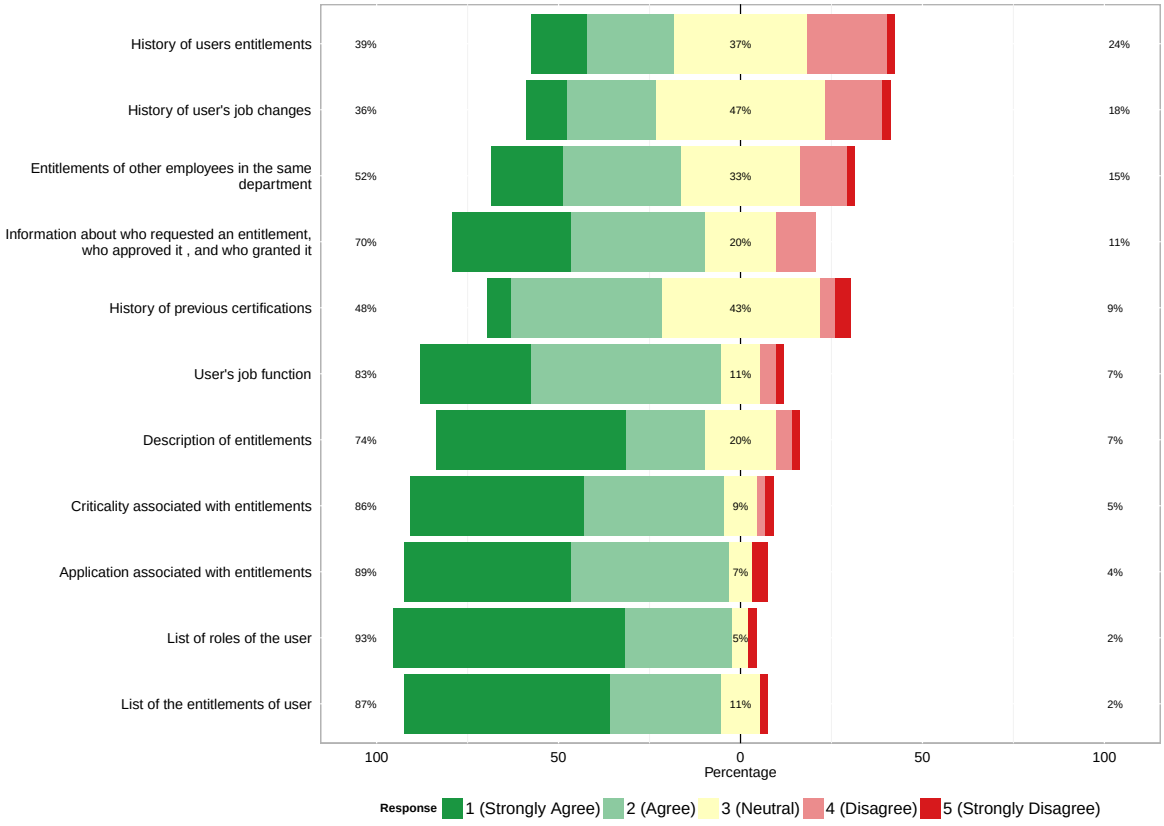


Figure 6.10: Usefulness of different pieces of contextual information during access review. Survey participants were asked to rate the usefulness on a 5-point likert scale.

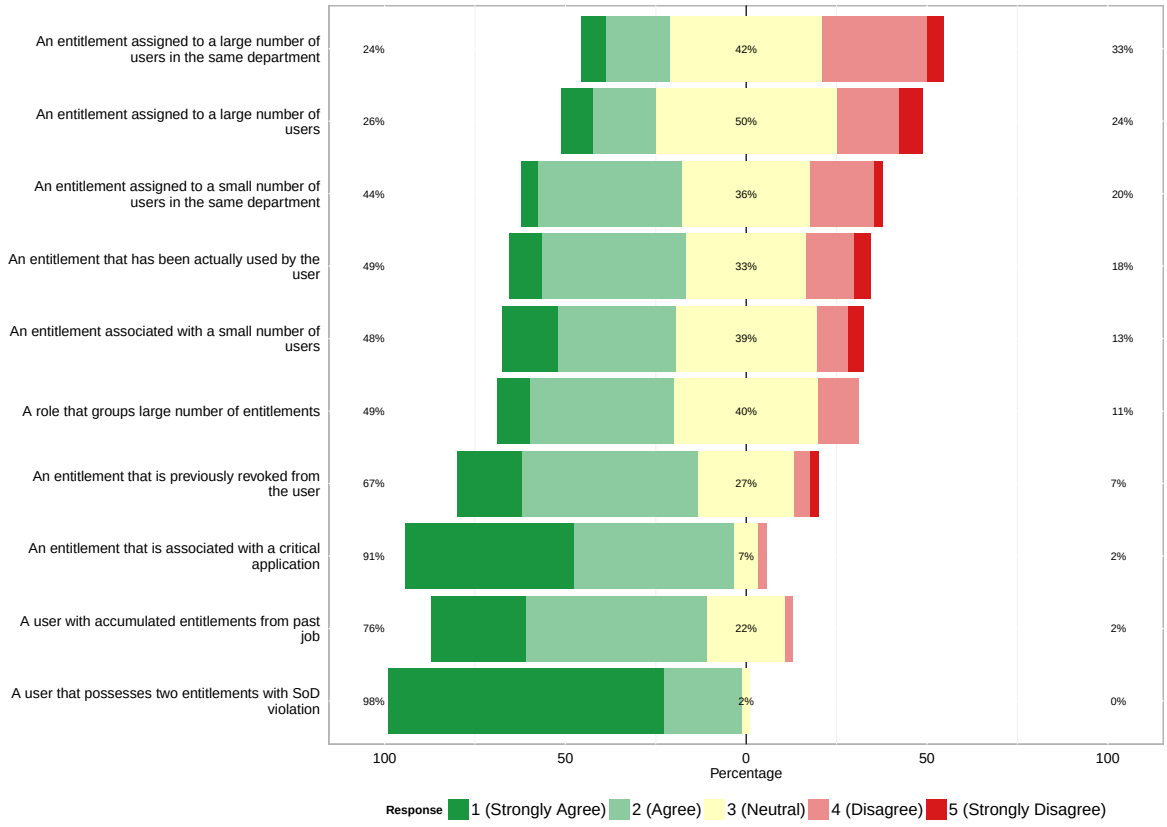


Figure 6.11: Risk indicators during access review. Survey participants were asked to rate the risk associated with each item on a 5-point likert scale.

We used the guidelines for designing ITSM tools (Chapter 3) to achieve each design goal. The mapping between guidelines and design goals is presented in Figure 6.12. Each row in the table corresponds to one of the guidelines, and each column corresponds to a design goal. The letters in each cell refer to the interface mechanism (labeled in Figure 6.13) used to realize each guideline. These interface mechanisms will be discussed in the next section. As the table shows, we elect not to use any of the guidelines related to configuration and deployment, as the access review did not involve any deployment related tasks.

AuthzMap Interface Design

To realize the goals discussed in the previous section, we designed a new interface and named it AuthzMap. We first built a low-fidelity prototype in Microsoft Visio (Appendix E), and

	Flexible support for review actions	Visibility of Job Changes	Visibility of other users' access	Visibility of previous reviews	Visibility of other users involved	Visibility of policy	Showing History of Access Privileges	Knowledge sharing
Make tools combinable		a						
Help task prioritization	b							
Provide customizability								
Use multiple levels of information abstraction	c		c					
Use different presentation / interaction methods	c		c				d	
Support knowledge sharing				e	f			g
Provide flexible reporting								
Provide an appropriate UI for stakeholders	j							
Provide communication integration								g
Facilitate archiving		a		e			k	
Support collaboration								
Work in a large workflow					f			
Provide customizable alerting						h		
Provide automatic detection								
Provide data correlation and filtering	i	a						
Make configuration manageable								
Support rehearsal and planning								
Make configuration easy to change								
Provide meaningful errors								

Figure 6.12: Mapping between ITSM guidelines, AuthzMap design goals, and the interface mechanisms used in Authzmap. Each non-empty cell indicates that the guideline in the corresponding row is used to achieve the design goal in the corresponding column. Letters refer to the actual interface mechanism (in Figure 6.13) used to realize the guideline.

improved it over multiple rounds of internal feedback. We then designed a medium-fidelity prototype (Appendix E) in Adobe Flash, and refined it by getting feedback from external usable security researchers, as well as our industrial partner in this project. Finally, we built a high fidelity prototype in Adobe Flash. It loads access control related data through XML files and allows the user to perform access review tasks. We depict the AuthzMap in Figure 6.13, and with more details in Appendix D.

The AuthzMap uses three levels of abstraction to integrate different contextual artifacts discussed in the previous section, and to present different views of the policy (Figure 6.13c). Level 1 shows users and access privileges in a grid that provides an overview of the overall review activity. The spatial layout of the interface was based on Lampson (1971) access matrix model. Users are sorted from top to bottom, based on the number of privileges they have. This allows reviewers to quickly identify users who have large number of privileges. Furthermore, reviewer can change the order of users or access privileges (Figure 6.13b) to compare users with similar job titles, or privileges that serve access to similar applications .

Authzmap includes a filtering functionality (Figure 6.13i) to filter users based on their name or job title. AuthzMap also provides batch certify accelerators (Figure 6.13j) to certify an access privilege for all users. Furthermore, we visualized the SoD violations between access privileges in the first level of the interface (Figure 6.13h) due to their importance. Reviewer can obtain the detailed access profile of a user using the second level of the interface (Figure 6.13c).

In Level 2, we integrated contextual information related to the user-to-access privilege assignments, the job changes of the user, and previous reviews. This level also uses a timeline metaphor (Figure 6.13d) to show the temporal relationship between the job changes (by integrating with an HR system) (Figure 6.13a), access privileges (Figure 6.13k), and previous reviews (Figure 6.13e). The reviewer can re-arrange the privileges based on the privilege name, active privileges, and the time the privilege is assigned to the user.

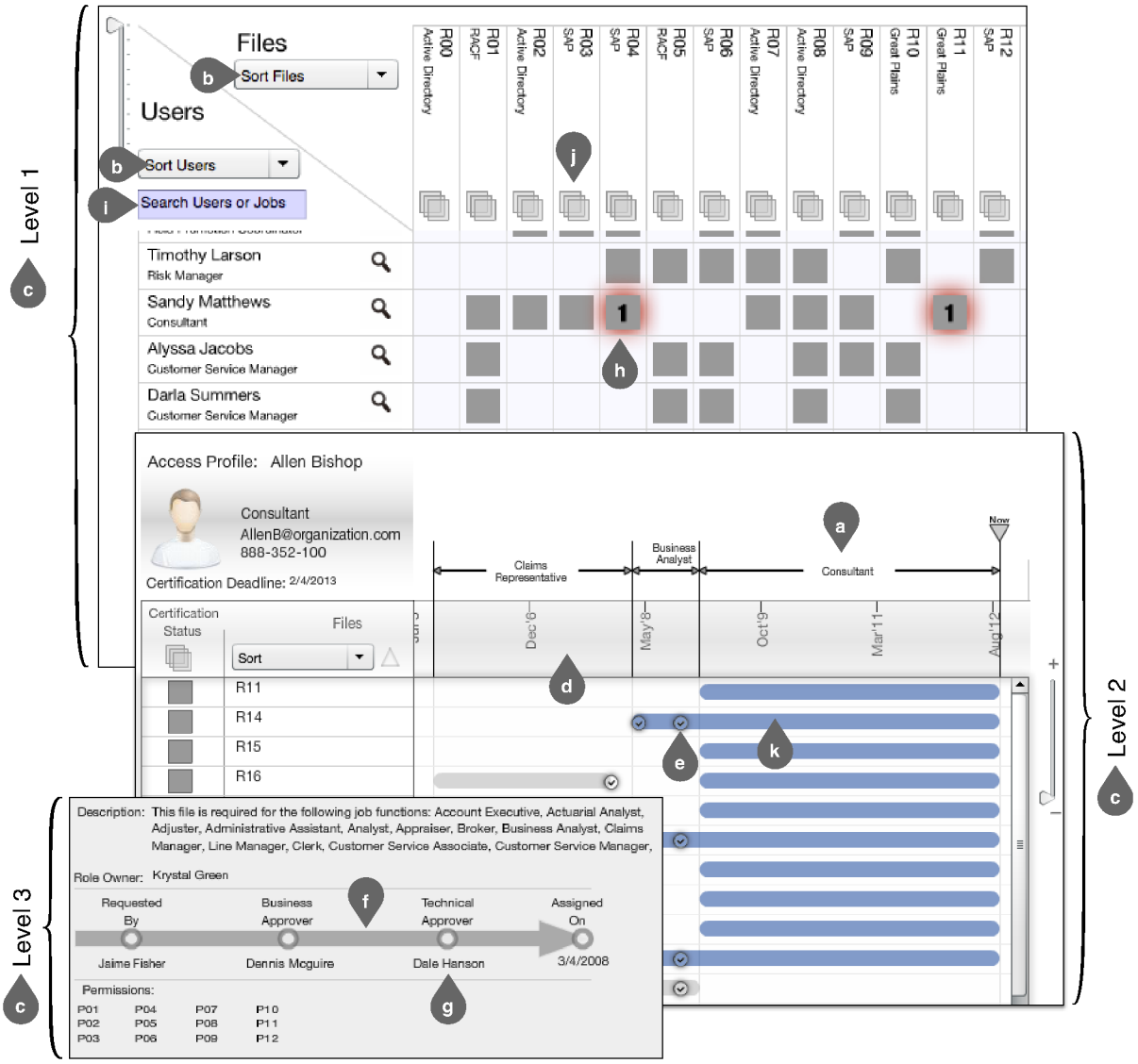


Figure 6.13: The three levels of the AuthzMap interface. The reviewer is presented with Level 1 of the interface. He can go into Levels 2 and 3 for making further sense of the accesses of the users. The access privileges are shown as files in this version of the interface for the purpose of a user study presented in Section 6.8. A more detailed description of the AuthzMap interface is available in Appendix D.

If the reviewer needs more details on one particular user-to-access privilege assignment, he can click on the access privilege bar to go to Level 3 of the interface, which shows the description of the access privilege, the privilege owner, the list of permissions if the access privilege was grouping a set of permissions, and the workflow (Figure 6.13f) through which the user obtained the access privilege. Level 3 allows the reviewer to learn about the meaning of the privilege. If the reviewer cannot articulate the meaning or the impact of the privilege using this information, he can use communication channels (Figure 6.13g) to seek help by clicking on the name of each stakeholder (e.g., owner, requester, approver, and implementer).

6.7 Study 3: Comparative Heuristic Evaluation

As a part of the usability engineering process for designing AuthzMap, we conducted a comparative heuristic evaluation of AuthzMap and two of the existing access review tools. The goal of the evaluation was twofold: (1) identify usability problems in AuthzMap (2) compare the number and severity of the problems between the three interfaces

6.7.1 Methodology

We designed a within-subjects study and asked HCI experts to perform heuristic evaluation on three interfaces. We chose within-subjects design (i.e., participants were asked to evaluate all of the three interfaces) to reduce the variance due to participant deposition, and to allow asking participants to compare the interfaces. To address the order effect, we performed counterbalancing by randomly assigning the order of the interfaces. Participants were asked to use two sets of heuristics for evaluation: Nielsen's heuristics, and ITSM heuristics (see Chapter 4 for details of the two sets of heuristics). The study session had three parts:

Training: At the beginning of the study, each participant was provided with the training material in the form of online slides with narratives. The training had three components: (1)

recap of heuristic evaluation method, (2) overview of the heuristics, and (3) overview of access review activity. We limited the training time to 15 minutes to avoid participants' fatigue.

Evaluation: After the training, we directed participants to the study website and asked them to fill out a background questionnaire to collect their demographics. Then we asked participants to start from the first interface, and spend 30 minutes for evaluation of each interface (90 minutes total evaluation time). During the evaluation, participants had access to an evaluation guide that described what they should do, a set of five example scenarios that they can perform on the interfaces, and the list of heuristics. During the inspection of the interfaces, participants were asked to report usability problems in an online form, and then identify the heuristic(s) that they used for finding each problem.

Post-Evaluation Feedback: After participants completed the evaluation of all three interfaces, we asked them to determine the severity of the problems they reported. The five-level scale recommended by Nielsen (2005b) was used for determining severity (0 = Not a usability problem at all, 1 = Need not be fixed unless extra time is available on project, 2 = Fixing this should be given low priority, 3 = Important to be fixed, 4 = Imperative to be fixed). Next, we asked participants to participate in a semi-structured interview. The goal of the interview was to further understand the positive and negative aspects of each interface and compare them with each other. The post-evaluation part took about 15 minutes.

6.7.2 Recruitment

We recruited 12 participants for the study and paid them $30 honorarium. The recruitment criteria for our study were having either academic and/or work experience in human-computer interaction. The recruitment notice was distributed over email to different mailing lists includ-

ing computer science and engineering graduate students at our university, other research groups that we have had collaboration with, user experience professionals association Facebook group, and a local user experience community. Those who responded to our notice were screened for their HCI background, and the eligible participants were invited to the study session. We organized in-person study sessions for local, and remote study sessions for distant participants. In terms of demographics, we had 8 females, and 4 males. The age of 3 participants was between 20 and 25, and the age of 9 participants was between 26 and 30. We had 11 participants with graduate degree, and one participant with undergraduate degree. All but one participant had professional or research experience in HCI (average years of experience was 0.83), and all but one had formal HCI training (university courses, tutorials, workshops). Four participants indicated that they had professional or research experience (average years of experience was 0.33) and formal training in the area of computer security.

6.7.3 Evaluated Interfaces

We compared the AuthzMap interface to two other interfaces, named Search (Figure 6.9) and List (Figure 6.14). The detailed description of each interface is provided in Appendix D. According to a market survey by Cser (2011), the List interface is known as one of the two access review market leaders, and Search as one of the two the strong performers. We choose not to reveal the actual names of Search and List interfaces as the purpose of the study is not to critique a particular commercial system, but rather compare three different approaches in the design of access review interfaces. The Search interface was first evaluated in Section 6.5. The Search interface does not reveal the context at all. The interface includes two levels. In the first level, a reviewer can search for users to obtain a list of users that match the search criteria. Then reviewer can select users one-by-one, see the profile of each user containing name, department, job, etc., and then review their access privileges. The List interface reveals certain contextual information such as the progress of reviewing individual users, the history of

previous reviews, and information about the individual users (such as their job and department) and the privileges (such as the date the user is assigned to a privilege, or the description of the privilege). But these contextual information pieces are not correlated with each other or immediately accessible to the reviewer. The List interface consists of two levels. In the first level, the list of all users and the number of privileges to be reviewed for each user is shown. Reviewer can select users one-by-one, and review the list of privileges in level two of the interface. In level two, certain contextual information such as the application that uses each privilege, the description, and the name and job title of the user under review is shown. Also, information about the review history, and rejection history on a particular privilege can be accessed through small icons. We chose to build a prototype of Search and List interfaces over using their full versions for two reasons. First, we wanted the three interfaces to be at the same level of granularity. Second, we did not have access to the installable version of the List interface. Third, prototyping allowed us to instrument the interfaces for our user study presented in Section 6.8.

6.7.4 Results

Table 6.2 shows the classification of the problems for each condition. The *"Problem Reports"* column shows the initial number of problems reported by the evaluators. The *"Tokens"* column shows the number of valid reported problem tokens after removing unknown problems, and false positives, and decomposing problems to their finest level of granularity. The *"Known"* column shows the number of problems after combining problems that are reported by multiple evaluators. That is, if a problem is found by multiple evaluators, we counted it as a single known problem. Table 6.2 also shows the classification of known problems based on their range of average severity.

First, we compare the number, severity, and evaluator consensus between the three interfaces. This will give us an indication of the usability of the interfaces in general. Table 6.2 shows that participants were able to report more problem tokens for Search and List interfaces compared

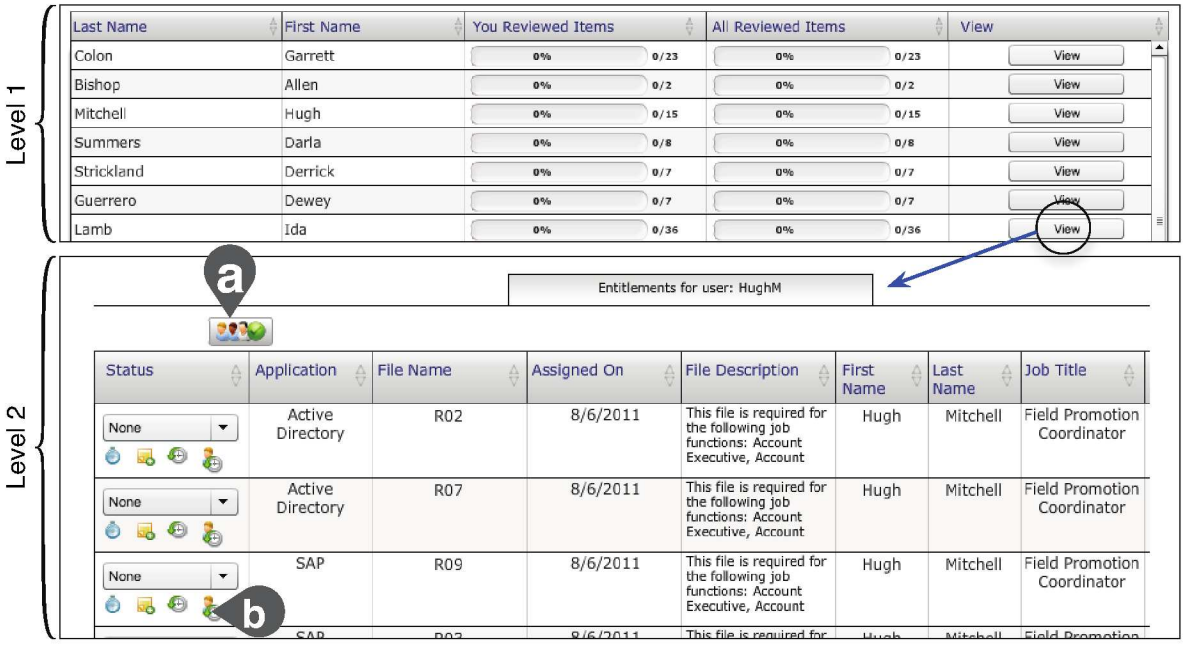


Figure 6.14: A screenshot of the List interface. Reviewer identifies the user and clicks on the View button. Reviewer is presented with the second level of the interface that includes the list of user's access privileges. The icon marked as (a) allows batch actions on privileges, and the four small icons (marked as b) do the following (from left to right): sets the access expiry time, writes notes for each privilege, shows history of actions on each privilege, and shows history of rejections for each privilege. A more detailed description of the List interface is available in Appendix D.

Table 6.2: Overview of the identified usability problems for each of the three evaluated interfaces

Interface	Problem Reports	Problem Tokens	Known Problems	Severity				FP	Unknown
				1-2	2-3	3-4	4		
AuthzMap	55	55	32	5	14	12	1	2	0
List	83	83	37	2	18	14	3	4	2
Search	105	107	33	3	9	16	5	3	1

to AuthzMap. Although, after combining similar problems, AuthzMap and Search had 32 and 33 unique usability problems while List had slightly more (37). In terms of classification of known problems into different severity categories, AuthzMap had only 1 problem with average severity of 4 while List and Search interfaces had 3, and 5 problems with the highest level of severity. Figure 6.15 shed light on the relationship between identified problem tokens, known problems, and the evaluators who found them. The figure shows that for AuthzMap, only 10

and 5 out of 32 problems were found by more than one and two evaluators. On the other hand, for List 18 and 11 out of 37 problems found by more than one and two evaluators. For Search, 20 and 16 out of 32 were found by more than one and two evaluators. This suggests that the evaluators consensus on Search interface was the highest followed by the List interface.

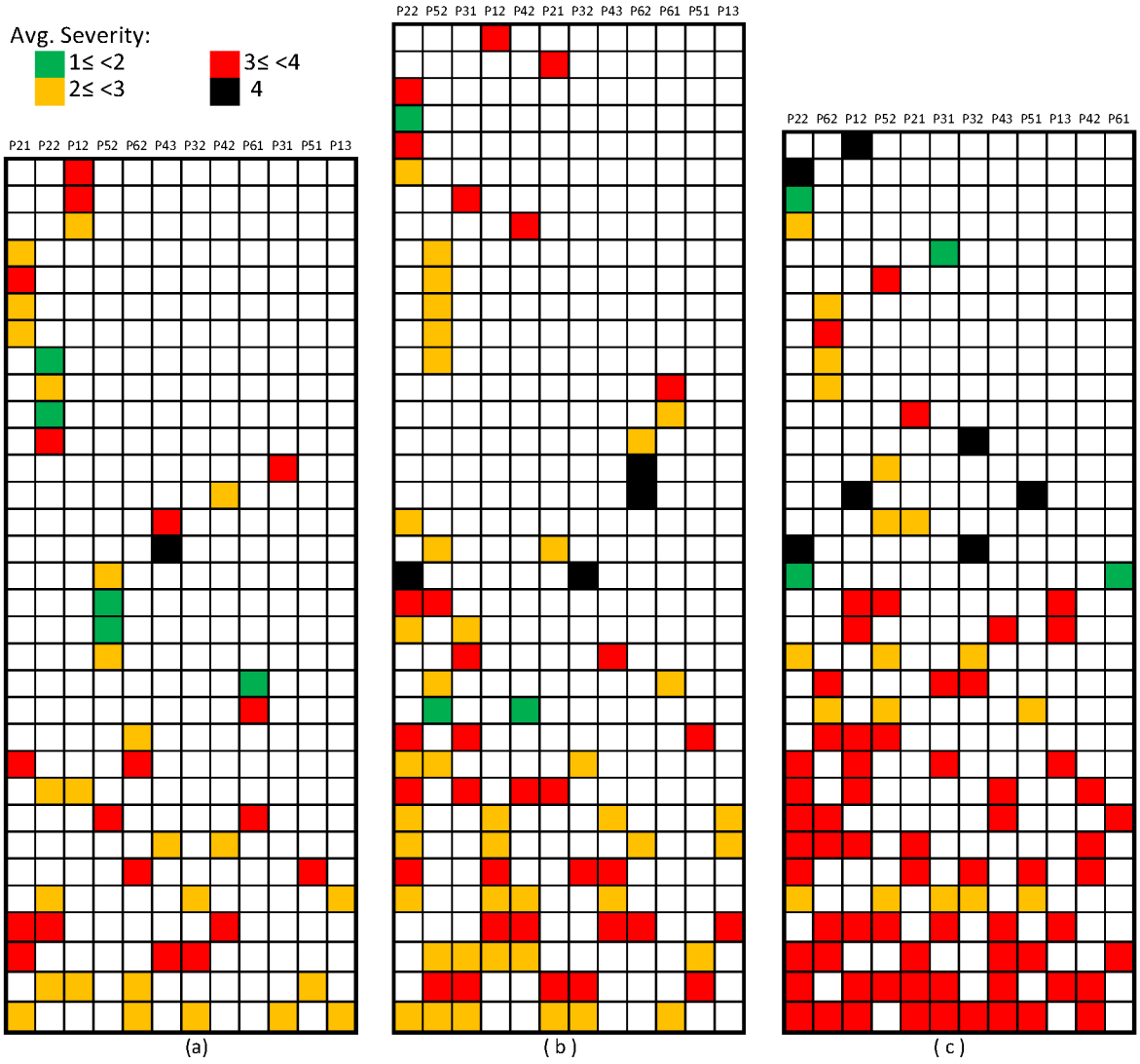


Figure 6.15: Problems identified in (a) AuthzMap, (b) List, and (c) Search. Each row in the grids indicates a known problem, and each column represents an evaluator (there are no overlapping problems between grids). Each cell shows a problem token, color coded by severity. The known problems are sorted from top to bottom based on the number of evaluators who found them. The evaluators are sorted from left to right based on the number of problems they found.

We also tested the following hypothesis to compare the number and severity of the problems

in each interface: (1) H_1: The individuals randomly selected from the population we drew our participant pool from will report more problems for List and Search interfaces than AuthzMap. H_0: There is no difference in the number of reported problems. The result of a repeated measures ANOVA test shows that there is a significant difference between the three interfaces (p=0.0056). Pairwise comparison shows that statistically significant difference exists between AuthzMap and Search (p=0.0059). (2) H_1: The average severity of the problems reported by individuals randomly selected from the population we drew our participant pool from will be higher if those individuals evaluate the List and Search interfaces. H_0: There is no difference in the average severity. The result of a repeated measures ANOVA test shows that there is a significant difference between the three interfaces (p=0.0103). Pairwise comparison shows that statistically significant difference exists between AuthzMap and Search (p=0.019).

Comparison of problems based on heuristics: We showed that AuthzMap contained fewer and less severe problems compared to List and Search, and we showed that the difference between AuthzMap and Search was statistically significant. Also, we showed that there was less consensus between evaluators on the AuthzMap problems. To understand what type of problems caused this difference, we analyzed problems found by each heuristic for each interface. This will help us understand if the design decisions we made caused these differences between the interfaces. The overview of this analysis is shown in Figure 6.16.

Using most of the heuristics, our participants reported fewer problems for AuthzMap compared to the the two other interfaces. Figure 6.16a shows a large gap between the number of reported problems using heuristics: ITSM1, and ITSM3. Heuristic ITSM1 refers to visibility of activity status, and the difference in number of reported problems shows that our participants found that the AuthzMap is doing a better job in revealing context. Heuristic ITSM3 refers to flexibility in representing information. The graph confirms that our participants found the use of different visualizations for representing access control data helpful.

Figure 6.16b shows that there is a large gap between the number of problems reported for

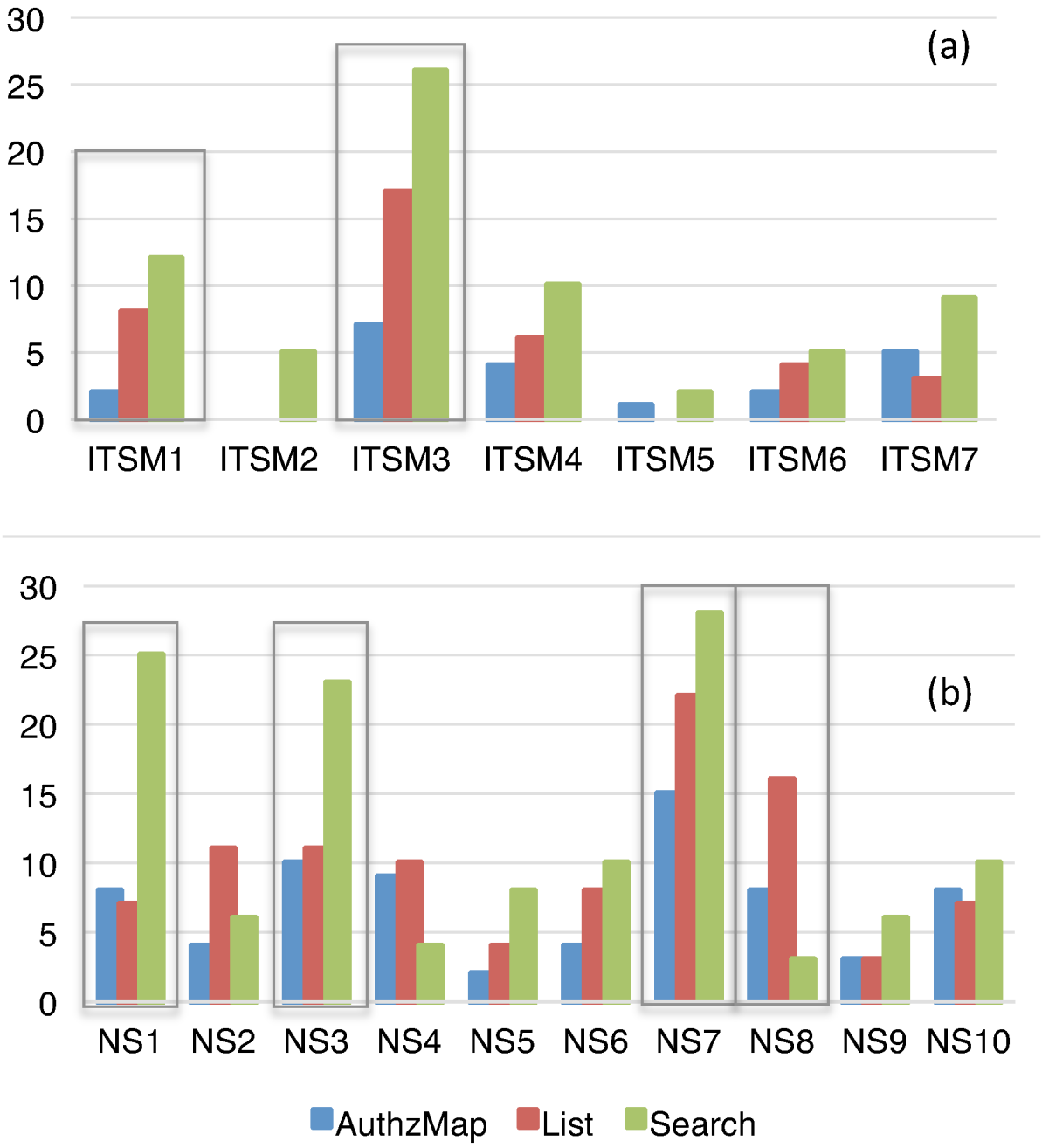


Figure 6.16: Number of problems per heuristics. The horizontal axis shows lists of ITSM heuristics in graph (a) and Nielsen's heuristics in graph (b). The vertical axis shows the number of raw problems that were associated to each heuristic by evaluators.

Search and the two other interfaces when participants used NS1 and NS3. Since the Search interface did not provide visibility of all users and the details of their roles, the evaluators reported large number of problems using NS1 (visibility of system status). Also, the Search interface did not allow changing the access review decisions after the user submitted the decision. Therefore, many participants reported problems related to NS3 (recovery from errors). Figure6.16b also shows a large gap between problems reported for the three interfaces using heuristics NS7 and NS8. Heuristic NS7 refers to flexibility and efficiency of use, and AuthzMap contained fewest problems in this category as it used different levels of abstraction, and allowed using accelerators for performing the tasks. The difference in the number of problems related to NS8, aesthetic and minimalist design, was also interesting. The Search interface, which provided a minimal interface with as few features as possible, contained fewest problems related to minimalism. AuthzMap ranked between Search and List and List contained most of the problems.

While the exhaustive list of 102 reported problems is beyond the scope of this chapter, we will discuss the top 5 most reported problems for each interface. For AuthzMap, top 5 reported problems were: (1) lack of visibility of role information in Level 1, (2) lack of batch certifying all roles of a user at Level 1, (3) lack of a confirmation message when user make review decisions in Level 1, (4) Lack of searching and filtering users, (5) lack of help and documentation. For List, the top reported problems were: (1) unclear icons, (2) lack of an immediate history, (3) lack of ability to organizing users by job title, (4) lack of ability to certify one role for multiple users, and (5) lack of ability to compare different users. For Search interface, the top reported problem were (1) lack of ability to verify the review decisions, (2) lack of ability to find users with same job title, (3) lack of undo for review decisions, (4) lack of ability to certify a role for multiple users, (5) lack of visibility of role details.

After participants performed the evaluation, we asked them to rank the three interfaces based on their overall experience. All but two of the participants ranked the AuthzMap as the best

followed by the List and Search interfaces. One participant ranked Search as the best while she had reported more problems for it than the other two interfaces. One participant ranked List as the best interface because she was familiar with the List based design, but found Grid based interface confusing. We also asked participants about the most difficult interface in terms of evaluation. Seven participants chose AuthzMap. They said the better design of the interface made it more difficult to find problems. The rest of the participants chose Search interface and they said it was so basic that they did not know whether they are doing anything wrong or not.

6.7.5 Discussion

Our results show that the AuthzMap contains fewer and less server problems compared to the other two interfaces. We also analyzed the categories in which the number of problems was different between three interfaces. We showed that the number of problems related to visibility and flexibility were different between interfaces. On the other hand, we did not see a difference between problems related to knowledge sharing. Multiple factors might lead to this observation. First, the participants might not realize the importance of knowledge sharing or use of communication channels to get help from other colleagues. Therefore, they reported very few problems related to knowledge sharing. Second, looking at the few reported problems in this category suggest that the use of communication channels was confusing for the participants, and some of them did not notice the existence of such feature in AuthzMap due to lack of an affordance. This problem can be addressed by providing a clear affordance that suggest clicking on different stakeholder names will allow e-mail or phone communications with them.

The results of this heuristic evaluation motivated us to continue working on AuthzMap, and prepare for a user study of the interface. We fixed a number of issues reported during the heuristic evaluation including: (1) adding the ability to search for users and jobs, (2) Adding further information about roles at the Level 1, (3) Adding a confirmation message for submitting the review decisions at the Level 1, (4) Ability to sort roles. We also elect not to implement

some of the suggested improvements such as (1) Adding ability to approve all the roles for a single user at level 1, as we believed this could lead reviewers to approve all access privileges for one user without explicitly looking at them. (2) Help and documentation, as the proposed interface was a prototype rather than a finished product. (3) Ability to move users in the grid to facilitate comparison of two or more users, as it was not technically feasible. (4) Improvements to the communication channels including better affordances, as it was out of the scope of our future lab study plan.

The results of this study also have broader implications for usability evaluation of systems that are hard to evaluate in the lab of field studies. It shows that discount usability evaluation methods can be used to check whether the design achieved the intended goals, before investing in performing costly studies. This is particularly useful in contexts like IT security in which recruiting participants and designing an ecologically valid study is challenging.

6.8 Study 4: Evaluation of AuthzMap

After the Study #3, we further improved AuthzMap by addressing as many identified problems as we could. Then we conducted a lab study to compare AuthzMap to Search and List. The goal of our evaluation was to test if AuthzMap is more usable than the two existing systems. Nielsen (2012) defines usability by five quality components: (1) Learnability, (2) Efficiency, (3) Memorability, (4) Errors, and (5) Satisfaction. In this study, we identify efficiency and errors as two main usability goals of the interface, as they are directly related to the challenges described in Section 6.4.3. At the end of the study, we also collect data about subjective satisfaction of the participants.

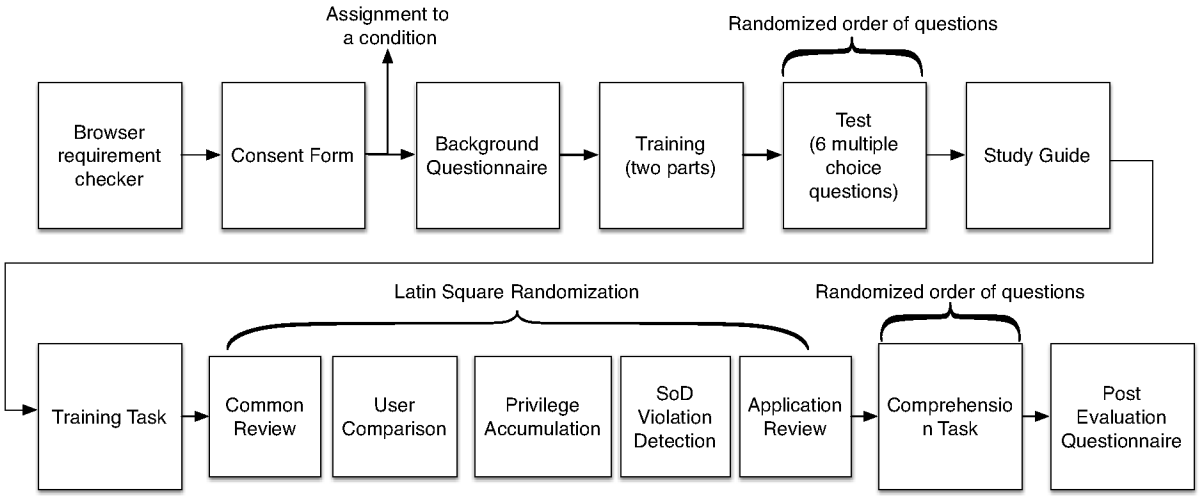


Figure 6.17: An overview of the comparative user-study protocol. Participants had to complete each step in the provided order, except the five study tasks that were randomized using a Latin square technique.

6.8.1 Evaluation Methodology

To evaluate three interfaces, we designed a between-subjects study with 3 conditions (one condition per interface). We asked participants of each condition to perform seven tasks. For each task, the interface was the independent variable, and we measured the following dependent variables: (1) efficiency, by recording time to completion (TTC), and (2) accuracy, by recording correctness of the critical components of the task. The overview of the study protocol is shown in Figure 6.17, and the screenshots of the study material are shown in Appendix F.

Evaluated Interfaces

We compared the AuthzMap interface to two other interfaces, named Search (Figure 6.9) and List (Figure 6.14). The details of each interface is given in Section 6.7.3. We instrument all three interfaces to allow collection of efficiency and accuracy data.

Participants

We used Amazon Mechanical Turk (MTurk) for recruitment, and gave each participant a $2 honorarium for their participation. MTurk has been used as a user study platform for HCI (see (Kittur et al., 2008)) and usable security research (see Wang et al. (2011)). Participants were asked to play the role of managers responsible for access review. Because we did not specifically recruit managers, we used an approach similar to the one by Convertino et al. (2011), to provide participants with the beliefs and knowledge of managers. Using our interview data, we first determined managers' level of computer security, review tool, and organizational knowledge. Interviews showed that managers do not have an extensive computer security knowledge, but they understand the concept of access review, and they know the steps for performing it. In addition, managers are trained on using the access review tool (i.e., they are not the first time users of a novel tool). We also assume they are not daily users of the tool (they use it four to two times a year or on an ad-hoc basis). To help participants have similar level of knowledge, we trained them on the basics of access review, and the use of tool to perform reviews (see Section 6.8.1 for the details of our training procedure). We further allowed them to explore the tool and familiarize themselves with it.

Training Material

We designed training material to ensure participants understood the concept of access review, and could apply that understanding using the system. The participants were given a brief training on access control and access review. We followed the recommendations from previous research on designing training materials:

Brief, up to the tasks: According to Carroll et al. (1987), users will learn tools faster when the training focuses on performing the task rather than understanding the rationale behind the task. We avoided training users on details of role-based access control, and concepts such

as roles, and entitlements. Instead of using the notion of roles, entitlements, or access privileges, we used the notion of access to files. Previous studies (Beckerle and Martucci, 2013; Reeder et al., 2008, 2011) show that participants can understand the meaning of file access control, and they are able to comprehend file access control policies.

Use of examples: We used examples throughout the training to explain the access review concepts. We also provided instances of how the interface can be used in interpretation of users' accesses.

Use of text-based material: We initially built a set of online videos to train participants. After piloting of the training, we switched to text-based material. Forget et al. (2012) show that online participants can do better with short textual instructions, rather than videos, or demos as it gives participants the opportunity to easily revisit the training during the study.

Use of multi-staged training Carroll et al. (1987) suggested to avoid overloading users with training material, and to help them start working on tasks as soon as possible. Therefore, we only taught them the basic access review concepts during the training. Task specific topics such as separation of duties (SoD) violations, privilege accumulation, etc. were taught as parts of the scenarios.

After the training, participants were asked to complete a test to check if they have the required knowledge to do the tasks. We tested the understanding of access control and access review using six multiple choice questions. According to Considine et al. (2005), multiple choice questions are a reliable and objective way to assess the outcome of the learning, while the answers can be checked automatically. We used standard techniques for designing multiple choice questions suggested by Considine et al. (2005), and piloted them to ensure their effectiveness.

Study Material

Actual users of access review tools also posses the organizational and contextual knowledge that our participants lacked. For example, a manager may have an understanding of the consequences of having access to a resource, or awareness of the access privileges for doing certain job. Such knowledge is context dependent, that is, we cannot have a clear assumption that a manager always has or lacks such understanding. In the study tasks, we simulated both situations where reviewer has or does not have contextual knowledge and provided participants with documents and material as external knowledge sources (similar to Convertino et al. (2011)).

We presented participants with three documents: *file catalog*, *application catalog*, and *SoD catalog*. The file catalog showed the list of files that each job function was allowed to access. The interview participants talked about entitlement catalogs, which we changed to the file catalog for the purpose of this study. According to P7: *"One of the things we have been doing is also building a catalog of access requests that people can make [based on their job]."* The application catalog listed all the applications and their files (entitlements). According to P3, they kept the track of this information in a knowledge base: *"our access procedures state that every application that has any level of criticality is supposed to have a published knowledge-based document in our service desk knowledge base that defines what the application is ..."* The SoD catalog showed pairs of files that caused SoD violations. P12 said they document these rules: *"And again whether they be SOD policies that say you can't have A if you have B or what we call 'restricted access' policies that say you can't have entitlement X if you are not cost center Y or division X or whatever the rule is, the ability to define that rule it lives with the entitlement in the resource catalog."* (P12)

Norman (1988) describes that people can rely on *knowledge in the world*, *knowledge in the head* or a combination of both in their activities. To determine the validity of a user's access, reviewers may completely rely on the above documents (knowledge in the world), they may

completely rely on their own knowledge (knowledge in the head), or they may use a combination of both. The lab study participants did not have the knowledge in the head of the hypothetical organization. Therefore, all the required knowledge for performing the tasks was included as knowledge in the world in the form of provided materials during tasks.

Study Tasks

After completing the training and the training test, participants were asked to perform seven tasks. We aimed to design tasks with three characteristics suggested by McCloskey (2014): (1) Realistic; (2) Actionable; (3) Avoid Clues or Steps. In order to achieve realism, we designed the tasks based on interview data and a survey we previously performed (see Section 6.4.1 for survey details). Tasks #2 and #3 simulate conditions where the manager knows which access privileges are appropriate for users, and only needs to identify users, and certify or revoke the privileges. Tasks #4 to #6 simulate scenarios where the manager tries to detect access privileges with high risk. To further understand what type of access privileges are risky, we conducted a survey and asked participants to rate the risk associated with certain types of access privileges. We chose to use the top three, which were SoD violations, accumulated privileges, and access privileges to critical applications, and used them to design tasks #4 to #6. The task #7 was a combination of previous scenarios to simulate a more uncertain and complex situation.

Training Task: The goal of this task was to familiarize participants with the interface. As we described in Section 6.8.1, managers will be familiar with their access review tool. This task gave participants an opportunity to perform a guided exploration of the interface, and understand how they can find pieces of information required in the upcoming study tasks. Participants were given the following scenario: "You are asked to identify the following information about "Clay Warren" : (1) his current job title, (2) list of files he has access to, (3) his previous job title, (4) the date of the last access review performed on the user." They were expected to select the correct answer to questions #1, #3, and #4 from seven possible options (including "I

do not know"). They should also type the answer to question #2 in a text box.

Common Review: (P1) explained that a common access review scenario is when a manager reviews one user: *"[Manager says:] what access does Jim have? I'd like to review Jim's access because he's changing roles within my department, there's no official job posting but I'm doing a realignment and I would like to review Jim's access."* Therefore, participants were given the following task: "You are asked to review the files *Timothy Larson* has access to. Check the user's access to files, certify the access to those files the user requires to perform his job and revoke those he does not require. Feel free to use the *File Catalog* in the top menu to find the list of files required for performing each job." In this scenario we made an assumption that the manager can determine the correct set of access for users. This is simulated by providing participants with access to a *File Catalog* that shows what files are required for performing each job. Participants are expected to revoke access to two files that are not necessary for Timothy Larson's job.

User comparison: P2 explained that similarity between users with the same job is used to detect excessive and unnecessary access: *"if you've got a group of 15 case managers and you bring them into the system, it'll say: ok, 12 out the 15 have 80% of access in common and these two people only have 20%. [...] oh this person has access they should not have, that has been carried over from somewhere else."* To simulate this scenario, participants were given the following task: "In this task you need to certify the access of three users. The certification is only limited to employees with *Loss Control Consultant/Specialist* job function. Identify such users, certify the files that users require to perform their job and revoke the access to the files they do not need. The catalog of jobs, and the required files to perform each job will be provided." In this scenario, we made an assumption that the manager can determine the set of access privileges required for the job and therefore provided participants with *file catalog*. In this task, there were three users with the "Loss Control Consultant/Specialist" job, and one of them had an unnecessary access to a file. Participants were expected to revoke the access to

that file.

Privilege Accumulation: Many of our interview participants discussed the privilege accumulation problem in large companies. For example, P6 explained: *"I was warehouse worker, I became public relations. They would request the public relations roles, nobody would take away the other ones and you would wind up with somebody having 50 roles."* Therefore, this task evaluated the interface in finding and resolving accumulated privileges. We gave the participants the following scenario: "Assume you do not know the list of files required for performing each job. In this case, you need to evaluate each user's access to files based on the following rule: *If the user changes job, he should not keep any access from his previous job. Any access that is kept from a previous job should be revoked.* Please review the following users, and revoke invalid accesses according to the above rule: (1) Derrick Strickland, (2) Lynda Robertson." The two target users had two and one permission accumulated from their past job, and participants were expected to revoke those permissions.

SoD Violation Detection: P6 described SoD violations as one of the highest access related risks. He described a case that someone is moving from accounts receivable (AR) to accounts payable (AP), and access to AR and AP systems causes SoD violations: *"So you are going from - you are the AP person, you are going to AR and your AP person needs to be trained [by you] – your replacement. Then we don't like it and it becomes very problematic and we usually want lots and lots of controls if you want the person to have the access."* Therefore, the goal of this task was to evaluate the proposed interface in the detection of SoD violations. We gave the participants the following scenario: "Sometimes a user should not have access to two specific files at the same time. For example, a user can have access to either file A or B but not both, at the same time. This rule is called *Separation of Duties (SoD)*, and having access to those files at the same time is called an *SoD violation*. In this scenario, you are asked to review the files of two users, and detect and eliminate SoD violations. To do so, you should first identify the two files that cause SoD violations, and remove access to one of the files to eliminate the violation.

Please check the following users for SoD violations: (1) Ida Lamb, (2) Maryann Weaver." In this task, each of the users had access to two files that caused SoD violation, and participants were expected to revoke access to one of the files causing SoD violation.

Application Review: P3 noted that they sometimes prioritize the access review according to applications. Critical applications are reviewed first, and in some instances non-critical applications are excluded from the review: *"They run a process which goes out to a subset of all those applications - the ones that we call critical which is SOX applications plus other [...] It goes out and it collects from these 80 or so applications what the access lists are, what the right are, it creates a report, we put it in a service desk ticket. Then it goes out to the [reviewers] and they review it."* In this task, we evaluated interfaces for application specific reviews. We gave the participants the following scenario: "The company uses four applications for running the business: *Active Directory*, *Great Plains*, *RACF*, and *SAP*. Each of these applications uses a subset of the available files. You are asked to review the following users, and revoke access to the files related to the *Great Plains* application: Edmund Johnston, Nelson Murphy, Jane Hoffman, Olive Morris." The four users in the scenario had access to 27, 21, 15, 2 files respectively, out of which 7, 5, 3, and 0 files were related to "Great Plains". Participants were expected to revoke access related to the *Great Plains* application.

Comprehension Task: In the previous tasks, we evaluated interfaces for specific scenarios, and told participants to look for a specific situation. In reality, reviewers may deal with a combination of various scenarios and need to integrate various cues to make decisions. This task aimed to evaluate the interface for situations where reviewer needs to evaluate the risk of particular access in the presence or absence of various indicators of risk and safety. We gave the participants the following scenario: "You are provided with a list of users and their accesses, and you are asked to determine how risky access to each file is. Use the knowledge you gained during the previous tasks to determine the risk associated with each file: (1) Francisco Lee, Director, R06; (2) Marcella Owens, Claims Manager, R02; (3) Margaret Estrada, Customer

Service Associate, R11; (4) Alyssa Jacobs, Customer Service Manager, R09"

For each of the four user/file pairs, participants were asked to rate the risk associated with the user having access to the file using a five point likert scale (1= Very Safe, 5 = Very Risky). The order of the four likert scale questions was randomized. Four user/file pairs had different levels of risk associated with them: (1) *Marcella Owens, R02*: Access to R02 caused a separation of duties violation with R44. We expected the participants to rate the risk at 5 (High risk). (2) *Francisco Lee, R06*: Access to R06 was given to the user during his previous job. Also there was no previous review of the user's access. On the other hand, there was another user with the "Director" job title who also had access to R06. We expected participants to rate the risk at 2, 3, or 4, as this access was associated with both indicators of risk and safety. (3) *Margaret Estrada, R11*: Access to R11 was given to the user as part of her current job, the access was certified twice during past reviews, and the two other users with the same job as Margaret had the same access. We expected participants to rate the risk at 1 (High safety). (4) *Alyssa Jacobs, R09:* Access to R09 was revoked from the user during a previous review, but the user gained access again after a while. Furthermore, other users with the same job did not have access to R09. We expect participants to rate the risk at 5 (High risk).

Exit Questionnaire

After participants finished the tasks, we collected their subjective satisfaction. The participants were asked to rate their agreement with the following statements on a five-point likert scale (1 = Strongly Agree, and 5 = Strongly Disagree): (1) The tasks were easy to perform using the provided system. (2) It was easy to find the required information for performing the tasks. (3) I feel that I performed the tasks correctly. (4) The access review tool helped me understand who has access to what. (5) I feel that I can easily find access violations using the access review tool. (6) I find the user interface of the access review tool intuitive. (7) Use of the provided tool for access review was engaging. We also asked participants to provide feedback about the

access review tool in a textual format.

6.8.2 Analysis

The goal of our analysis is to compare the three tested interfaces in terms of efficiency, accuracy, and subjective satisfaction.

Efficiency: We used time-to-completion (TTC) as a metric for efficiency. To capture TTC, we automatically logged the time users spent between starting and finishing each task. Then for each task, we tested the following null hypothesis: H_o: There is no difference between the median time to completion when using any of the three interfaces. H_1: There is a difference between time to completions. We used Kruskal-Wallis test, which is a non-parametric alternative to ANOVA, since we found that the time to completion was not normally distributed, and we could not normalize the distribution using transformation. Whenever we rejected the null hypothesis, we used pairwise Wilcox test with Bonferroni adjustment to test the following three null hypotheses: (A=L) There is no difference between AuthzMap and List. (A=S) There is no difference between AuthzMap and Search. (L=S) There is no difference between List and Search. For each test, we report the p value and the effect size (r). We also discuss the practical significance of the difference between AuthzMap and the other interfaces by showing the percentage of improvement or declination of median TTC over the other interfaces.

Accuracy: We identified those critical components of each task in which participants can commit dangerous errors. An error is dangerous if it puts the system in insecure state (i.e., leaves user with excessive privileges). For each critical component, we calculated the total number of participants who did and did not commit the error. Then we tested the following null hypotheses: (1) (A=L) There is no difference between the correctness of answers of AuthzMap and List participants. (2) (A=S) There is no difference between the correctness of answers of AuthzMap and Search participants. We used two-sided Fisher's exact test with Bonferroni

adjustment to test the above hypotheses.

Subjective Satisfaction: We performed pairwise comparison of participants' ratings using Mann-Whitney-U tests with bonferroni correction to test the following hypotheses: (A=L) There is no difference in ratings of participants in AuthzMap and List conditions. (A=S) There is no difference in ratings of participants in AuthzMap and Search conditions.

6.8.3 Results

In this section, we present the results of our data analysis. First, we provide a summary of participants' demographics and experience. Then we present the findings of the study. In this section, we use abbreviated condition names when presenting the results (A = AuthzMap, L = List, S = Search). Table 6.3 shows the number of participants who consented to the study, attempted the study, finished the study (received a return code for compensation), and those who provided valid results. If participants clicked on the consent form, we counted them as a consented participant. If a participant at least started the background questionnaire, we counted them as an attempted participant. If a participant completed all of the stages of the study, we counted them as a finished participant. Some of the finished participants skimmed through the study (our system recorded their time to completion for certain tasks at 0 seconds), or intentionally or unintentionally bypassed our system in order to get to the finish page without completing all of the tasks. We eliminated these participants, and reduced the pool of participants to a set of valid participants. We made use of data from 430 valid participants in this section.

We also tested the following null hypothesis: H_o: The validity of participants is independent from the interface they were using. To test this hypothesis, we divided the attempted participants in each condition into two groups: those who were valid participants, and those who were not valid participants. Our chi-square test revealed that the validity of the participants depends

on the interface ($\chi^2(2, N = 1030) = 20.424, p = 3.7e-05, Cramer's V = 0.141$). However the effect size is small.

Table 6.3: Classification of participants according to their progress in the study. "Consented" indicates those participants who consented to the study. "Started" indicates those participants who at least started the background questionnaire. "Finished" indicates those participants who completed all of the study steps. "Valid" are those participants that we used their data for analysis.

	AuthzMap	List	Search	Total
Consented	355	355	354	1064
Started	341	341	350	1032
Finished	190	156	151	497
Valid	174	135	121	430

We show the total time needed to complete the entire study for the valid participants in Figure 6.18. We tested the following null hypothesis for the time to completion of the study: H_o: The choice of the interface does not impact the total time needed for completion of the study. A Kruskal-Wallis test revealed a significant effect of interface on the time to completion of the study ($\chi^2(2) = 48.033, p = 3.7e-11$). A post-hoc test using Mann-Whitney tests with Bonferroni correction showed significant differences between AuthzMap and List ($p = 4.8e-10, r = 0.31$) and between AuthzMap and Search ($p = 6.4e-07, r = 0.25$).

Participants Demographics

In the beginning of the study participants were asked to answer the background questionnaire. We show the overview of the participants' responses in Tables 6.4.

Training

Participants were asked to complete the post-training test before proceeding to the study tasks. We summarized the number of attempts to complete the test in Figure 6.19. The results showed that nearly half of the participants in each condition could pass the test in the first attempt.

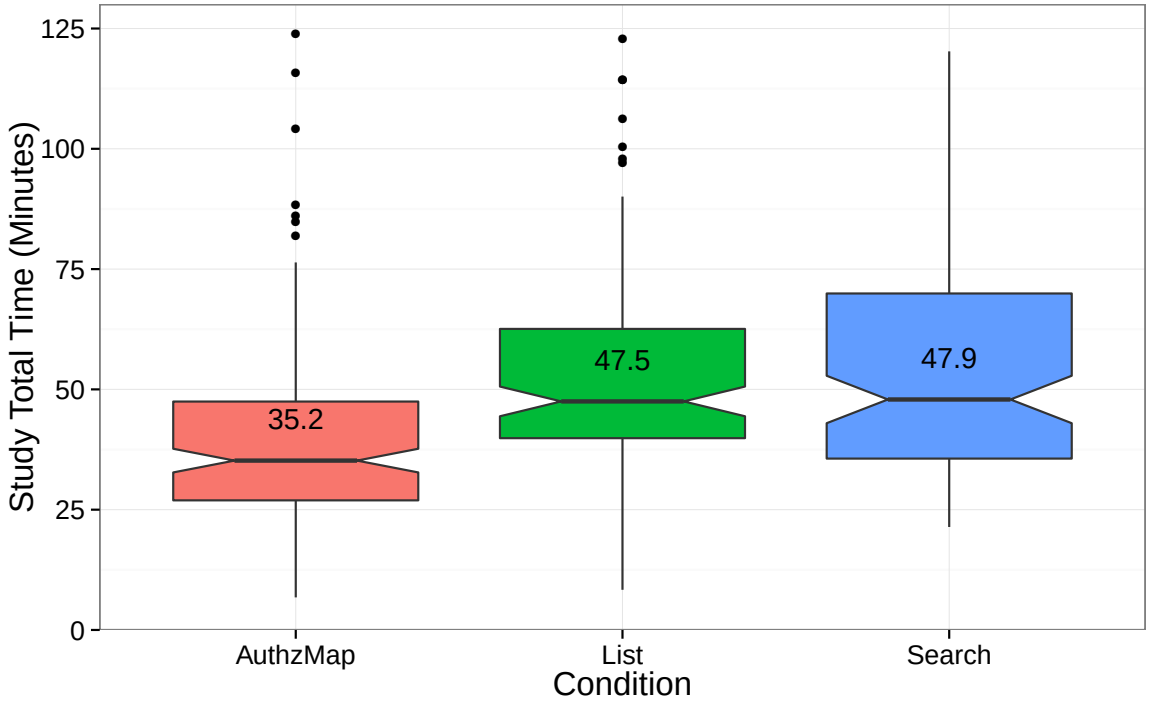


Figure 6.18: Total time needed to complete the study for participants in each condition. The numbers shown on the box plots are the median TTCs. Notches on the box plots indicate 95% confidence interval for the medians.

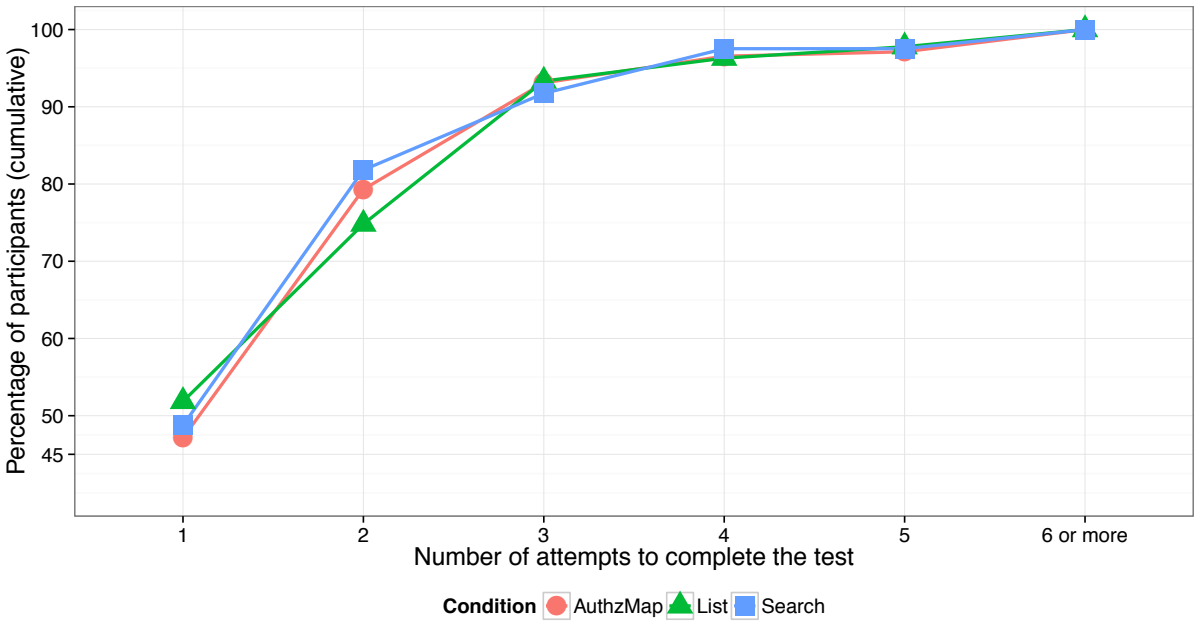


Figure 6.19: Number of attempts in completion of training test.

Table 6.4: Participants Demographics

		AuthzMap	List	Search	Total
Gender	Female	46.6%	52.6%	56.2%	51.2%
	Male	53.4%	47.4%	43.8%	48.8%
Education	Less than High School	2.3%	0.7%	0.8%	1.4%
	High School, diploma	8%	12.6%	10.7%	10.2%
	University/College Deg.	86.8%	85.9%	86.8%	86.5%
	Professional Deg.	2.9%	0.7%	1.7%	1.9%
Age	18-24 years old	30.5%	25.9%	42.1%	32.3%
	25-34 years old	43.7%	54.8%	39.7%	46.0%
	35-44 years old	15.5%	11.9%	10.7%	13.0%
	45-54 years old	6.3%	5.9%	5.8%	6.0%
	55-64 years old	3.4%	1.5%	1.7%	2.3%
	65-74 years old	0.6%	0%	0%	0.2%

Table 6.5: Median time to completion (TTC) for each of the tasks (in seconds), and pairwise comparison of TTCs. "A" stands for AuthzMap, "L" stands for List, and "S" stands for Search. The last three columns indicate null hypotheses (e.g., "A=L" indicates the following null hypothesis: there is no difference between TTC in AuthzMap (A) condition and List (L) condition). The highlighted cells show the cases where the null hypothesis was rejected and median TTC for AuthzMap condition was lower than the other condition.

	Task	A	L	S	A = L	A = S	S=L
1	Training	192.5	243.0	259.0	$p < 0.01, r = 0.24$	$p < 0.01, r = 0.22$	-
2	Common Review	117.5	144.0	96.0	-	$p = 0.01, r = 0.10$	$p < 0.01, r = 0.14$
3	User Comparison	109.5	225.0	195.0	$p < 0.01, r = 0.45$	$p < 0.01, r = 0.37$	-
4	Privilege Accumulation	89.5	256.0	190.0	$p < 0.01, r = 0.50$	$p < 0.01, r = 0.42$	$p < 0.01, r = 0.15$
5	SoD Violation Detection	92.0	293.0	165.0	$p < 0.01, r = 0.57$	$p < 0.01, r = 0.35$	$p < 0.01, r = 0.39$
6	Application Review	181.0	185.0	280.0	-	$p < 0.01, r = 0.34$	$p < 0.01, r = 0.33$
7	Comprehension	247.5	426.0	469.0	$p < 0.01, r = 0.30$	$p < 0.01, r = 0.32$	-

Per Task Results

In this section, we compare three conditions per task. Table 6.5 shows the median time to completion of individual tasks. The result of Kruskal-Wallis test for each task showed a statistically significant difference between three conditions. Therefore, we only show the result of three pairwise comparisons between conditions in Table 6.5.

Training Task: Table 6.5, and Figure 6.20 show that AuthzMap improved efficiency over the two other interfaces, although the effect size was medium. In terms of practical significance, AuthzMap reduced time to completion by about 20% compared to List, and by 25% compared

to Search. Table 6.6 shows the results of the accuracy analysis. Fewer participants in AuthzMap condition committed errors in identifying the last job function of the user, compared to List condition, and in identifying the date of last access review, compared to Search condition.

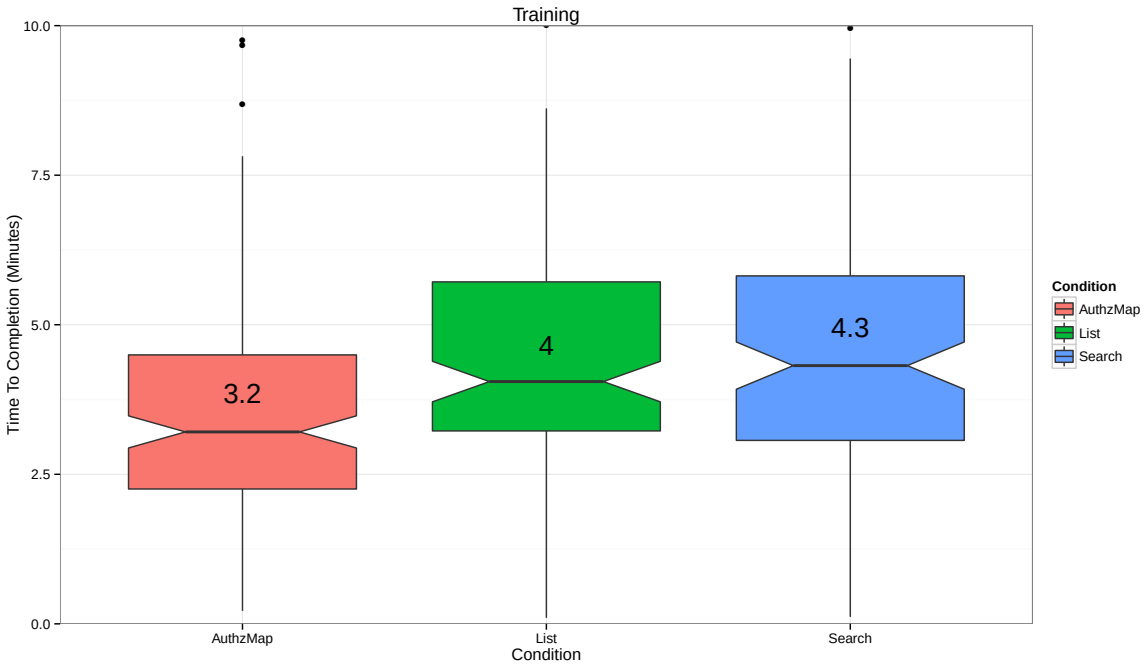


Figure 6.20: Time needed to complete the training task for participants in each condition.

Table 6.6: Comparing the correctness of participants' responses to the four components of the training task. The highlighted cells show the cases where the accuracy in AuthzMap condition was higher than the accuracy in the other condition, and the difference was statistically significant.

	A	L	S	A=L	A=S
Job Title	97.1%	97%	98.3%	1	1
List of files	87.4%	80%	90.1%	0.443	1.000
Last Job	87.4%	54.1%	81%	<0.05	0.738
Last Review	75.9%	66.7%	35.5%	0.394	<0.05

Common Review Task: Table 6.5, and Figure 6.21 indicate that for reviewing a single user, while the reviewer knows the files the user should have access to, Search is the fastest interface. Yet, looking at the effect size reveals that the size of the difference between AuthzMap and Search is small. In other words, AuthzMap reduces the median time-to-completion by approximately 17%, compared to list, but increases time to completion by approximately 25%,

compared to Search. In this task participants could commit two dangerous errors (i.e., not revoking invalid access), and we show the proportion of participants who correctly revoked such access in Table 6.7. Table 6.7 shows that we rejected all four accuracy hypotheses, and shows that participants in AuthzMap condition had more errors than the two other conditions.

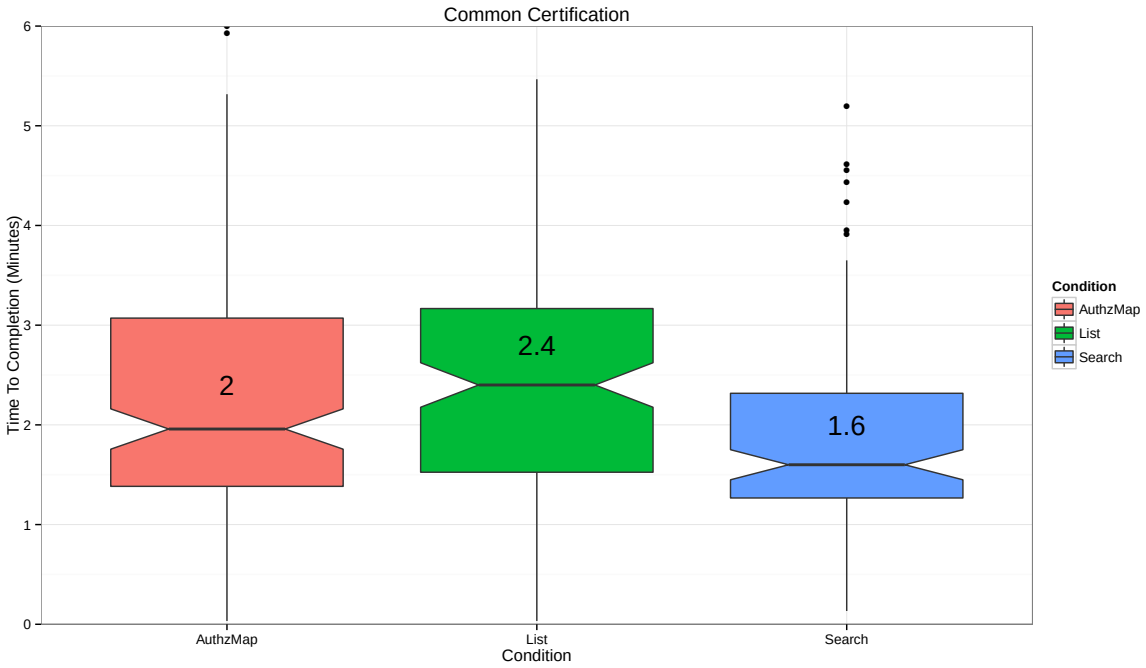


Figure 6.21: Time needed to complete the common review task for participants in each condition

Table 6.7: Comparing the correctness of participants' choices in common review task.

	A	L	S	A=L	A=S
Revoked R19	70.7%	86.7%	88.4%	<0.01	<0.01
Revoked R10	70.1%	87.4%	87.6%	<0.01	<0.01

User Comparison Task: Table 6.5, and Figure 6.22 show that AuthzMap improves efficiency over the two other tasks. In terms of practical significance, AuthzMap decreased the time to completion by about 105%, compared to List, and by about 78%, compared to Search. The accuracy analysis (Table 6.8) did not reject any of the accuracy null hypotheses.

Privilege Accumulation Task: Table 6.5, and Figure 6.23 shows that AuthzMap improves efficiency over the two other tasks. In terms of practical significance, AuthzMap improved time

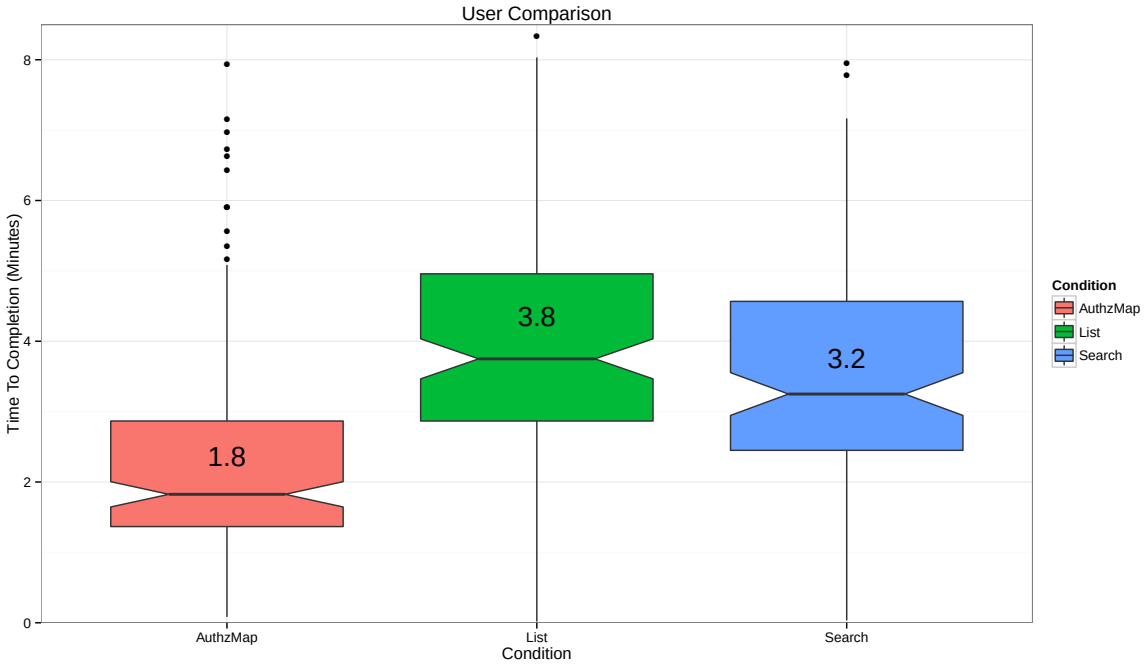


Figure 6.22: Time needed to complete the user comparison task for participants in each condition

Table 6.8: Comparing the correctness of participants' choices in user comparison task.

	A	L	S	A=L	A=S
Revoked R13	84.5%	88.9%	86.8%	0.632	1.000

to completion by about 186%, compared to List, and by about 112%, compared to Search. Table 6.9 shows the result of accuracy tests. We rejected three of the null hypothesis for comparing AuthzMap and List, but we did not reject any of the hypotheses for comparing AuthzMap and Search.

Table 6.9: Comparing the correctness of participants' choices in privilege accumulation task.

	A	L	S	A=L	A=S
R06, LyndaR	86.8%	68.9%	79.3%	<0.05	0.652
R03, DerrickS	88.5%	71.9%	80.2%	<0.05	0.4
R12, DerrickS	86.2%	71.1%	81.8%	<0.05	1

SoD Violation Detection Task: Table 6.5, and Figure 6.24 show that AuthzMap improves the efficiency of detecting SoD violations. In terms of practical significance, AuthzMap reduced

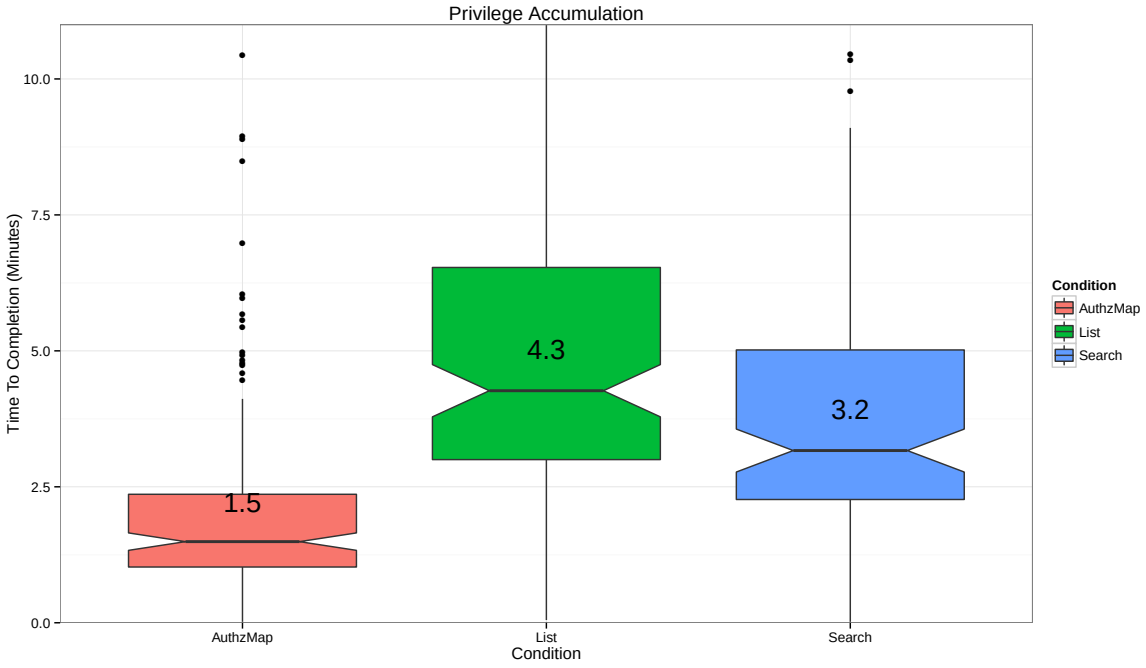


Figure 6.23: Time needed to complete the privilege accumulation task for participants in each condition

the time to completion by about 218%, compared to List, and about 165%, compared to Search.

The result of the accuracy analysis (Table 6.10) rejected two of the null hypothesis for comparing AuthzMap and List, but did not reject any of the hypotheses for comparing AuthzMap and Search.

Table 6.10: Comparing the correctness of participants' choices in SoD violation detection task.

	A	L	S	A=L	A=S
SoD (R36, R11)	92.5%	83%	91.7%	<0.05	1
SoD (R14, R00)	94.8%	85.9%	89.3%	<0.05	0.451

Application Review Task: Table 6.5, and Figure 6.25 show that AuthzMap and List did similarly in terms of efficiency, while Search did worse. In terms of practical significance, AuthzMap reduced the time to completion by about 35%, compared to Search. The accuracy analysis (Table 6.11) rejected all the null hypotheses for comparing AuthzMap and List in favor of List, and rejected one of the 15 hypotheses for comparing AuthzMap and Search in favor of Search.

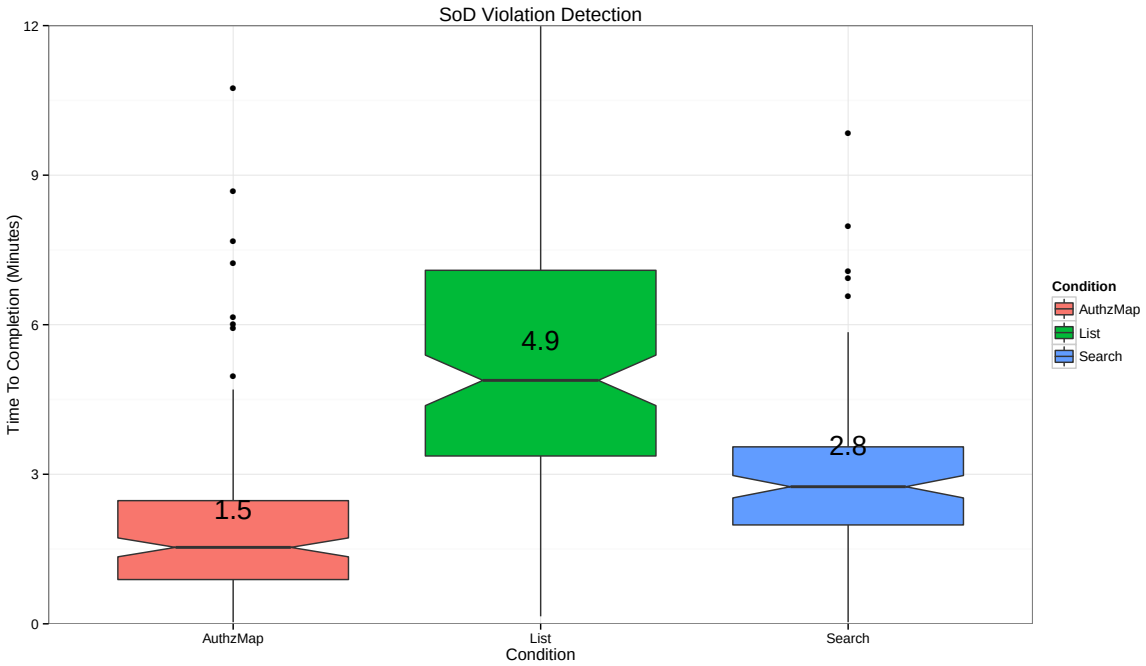


Figure 6.24: Time needed to complete the SoD violation detection task for participants in each condition

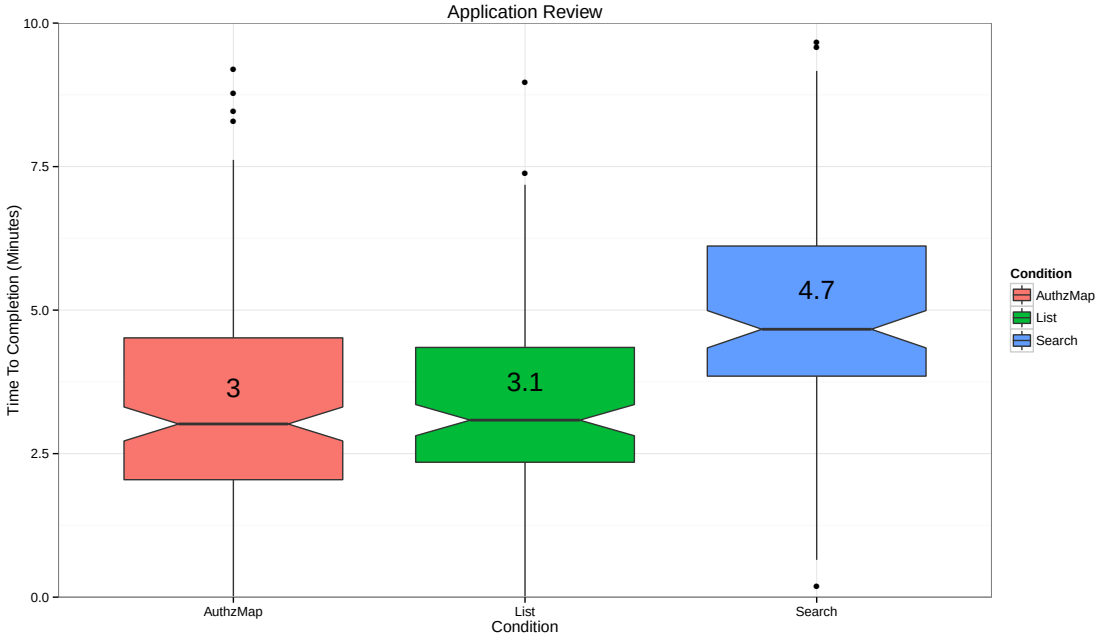


Figure 6.25: Time needed to complete the application review task for participants in each condition

Table 6.11: Comparing the correctness of participants' choices in application review task.

	A	L	S	A=L	A=S
EdmundJ, R10	82.8%	97%	78.5%	<0.05	1
EdmundJ, R15	81.6%	96.3%	86%	<0.05	1
EdmundJ, R23	81%	97%	76.9%	<0.05	1
EdmundJ, R11	83.3%	97%	80.2%	<0.05	1
EdmundJ, R22	83.3%	97%	78.5%	<0.05	1
EdmundJ, R30	70.7%	97%	86.8%	<0.05	<0.05
EdmundJ, R28	69%	97%	78.5%	<0.05	1
NelsonM, R10	82.2%	97%	78.5%	<0.05	1
NelsonM, R15	82.8%	97%	86%	<0.05	1
NelsonM, R23	79.9%	96.3%	78.5%	<0.05	1
NelsonM, R33	69.5%	96.3%	78.5%	<0.05	1
NelsonM, R35	70.7%	95.6%	85.1%	<0.05	0.149
JaneH, R10	83.3%	96.3%	81.8%	<0.05	1
JaneH, R23	82.8%	97%	81%	<0.05	1
JaneH, R35	71.3%	95.6%	86%	<0.05	0.0909

Comprehension Task: Our analysis (Table 6.5, and Figure 6.26) suggests that AuthzMap does better in terms of efficiency than the two other interfaces. It also practically improves efficiency by about 72%, compared to List, and by 89%, compared to Search. This task involved the assessment of risk for users having specific access privileges. The summary of participants' responses to risk assessment questions is presented in Figure 6.27. We used pair-wise two-sided fisher's exact tests with Bonferroni correction, to test the following hypothesis for each of the risk assessment: (A=L) The choice of AuthzMap or List does not impact the accuracy of risk assessment. (A=S) The choice of AuthzMap or Search does not impact the accuracy of risk assessment. The result of the test rejected ($p < 0.05$) the all four (A=L) hypotheses, and rejected ($p < 0.05$) three of the (A=S) hypothesis (in assessment of risk related to R02, R09, and R11).

Post Evaluation Questionnaire

We asked participants to answer seven likert scale questions related to their subjective satisfaction of using the interface. An overview of the responses is shown in Figure 6.28. The results

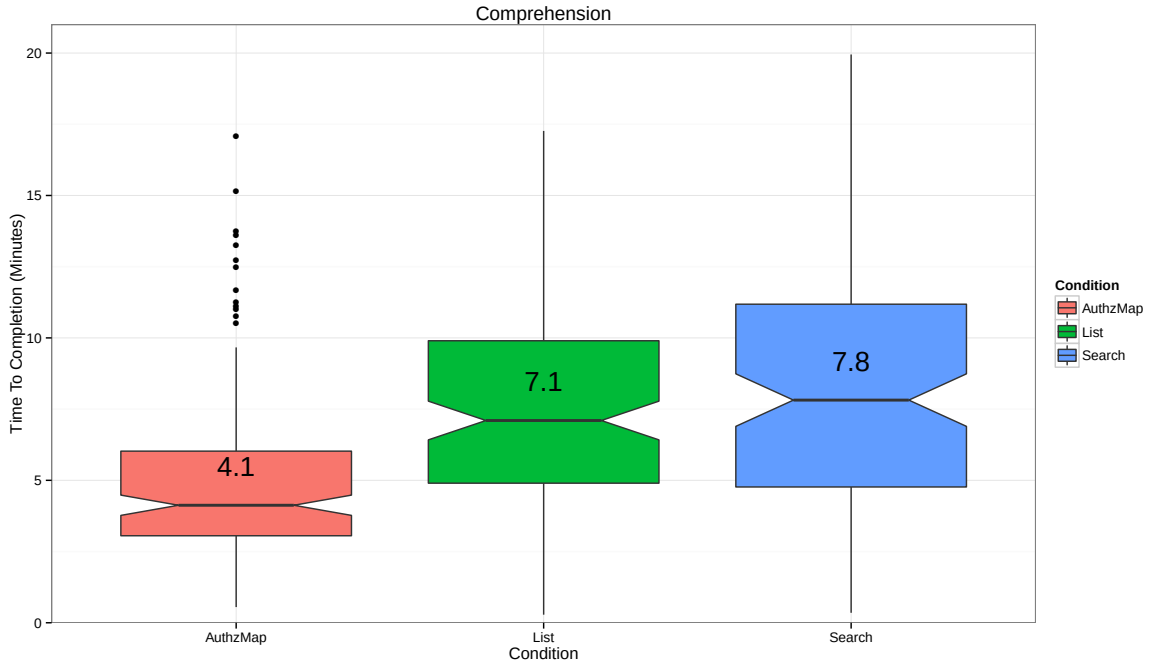


Figure 6.26: Time needed to complete the comprehension task for participants in each condition

of the hypothesis testing for each question is presented in Table 6.12.

Table 6.12: Pairwise comparison of participants' responses to post-evaluation questionnaire. The highlighted cells show the cases where the null hypothesis was rejected and the AuthzMap ratings were higher than the other interface.

	Question	A=L	A=S
Q1	Ease of performing tasks	<0.05	<0.05
Q2	Ease of finding information	<0.05	<0.05
Q3	Perceived correctness	0.42	0.33
Q4	Understanding who has access to what	0.59	<0.05
Q5	Perceived access violation detection	0.072	<0.05
Q6	Perceived intuitiveness	0.34	<0.05
Q7	Perceived Engagement	0.3	<0.05

6.9 Discussion

In this section, we summarize, interpret, and discuss the findings of the user study. We first discuss the efficiency, accuracy, and subjective satisfaction findings. Then we will discuss

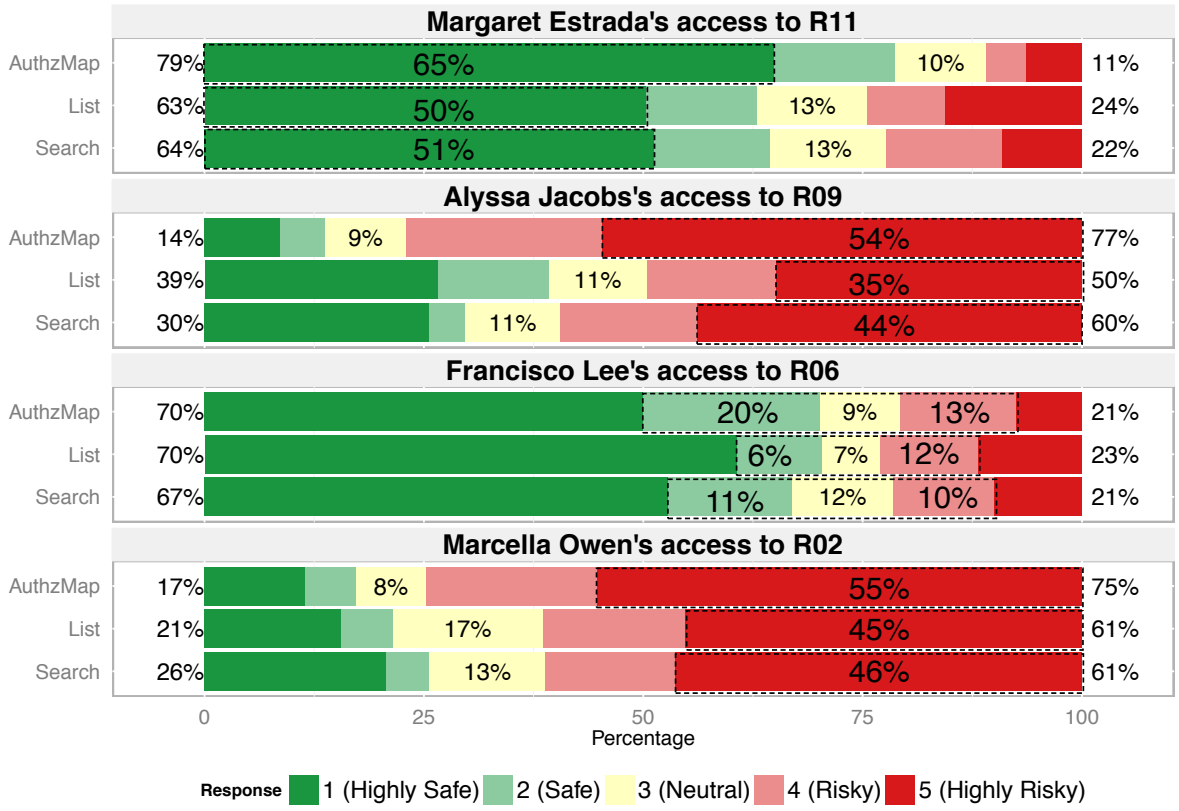


Figure 6.27: Summary of participants responses to comprehension questions. Participants were asked to rate the risk associated with each access on a five-point likert scale. The areas with a dashed border indicate correct responses.

the limitations of the user study, including the use of non-expert participants, and a synthetic dataset.

6.9.1 Efficiency

In Section 6.8.3, we show that participants in AuthzMap condition could finish the study faster than those in two other conditions. We also compared the use of three interfaces in various access review scenarios. We showed that AuthzMap improved the efficiency, compared to both of the other interfaces in five of the seven tasks, and compared to one of the interfaces in the two other remaining tasks. This finding require further discussion.

AuthzMap participants' performance in *Common Review* was not as efficient as Search, but

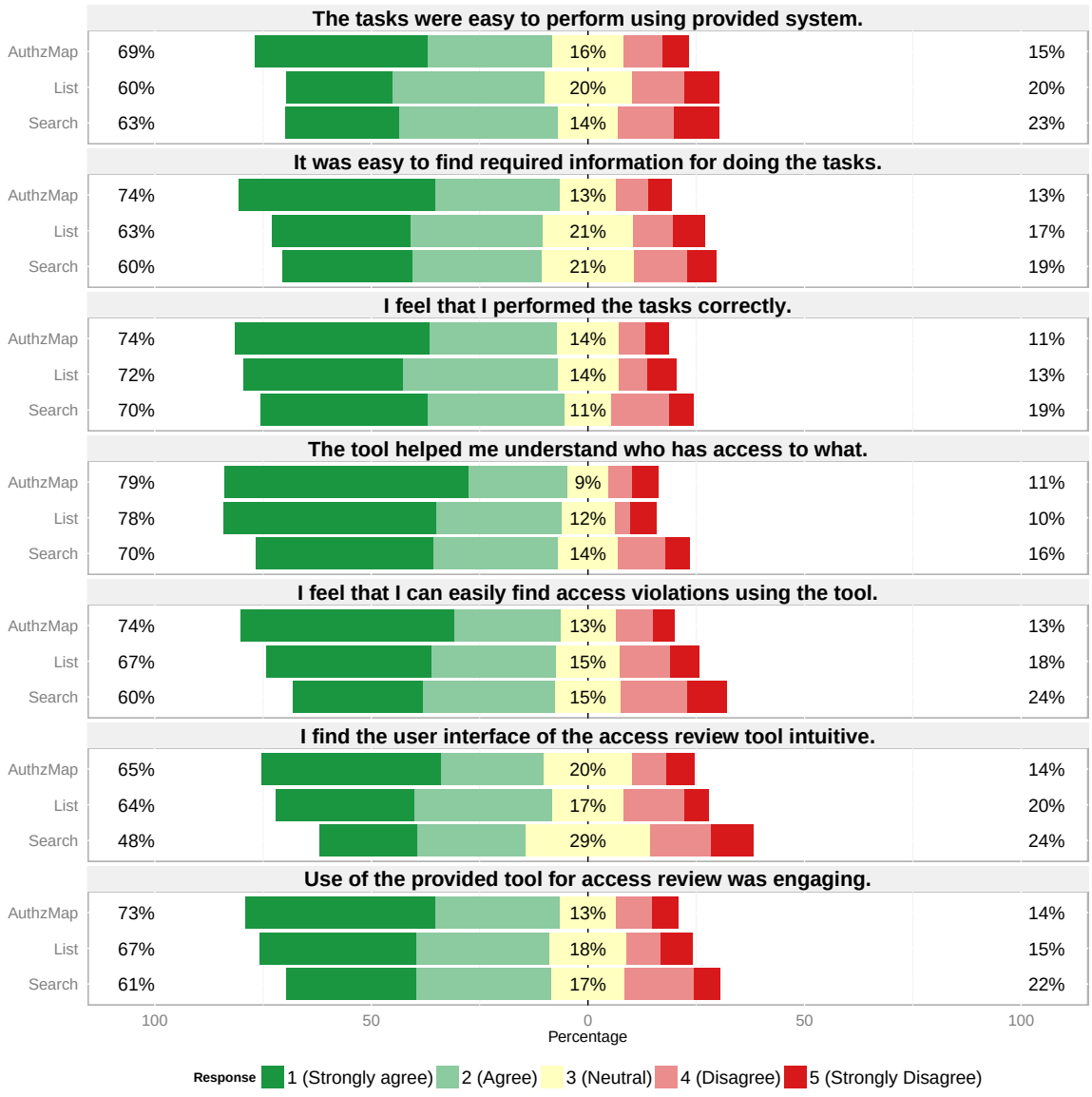


Figure 6.28: An overview of post-evaluation questionnaire responses. Participants were asked to rate their agreement with each statement on a five-point likert scale.

was more efficient than List. We can provide three explanations for this: (1) The task involved reviewing only one user. The additional contextual information in the AuthzMap fisheye view could increase user's cognitive load, and hence reduce the performance. (2) The Search interface by default set the status of files to "certify". This helped the participants to change the status of two unauthorized files, and keep the rest of the files intact. Meanwhile, AuthzMap participants had to explicitly set the review status of each file. These two suspected issues

provide an opportunity for further improvement. To address the first issue, we can use the focus plus context visualization developed by Leung and Apperley (1994) to highlight the user that the reviewer is currently working on, while still showing the contextual information in the background (e.g., by highlighting the current user and fading the rest of the users). The second possible issue was a design decision that we made in AuthzMap to prevent reviewers from using the default option, and rather make an explicit decision for each access privilege.

The *User Comparison* task was similar to *Common Review*, but it involved three users with identical jobs instead of one. AuthzMap participants did better than participants in two other conditions. We attribute the improvement to AuthzMap's ability to categorize, and filter users according to their job, and then use the contextual information (access of users with the same job) to quickly find the excessive access. Our analysis of study logs confirmed this, as participants used the *sort user* feature of the AuthzMap in this task significantly more than in other tasks. Furthermore, comparing the TTC of this task to the TTC of *Common Review* shows that increasing the number of participants did not impose an additional burden on AuthzMap participants, unlike Search and List participants.

In the *Privilege Accumulation*, *SoD Violation Detection*, and *Comprehension* tasks, AuthzMap performed better than the other two interfaces. We can attribute this to the visibility of context and history in the interface. For example, AuthzMap integrates the employment history and access privileges, and makes it accessible to users. The two other interfaces required participants to collect information from the HR and access review systems, and perform a mental process to formulate the relationship between access control and employment data. Additionally, AuthzMap integrated the SoD policy information with the existing access control data, and helped users quickly identify and resolve the SoD violations. Participants in two other conditions had to use an SoD catalog. Therefore, they needed to mentally associate the policy with the access control data.

In *Application Review* task, AuthzMap participants performed similar to List and better than

Search. This is an expected result as both List and AuthzMap clearly integrate information about the application in the interface, but Search participants had to use the auxiliary application catalog to find the files related to a certain application.

6.9.2 Accuracy

In Section 6.8.3, we report that AuthzMap participants could achieve more accurate results than the other two interfaces in only one of the tasks.

Accuracy results of *Common Review* task were unexpected. AuthzMap participants committed significantly more errors than participants in two other conditions. Examining the data closely shows many of these participants committed identical errors. After further investigation, we realized that AuthzMap's detail interface showed the user had accumulated privileges from a prior job. And a subset of participants who received the *Privilege Accumulation* task before *Common Review*, did not use the information in File Catalog, but rather did access review based on what they learned from *Privilege Accumulation* (about 15% of the participants in AuthzMap condition). This was our mistake in designing task data, and we should have controlled the privilege accumulation in the policy for this task. If we count the correct answers from the participants who looked at the task from privilege accumulation perspective as valid, there is no statistically significant difference between three conditions in accuracy.

Accuracy results for "User Comparison" task do not show a difference between three conditions. These results suggest that the increase in efficiency did not impact the accuracy of participants in AuthzMap condition.

For two tasks that required decision making in uncertain conditions, *Privilege Accumulation* and *SoD Violation Detection*, AuthzMap positively affected accuracy, compared to List but not Search. These two tasks required contextual information that unlike AuthzMap was not integrated in List and Search interfaces. Search participants did surprisingly well in the collection

and integration of the context with the information available in the interface but not the Lits participants. One explanation for this observation is that the List interface contains redundant information that could mentally overload users. On the other hand, Search is rather straightforward, and while it requires user to spend more time collecting and integrating information, it does not reduce the accuracy.

Accuracy results for "Application" task were rather surprising. AuthzMap and Search participants produced less accurate results than List. While we expected the List participants to do better than Search (List clearly showed the application associated with each access privileges, as one of the columns in the list of privileges), we expected AuthzMap to perform as good as List. Further looking at the participants errors, we did not find any patterns or evidence that participants committed mistakes rather than slips. There are three possible explanations of such slips: (1) The names of the applications were presented in a small text, and it was rotated 90 degrees. Jolicoeur (1985) shows that text rotation can have a negative impact on human cognition, and requires mental rotation, before a human can recognize an object. To address this, we can use slightly less rotated text (e.g., 45 degrees), as it is shown that rotation is positively correlated with cognitive load. (2) Complexity of the grid: to complete the task, users had to recognize the file related to an application (located in columns of the grid), and then check the target user for having access to the file. This process can be prone to errors due to the proximity of grid cells. To address this, we can utilize the focus plus context visualization by Leung and Apperley (1994), by allowing users click on the column to focus on a specific file, and then perform the actions.

Unlike other tasks, AuthzMap participants provided more accurate responses to three of the four questions in the *Comprehension* task. We expected this result, as participants in other conditions should have used multiple information sources to complete the task, and they needed to build the correct model of the policy in their memory. Yet, AuthzMap participants could see the complete picture of the policy. The only question that we did not see a significant difference

between AuthzMap and both conditions was the assessment of R06 risk, which we did not see a difference between AuthzMap and Search. R06 could be both safe or unsafe, therefore, we expect participants not to choose highly safe or highly risky. Further look at the graphs in Figure 6.27 shows that Search participants assessed R06 and R11 (which was highly safe) similarly. However, AuthzMap participants assessed R06 rather differently from how they assessed R11. This suggests that maybe Search participants naively chose unsure responses, but AuthzMap participants made a more informed choice.

6.9.3 Subjective Satisfaction and Learnability

AuthzMap is rated significantly higher than Search in all of the subjective satisfaction questions but one. It is also rated significantly higher in two of the questions compared to List. The AuthzMap ratings are not lower than List or Search for any of the questions. This suggests that participants preferred using AuthzMap over the two existing interfaces.

Although we did not measure learnability directly, we can use the results of the *Training* task to show that the interface was not hard to learn for participants. AuthzMap participants finished the *Training* task faster and more accurate than the participants in other conditions (i.e., they quickly learned how to use the interface in their first encounter with it).

6.9.4 User Study Limitations

Ideally we would have evaluated AuthzMap by asking managers to use AuthzMap to review access of actual users in their company. But our experience from this study, and our past field-studies in HOT-Admin project suggests that conducting a field experiments in real organizations is extremely challenging. We faced similar difficulties to what has been previously discussed by Sedlmair et al. (2011) as challenges of conducting studies in large organizations. First, AuthzMap is a prototype, and integrating it with real access management systems in or-

ganizations is a software engineering challenge. Second, asking managers to budget time to evaluate AuthzMap is challenging, particularly because access review is not their day-to-day task. Third, AuthzMap requires identity and access control data, which are considered extremely sensitive. Our experience shows that even getting permission to conduct an interview requires approval from the legal department of a large company, as well as multiple managers, let alone conducting experiments using the sensitive data.

Due to the above challenges, we adopted an approach similar to Sedlmair et al. (2011), and conducted a set of during-design, exploratory studies before committing to a costly field-study. First, we received feedback on AuthzMap from a large domain expert audience (employees of our industry partner). We also had two small group discussions with the engineering team, and usability team of our industry partner. Second, we conducted 12 heuristic evaluation sessions (using Nielsen (1992) and ITSM heuristics) with independent usability experts to identify usability issues with AuthzMap, and further improve the interface. Third, we conducted a lab study (Section 6.8) with non-domain experts to further evaluate the interface and compare it to existing systems. Sedlmair et al. (2011) show that conducting during-design experiments could be very helpful and lead to tools with higher usability, and eventually become a major reason for the tool being deployed in the field. Therefore, we conducted an exploratory study with MTurk participants to be confident that the tool does not have obvious usability problems, and fares well against existing systems. But the next step in evaluation would be to conduct an in-depth long-term case study as suggested by Shneiderman and Plaisant (2006), by integrating AuthzMap with existing access management systems, asking managers to use AuthzMap, and then get qualitative feedback. Such a field-study can show if the tool will be adopted by managers, and could solve access review problem.

We used an automatically generated dataset. Using a real-world dataset was not feasible, as there are very few real-world enterprise access control data sets available to the research community. We examined five common datasets used regularly by access control community such

as: americas_small, apj, healthcare, domino, firewall1 and firewall2 (see (Ene et al., 2008) for details of the datasets). These datasets only contained lists of users, permissions, and user-to-permission assignments. Our study required contextual data, such as users' job, employment history, access history, and review history. Adding meaningful context to existing datasets was not possible, therefore, we elect to generate a dataset that best matched our interview study findings.

6.9.5 Moving from Prototype to an Actual System

In order to implement AuthzMap as an actual access review system, it should be integrated with two systems already available in many organizations: HR system to provide employee data, and IAM system to provide the access policy data. These systems can provide all the necessary information for AuthzMap, including the name, job, and employment history of users, and the user-to-access privilege assignments, and their history. Currently, AuthzMap is developed in Flash, but in order to implement it as an actual system, it should be implemented as a standard HTML web application, as many large enterprises disallow browser plugins such as Adobe Flash (Dormann and Rafail, 2008).

6.10 Conclusion

In this chapter, we studied how access policies are reviewed in large organizations using interview and survey data. We then identified a set of five challenges that organizations face during access review, and suggested four design goals to deal with those challenges. We then realized the design goals by building AuthzMap, a novel user interface for making sense of access policies. We then performed a heuristic evaluation of AuthzMap to identify problems and improve the interface further. We then conducted an exploratory user study with 430 MTurk participants to compare AuthzMap to two of the existing access review systems. Our results show

that AuthzMap improved efficiency of access review in five of the seven, and accuracy in one of the seven tasks. The design of AuthzMap can serve as an example of how the contextual information can be integrated with access policy in a user interface, and our user study suggested that the design was successful; As we showed that such integration will improve efficiency of accessing contextual information, and in complex decision making processes (such as *Comprehension* task in our study) can improve better understanding of policy, and therefore, facilitate making more accurate decisions. The bigger HCI implications of this work are exploring the importance of context in access control, and proposing an effective approach for integrating context in access control interfaces. As the next step, AuthzMap should be deployed in a real organizational setting, and its impact should be evaluated.

Chapter 7

Discussion

While we discussed the details of each project separately, we provide an overall discussion of the findings in this short chapter.

7.1 Revisiting Social-technical Gap in IAM

This dissertation spans over engineering and social science disciplines. On the engineering side, we attempt to construct a technological solution that can address a practical problem. On the social science side, we tried to understand the basis for building our proposed system in the social phenomena. Furthermore, in this process we developed a set of methods, ITSM guidelines and heuristics, that we used during the engineering of our proposed solution.

Our main goal of this thesis was to understand and narrow the social-technical gap in identity and access management. We showed that the core of IAM is access provisioning, and it happens in a social setting of an organization. But there is a gap between what socially must be supported (i.e., giving users access to resources they need, and preventing unauthorized access), and what technology supports. While some of the literature in access management, and

even some of our interview participants suggest automated role-based provisioning could be a technological solution to support the social practice, we showed that it is far from ideal, and it is not capable of addressing all of the nuances, and complexities of the social practice such as exceptions, and business changes. While according to Ackerman (2000), bridging this gap (i.e., supporting the social activity by technology) is considered a grand challenge in the field of HCI and CSCW, we saw that a set of practices has already been developed by the participants to work around the gap and narrow it. For example, participants documented the meaning of privileges to create a common terminology, or they used technological solutions such knowledge bases or access catalogs to *partially* support the social process. We also found that automated provisioning is a useful solution that can *substitute* a part of the social phenomena. Access review was the third solution that *supported* and *augmented* (Hutchins, 1995) the social practice to address some of the challenges such as human errors. Ackerman (2000) names this type of solutions first-order approximations, which "are tractable solutions that partially solve specific problems with known trade-offs."

Our integrated model of IAM activities and challenge guided us in better understanding and explaining the social-technical gap, and understanding the practices developed by our participants to narrow the gap. Armed with this understanding, we suggested a set of implications for narrowing the gap, by improving technology or practices. Our proposed implications were, in the words of Sas et al. (2014), "*succinct descriptions for communicating core field-study findings*", and were not concrete solutions. While realizing and testing all of the design suggestions were beyond the scope of this thesis, we chose one recommendation, "allow understanding and making sense of access policy", and focused on realizing it through building and testing an example system.

We used a combination of methods to narrow the gap by providing more usable access review systems. First, we looked at the field study data, and modeled it in the activity theory framework. This enabled us to use activity theory as sensitizing concepts (see (Blumer, 1954) for the

definition of sensitizing concepts, and (Hughes et al., 1992) for the use of them in interpreting field data), and identify the areas where technology can address a problem. After identifying the areas to be improved, we used ITSM design guidelines to propose solutions for improving the technology. Although the guidelines were still abstract, and we needed to instantiate them. We realized each guideline based on our HCI experience, and came up with an exemplar system, AuthzMap. The instantiation of guidelines was a subjective, and creative process. Therefore, its outcome should be evaluated. We used our ITSM usability heuristics to evaluate and improve our exemplar system. Finally, to show if the proposed interface is seen as an improvement over existing systems, we conducted a user study and showed that our interface is more usable than two of the existing commercial systems.

7.2 Implications Beyond IAM or Access Review

While the focus of this thesis is on IAM, our findings are useful beyond that. First, our proposed guidelines and heuristics are general to all of the ITSM tools, and therefore, they can be useful for other researchers or practitioners who are designing interfaces for those tools. Additionally, each individual ITSM heuristic might be applicable to other work domains. For example, the "visibility of activity status" and "planning and dividing work" heuristics can play an important role in the evaluation of any collaborative software. "History of actions on artifacts", and "flexible representation of information" are applicable to domains that require intensive inferential analysis, pattern recognition, and addressing previously unknown conditions. "Knowledge sharing" and "rules and constraints" are particularly important in evaluating software that is deployed in the organizations. "Verification of knowledge" is important to software that operates on critical information.

Second, our process of systematically studying the social-technical gap, and narrowing the gap can be used by other researchers who work in a similar problem domain. We tried to make our research process clear, and repeatable. We explained all the steps we used to develop our set of

guidelines and heuristics so that other researchers can replicate the process to create guidelines and heuristics for their domain of interest. Additionally, we clearly described our usability engineering process, including understanding of the activity, mapping the understanding to design goals, and design goals to actual design. Finally, we laid out all of the steps we use to conduct our lab study.

Third, our field study and survey findings have larger implications than just understanding access review activity. Our findings suggest that while access control policies are usually composed of users, access privileges, and the assignment of users to privileges, these three components are only parts of a larger context, and they evolve and change over time. Therefore, a snapshot of a user's access privileges does not provide a complete picture of access policy. We further determined the context of a users' access privileges, which includes other users' access, other policies that impact such access (such as SoD policies), user's job, and other stakeholders involved in the access control decisions, such as those who requested or approved the user's access. We also demonstrated that access control policies evolve over time, and identified users' job, access privileges, and previous reviews as important historical artifacts. Although our focus was on access control in large organizations, the concept of context for access control policies is still applicable to access control in other domains such as file systems, multimedia, etc. We should note that each domain should be studied separately, as the contextual information for enterprise domain (such as job or approval workflow) may not be applicable in other domains. For example, findings by Vaniea et al. (2012a,b) suggest that proximity of access control displays and photos helps users notice and correct access control errors. In this case, the photo (visual representation of the asset) is part of the access control context.

Fourth, our design of AuthzMap included many examples of implementing our suggested ITSM guidelines. Therefore, AuthzMap can be used as a design concept for practitioners or researchers who are interested in applying ITSM guidelines to ITSM or similar domains.

7.3 Validity of the Research

We used a mixture of qualitative and quantitative research throughout this dissertation. Therefore, we employed different approaches to address threats to validity. In this section, we review the validity of the major components of the thesis, discuss threats to validity, and our approach to reduce them.

ITSM Heuristics: Sas et al. (2014) propose criteria for validity of design implications, and usability heuristics falls under the design implications definition. The criteria include empirical, theoretical, and external validity as well as originality, generativity, inspirability, actionability. Our proposed heuristics were grounded in the literature (*empirical validity*), and were supported by activity theory (*theoretical validity*). Furthermore, in our heuristic evaluation experiment, we prove the *originality* of the heuristics, as we show that they are more powerful than Nielsen's in evaluating ITSM tools. Our heuristic evaluation experiment also show that the heuristics can actually be useful and effective in evaluation (*actionability*). There are also certain threats to validity of our results. We did not explicitly evaluate the *external validity* of heuristics, as we only tested them in evaluation of one system. Although, the threat to external validity is reduced by building the heuristics based on a large set of publications related to a wide range of ITSM tools, and also supporting the heuristics with an HCI theory that is applicable in all social contexts. Additionally, we did not formally evaluate the generativity, and inspirability of heuristics, but we showed that the use of heuristics in the engineering process of AuthzMap inspired and generated a novel user interface.

ITSM Heuristics User Study: We used a combination of techniques to reduce the threats to *internal validity* of our study. To eliminate the impact of participants' background on the study outcome, we balanced the knowledge and background of the participants between two conditions. Furthermore, participants in both conditions were given exactly same

type of training, and the training did not involve the researcher. To eliminate our own bias in rating the severity of the problems, we asked external usable security experts to rate the severity of the problems. The problems aggregation phase was performed by the research team, and therefore, could be impacted by researcher bias. Although to reduce bias, we used two researchers to perform this process. In terms of *external validity* of the findings, we did not replicate the study for other types of ITSM systems, therefore, our statistical conclusions are limited to the evaluated interface, and to evaluators with a background similar to recruited participants. We also tried to make the study ecologically valid by asking people with HCI expertise to perform heuristic evaluation.

Field-study: The field study was a qualitative research, and it should be judged using a qualitative research validity criteria. We conducted a set of validity procedures suggested by Creswell and Miller (2000). We performed three types of *triangulation* at different stages of research. First, we compared our findings to the existing literature in the area of access management, and showed that our findings support and explain prior literature. Second, for the access review part, we used quantitative data for triangulation purpose as suggested by Jick (1979). We supported qualitative data through quantitative data, and showed that the survey findings are consistent with the field study findings. Third, we performed researcher triangulation in the initial stages of the study, as two researchers coded the interviews, and compared their findings to each other. In addition to triangulation, we provided a *thick description* by including the actual quotes from participants so that the reader can judge the validity of our interpretations from the data. We used *peer briefing* by presenting the results of our research to various audiences, including computer security community (on multiple occasions in conferences and workshops), usable security community, local Vancouver IT security professionals groups, and our industrial project partner, and by allowing them to challenge and critique our findings. We also attempted *member checking* by asking the participants to do follow-up interviews with us, but most of our participants were busy, or changed jobs or projects, which prevented us

from scheduling another interview with them. We also sent our initial field study publication (Jaferian et al., 2009) to the company that we performed our case study in, and asked for comments. But they just confirmed our findings without providing any specific feedback. Finally, as a part of the validity procedure, we should explain our assumptions, beliefs, and biases that could have an impact on the outcome (i.e., *researcher reflexivity*). Before conducting our field study, we had experience with the HOT-Admin project (see Chapter 2) and were familiar with IT security domain and its challenges. Furthermore, the author had experience with using activity theory, and could have a bias towards looking at the data through a theoretical lens. We tried to overcome some of these biases by using two researchers for the initial data analysis.

Evaluation of AuthzMap: To reduce threats to internal validity, we assigned participants randomly to the conditions, and we used participants without prior background in using access review tools. Furthermore, we randomized the order of the tasks during the study, to eliminate the order effect. To avoid the researcher bias during study sessions, we built an online website so that participants can complete the study without the researcher's involvement. Furthermore, we provided same training, tasks, and access control datasets to participants in all three conditions. There are areas that there might be a threat to internal validity of the study. The study tasks were designed by the research team, and therefore, they can be biased towards the AuthzMap. To reduce this threat, we built and justified the tasks using field study and survey data. There are also threats to external validity of our findings as our evaluation was limited to the tested dataset, and tested tasks, although we tried to include various possible access review scenarios to reduce this threat. Additionally, there is a threat to ecological validity of the study, as we used the MTurk participants. Although, to reduce this threat, we trained the participants, and provided them with the knowledge to perform the tasks.

According to McGrath (1995), some of the threats to validity discussed above are a result of the inherent limitations of the chosen strategies. Our use of multi-method approach was particularly helpful in reducing those threats. For example, in ITSM heuristics creation, we combined field studies (literature and interviews), which maximizes the realness of findings with HCI theories, which maximizes the generalizability. Similarly, in design and evaluation of AuthzMap, we used field study data to increase realness, survey to increase the generalizability, and a lab study to increase the precision of findings.

Chapter 8

Conclusion

This dissertation has addressed the social technical gap in the area of identity and access management (IAM). We have shown that IAM tools are essential components of IT security in organizations, but there is a lack of attention to usability of those tools by the research community or industry. To address the problem, we adopted a three phase approach. In phase one, we developed a set of guidelines and a set of heuristics that can be used for designing and evaluating the usability of ITSM tools. In phase two, we focused on the understanding of the social phenomena under study, identity and access management. In phase three, we focused on one specific IAM related activity, access review. We identified the shortcomings with existing technologies, and challenges in current access review practice, and proposed and evaluated a new user interface that supports the current practices, and addresses the shortcomings of existing technologies.

In phase one of this dissertation, we approached the problem with a broad focus, and tried to understand how usable IT security tools should be built. Through use of a rigorous literature survey, and interview analysis, we developed a set of 19 usability guidelines for designing IT security management (ITSM) tools. The identified guidelines were built on collected data and

were specific and limited to the data we analyzed. Furthermore, the guidelines were leaned toward design of tools, rather than evaluation of them. Therefore, we analyzed the guidelines using the theoretical leverage provided by activity theory. Using theory leveraged our interpretation of data and added another level of validation to our findings. This additional analysis led to seven usability heuristics for evaluation of IT security management tools. To validate the heuristics, we conducted a lab study to compare the use of ITSM heuristics with Nielsen's heuristics. Our results showed that the severity of the problems found by participants in the ITSM condition was higher than those found in Nielsen condition. Furthermore, our participants found the ITSM heuristics to be as relevant, easy to apply, and easy to learn as Nielsen's. We also showed that Nielsen's heuristics can also be effective in finding a class of problems in ITSM tools that cannot be found by the ITSM heuristics.

In phase two, we zoomed our focus on identity and access management (IAM), as one subarea of IT security. To understand IAM activities and challenges, we adopted a grounded theory approach. We performed a set of 19 semi-structured interviews with security practitioners involved in IAM. We used grounded theory coding techniques such as open, axial, and selective coding to analyze the data. Our results provided a detailed description of IAM activities, and their challenges. Furthermore, we built an integrated model of IAM activities and challenges. Our model showed that the core of IAM activities are automatic and manual provisioning. Automatic provisioning, and partly manual provisioning are orchestrated by identity management activity. We also showed that manual provisioning involves challenges such as errors, communication and collaboration complexity, and inefficiency. To cope with the challenges, companies adopt three practices: access review, understanding of access privileges, and transitioning to automatic provisioning. On the other hand, our model shows that only transitioning can completely eliminate manual provisioning challenges. We show that companies like to automate their provisioning, and perform a set of difficult activities to achieve that. But our model suggests that automated provisioning involves challenges as well, and leads to companies revisiting manual provisioning. We also showed that companies understanding of

access privileges and business processes expand while cycling between manual and automatic provisioning.

In phase three, our goal was to narrow the social technical gap in one of the IAM activities. Therefore, we further zoomed our focus on access review that we found particularly challenging during phase two. To bridge the gap, we studied existing social practice, and existing technology, and based on that understanding developed and evaluated a new user interface for access review. First, we further analyzed our interview data with the focus on *access review* activity, and conducted a survey of 49 security practitioners to collect quantitative data about access review. Using the interview and survey data, we provided a detailed description of access review, and identified its challenges. Second, we also analyzed the usability problems with the access review interface of a commercial IAM system. Our findings show that while access policy in organizations usually consists of users, privileges, and user-to-privilege assignments, contextual information is needed to comprehend the policy. We found that the context of access policy includes other users access, history of the policy, uesrs' employment information, and other policies such as SoD. But the existing access review tools usually hide this information, and only reveal the immediate policy. Based on this understanding, we suggested a set of design goals to improve the usability of existing access review tools. To achieve these goals, we built a new user interface named *AuthzMap* through multiple iterations, including extensive heuristic evaluation sessions. Fourth, we conducted a large user study to compare the efficiency, and accuracy of access review tasks using AuthzMap to two of the existing systems in this area. Our results showed that AuthzMap improved efficiency in five of the seven tested tasks, and improved accuracy in one of the seven tested tasks, while only negatively affected accuracy in one of the tested tasks.

In summary, this dissertation shows how multiple methods including usability guidelines and heuristics, field-studies, surveys, and lab studies can be combined together to address the social technical gap.

8.1 Contributions

The original contributions of this thesis are:

Guidelines for designing ITSM tools: We gathered guidelines and recommendations related to IT security management tools from the literature as well as from our own prior studies of IT security management. We categorized and combined these into a set of high level guidelines and identified the relationships between the guidelines and challenges in IT security management. We also illustrated the need for the guidelines, where possible, with quotes from interviews with five security practitioners. Our framework of guidelines can be used by those developing IT security tools, as well as by practitioners and managers evaluating tools.

Heuristics for evaluation of ITSM tools: We developed seven domain-specific usability heuristics for evaluation of IT security management tools, and named them ITSM heuristics. Furthermore, with a between-subjects study, we compared the employment of the ITSM and Nielsen's heuristics for evaluation of a commercial identity and access management system. The results of our evaluation validated the effectiveness of the heuristics. We showed that participants who used the ITSM set found more problems categorized as severe than those who used Nielsen's. We analyzed several aspects of our heuristics including the performance of individual participants using the heuristic, the performance of individual heuristics, the similarity of our heuristics to Nielsen's, and the participants' opinion about the use of heuristics for evaluation of IT security tools. Our heuristics can be used by usability experts, or other practitioners in usability evaluation of ITSM tools.

A detailed process for building usability heuristics: We explored how domain specific usability heuristics are created by examining prior research in the area of heuristic and guideline creation. We classified prior heuristic creation literature according to its methodology, and discussed the benefits and drawback of each approach. We then described our

methodical way of creating domain specific usability heuristics, which we applied to create a set of usability heuristics for evaluation of ITSM tools. Our method in heuristic creation can be adopted by other researchers for creation of domain specific heuristics.

A thick description of IAM and its challenges: We conducted 19 interviews with security practitioners to understand how identity and access management is done in organizations, and why it is challenging. We used grounded theory to analyze and understand data. We then provided a thick description of IAM and its challenges. Our findings can be used by researchers in the area of access control to identify open problems, and by security practitioners to better understand the IAM activities before getting involve in IAM adoption projects.

An abstract model of IAM activities and challenges: Using grounded theory methodology, we built two explanatory models to show the relationship between IAM activities, and the challenges in doing IAM. Our models explained why IAM is challenging, and why organizations choose to perform certain IAM related activities. Based on this understanding, we suggested a set of recommendations, and implications for design. Our results can be used by organizations or IAM tool developers to improve their practice or technology.

A detailed model of access review activity: We used our interview and survey data to explore the access review activity and identify its challenges. The study data revealed that access review involves challenges such as scale, technical complexity, the frequency of reviews, human errors, and exceptional cases. We also modeled access review in the activity theory framework. The model shows that access review requires an understanding of the activity context including information about the users, their job, their access privileges, and the history of access policy.

AuthzMap: We used our proposed ITSM guidelines to design a novel user interface named AuthzMap for understanding and reviewing users access. We improved AuthzMap through

multiple rounds of prototyping, informal feedback, and heuristic evaluation. We then conducted an exploratory user study with 430 participants to compare the use of AuthzMap with two existing commercial systems for access review. Our results show that AuthzMap improved efficiency of access review in five of the seven, and accuracy in one of the seven tasks. Our design of AuthzMap can be adopted by access review tool developers. Furthermore, our work highlighted the importance of context in access control, and suggested an effective approach for integrating contextual information in access control interfaces.

8.2 Future Work

Although this dissertation has presented several research projects toward improving the usability of ITSM and IAM tools, there are certain areas than can be studied in future research. We detail each future research direction in this section.

8.2.1 Improvements on AuthzMap

While our user study suggested that AuthzMap was more effective than existing interfaces, there are still certain areas to improve the interface:

Cosmetic Improvements: Our user study suggested that a set of cosmetic improvements may increase the efficiency of using AuthzMap, and prevent errors. First, we show that the names of the applications were presented in a small vertical text and it can have a negative impact on human cognition. To address this, we can use slightly less rotated text (e.g., 45 degrees), as it is shown that rotation is positively correlated with cognitive load. Furthermore, we show that the large number of grid cells that are in close proximity can cause slips, and reduce the performance by inducing cognitive load on users. Therefore, Authzmap can be improved with a focus plus context visualization developed by Leung

and Apperley (1994), by allowing users click on the column to focus on a specific file, and then perform the actions. Alternatively, rows and columns of the current grid cell can be highlighted as the user hovers the mouse over the grill cells.

Detail in Context: AuthzMap used three levels of abstraction for visualizing access control data. We used hierarchical windowing to show different levels of the interface. According to Leung and Apperley (1994), windowing is considered a nondistortion-oriented technique, and while it helps in navigating the access control data, it may not provide adequate context. Alternatively, a focus plus context technique (also called detail in context, or distortion-oriented) can be used. Such a technique shows the entire information (context), while showing the details of specific items (focal point) as well. One way to implement focus plus context in AuthzMap is to use a similar approach to Table Lens by Rao and Card (1994) and increase the size of the cells in the focal area, and decrease the size of the cells in the context area.

Use of Brushing: We designed a set of accelerators in AuthzMap to help users certify or revoke access to one access privilege for multiple users. In the user comparison task, we saw that the use of accelerators on similar users impacted the performance of participants. We can extend the use of accelerators in the interface by adopting the brushing technique by Roberts and Wright (2006). Brushing would allow reviewers to select a subset of user-to-privilege assignments by drawing a rectangle directly on the AuthzMap interface, and applying the appropriate certification decision on the chosen set of user-to-privilege assignments.

Extending the context: In Chapter 6, we introduced a set of contextual information to be visualized in the interface. While we chose the information based on the interviews, we saw cases were companies expressed their need for other contextual information. Therefore, one future improvement for AuthzMap is to visualize information such as: projects the user is working on, geographical location of the users, the risk associated with each

access privilege, etc. on the interface. Ideally, the production version of the interface should allow reviewers to customize the interface to show the information appropriate in the context of their organization.

Applying the expandable grid concept: In designing AuthzMap we assumed that users, and access privileges have a flat structure. In many organizations, access privileges can be combined as roles, and users can be combined in groups. Therefore, AuthzMap can adopt an approach similar to expandable grid by Reeder et al. (2008) and allow users to expand groups and roles in a tree structure, and see the individual members of each role and group.

Automation opportunities: On of the proposed guidelines in Chapter 3 is *Providing Automatic Detection*. We did not apply this guideline in the design of AuthzMap, but there are opportunities for automating certain parts of decision making during access review. First, risk associated with each user-to-privilege assignment can be automatically determined using a machine-learning technique by analyzing previous access review decisions. In addition, a heuristic based approach can be used to determine the risk. Second, safe privileges (e.g., access to the email) can be automatically certified, so that the reviewer can focus only on the high risk ones. Third, risk associated with each user can be calculated, and users can be prioritized based on their risk. While automation will free cognitive resources, there is a danger of false-positive, and false negatives.

8.2.2 Field-study of IAM

During our field study, we identified a set of IAM related activities that were outside the scope of our theory. We filtered the categories identified about these activities during our selective coding stage. As a future direction, one can focus on further understanding of the following two activities:

IAM deployment and configuration: We found that IAM deployment is a challenging activity. It involved satisfying a set of pre-requisites, choosing a vendor, and going through an intensive deployment, configuration, and testing.

IAM from end-users perspective: We observed that end-users were involved in IAM activities by performing few tasks: (1) Changing and Reseting Passwords; and (2) Access Delegation. We found that during these two tasks, end-users usually broke the company's policy by choosing weak passwords, or sharing their credentials with other employees.

8.2.3 Heuristic Evaluation of Other IT Security Tools

There are few areas that we can extend our research of ITSM heuristics. As we discussed in Chapter 4, we did not evaluate the downstream utility of ITSM heuristics. As a future work, we can design a study to evaluate the downstream utility of ITSM heuristics. To do this, an IT security tool deployed in an actual organization can be selected, and evaluated using ITSM heuristics. After addressing the identified problems, the tool can be re-deployed in the field. The users of the tool can be surveyed (or interviewed) before and after deployment of the tool, in order to get quantitative or qualitative feedback on the impact of evaluating and improving the tool using heuristic evaluation.

Another future step in evaluation of ITSM heuristics would be to choose another ITMS tool in one of the nine areas listed in Chapter 2, and the study presented in Chapter 4 can be replicated using the new system. This will contribute to the generalizability of our results.

8.2.4 Future Work on Guidelines for ITSM Tools

In Chapter 3, we have attempted to show the relationship between guidelines and challenges to ITSM. These relationships could help SPs and tool developers to decide about the importance of each guideline. However, the understanding of each guideline importance can be

investigated further. Based on Koyani et al. (2006) successful experience with the development of design guidelines, we suggest a survey of SPs on the importance of each guideline. Furthermore, the strength of evidence for each guideline can be determined by evaluating the methodology used in each cited source generating guidelines, as well as whether the guidelines were generated by studying specific user populations (e.g., security practitioners, system administrators) and whether they were generated considering specific security tools (e.g., IDS).

Another future direction of developing guidelines is to build a database of usability patterns that can realize each guideline. An approach similar to Garfinkel (2005) can be used to relate each guideline with multiple patterns. Then for each pattern, information such as the name, intent, motivation, illustrative image, applicability, implementation, and known uses can be incorporated into the database. Such a database can help developers without usability knowledge to use proven patterns rather than relying on their own knowledge to realize each guideline.

Bibliography

Abdullah, K., Lee, C., Conti, G., Copeland, J. A., and Stasko, J. (2005). IDS RainStorm: Visualizing ids alarms. In *VIZSEC '05: Proceedings of the IEEE Workshops on Visualization for Computer Security*, pages 1–10, Minneapolis, MN, USA. IEEE Computer Society. → pages 29, 30, 39

Ackerman, M. S. (2000). The intellectual challenge of CSCW: the gap between social requirements and technical feasibility. *Hum.-Comput. Interact.*, 15(2):179–203. → pages 2, 230

Andrew, C. (2005). The five ps of patch management: Is there a simple way for businesses to develop and deploy an advanced security patch management strategy? *Computers and Security*, 24(5):362–363. → pages 36, 37

Baker, K., Greenberg, S., and Gutwin, C. (2002). Empirical development of a heuristic evaluation methodology for shared workspace groupware. In *Proceedings of the 2002 ACM conference on Computer supported cooperative work*, CSCW '02, pages 96–105, New York, NY, USA. ACM. → pages xvi, 3, 45, 47, 49, 62, 74, 83, 84, 88

Baldonado, M. Q. W., Woodruff, A., and Kuchinsky, A. (2000). Guidelines for using multiple views in information. In *AVI '00: Proceedings of the working conference on Advanced visual interfaces*, pages 110–119, Palermo, Italy. ACM. → pages 20, 29, 30

Ball, R., Fink, G. A., and North, C. (2004). Home-centric visualization of network traffic for security administration. In *VizSEC/DMSEC '04: Proceedings of the 2004 ACM workshop on Visualization and data mining for computer security*, pages 55–64, Fairfax, VA, USA. ACM. → pages 29, 30

Barrett, R., Kandogan, E., Maglio, P. P., Haber, E. M., Takayama, L. A., and Prabaker, M. (2004). Field studies of computer system administrators: Analysis of system management tools and practices. In *Proceedings of the 2004 ACM Conference on Computer Supported Cooperative Work*, CSCW '04, pages 388–395, New York, NY, USA. ACM. → pages 11, 22, 32, 36, 37

Barrett, R., Maglio, P. P., Kandogan, E., and Bailey, J. (2005). Usable autonomic computing systems: The system administrators' perspective. *Advanced Engineering Informatics*, 19(3):213–221. → pages 11, 22, 32, 36, 37

Bauer, L., Cranor, L. F., Reeder, R. W., Reiter, M. K., and Vaniea, K. (2009). Real life challenges in access-control management. In *CHI '09: Proceedings of the 27th international conference on Human factors in computing systems*, pages 899–908, New York, NY, USA. ACM. → pages 3, 13, 56, 147, 148, 150, 157, 160

Beal, B. (2005). IT security: the product vendor landscape. *Network Security*, 2005(5):9–10. → pages 19, 27, 35, 43, 57

Beckerle, M. and Martucci, L. A. (2013). Formal definitions for usable access control rule sets from goals to metrics. In *Proceedings of the Ninth Symposium on Usable Privacy and Security*, SOUPS '13, pages 2:1–2:11, New York, NY, USA. ACM. → pages 3, 161, 201

Blakley, B. (1996). The emperor's old armor. In *Proceedings of the 1996 workshop on New security paradigms*, NSPW '96, pages 2–16, New York, NY, USA. ACM. → pages 147

Blum, D. (2005). Identity management - concepts and definitions. Burton Group. → pages 10, 90

Blumer, H. (1954). What is wrong with social theory? *American Sociological Review*, 19(1):3–10. → pages 230

Botta, D., Muldner, K., Hawkey, K., and Beznosov, K. (2011). Toward understanding distributed cognition in it security management: the role of cues and norms. *Cognition, Technology & Work*, 13(2):121–134. → pages 12, 54, 57, 58, 60, 61

Botta, D., Werlinger, R., Gagné, A., Beznosov, K., Iverson, L., Fels, S., and Fisher, B. (2007). Towards understanding IT security professionals and their tools. In *Proc. of Symp. On Usable Privacy and Security (SOUPS)*, pages 100–111, Pittsburgh, PA. → pages 1, 2, 11, 12, 18, 22, 27, 28, 30, 33, 34, 35, 44, 56, 57, 59, 61

Brodie, C., Karat, C.-M., Karat, J., and Feng, J. (2005). Usable security and privacy: a case study of developing privacy management tools. In *SOUPS '05: Proceedings of the 2005 symposium on Usable privacy and security*, pages 35–43, New York, NY, USA. ACM. → pages 3, 160

Brown, G. and Smith, R. (2007). Identity management survey 2007 - findings report. Technical report. → pages 13

Burns, C. M., Kuo, J., and Ng, S. (2003). Ecological interface design: a new approach for visualizing network management. *Comput. Netw.*, 43(3):369–388. → pages 30

Burton Group (2005). Root document enterprise identity management: Moving from theory to practice. Technical report, Burton Group. → pages 12

Carroll, J. M., Neale, D. C., Isenhour, P. L., Rosson, M. B., and McCrickard, D. S. (2003). Notification and awareness: synchronizing task-oriented collaborative activity. *International Journal of Human-Computer Studies*, 58(5):605 – 632. Notification User Interfaces. → pages 55

Carroll, J. M. and Rosson, M. B. (1992). Getting around the task-artifact cycle: how to make claims and design by scenario. *ACM Trans. Inf. Syst.*, 10(2):181–212. → pages 47, 49

Carroll, J. M., Smith-Kerker, P. L., Ford, J. R., and Mazur-Rimetz, S. A. (1987). The minimal manual. *Human–Computer Interaction*, 3(2):123–153. → pages 200, 201

Centers for Medicare & Medicaid Services (1996). The Health Insurance Portability and Accountability Act of 1996 (HIPAA). Online at http://www.cms.hhs.gov/hipaa/. → pages 158

Charmaz, K. (2006). *Constructing Grounded Theory*. SAGE publications. → pages 23, 47, 92, 95

Chiasson, S., van Oorschot, P. C., and Biddle, R. (2007). Even experts deserve usable security: Design guidelines for security management systems. In *SOUPS Workshop on Usable IT Security Management (USM)*, pages 1–4, Pittsburgh, PA. → pages 2, 18, 20, 22, 28, 30, 44

Coffey, A. and Atkinson, P. (1996). *Making Sense of Qualitative Data: Complementary Research Strategies*. SAGE Publications. → pages 24

Connor, A. C. O. and Loomis, R. J. (2010). 2010 economic analysis of role-based access control. National Institute of Standards and Technology. → pages xiv, xvii, 144, 145

Considine, J., Botti, M., and Thomas, S. (2005). Design, format, validity and reliability of multiple choice questions for use in nursing research and education. *Collegian*, 12(1):19 – 24. → pages 201

Convertino, G., Mentis, H. M., Slavkovic, A., Rosson, M. B., and Carroll, J. M. (2011). Supporting common ground and awareness in emergency management planning: A design research project. *ACM Trans. Comput.-Hum. Interact.*, 18(4):22:1–22:34. → pages 200, 202

Corbin, J. and Strauss, A. (1990). *Basics of Qualitative Research: Grounded Theory Procedures and Techniques*. Sage, Newbury Park, CA. → pages 16, 17, 95

Creswell, J. W. and Miller, D. L. (2000). Determining validity in qualitative inquiry. *Theory Into Practice*, 39(3):124–130. → pages 234

Cser, A. (2009). The Forrester Wave: Identity And Access Management, Q4 2009. Technical report, Forrester Research. → pages xiii, 10, 11

Cser, A. (2011). The forrester wave: Role management and access recertification, Q3 2011. Technical report, Forrester Research, inc. → pages 158, 190

DiGioia, P. and Dourish, P. (2005). Social navigation as a model for usable security. In *SOUPS '05*, pages 101–108, Pittsburgh, Pennsylvania. ACM. → pages 20, 28

Dijker, B. (2006). A day in the life of system administrators. http://sageweb.sage.org. → pages 31

Dormann, W. and Rafail, J. (2008). Securing your web browser. Technical report, US-CERT (United States Computer Emergency Readiness Team). → pages 227

Dourish, P. (2001). Seeking a foundation for context-aware computing. *Hum.-Comput. Interact.*, 16(2):229–241. → pages 51

Dourish, P. and Redmiles, D. (2002). An approach to usable security based on event monitoring and visualization. In *NSPW '02: Proceedings of the 2002 workshop on New security paradigms*, pages 75–81, New York, NY, USA. ACM. → pages 57

Elliott, M. and Kling, R. (1997). Organizational usability of digital libraries: Case study of legal research in civil and criminal courts. *American Society for Information Science*, 4(11):1023–1035. → pages 42

Ene, A., Horne, W., Milosavljevic, N., Rao, P., Schreiber, R., and Tarjan, R. E. (2008). Fast exact and heuristic methods for role minimization problems. In *SACMAT '08: Proceedings of the 13th ACM symposium on Access control models and technologies*, pages 1–10, New York, NY, USA. ACM. → pages 227

Engeström, Y. (1999). Activity theory and individual and social transformation. *Perspectives on activity theory*, pages 19–38. → pages 15, 61, 62, 138, 166

Engeström, Y. (2001). Expansive learning at work: Toward an activity theoretical reconceptualization. *Journal of Education and Work*, 14(1):133–156. → pages 15, 52

Erickson, T. and Kellogg, W. A. (2000). Social translucence: an approach to designing systems that support social processes. *ACM Trans. Comput.-Hum. Interact.*, 7(1):59–83. → pages 55

Ferraiolo, D. F., Sandhu, R., Gavrila, S., Kuhn, D. R., and Chandramouli, R. (2001). Proposed NIST standard for role-based access control. *ACM Transactions on Information and System Security*, 4(3):224–274. → pages 142

Ferreira, A., Cruz-Correia, R., Antunes, L., Farinha, P., Oliveira-Palhares, E., Chadwick, D. W., and Costa-Pereira, A. (2006). How to break access control in a controlled manner. *Computer-Based Medical Systems, IEEE Symposium on*, 0:847–854. → pages 148, 149

Fogg, B. J. (2002). Persuasive technology: using computers to change what we think and do. *Ubiquity*, 2002(December):2. → pages 20

Forget, A., Chiasson, S., and Biddle, R. (2012). Supporting learning of an unfamiliar authentication scheme. In *World Conference on E-Learning in Corporate, Government, Healthcare, and Higher Education*, volume 2012, pages 1002–1011. → pages 201

Fuchs, L., Pernul, G., and Sandhu, R. (2011). Roles in information security – a survey and classification of the research area. *Computers & Security*, 30(8):748 – 769. → pages 13

Furniss, D., Blandford, A., and Curzon, P. (2011). Confessions from a grounded theory phd: experiences and lessons learnt. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '11, pages 113–122, New York, NY, USA. ACM. → pages 16

Gagné, A., Muldner, K., and Beznosov, K. (2008). Identifying differences between security and other IT professionals: a qualitative analysis. In *HAISA'08: Human Aspects of Information Security and Assurance*, pages 69–80, Plymouth, England. → pages 2, 12, 20, 22, 32, 41, 55

Gallaher, M. P., Oconnor, A. C., and Kropp, B. (2002). The Economic Impact of Role-Based Access Control. → pages 144

Garfinkel, S. L. (2005). *Design principles and patterns for computer systems that are simultaneously secure and usable*. PhD thesis, Massachusetts Institute of Technology, Cambridge, MA, USA. Adviser-David D. Clark and Adviser-Robert C. Miller. → pages 246

Garigue, R. and Stefaniu, M. (2003). Information security governance reporting. *The EDP Audit, Control, and Security Newsletter(EDPACS)*, 31(6):11–17. → pages 34

Glaser, B. and Holton, J. (2004). Remodeling grounded theory. *Forum Qualitative Sozialforschung / Forum: Qualitative Social Research*, 5(2). → pages 16

Glaser, B. and Strauss, A. L. (1967). *The Discovery of Grounded Theory, Strategies for Qualitative Research*. Aldine Publishing Company, Chicago, Illinois. → pages 16, 47, 95

Glaser, B. G. (1978). *Theoretical sensitivity : advances in the methodology of grounded theory*. Sociology Press, Mill Valley, CA. → pages 95

Goodall, J. R., Lutters, W. G., and Komlodi, A. (2004). I know my network: Collaboration and expertise in intrusion detection. In *CSCW '04*, pages 342–345. → pages 11

Grance, T., Stevens, M., and Myers, M. (2003). NIST Special Publication 800-36, Guide to selecting information technology security products. Technical report, National Institute for Standards and Technology. → pages 9

Greenberg, S., Fitzpatrick, G., Gutwin, C., and Kaplan, S. (2000). Adapting the locales framework for heuristic evaluation of groupware. *Australian Journal of Information Systems*, 7(2):102–108. → pages 45, 49, 62

Grunwald, T. and Corsbie-Massay, C. (2006). Guidelines for cognitively efficient multimedia learning tools: educational strategies, cognitive load, and interface design. *Academic medicine*, 83(3):213–223. → pages 20

Haber, E. M. and Bailey, J. (2007). Design guidelines for system administration tools developed through ethnographic field studies. In *CHIMIT '07: Proceedings of the 2007 symposium on Computer human interaction for the management of information technology*, pages 1:1–1:9, New York, NY, USA. ACM. → pages 2, 11, 20, 22, 27, 28, 35, 36, 37, 38, 39, 54, 57, 61

Halverson, C. A. (2002). Activity theory and distributed cognition: Or what does cscw need to do with theories? *Comput. Supported Coop. Work*, 11(1-2):243–267. → pages 14

Halverson, C. A. (2004). The value of persistence: A study of the creation, ordering and use of conversation archives by a knowledge worker. In *HICSS '04: Proceedings of the 37th Annual Hawaii International Conference on System Sciences*, pages 1–10, Washington, DC, USA. IEEE Computer Society. → pages 33

Hartson, H. R., Andre, T. S., and Williges, R. C. (2001). Criteria for evaluating usability evaluation methods. *International Journal of Human-Computer Interaction*, 13(4):373–410. → pages 63, 70, 73

Hawkey, K., Botta, D., Werlinger, R., Muldner, K., Gagne, A., and Beznosov, K. (2008a). Human, Organizational, and Technological Factors of IT Security. In *CHI'08 extended abstract on Human factors in computing systems*, pages 3639–3644, Florence, Italy. → pages 11

Hawkey, K., Muldner, K., and Beznosov, K. (2008b). Searching for the Right Fit: Balancing IT Security Model Trade-offs. *Special Issue on Useful Computer Security, IEEE Internet Computing*, 12(3):22–30. → pages 12, 35

Heckle, R., Lutters, W. G., and Gurzick, D. (2008). Network authentication using single sign-on:the challenge of aligning mental models. In *CHIMIT '08: Proceedings of the 2008 symposium on Computer Human Interaction for the Management of Information Technology*, Cambridge, Massachusetts. ACM. → pages 14

Herzog, A. and Shahmehri, N. (2007). User help techniques for usable security. In *CHIMIT '07: Proceedings of the 2007 symposium on Computer Human Interaction for the Management of Information Technology*, pages 93–102, Cambridge, Massachusetts. ACM. → pages 27

Hollan, J., Hutchins, E., and Kirsh, D. (2000). Distributed cognition: toward a new foundation for human-computer interaction research. *ACM Trans. Comput.-Hum. Interact.*, 7(2):174–196. → pages 51, 56, 181

Hornbaek, K. and Frokjaer, E. (2001). Reading of electronic documents: the usability of linear, fisheye, and overview+detail interfaces. In *CHI '01: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 293–300, New York, NY, USA. ACM. → pages 30

Hughes, J. A., Randall, D., and Shapiro, D. (1992). Faltering from ethnography to design. In *Proceedings of the 1992 ACM Conference on Computer-supported Cooperative Work*, CSCW '92, pages 115–122, New York, NY, USA. ACM. → pages 231

Hutchins, E. (1995). *Cognition in the Wild*. MIT Press, Cambridge, MA. → pages 230

Inglesant, P., Sasse, M. A., Chadwick, D., and Shi, L. L. (2008). Expressions of expertness: the virtuous circle of natural language for access control policy specification. In *SOUPS*

'08: Proceedings of the 4th symposium on Usable privacy and security, pages 77–88, New York, NY, USA. ACM. → pages 3, 160

ISO (2009a). Information technology – Security techniques – A framework for identity management. ISO ISO/IEC 24760-1:2011, International Organization for Standardization, Geneva, Switzerland. → pages 10

ISO (2009b). Information technology - Security techniques - A framework for access management. ISO ISO/IEC CD 29146, International Organization for Standardization, Geneva, Switzerland. → pages 10

Jaferian, P., Botta, D., Hawkey, K., and Beznosov, K. (2009). A case study of enterprise identity management system adoption in an insurance organization. In *Proceedings of the Symposium on Computer Human Interaction for the Management of Information Technology*, pages 46–55. ACM. → pages 235

Jeffries, R., Miller, J. R., Wharton, C., and Uyeda, K. (1991). User interface evaluation in the real world: a comparison of four techniques. In *CHI '91: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 119–124, New York, NY, USA. ACM. → pages 45

Jick, T. D. (1979). Mixing qualitative and quantitative methods: Triangulation in action. *Administrative Science Quarterly*, 24(4):pp. 602–611. → pages 234

Jolicoeur, P. (1985). The time to name disoriented natural objects. *Memory & Cognition*, 13(4):289–303. → pages 224

Kandogan, E. and Haber, E. M. (2005). Security administration tools and practices. In Cranor, L. F. and Garfinkel, S., editors, *Security and Usability: Designing Secure Systems that People Can Use*, chapter 18, pages 357–378. O'Reilly Media, Inc. → pages 11, 22, 27, 28, 39, 40

Kaptelinin, V. and Nardi, B. (2006). *Acting with technology: Activity theory and interaction design*. MIT Press. → pages 15, 44, 51, 52, 59, 181, 182

Kaptelinin, V., Nardi, B., Bodker, S., Carroll, J., Hollan, J., Hutchins, E., and Winograd, T. (2003). Post-cognitivist HCI: second-wave theories. In *CHI '03: CHI '03 extended abstracts on Human factors in computing systems*, pages 692–693, New York, NY, USA. ACM. → pages 51

Kaptelinin, V., Nardi, B. A., and Macaulay, C. (1999). Methods & tools: The activity checklist: a tool for representing the space of context. *interactions*, 6(4):27–39. → pages 178

Kern, A. (2002). Advanced features for enterprise-wide role-based access control. In *Computer Security Applications Conference, 2002. Proceedings. 18th Annual*, pages 333–342. → pages 146, 151

Kern, A., Kuhlmann, M., Schaad, A., and Moffett, J. (2002). Observations on the role life-cycle in the context of enterprise security management. In *Proceedings of the 7th ACM symposium on Access control models and technologies (SACMAT)*, pages 43–51. → pages 4, 143, 151, 152

Kesh, S. and Ratnasingam, P. (2007). A knowledge architecture for it security. *Commun. ACM*, 50(7):103–108. → pages 28, 60

Killcrece, G., Kossakowski, K.-P., Ruefle, R., and Zajicek, M. (2003). Organizational models for computer security incident response teams (CSIRTS). Technical Report CMU/SEI-2003-HB-001. → pages 32

Kittur, A., Chi, E. H., and Suh, B. (2008). Crowdsourcing user studies with mechanical turk. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '08, pages 453–456, New York, NY, USA. ACM. → pages 200

Komlod, A., Rheingans, P., Ayachit, U., Goodall, J., and Joshi, A. (2005). A user-centered look at glyph-based security visualization. In *VIZSEC '05: Proceedings of the IEEE Workshops on Visualization for Computer Security*, pages 21–28, Minneapolis, MN, USA. → pages 29

Kotulic, A. G. and Clark, J. G. (2004). Why there aren't more information security research studies. *Information & Management*, 41(5):597–607. → pages 44

Koyani, S. J., Bailey, R. W., and Nall, J. R. (2006). *Research-Based Web Design & Usability Guidelines*. U.S. Dept. of Health and Human Services. → pages 20, 246

Kraemer, S. and Carayon, P. (2007). Human errors and violations in computer and information security: The viewpoint of network administrators and security specialists. *Applied Ergonomics*, 38:143–154. → pages 11, 31

Kuhn, D. R., Coyne, E. J., and Weil, T. R. (2010). Adding attributes to role-based access control. *Computer*, 43(6):79–81. → pages 143

Kuutti, K. (1995). *Activity theory as a potential framework for human-computer interaction research*, pages 17–44. Massachusetts Institute of Technology, Cambridge, MA, USA. → pages 14

Lampson, B. W. (1971). Protection. In *5th Princeton Conference on Information Sciences and Systems*, page 437, New York, NY, USA. ACM Press. → pages 141, 186

Lee, C. P. and Copeland, J. A. (2006). Flowtag: a collaborative attack-analysis, reporting, and sharing tool for security researchers. In *VizSEC '06: Proceedings of the 3rd international workshop on Visualization for computer security*, pages 103–108, Alexandria, VA, USA. ACM. → pages 28

Leontiev, A. (1974). The problem of activity in psychology. *Journal of Russian and East European Psychology*, 13(2):4 – 33. → pages 14

Leung, Y. K. and Apperley, M. D. (1994). A review and taxonomy of distortion-oriented presentation techniques. *ACM Trans. Comput.-Hum. Interact.*, 1(2):126–160. → pages 222, 224, 242, 243

Maglio, P. P., Kandogan, E., and Haber, E. (2003). Distributed cognition and joint activity in collaborative problem solving. In *Proceedings of the Twenty-fifth Annual Conference of the Cognitive Science Society*. → pages 57

Mankoff, J., Dey, A. K., Hsieh, G., Kientz, J., Lederer, S., and Ames, M. (2003). Heuristic evaluation of ambient displays. In *Proc. CHI '03*, pages 169–176, New York, NY, USA. ACM. → pages 3, 45, 47, 49, 62

Matavire, R. and Brown, I. (2008). Investigating the use of "grounded theory" in information systems research. In *Proceedings of the 2008 Annual Research Conference of the South African Institute of Computer Scientists and Information Technologists on IT Research in Developing Countries: Riding the Wave of Technology*, SAICSIT '08, pages 139–147, New York, NY, USA. ACM. → pages 16

McCloskey, M. (2014). Turn user goals into task scenarios for usability testing. → pages 203

McGann, S. and Sicker, D. C. (2005). An analysis of security threats and tools in SIP-based VoIP systems. In *2nd VoIP Security Workshop*, pages 1–8, Washington DC, USA. → pages 31, 34

McGrath, J. E. (1995). Methodology matters: doing research in the behavioral and social sciences. *Human-computer interaction: toward the year 2000*, pages 152–169. Morgan Kaufmann Publishers Inc. → pages 236

McLeish, S. (2007). Institutional audit case studies process final report. Technical report. → pages 13

Microsoft Corporation (2006). Microsoft identity and access management series: Fundamental concepts. → pages 90

Muller, M. J. and McClard, A. (1995). Validating an extension to participatory heuristic evaluation: quality of work and quality of work life. In *CHI '95: Conference companion on Human factors in computing systems*, pages 115–116, New York, NY, USA. ACM. → pages 45

Nardi, B. A., editor (1995). *Context and consciousness: activity theory and human-computer interaction*. Massachusetts Institute of Technology, Cambridge, MA, USA. → pages 51

Neale, D. C., Carroll, J. M., and Rosson, M. B. (2004). Evaluating computer–supported cooperative work: models and frameworks. In *CSCW '04*, pages 112–121. ACM Press. → pages 44

Nielsen, J. (1992). Finding usability problems through heuristic evaluation. In *Proc. CHI '92*, pages 373–380, New York, NY, USA. ACM. → pages 69, 78, 226

Nielsen, J. (1994). Enhancing the explanatory power of usability heuristics. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, CHI '94, pages 152–158, New York, NY, USA. ACM. → pages 46, 47, 49, 62, 65

Nielsen, J. (1995). Applying discount usability engineering. *IEEE Software*, 12(1):98–100. → pages 3, 26, 38

Nielsen, J. (2005a). How to conduct a heuristic evaluation. http://www.useit.com/papers/heuristic/ heuristic_evaluation.html. → pages 45, 46, 67, 69, 153

Nielsen, J. (2005b). Severity ratings for usability problems. http://www.useit.com/papers/heuristic/ severityrating.html. → pages 189

Nielsen, J. (2012). Usability 101: Introduction to usability. http://www.nngroup.com/articles/usability-101-introduction-to-usability/. → pages 198

Nielsen, J. and Molich, R. (1990). Heuristic evaluation of user interfaces. In *CHI '90: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 249–256, New York, NY, USA. ACM. → pages xvi, 3, 44, 45, 73, 74, 83, 84, 175

Nohlberg, M. and Backstrom, J. (2007). User-centred security applied to the development of a management information system. *Information Management & Computer Security*, 15(5):372–381. → pages 33, 34

Norman, D. A. (1988). *The Psychology of Everyday Things*. Basic Books. → pages 202

Norman, D. A. (1991). Cognitive artifacts. *Designing interaction: Psychology at the human-computer interface*, pages 17–38. → pages 58

Norman, D. A. and Draper, S. W. (1986). *User Centered System Design; New Perspectives on Human-Computer Interaction*, chapter 3, pages 31–61. L. Erlbaum Associates Inc., Hillsdale, NJ, USA. → pages 45

Olson, G. and Moran, T. (1998). Commentary on "Damaged Merchandise?". *Human-Computer Interaction*, 13(3):263–323. → pages 63

Osborn, S., Sandhu, R., and Munawer, Q. (2000). Configuring role-based access control to enforce mandatory and discretionary access control policies. *ACM Transactions on Information and System Security (TISSEC)*, 3(2):85–106. → pages 142

Penn, J. (2009). Market overview: IT security in 2009. Technical report, Forrester Research. → pages 1, 9

Pinelle, D., Wong, N., and Stach, T. (2008). Heuristic evaluation for games: usability principles for video game design. In *Proc. CHI '08*, pages 1453–1462, New York, NY, USA. ACM. → pages 45, 47, 49, 62

Polson, P. G., Lewis, C., Rieman, J., and Wharton, C. (1992). Cognitive walkthroughs—a method for theory-based evaluation of user interfaces. *International Journal of Man-Machine Studies*, 36(5):741–773. → pages 45

Post, G. V. and Kagan, A. (2007). Evaluating information security tradeoffs: Restricting access can interfere with user tasks. *Computers & Security*, 26(3):229 – 237. → pages 14

Povey, D. (2000). Optimistic security: A new access control paradigm. In *Proceedings of the 1999 Workshop on New Security Paradigms*, NSPW '99, pages 40–45, New York, NY, USA. ACM. → pages 147, 148

Rabardel, P. and Bourmaud, G. (2003). From computer to instrument system: a developmental perspective. *Interacting with Computers*, 15(5):665 – 691. From Computer Artefact to Instrument for Mediated Activity. Part 1 Organizational Issues. → pages 58

Rao, R. and Card, S. K. (1994). The table lens: Merging graphical and symbolic representations in an interactive focus + context visualization for tabular information. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '94, pages 318–322, New York, NY, USA. ACM. → pages 243

Rayford B. Vaughn Jr., R. H. and Fox, K. (2001). An empirical study of industrial security-engineering practices. *The Journal of Systems and Software*, 61:225–232. → pages 18

Reeder, R. W., Bauer, L., Cranor, L. F., Reiter, M. K., Bacon, K., How, K., and Strong, H. (2008). Expandable grids for visualizing and authoring computer security policies. In *Proc. CHI '08*, pages 1473–1482, New York, NY, USA. ACM. → pages 3, 157, 160, 201, 244

Reeder, R. W., Bauer, L., Cranor, L. F., Reiter, M. K., and Vaniea, K. (2011). More than skin deep: measuring effects of the underlying model on access-control system usability. In *Proceedings of the 2011 annual conference on Human factors in computing systems*, CHI '11, pages 2065–2074, New York, NY, USA. ACM. → pages 141, 142, 201

Roberts, J. and Wright, M. (2006). Towards ubiquitous brushing for information visualization. In *Information Visualization, 2006. IV 2006. Tenth International Conference on*, pages 151–156. → pages 243

Rogers, Y. (1992). Ghosts in the network: distributed troubleshooting in a shared working environment. In *CSCW '92: Proceedings of the 1992 ACM conference on Computer-supported cooperative work*, pages 346–355, Toronto, ON, Canada. ACM. → pages 28, 60

Rogers, Y. (2012). HCI theory: classical, modern, and contemporary. *Synthesis Lectures on Human-Centered Informatics*, 5(2):1–129. → pages 14

Rosson, M. B. and Carroll, J. M. (2002). *Usability engineering: scenario-based development of human-computer interaction*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA. → pages 67

Samarati, P. and Vimercati, S. D. C. d. (2001). Access control: Policies, models, and mechanisms. In *FOSAD '00: Revised versions of lectures given during the IFIP WG 1.7 International School on Foundations of Security Analysis and Design on Foundations of Security Analysis and Design*, pages 137–196, London, UK. Springer-Verlag. → pages 140

Sandhu, R. and Samarati, P. (1994). Access control: Principles and practice. *IEEE Communications Magazine*, 32(9):40–48. → pages 140, 141, 142

Sarbanes, P. (2002). Sarbanes-Oxley Act of 2002. In *The Public Company Accounting Reform and Investor Protection Act. Washington DC: US Congress*. → pages 55

Sas, C., Whittaker, S., Dow, S., Forlizzi, J., and Zimmerman, J. (2014). Generating implications for design through design research. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '14, pages 1971–1980, New York, NY, USA. ACM. → pages 230, 233

Schaad, A., Moffett, J., and Jacob, J. (2001). The role-based access control system of a european bank: a case study and discussion. In *SACMAT '01: Proceedings of the sixth ACM symposium on Access control models and technologies*, pages 3–9, New York, NY, USA. ACM. → pages 4, 149, 151

Scholtz, J. and Consolvo, S. (2004). Toward a framework for evaluating ubiquitous computing applications. *Pervasive Computing, IEEE*, 3(2):82 –88. → pages 45, 47, 49, 62

Scott, S. D., Grant, K. D., and Mandryk, R. L. (2003). System guidelines for co-located, collaborative work on a tabletop display. In *ECSCW'03: Proceedings of the eighth European Conference on Computer Supported Cooperative Work*, pages 159–178, Norwell, MA, USA. Kluwer Academic Publishers. → pages 20

Sedlmair, M., Isenberg, P., Baur, D., and Butz, A. (2011). Information visualization evaluation in large companies: Challenges, experiences and recommendations. *Information Visualization*, 10(3):248–266. → pages 225, 226

Shneiderman, B. (2000). Creating creativity: user interfaces for supporting innovation. *ACM Trans. Comput.-Hum. Interact.*, 7(1):114–138. → pages 55, 56

Shneiderman, B. and Plaisant, C. (2006). Strategies for evaluating information visualization tools: Multi-dimensional in-depth long-term case studies. In *Proceedings of the 2006 AVI Workshop on BEyond Time and Errors: Novel Evaluation Methods for Information Visualization*, BELIV '06, pages 1–7, New York, NY, USA. ACM. → pages 226

Shneiderman, B. and Plaisant, C. (2010). *Designing the User Interface: Strategies for Effective Human-Computer Interaction*. ADDISON WESLEY Publishing Company Incorporated. → pages 46

Siegel, D. A., Reid, B., and Dray, S. M. (2006). IT Security: Protecting Organizations In Spite of Themselves. *Interactions*, pages 20–27. → pages 11

Smetters, D. K. and Good, N. (2009). How users use access control. In *SOUPS '09: Proceedings of the 5th Symposium on Usable Privacy and Security*, pages 1–12, New York, NY, USA. ACM. → pages 4, 157, 160

Smith, S. L. and Mosier, J. N. (1986). Guidelines for designing user interface software. Technical Report ESD-TR-86-278, The MITRE Corporation Bedford MA. → pages 19, 26

Somervell, J. (2004). *Developing heuristic evaluation methods for large screen information exhibits based on critical parameters*. PhD thesis. AAI3136384. → pages 45, 47, 48, 49, 62

Somervell, J. and McCrickard, D. S. (2005). Better discount evaluation: illustrating how critical parameters support heuristic creation. *Interacting with Computers*, 17(5):592 – 612. → pages 3

Stevens, G. and Wulf, V. (2009). Computer-supported access control. *ACM Trans. Comput.-Hum. Interact.*, 16(3):1–26. → pages 4, 147, 148

Sutcliffe, A. and Gault, B. (2004). Heuristic evaluation of virtual reality applications. *Interacting with Computers*, 16(4):831 – 849. Human Computer Interaction in Latin America. → pages 45

Swensen, T. (2011). Wikileaked! wikileaked! what we can all learn from the bradley manning debacle. Novell connections magazine. → pages 158

Te'eni, D., Carey, J., and Zhang, P. (2007). *Human Computer Interaction: developing effective organizational information systems*. Wiley. → pages 46

Theng, Y. L., Duncker, E., Mohd-Nasir, N., Buchanan, G., and Thimbleby, H. W. (1999). Design guidelines and user-centred digital libraries. In *ECDL '99: Proceedings of the Third European Conference on Research and Advanced Technology for Digital Libraries*, pages 167–183, London, UK. Springer-Verlag. → pages 20

Thomas, R. and Sandhu, R. (1997). Task-based authorization controls (TBAC): A family of models for active and enterprise-oriented authorization management. In *Database Securty XI: Status and Prospects. Proceedings of the IFIP TC11 WG11.3 Eleventh International Conference on Database Security*, pages 166–181. → pages 143

Thomas, R. K. (1997). Team-based access control (TMAC): a primitive for applying role-based access controls in collaborative environments. In *Proceedings of the second ACM workshop on Role-based access control*, RBAC '97, pages 13–19, New York, NY, USA. ACM. → pages 143

Thompson, R. S., Rantanen, E., and Yurcik, W. (2006). Network intrusion detection cognitive task analysis: Textual and visual tool usage and recommendations. In *Proc. of Human Factors and Ergonomics Society Ann. Meeting (HFES)*, pages 669–673. → pages 29, 30, 39

Thompson, R. S., Rantanen, E. M., Yurcik, W., and Bailey, B. P. (2007). Command line or pretty lines?: comparing textual and visual interfaces for intrusion detection. In *CHI '07: Proceedings of the SIGCHI conference on Human factors in computing systems*, page 1205, San Jose, CA, USA. ACM. → pages 29, 57

United States Code (2002). Sarbanes-Oxley Act of 2002, pl 107-204, 116 stat 745. Online Document. → pages 158, 159

Vaidya, J., Atluri, V., and Guo, Q. (2007). The role mining problem: Finding a minimal descriptive set of roles. In *Proceedings of the 12th ACM Symposium on Access Control Models and Technologies (SACMAT)*, pages 175–184, Sophia Antipolis, France. ACM Press. → pages 129

Vaniea, K., Bauer, L., Cranor, L. F., and Reiter, M. K. (2012a). Out of sight, out of mind: Effects of displaying access-control information near the item it controls. In *Proceedings of the 2012 Tenth Annual International Conference on Privacy, Security and Trust (PST)*, PST '12, pages 128–136, Washington, DC, USA. IEEE Computer Society. → pages 160, 232

Vaniea, K., Bauer, L., Cranor, L. F., and Reiter, M. K. (2012b). Studying access-control usability in the lab: lessons learned from four studies. In *Proceedings of the 2012 Workshop on Learning from Authoritative Security Experiment Results*, LASER '12, pages 31–40, New York, NY, USA. ACM. → pages 160, 232

Velasquez, N. F. and Durcikova, A. (2008). Sysadmins and the need for verification information. In *CHiMiT '08: Proceedings of the 2nd ACM Symposium on Computer Human Interaction for Management of Information Technology*, pages 1–8, New York, NY, USA. ACM. → pages 55, 56

Velasquez, N. F. and Weisband, S. P. (2008). Work practices of system administrators: implications for tool design. In *CHiMiT '08: Proceedings of the 2nd ACM Symposium on Computer Human Interaction for Management of Information Technology*, pages 1–10, New York, NY, USA. ACM. → pages 57

Vicente, K. and Rasmussen, J. (1992). Ecological interface design: theoretical foundations. *Systems, Man and Cybernetics, IEEE Transactions on*, 22(4):589–606. → pages 20, 29

Vicente, K. J. (2000). HCI in the global knowledge-based economy: designing to support worker adaptation. *ACM Trans. Comput.-Hum. Interact.*, 7(2):263–280. → pages 59

von Solms, B. and von Solms, R. (2004). The 10 deadly sins of information security management. *Computers security*, 23(5):371. → pages 33

Vredenburg, K., Mao, J.-Y., Smith, P. W., and Carey, T. (2002). A survey of user-centered design practice. In *CHI '02: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 471–478, New York, NY, USA. ACM. → pages 45

Wang, Y., Norcie, G., Komanduri, S., Acquisti, A., Leon, P. G., and Cranor, L. F. (2011). 'i regretted the minute i pressed share': a qualitative study of regrets on facebook. In

Proceedings of the Seventh Symposium on Usable Privacy and Security, SOUPS '11, pages 10:1–10:16, New York, NY, USA. ACM. → pages 200

Werlinger, R., Hawkey, K., and Beznosov, K. (2008a). Human, Organizational and Technological Challenges of Implementing IT Security in Organizations. In *HAISA'08: Human Aspects of Information Security and Assurance*, pages 35–48, Plymouth, England. → pages 18, 21, 22, 27, 31, 33, 36, 37, 38, 39

Werlinger, R., Hawkey, K., and Beznosov, K. (2008b). Security practitioners in context: their activities and interactions. In *CHI '08 extended abstracts on Human factors in computing systems*, pages 3789–3794, Florence, Italy. → pages 22, 30, 31, 32, 35

Werlinger, R., Hawkey, K., and Beznosov, K. (2009a). An integrated view of human, organizational, and technological challenges of IT security management. *Journal of Information Management & Computer Security*, 17(1):4–19. → pages 1, 2, 12, 13, 58, 59, 60

Werlinger, R., Hawkey, K., Botta, D., and Beznosov, K. (2009b). Security practitioners in context: Their activities and interactions with other stakeholders within organizations. *International Journal of Human-Computer Studies*, 67(7):584–606. → pages 1, 2, 12, 43, 54, 57

Werlinger, R., Hawkey, K., Muldner, K., and Beznosov, K. (2009c). Towards understanding diagnostic work during the detection and investigation of security incidents. In *Proceedings of HAISA: Human Aspects of Information Security and Assurance*, pages 119–132, Athens, Greece. → pages 12

Werlinger, R., Hawkey, K., Muldner, K., Jaferian, P., and Beznosov, K. (2008c). The challenges of using an intrusion detection system: Is it worth the effort? In *Proceedings of the 4th Symposium On Usable Privacy and Security (SOUPS)*, pages 107–116, Pittsburgh, PA. → pages 2, 12, 38, 41

White, K. F. and Lutters, W. G. (2007). Midweight collaborative remembering: wikis in the workplace. In *CHIMIT '07: Proceedings of the 2007 symposium on Computer Human Interaction for the Management of Information Technology*, pages 111–112, Cambridge, MA, USA. ACM. → pages 28

Wright, J. (2007). Final progress reports. http://www.jisc.ac.uk/media/documents/programmes /einfrastructure/tidpfinalreport.pdf. → pages 13, 90

Yurcik, W., Barlow, J., and Rosendale, J. (2003). Maintaining perspective on who is the enemy in the security systems administration of computer networks. In *ACM CHI Workshop on System Administrators Are Users, Too. Proceedings of the Tenth Americas Conference on Information Systems*. → pages 29

Yurcik, W., Thompson, R. S., Twidale, M. B., and Rantanen, E. M. (2007). If you can't beat 'em, join 'em: combining text and visual interfaces for security-system administration. *Interactions*, 14(1):12–14. → pages 29

Zager, D. (2002). Collaboration as an activity coordinating with pseudo-collective objects. *Computer Supported Cooperative Work (CSCW)*, 11(1):181–204. → pages 51, 54

Zhang, J., Johnson, T. R., Patel, V. L., Paige, D. L., and Kubose, T. (2003). Using usability heuristics to evaluate patient safety of medical devices. *Journal of Biomedical Informatics*, 36(1-2):23 – 30. Patient Safety. → pages 45

Zhou, A. T., Blustein, J., and Zincir-Heywood, N. (2004). Improving intrusion detection systems through heuristic evaluation. In *in IEEE Canadian Conf. on Electrical B. and Computer Engineering (CCECE)*, pages 1641 – 1644. → pages 45

Appendices

Appendix A

Heuristic Evaluation Study Material

A.1 Evaluation Guide

Evaluation Steps

- Go through the list of heuristics to have a sense of each.
- Read the description of the scenario and understand the business logic.
- Perform each task as described on the IdM system.
- Identify usability problems while doing each task or after finishing the task. For each problem, please record the task in which you found the problem, and the heuristic with which you identified the problem. Use the scenario description and heuristics to check if the system supports the activity described in the scenario.
- Please record the problems in Here
- If you want to edit any of the identified problems which you already entered in the form, you can do it from Here.

Recommendations

- I recommend exploring the IdM system first before going through specific tasks.
- When performing the tasks, you should login with different users as described in the scenario For example, if you want to login as a Security team member or a particular manager, use the organizational chart to find the right person to login as.
- If you want to login as a user, the user name is: first name + the first letter of the family name (e.g. James Beers -¿ jamesb) and password is q1w2e3.
- If you couldnt finish a task or get the desired result, don't worry! The real user may have the same problem.
- If you face any problems, you can ask the person conducting the study. Scenarios

Description of the actors

Steve Barlow is an employee in the operations department. He is responsible for reviewing the information about the contractors. He does not have technical information about the Identity Management System or role based access control.

James Beers is the manager of operations. His day mostly involves meeting with different stakeholders in the organization. He receives lots of emails and telephone calls every day therefore he needs lots of discipline to prioritize his tasks. He does not know technical information about the IdM system or the role based access control, but he knows if an employee should have access to some resources or not.

Kevin Klien and Sandra Tsai are both members of the security team. They are responsible for managing access to the resources in the organization and solving problems of different stakeholders. They work in the same office and they are very busy with these tasks.

Larry Gomez is a contractor that needs to work in NeteAuto for one month. He barely knows the structure of the company or other employees.

Scenario 1: Self-serve user registration Larry Gomez is a contractor for the NeteAuto Company and just started his job. To be able to access the Internet, he wants to create a user account in the IdM system. Using company's intranet, he finds the link to the IdM system and creates a new user account. His request is directed to the security department. All members of the security team receive the request in their task list, and they can review or edit the user information. Finally they can approve, reject, or reserve the task (reserving the task will remove it from the worklist of other security admins).

Steps for performing the scenario: Larry Gomez accesses the IdM system. He uses the create an account link on the IdM login page and enters the required information. Kevin Klien receives the request and after reviewing the information approves the request.

Scenario 2: Bulk loader When an employee is hired by the NeteAuto Company, or information about an employee changes, or an employee leaves the company, the first system in which the changes are reflected is the HR (Human Resources) system. The HR system is separate from the IdM system; therefore, the changes in the HR system need to also be applied to the IdM system. Transferring changes from the HR system to the IdM system is performed by the security team. The security team receives a file containing all the changes (additions, modifications, and deletions) from the HR system. Every morning, Sandra Tsai, a member of the security team, downloads the HR file from the HR website and uploads the file to the IdM system to apply all the changes made in the HR system. She uses the "Bulk Loader" feature in the IdM system to upload the HR file. Then she configures the system to respond to different actions defined in the HR file. An important step after submitting the changes is to review the result of submission. She goes through the system logs, finds appropriate records, and identifies and fixes the problems, if any. Based on the organization's policy, if the number of changes in the HR file is more than 500, applying the changes should be postponed until

further clarification by HR.

Steps for performing the scenario: Sandra Tsai should first upload the HR file using the Bulk Loader in the System tab. In the next screen she chooses which field in the HR file describes the action that should be performed for each row in the file (in the example HR file it is the "action" row). Also she chooses which field uniquely identifies each row in the HR file (in the example HR file it is the "$USER_ID''$ row). In the next screen she identifies the primary object that HR file contains (choose USER as the file contains user information) and the mapping between actions in the HR file and actions in the IdM system (choose Create User, Modify User, and Delete User for any create, modify, and delete actions respectively).

Scenario 3: Requesting a role Steve Barlow is going on a last minute vacation. He realizes that he does not have the required privileges to delegate his tasks to Jason Halpin, another member of the operations department. He does not have any technical information about the privileges required to perform the delegation. But, he knows that he can generate request for privileges in the identity management system. Therefore, he uses the IdM system to write a request. In the request, he describes that he needs the ability to delegate his role to another employee in his department. When Steve submits the request, his manager needs to approve it before the request is implemented. The manager uses the IdM system to review and approves the request.

Once the manager approves the request, the request is directed to a member of security team who reviews the request, and, if it does not conflict with the security policy of the organization, tries to implement the request. Implementing the request requires the security admin to understand the content of the request (in this case, learn that Steve wants to delegate his role) and find the appropriate role that corresponds to the request (in this case, the Delegation Manager role). Then he can add Steve Barlow as a member of that role.

Steps for performing the scenario: Steve Barlow: generate the request using the "Users¿Manage Users¿Create Online Request" and then select himself as the target user. Then he can describe and submit his request. James Beers (Steve's manager): log into the IdM system. Identify, review, and approve the request. Kevin Klien (or other members of Security): log into the IdM system. Identify, review, and implement the request. To implement the request, he needs to modify the user and provision the user with the "Delegation Manager" role.

Scenario 4: Certification As a part of the organization's policy, the security team should certify the roles of the employees in each department every 6 months. The security team uses a shared calendar to mark the dates that they should perform the certification and the deadline for finishing the certification. Each member in the security team is able to start a "Certification Process" in the IdM system. When the certification date approaches, a member of the security team (Kevin Klein in this scenario) logs into the IdM system and chooses employees that should be certified. The manager of each department receives the notification about certification of his employees. In this scenario, the manager of operations (James Beers) receives an email that he should certify the roles of the employees of operations department. James put the email in his todo list. After a while, James logs into the IdM system and tries to certify the roles of the employees. For all of the employees, he checks the roles and validates if the employee should possess the role or not. It is important for the manager to perform the certification before the deadline. If the certification does not happen before the deadline, all the uncertified roles will be revoked from the employees. Therefore, before the deadline, a member of the security team sends reminders to perform the certification. On the certification deadline, a member of the security team ends the certification process.

Steps for performing the scenario: Kevin Klein: Login to the IdM system. Go to the certification tab and start the certification process for the employees in the Operations department. Also, send reminders about the certification. James Beers: Assume you are going to certify users in your department. Login to the IdM system and search for the users that require certifi-

cation using “Users¿Manage Users¿Certify Users”. Select users one by one, go to the “Certify Roles” tab, review their roles, and approve them. Sandra Tsai: Login to the IdM system and end the certification process.

A.2 Usability Problem Specification Form (ITSM Condition)

Problem specification: Please specify the identified usability problem.

Task: Please specify the task in which you identified the problem.

- ○ (1) Self-serve user creation
- ○ (2) Bulk user creation
- ○ (3) Requesting a role workflow
- ○ (4) Certification Process

Heuristic: Please choose the heuristic using which you identified a problem. If you can't associate the problem with a heuristic, please choose "Can't Specify"

- ☐ 1- Visibility of activity status
- ☐ 2- History of actions and changes on artefacts
- ☐ 3- Flexible representation of information
- ☐ 4- Rules and constraints
- ☐ 5- Planning and dividing work between users
- ☐ 6- Capturing, sharing, and discovery of knowledge
- ☐ 7- Verification of knowledge
- ☐ Can't Specify

Figure A.1: Usability Problem Specification Form (ITSM Condition)

A.3 Background Questionnaire

PART I - General Information

Gender

☐Male ☐Female

Age		**Last educational degree**	
Major		**Current Occupation**	

PART II – Human computer interaction background

- How many years of professional or research experience do you have in the area of Human Computer Interaction (HCI)?
- Do you have formal training in human computer interaction (university courses, tutorials, workshops)? Please answer with "Yes" or "No". If your answer is "Yes" please specify the list of courses.
- Do you have professional experience in the area of human computer interaction? Please answer with "Yes" or "No". If your answer is "Yes", provide a summary of your experience in this field.
- Do you have research experience in the area of human computer interaction? Please answer with "Yes" or "No". If your answer is "Yes", provide a summary of your experience in this field.
- Have you specifically been trained to perform a heuristic evaluation?
  ☐Yes ☐No
- Have you performed heuristic evaluation before? Please answer with "Yes" or "No". If your answer is "Yes", provide a summary of your previous experience in performing heuristic evaluation including number and type of systems you have evaluated

Part III – Computer security background

- How many years of research experience do you have in the area of computer security?
- How many years of professional experience do you have in the area of computer security?
- Do you have formal training in computer security (university courses, tutorials, workshops)? Please answer with "Yes" or "No". If your answer is "Yes" please specify the list of courses.
- Do you have professional experience in the area of computer security? Please answer with "Yes" or "No". If your answer is "Yes", provide a summary of your experience in this field.
- Do you have research experience in the area of computer security? Please answer with "Yes" or "No". If your answer is "Yes", provide a summary of your experience in this field.
- Have you ever worked in an organization that uses role-based access control to manage users and their privileges?
  ☐Yes ☐No
- Have you ever used to manage users and their privileges using role-based access control?
  ☐No ☐Yes in a small group (less than 10 users)

  ☐Yes in a small organization (between 10 to 50) ☐Yes in a large organization (more than 50 users)

Figure A.2: Background Questionnaire

A.4 Post-Evaluation Questionnaire (ITSM Condition)

Please indicate the extent to which you agree with each of the following statements about the heuristics that you used in this study by using the scale below: 1= Strongly Agree, 2= Agree, 3= Undecided or unsure, 4= Disagree, 5= Strongly Disagree

The heuristics were very useful in finding all of the problems that you found in the IdM system.

	1	2	3	4	5	
Strongly Agree	○	○	○	○	○	Strongly Disagree

The heuristics were very easy to learn and understand.

	1	2	3	4	5	
Strongly Agree	○	○	○	○	○	Strongly Disagree

The heuristics were very easy to apply on the IdM system.

	1	2	3	4	5	
Strongly Agree	○	○	○	○	○	Strongly Disagree

The following heuristics were very useful in identifying problems that you found in the IdM system:

	1	2	3	4	5
1- Visibility of activity status	○	○	○	○	○
2- History of actions and changes on artifacts	○	○	○	○	○
3- Flexible representation of information	○	○	○	○	○
4- Rules and constraints	○	○	○	○	○
5- Planning and dividing work between users	○	○	○	○	○
6- Capturing, sharing, and discovery of knowledge	○	○	○	○	○
7- Verification of knowledge	○	○	○	○	○

Figure A.3: Post-Evaluation Questionnaire - Part I

The following heuristics were very easy to learn and understand.

	1	2	3	4	5
1- Visibility of activity status	○	○	○	○	○
2- History of actions and changes on artifacts	○	○	○	○	○
3- Flexible representation of information	○	○	○	○	○
4- Rules and constraints	○	○	○	○	○
5- Planning and dividing work between users	○	○	○	○	○
6- Capturing, sharing, and discovery of knowledge	○	○	○	○	○
7- Verification of knowledge	○	○	○	○	○

The following heuristics were very easy to apply on the IdM system.

	1	2	3	4	5
1- Visibility of activity status	○	○	○	○	○
2- History of actions and changes on artifacts	○	○	○	○	○
3- Flexible representation of information	○	○	○	○	○
4- Rules and constraints	○	○	○	○	○
5- Planning and dividing work between users	○	○	○	○	○
6- Capturing, sharing, and discovery of knowledge	○	○	○	○	○
7- Verification of knowledge	○	○	○	○	○

Figure A.4: Post-Evaluation Questionnaire - Part II

Appendix B

IAM Field Study Material

B.1 Interview Guide

B.1.1 Organizational Context

General Information about Interviewee and Organization

- What is your position?
- Background: What is your IT/Security education/path?
- Can you briefly describe your organization? (size, sector)
- Describe security management within your organization
  - Who is responsible for security within your organization?
  - What is the security management model (centralized, distributed, etc.)? (With little help to the person)

- Can you describe the security policies in your organization (also probe for participant's role)
  - What formal (official, written) security guidelines/ policies/ architectures/ models are in place?
  - What is done in practice? (To see if the policy is completely enforced)
  - What is the process for developing policies?
  - How are policies communicated?
- To whom are policies communicated?
  - How are security-related policies enforced?
- What security risks/challenges do you perceive to be important for your organization?
  - What are the security risks or challenges in your organization?
  - What security incidents has your organization experienced as a result of these risks/challenges?
  - To what extent these incidents relate to access and identity management?
  - Are there security incidents or risks that are least priority?

Activities

- What are your responsibilities within the organization? (get overall, lead into security specific activities)
  - Actual duties/ official duties (Let them talk, probe anything not on list to confirm

that omissions are true negatives)

- Manage identities and accesses
- Perform and respond to security audits on the IT infrastructure?
- Develop security policies?
- Design and revise security services or projects?
- Implement security controls?
- Solve end user security issues?
- Educate and train?
- Respond to security incidents? (Skills, knowledge and strategies, resources (tools) used)
- Mitigate new security vulnerabilities?
- Prioritization (typical day)

B.1.2 Questions About IAM Process

AIM process (general)

- What do you consider to fall under the definition of access and identity management?
- What is the current process within your organization?
  - Activities? (policies, managing access, managing identities, audit, compliance, trouble shooting)

- Stakeholders? (management, HR, IT, security, employees, customers, external organizations...)
- What is your role?
- Knowledge required
- Importance?
- Frequency?
- Is it supported by tools?
- Can it be automated or supported better by the tool?
- How was this process before adopting an IdM solution ?

Compliance

- Is the organization required to comply with any standard? Which standard?
- What is the role of IDM solution in your compliance with the standard?

B.1.3 Probing Specific Activities (Depends on Participant's Role)

Managing accesses and identities

- Can you describe the lifecycle for managing accesses and identities? (From creation to destruction of an identity)
- Which parts of this lifecycle is supported by your IdM system?

- How you manage changes in user status? (extending access for a user, changing access, discontinuing access)
- How frequently you face exceptions in setting up accesses and how you handle them? (For example: Employees should normally access X but not Y. But for a specific case you should temporarily provide access to an employee to Y.)
- How complex are the policies and how do you handle complexity?
  - Number of users? Number of resources? Number of roles? Number of access rules (E.g. Role X has access Y to resource Z)
- Are there any cases that you don't want system access to be controlled by your IDM solution?

Entitlements

- Can you give us a definition for entitlement ? Can you give us examples from your organization?
- How entitlements are managed in your organization ? Is there a process in place?
- What stakeholders are involved in determining the meaning of an entitlement and deciding about associating entitlements to users ?
- What is the process of checking if users are assigned to a correct set of entitlements?

Audit

- How can you make sure that the correct access rights are set for the intended person? (that the policy is implemented correctly)

- What is the process for identifying and removing the unused or discontinued identities and accesses?
- Do you have any formal audit procedure in place? If so, describe?
- Is there any legislation that require your organization to perform audit ?

Role Management

- How do you create roles in your organization? (define business responsibilities as roles and association of roles to entitlements?
- How frequently roles are changed or added?
- How do you perform "role engineering" in your system?
  - What is difficult/easy about it?
  - What approach do you use (top down, bottom up, hybrid)?
- What stakeholders are involved in the process of managing roles?
- What tools do you use for managing roles?

End-user experience

- What are the ways of accessing the system for users? Is there just one, or many (different usernames, different portals, etc)?
- Can you recall any end-user complaints relating to the IdM solution?
- Is it possible for users to manage access?

- How do the end users understand the configuration implemented by security practitioners? How can an end-user know which resources he has access to?
  - Does the tool give feedback?
  - Do you need to provide explicit knowledge? (For example about how they can find-out their access rights, changing their personal information (password, etc.))?
  - Do end-users need to be aware of their access rights or policy at all?
- Do you think the end-user experience has changed after adoption of IdM system ?

Troubleshooting

- How frequently you deal with problems that require troubleshooting?
- Can you give an example? (get details: collaboration?, blow-by-blow account)
- While performing troubleshooting, what is the magnitude of information that you work with? (means logs about accesses) Do you cut things or prioritize because of the volume of information?

Archiving

- What kind of activities/incidents/interactions/communications do you document and how?
- Is there a need for recording/archiving of communications? In what circumstances?

Reporting

- Describe the reports that you generate that are related to access and identity management.

- For whom do you generate these reports?
- How are your reports used?
- What tools do you use to help compose and send your IdM reports?
- Do you generate reports for different people? Who?
- If you compose different kinds of reports (different content, different level of granularity) for different people, is it easy for you to compose different kinds?
- What makes it easy or tedious?
- Do any of your report help you prioritize? What information helps? Where does it come from?

B.1.4 Questions About IAM Technologies

- What is your definition of an ideal IdM solution? (Solution that manage accesses, control digital identities, enable checking who did what and who granted the access, checking the compliance of the system)
- Do you currently have such solution?
- Which parts exist in your current infrastructure?
- What are the driving forces for adopting IdM technology in your organization ?

Purchasing/Evaluation

- What was the process for selecting the IdM tool in use?
  - What stakeholders are involved in the process?

  - How did you evaluate the competing tools?

- What features do you look for in a tool? Which features are available in your current tools?

- What properties to you wish for in your tools? (quality, user interface, performance, service, vendor reputation).

Tool deployment

- What are the pre-requisites for deploying an IdM solution? I mean should any specific business processes in place? Should any technological infrastructure be in place? Is there any training required? Is there any kind of knowledge required?

- Who are the people involved in the IdM deployment? I mean is there any relation for example with managers, end-users, or external organizations?

- What are the difficulties in deployment of the product?

- Do you need to customize out of the box identity and access management tools to meet your needs? If yes, can you describe the process for that?

- Do you need to integrate any of your existing systems (Databases, Terminals, Web Applications, etc.) with your IdM solution? Does the solution perform this automatically?

- Do you have any recommendations for improving deployment process?

Tool maintenance

- What maintenance tasks do you perform to keep the IdM solution running and who is responsible for them?

- How much technical knowledge and effort do they need to maintain the solution?
- What is the process of updating or changing your IdM solution?

Tool Use

- How do you use tool X and what do you like/dislike about it? (if possible, get them to show the interface and probe their view of the functionality/usability afforded by the tool. Try to take photos or draw sketches from what the show.)
- In addition to tools that are part of your general IdM infrastructure, are there any other tools used for the various IdM activities? (i.e., excel sheet for creating reports related to IdM)
- Are there any tools do you no longer use? (why?)
- What is the most error prone part of your identity management solution?
  - How do you find out that a tool has made an error?
  - What do you do to recover from errors?

B.1.5 Working/Dealing with Other Stakeholders

Collaboration

- With whom do you interact during IdM activities? What are the circumstances?
- Do you need to Co-ordinate your work with other people?
  - Do you need to delegate some part of an IdM task to other people? Do you need to work with other people in order to accomplish an IdM task?

- What is your relationship with other people who are responsible for identity management? How closely do you work with them?
- Do the people who manage accesses or identities have knowledge about computer security? Do they know whether or not risks are involved in what they do? Do they understand these risks?
  - Tools to facilitate awareness: Do you use any tools to support awareness of activities of others (workflows, shared calendars, shared to-do lists, whiteboards)
  - Does the IdM tool provide any support for activities which require collaboration?

Communication and Common ground (negotiating a shared understanding?

- What type of information do you need to share?
- Are there new issues that arise through your on-going experience with IdM which are necessary to communicate to others?
  - How are they communicated? (Can give example of Documents, Wikis, or SharePoint)
  - Is your IdM tool integrated with any of these communication channels?
  - Do you use specific terminology to communicate with other people involved in IdM activities?
- How do they know that the information and your communication is understood?
- How people understand each other while communicating and how they make sure and let each other know that they understood each other?

- Can you give us an example of misunderstanding during communication with other stakeholders about IdM?
- When is it necessary to interact with people outside of the organization?

Appendix C

Access Certification Survey

UBC Access Certification Survey

Introduction

This survey is a part of a larger research project to improve identity and access management tools and processes in organizations. Your role in the complex business of IT security administration and identity management is important, and we believe that participation by people like you is essential for the improvement of identity and access management tools.

What will be expected from you

This survey should take about 15 minutes to complete. Your participation in this study is entirely voluntary and you may discontinue the survey at any time. There are no consequences for withdrawal from participation. If you have any concerns about the treatment or your rights, please telephone the Research Subject Information Line in the Office of Research Services, University of British Columbia, at +1 (XXX) XXX-XXXX.

Your Privacy and Confidentiality

Your responses will be stored in a secure server in Canada. The only personal or identifying information that will be collected is your name and email address. Entering your name and email address is completely optional and we only collect this information if you are interested to be considered for raffle or you want to participate in other components of our research project. Forrester Research, Inc. will be provided with the access to anonymized survey data, and will publish the highlights of survey results.

Remuneration/Compensation:

We are very grateful for your participation. You will be considered for a raffle of one iPad (WiFi, 128Gb) as compensation for participating in the study. You are eligible even if you do not complete the survey. If at any point you decide to withdraw from the survey, you will still be considered for the raffle. In order to be considered for the raffle, you should enter your email address and name so we can contact you in case you win. The expected odds of winning in this study is about 1/100. In addition, you will get a courtesy copy of a Forrester report on the findings of this survey.

Contact

Should you have questions or comments about this survey, please contact any of the research team members.

- Pooya Jaferian, PhD Candidate, Laboratory for Education and Research in Secure Systems Engineering (LERSSE), University of British Columbia, xxxxx@ece.ubc.ca
- Konstantin Beznosov, Associate Professor, Laboratory for Education and Research in Secure Systems Engineering (LERSSE), University of British Columbia, xxxxxx@ece.ubc.ca

Consent:

Your participation in this study is entirely voluntary and you may refuse to participate or withdraw from the study at any time. By choosing Agree option below you indicate that you have printed or saved a copy of this consent form for your own records and that you consent to participate in this survey.

1) Please enter your name and email address so we can contact you if you are selected in the raffle.

Name (Optional)_____
Email (Optional)_____

2) Do you agree to participate in the survey?

- ❍ Agree
- ❍ Decline

3) Please choose job titles that best fit the activities you perform.

- ❑ IT security manager
- ❑ IT manager
- ❑ Security analyst
- ❑ Security administrator
- ❑ Security consultant
- ❑ Auditor
- ❑ Other (please specify)

If you selected other, please specify
_____

4) How many years of experience do you have in IT security?

_____

5) How many years of experience do you have in Identity and Access Management (IAM)?

_____

6) Which IAM systems do you have experience working with (e.g., Oracle Identity Manager, CA Identity Manager, Novell Identity Manager)?

_____
_____
_____

7) What sector is your organization in?

- ❍ Aerospace
- ❍ Manufacturing
- ❍ Banking
- ❍ Finance / Accounting
- ❍ Insurance / Real Estate / Legal
- ❍ Federal Government (Including Military)
- ❍ State / Local Government
- ❍ Medical / Dental / Health
- ❍ Internet Access Providers / ISP
- ❍ Communications Carriers
- ❍ Transportation / Utilities
- ❍ Construction / Architecture / Engineering
- ❍ Data Processing Services
- ❍ Wholesale / Resale / Distribution
- ❍ Education
- ❍ Marketing / Advertising / Entertainment
- ❍ Research / Development Lab
- ❍ Business Service / Consultant
- ❍ Computer Manufacturer
- ❍ Computer / Network Consultant
- ❍ Computer Related Retailer / Wholesaler / Distributor
- ❍ VAR/VAD/Systems or Network Integrators
- ❍ Other (please specify)

If you selected other, please specify

_____

8) Which country is your primary work location ?

- ❍ List of all countries

9) Approximately how many employees work at your location? If you are a consultant working with multiple organizations, please choose one of them in order to answer the following questions.

- ❑ Less than 100
- ❑ 100 to 500
- ❑ 501 to 1000
- ❑ 1001 to 5000
- ❑ More than 5000

10) There are many IT standards that an organization may follow. For each type of standard, please indicate whether your organization is currently following the standard or considering implementing the standard.

- ❑ SOX or its equivalents in other countries (e.g., Bill 198, J-Sox)
- ❑ PCI
- ❑ HIPPA
- ❑ ISO 27001

❑ GLBA
❑ I do not know
❑ Other (please specify)

If you selected other, please specify
_____

11) If you are adopting or thinking about adopting an IAM system in your organization, please specify the stage at which your organization is in?

❑ Thinking about IAM adoption
❑ Currently in the process of deployment
❑ Already deployed part of the IAM system
❑ Successfully finished deployment of an IAM system
❑ Have no plan for adopting an AIM system
❑ Other (please specify)

If you selected other, please specify
_____

12) Are you using roles for access control in your organization?

❍ Roles are used throughout the organization, entitlements (access rights) are grouped in roles, and users are assigned to roles to get access to an application
❍ Roles are partly used. Some applications use roles and some use assignment of entitlements to users
❍ Roles are not used, and entitlements are assigned to users to provision them with access
❍ I do not know
❍ Other (please specify)

If you selected other, please specify
_____

13) How do you think access certification should be performed?

❑ Managers review and validate the entitlements (or roles) of their employees
❑ Application owners review and validate the users who have access to their applications
❑ Security team review and validate the entitlements (or roles) of users
❑ Auditors review and validate the entitlements (or roles) of users
❑ There is no need for manual certification as the policy is automatically enforced in the provisioning stage
❑ Other (please specify)

If you selected other, please specify
_____

14) How access certification is currently performed in your organization?

- ❑ Managers review and validate the entitlements (or roles) of their employees
- ❑ Application owners review and validate the users who have access to their applications
- ❑ Security team review and validate the entitlements (or roles) of users
- ❑ Auditors review and validate the entitlements (or roles) of users
- ❑ There is no need for manual certification as the policy is automatically enforced in the provisioning stage
- ❑ Access certification is not performed
- ❑ Other (please specify)

If you selected other, please specify

_____

15) How frequently access certification is performed in your organization?

- ❑ Annually
- ❑ Twice each year
- ❑ Quarterly
- ❑ On an ad-hoc basis
- ❑ I do not know
- ❑ Other (please specify)

If you selected other, please specify

_____

16) What triggers access certification in your organization?

- ❑ Scheduled, previously planned reviews
- ❑ Changes in users' job function
- ❑ Changes in role-entitlement assignments (if the organization uses roles)
- ❑ Additional access request by a user
- ❑ Mergers and acquisitions
- ❑ Reorganization of a business unit
- ❑ A security incident
- ❑ Requests by managers or application owners
- ❑ I don't know
- ❑ Other (please specify)

If you selected other, please specify

_____

17) How the access lists of users are generated for those who perform certification ?

- ❑ A report is generated manually by visiting different end-points
- ❑ A report is generated automatically by a homegrown solution
- ❑ An IAM system is used for access certification
- ❑ I don't know
- ❑ Other (please specify)

If you selected other, please specify

_____

18) Assume a certifier is in the process of certifying the entitlements of a user. Which of the following communication channels might be used during the process?

❑ When an application owners do the certification, they need to contact users' managers for help

❑ When users' managers do the certification, they need to contact application owners for help

❑ When security administrators do the certification, they need to contact application owners and users' managers for help

❑ When either of users' manager or application owners do the certification, they need to ask security administrators for help

❑ Certifiers need to contact previous certifiers for help

❑ I don't know

❑ Other (please specify)

If you selected other, please specify

_____

19) How much do you agree with the usefulness of the following pieces of information in performing access certification?

	1 (Strongly Agree)	2 (Agree)	3 (Neutral)	4 (Disagree)	5 (Strongly Disagree)	I don't know
List of roles the user is assigned to (if organization uses roles)	❍	❍	❍	❍	❍	❍
List of the entitlements of the user	❍	❍	❍	❍	❍	❍
Description of each entitlement	❍	❍	❍	❍	❍	❍
History of users entitlements including additions and deletions over time	❍	❍	❍	❍	❍	❍
Criticality or risk associated with each entitlement	❍	❍	❍	❍	❍	❍
Application associated with each entitlement	❍	❍	❍	❍	❍	❍
List of the entitlements of other employees in the same department	❍	❍	❍	❍	❍	❍
User's job function	❍	❍	❍	❍	❍	❍
History of user's job function changes	❍	❍	❍	❍	❍	❍
History of previous certifications and their result	❍	❍	❍	❍	❍	❍
Information about who requested an entitlement, who approved it, and who granted it	❍	❍	❍	❍	❍	❍

20) Please add additional comments on helpful information sources during access certification.

_____
_____
_____

21) Assume you are reviewing and validating possession of an entitlement (or assignment of a user to a role in case of role based access control) during access certification. How much do you agree that each of the following observations is an indication of risk?

	1 (Strongly Agree)	2 (Agree)	3 (Neutral)	4 (Disagree)	5 (Strongly Disagree)	I don't know
An entitlement (or a role) that is only associated with a small number of users	❍	❍	❍	❍	❍	❍
An entitlement (or a role) that is only assigned to a small number of users in the same organizational unit	❍	❍	❍	❍	❍	❍
An entitlement (or a role) that is assigned to a large number of users	❍	❍	❍	❍	❍	❍
An entitlement (or a role) that is assigned to a large number of users in the same department	❍	❍	❍	❍	❍	❍
A role that groups large number of entitlements	❍	❍	❍	❍	❍	❍
An entitlement (or a role) that is previously revoked from the user	❍	❍	❍	❍	❍	❍
A user that possesses two entitlements (or roles) that cause SoD (Separation of Duties) violations	❍	❍	❍	❍	❍	❍
A user with accumulated entitlements from past job	❍	❍	❍	❍	❍	❍
An entitlement (or a role) that is associated with a critical application (e.g., SOX critical application)	❍	❍	❍	❍	❍	❍
An entitlement that has been actually used by the user to access the system	❍	❍	❍	❍	❍	❍

22) Please add additional comments on observations or cues that show a user-entitlement assignment is risky, and observations or cues that show a user-entitlement assignment is safe.

_____
_____
_____

23) Approximately, how many applications do you have in your organization? (leave blank if you don't know)

_____Applications

24) On Average, how many entitlements a user has in your organization? (leave blank if you don't know)

_____Entitlements

25) On average, how many roles a user has in your organization? (leave blank if you don't know)

_____Roles

26) On average, how many users a manager (or an application owner) needs to certify in a certification activity? (leave blank if you don't know)

_____Users

27) Approximately how many users are in your organization ?

_____Users

28) Thank you very much for participating in this survey. Your feedback will help us investigate improvement of technological support for access certification problem.

As another component of our research, we are conducting interviews with security practitioners who have expertise in identity and access management. We would like to have a phone interview (between 30 to 60 minutes) with you to know more about your experience in IAM. If you are willing to participate in a phone interview, please provide your name and email address and we will contact you to schedule an interview. Providing this information is optional and you can skip this part.

name _____
email address _____

Appendix D

Detailed Description of AuthzMap, List, and Search

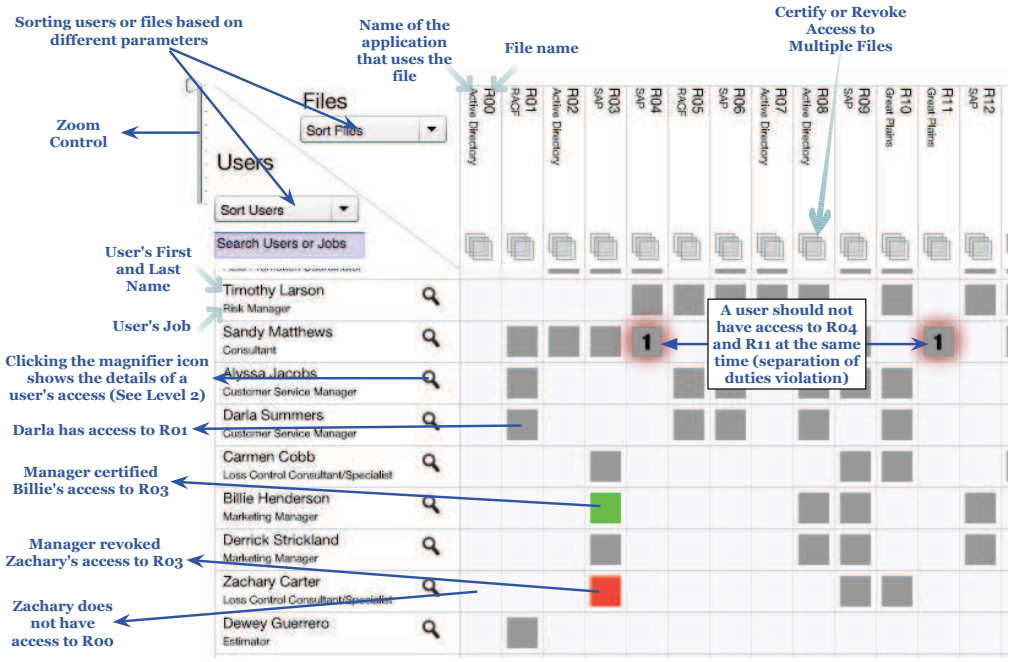


Figure D.1: Level one of the AuthzMap interface. We used the notion of files in the user study, but eventually columns in the grid indicate roles, permissions, files, or any other type of entitlements.

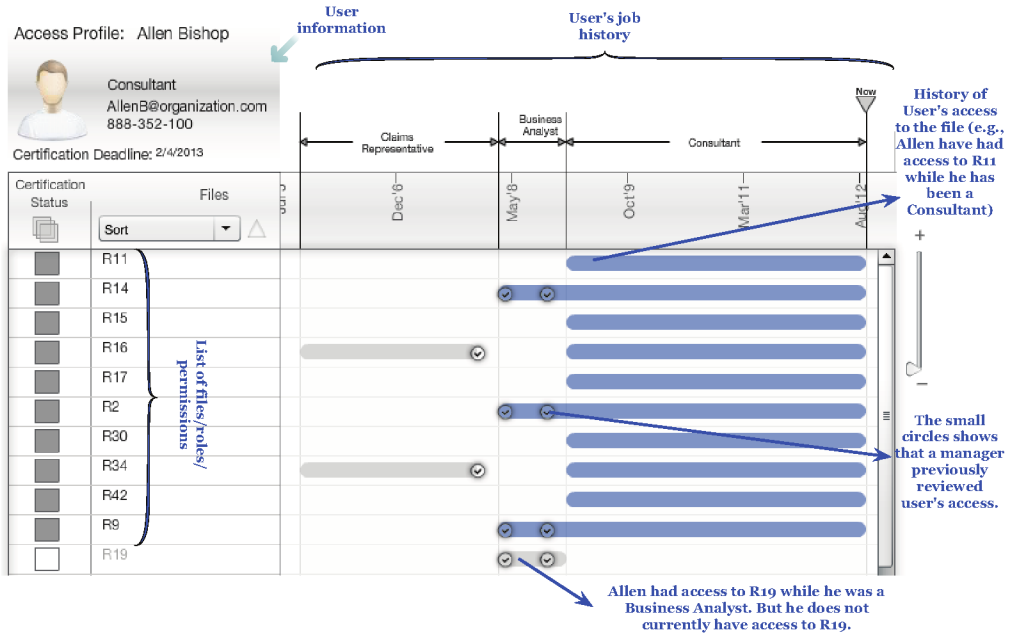


Figure D.2: Level two of the AuthzMap interface. Reviewer can access this level by clicking on the magnifier icon in the level 1 of the interface.

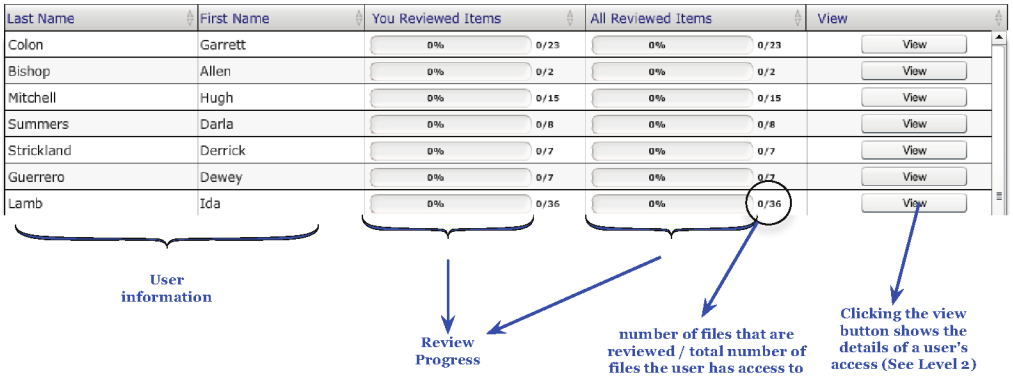


Figure D.3: Level one of the List interface. The original interface used the notion of "entitlements", but we changed it to files for the purpose of the user study.

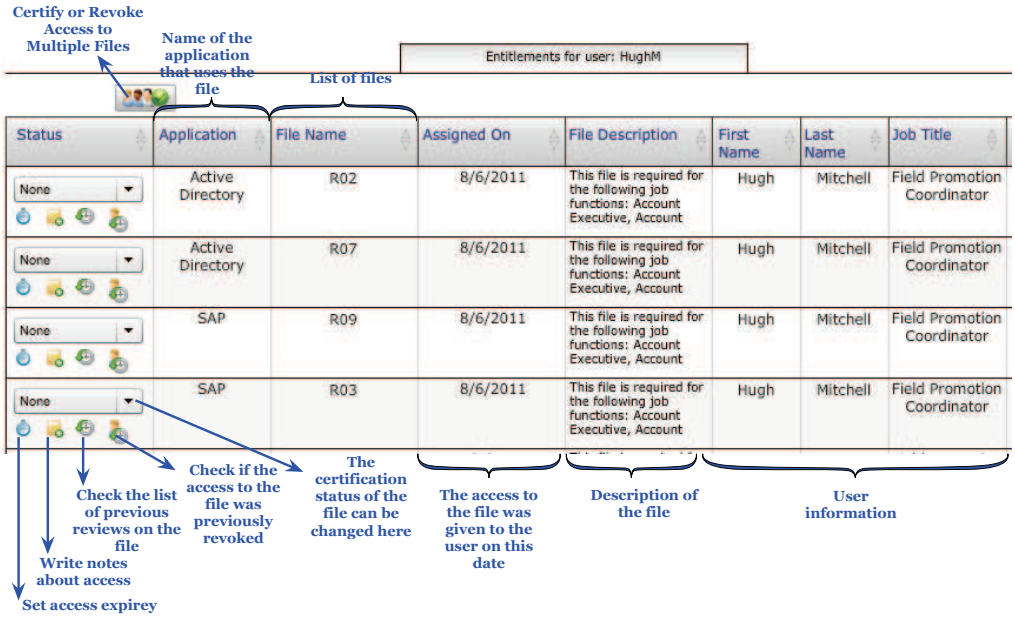


Figure D.4: Level two of the List interface.

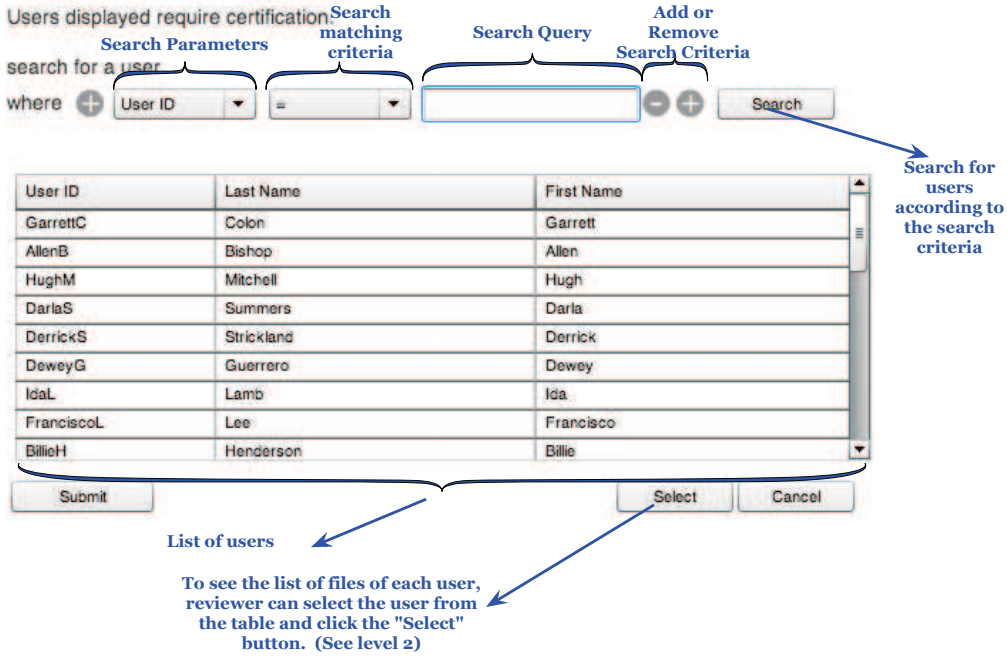


Figure D.5: Level one of the Search interface. The original interface used the notion of "Roles", but we changed it to files for the purpose of the user study.

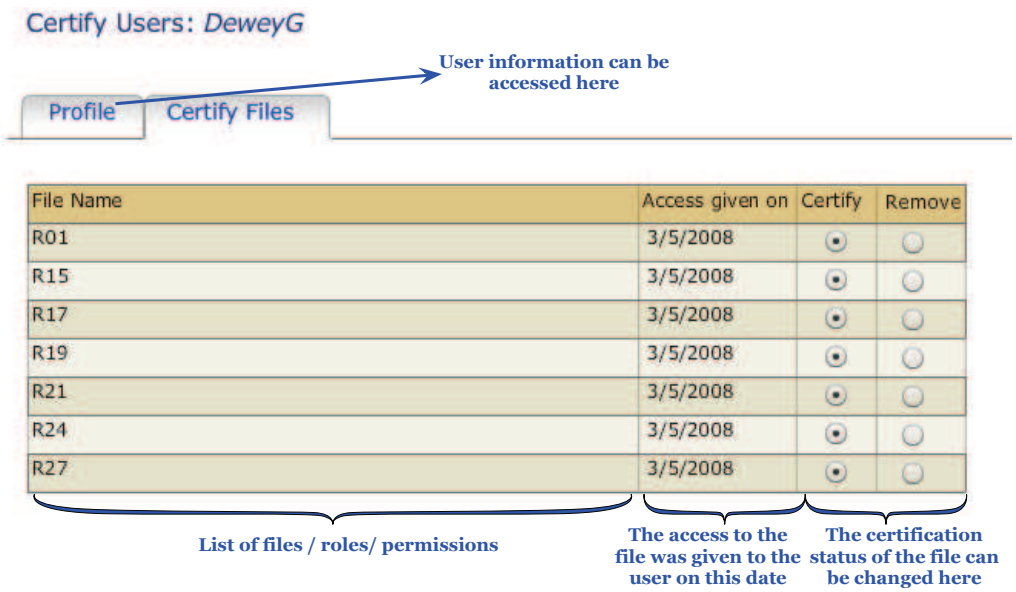


Figure D.6: Level two of the Search interface.

Appendix E

Authzmap Initial Prototypes at Different Levels of Fidelity

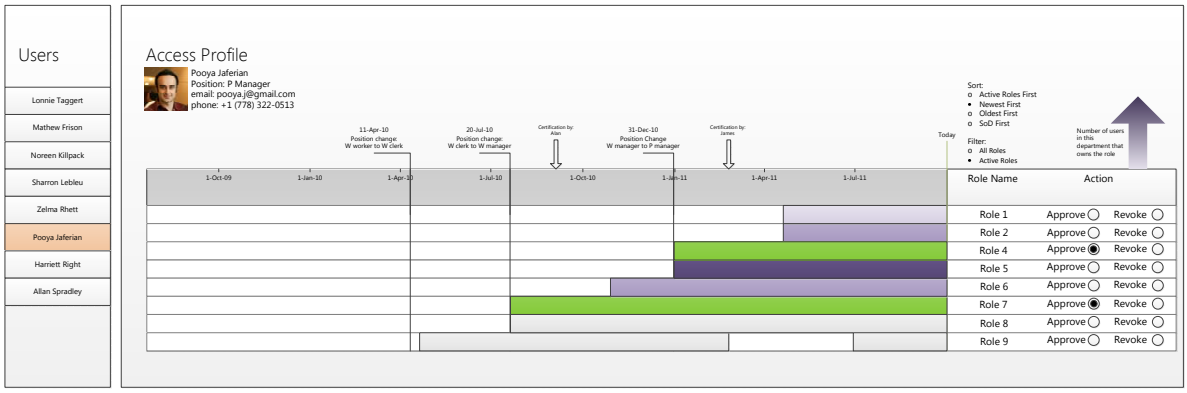


Figure E.1: Low fidelity prototype of AuthzMap (developed in MS Visio) showing the initial state of the interface.

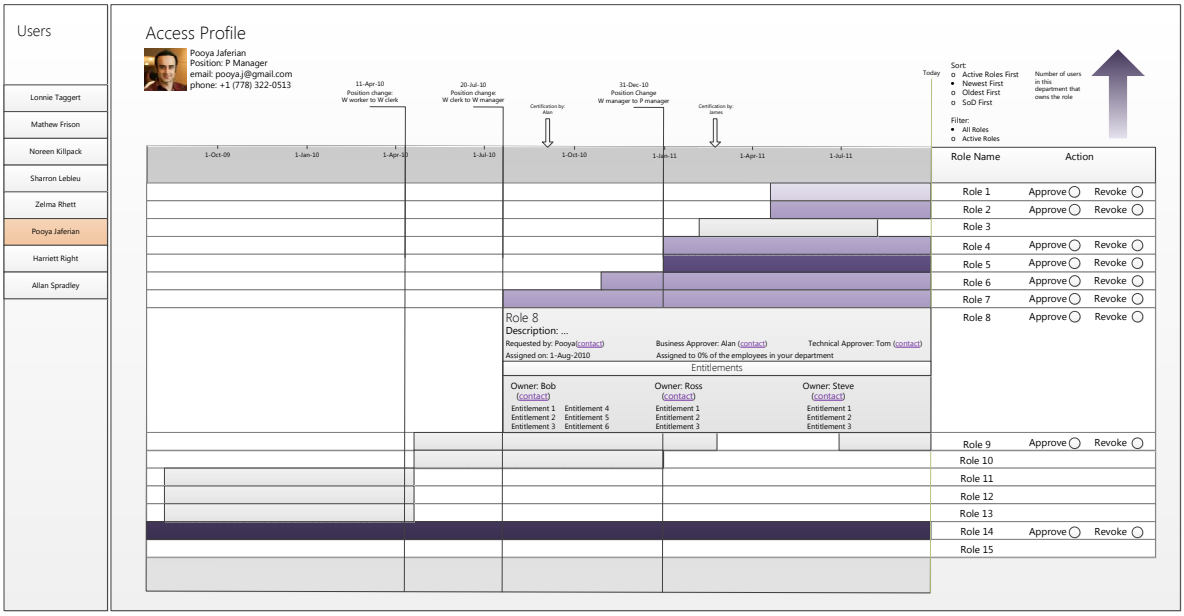


Figure E.2: Low fidelity prototype of AuthzMap (developed in MS Visio) showing the detail view.

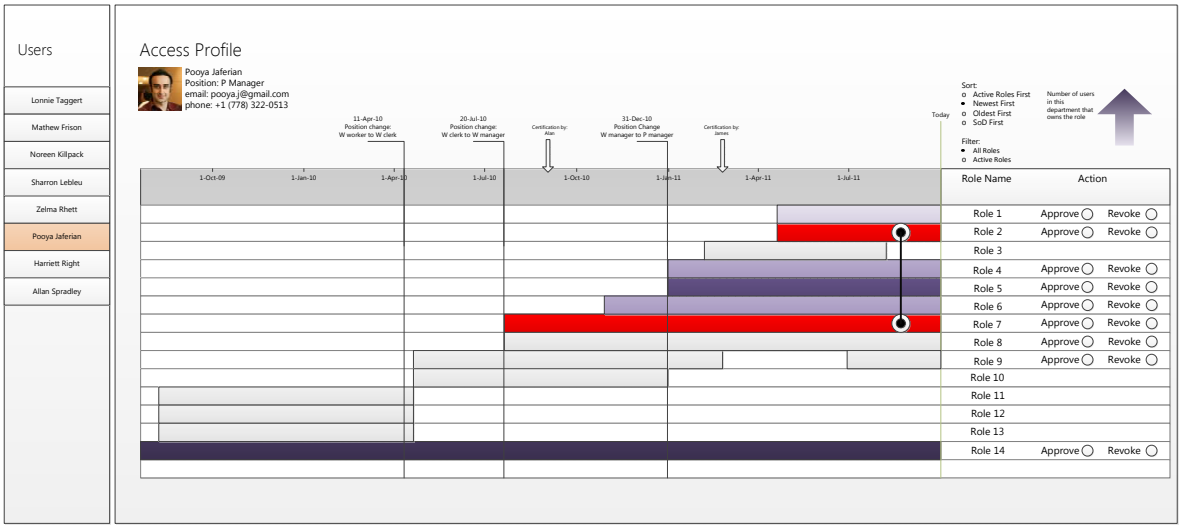


Figure E.3: Low fidelity prototype of AuthzMap (developed in MS Visio) showing the SoD violations.

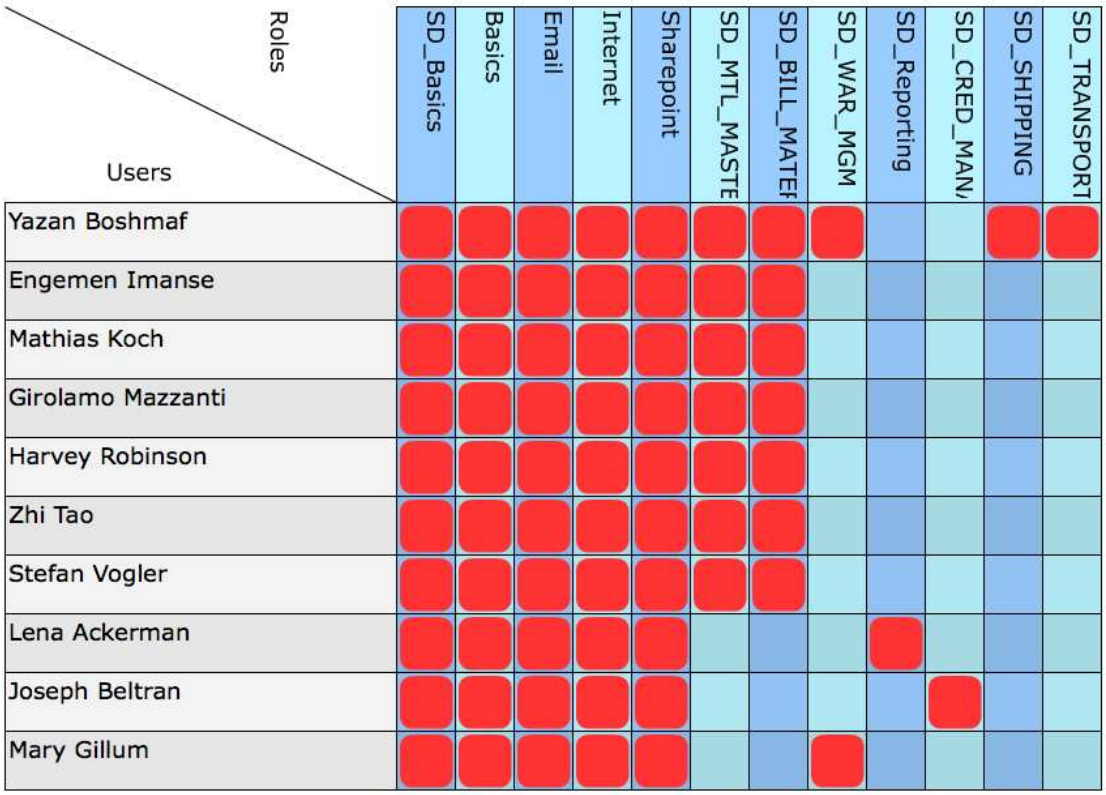


Figure E.4: Medium fidelity prototype of AuthzMap (developed in Flash) showing the fish-eye view.

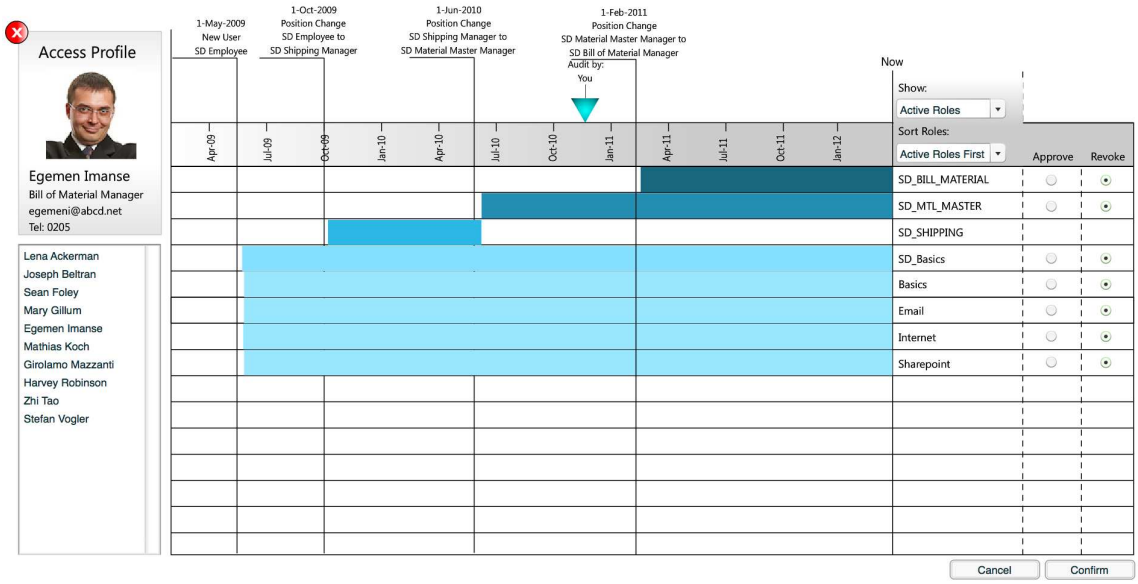


Figure E.5: Medium fidelity prototype of AuthzMap (developed in Flash) showing the history view.

Appendix F

AuthzMap User Study Material

Background Questionnaire

Please provide the following demographics information.

Gender

○ Male ○ Female

Age

Select one...

Last educational degree

Select one...

Academic Major (If have academic education)

Current Occupation

How many employees are in the organization you are working in ?

Select one...

Have you performed any of these activities (currently, or in the past):

○ Yes (please check the activities you preformed):

- ☐ I used a user name and password to use IT services at work
- ☐ I had multiple users on my computer
- ☐ I changed access to my files or resources at work
- ☐ I had a managerial position at work.
- ☐ I authorized people's access to resources at work

○ No, I have not performed any of the above activities.

Submit

© LERSSE
University of British Columbia

Training

Please complete the following training slides. After the training, you need to answer a short test, in order to verify your understanding of training material. You have to asnwer the test questions correctly to continue with the study.

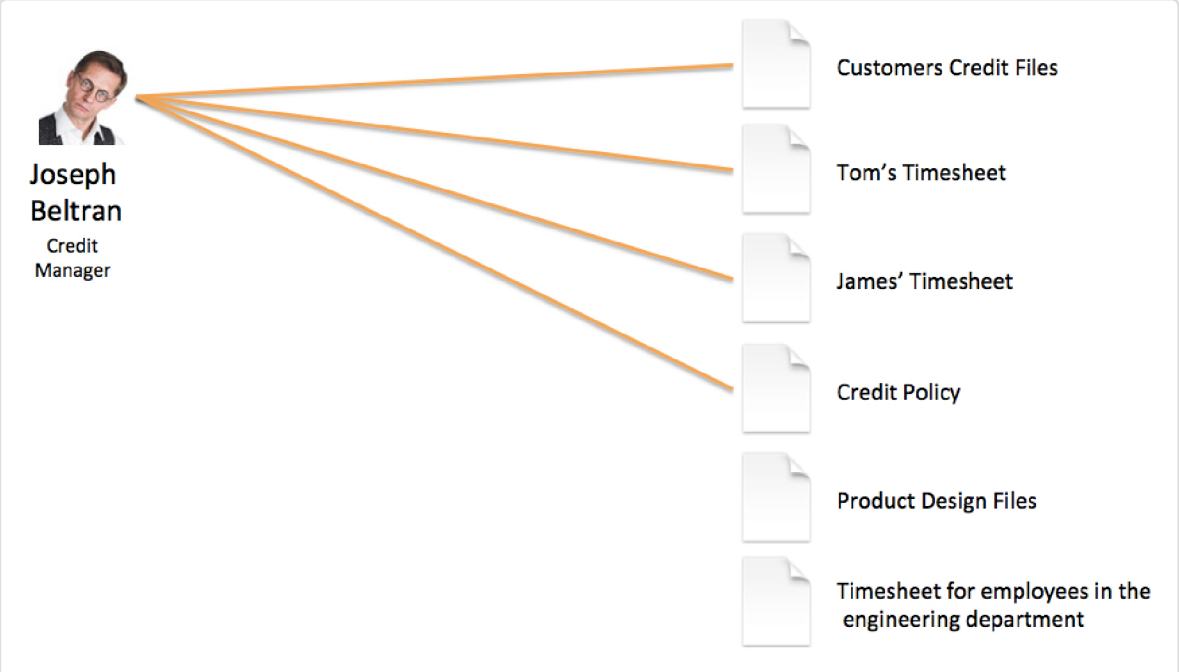
Part I | Part II

Introduction

Autoneat is a company that creates automobiles. Each employee in the company has a computer and can access various files on the company's network.

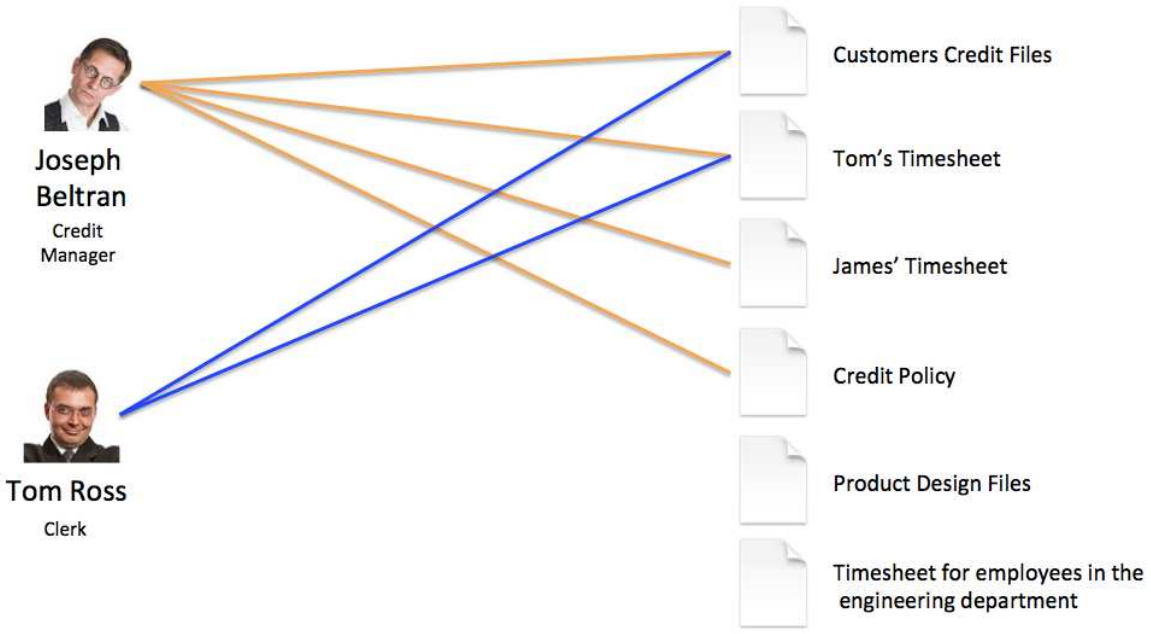
Employee's access to files

There are thousands of files on the company's network, but **each employee can only access a subset of these files.** If an employee **needs access to a file for his job**, he is provided with access to the file. Otherwise, he **should not have access** to the file.



Example 1

Joseph is a *Credit Manager* at the Autoneat company. As a credit manager, he needs access to customers' credit report files, time sheet files of his employees, and the credit policy document. On the other hand, he should not have access to the time sheets of employees in engineering department, and the design files of the design department because his job does not require him to access these files.

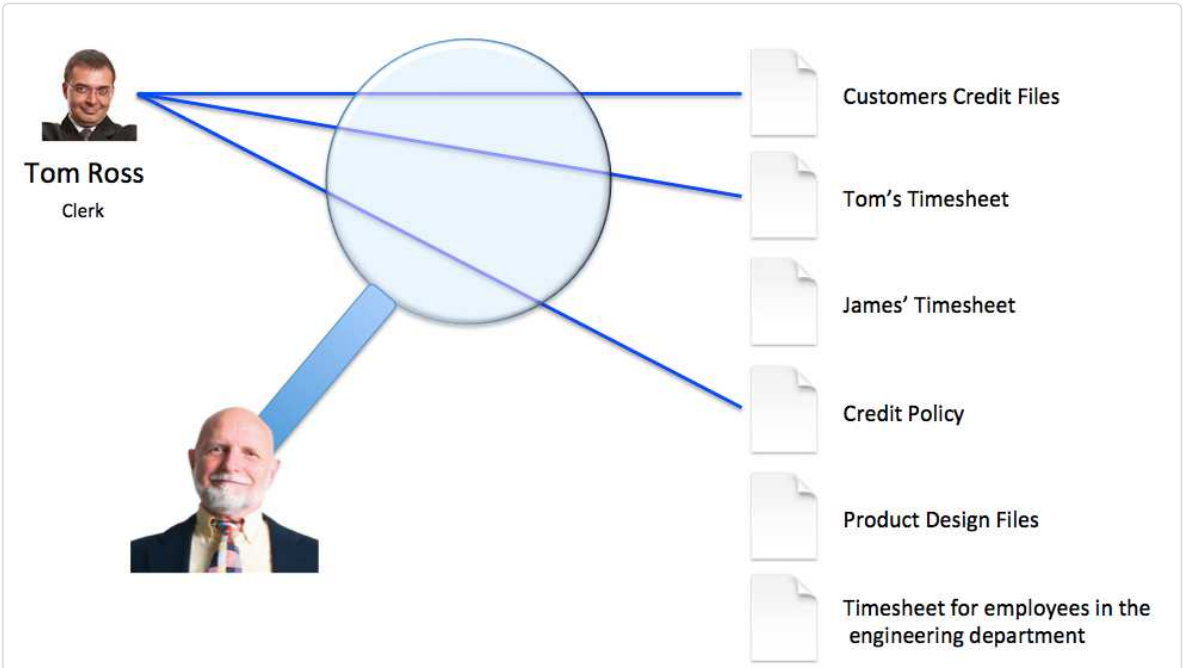


Example 2

Tom is a clerk in the Credit department. He needs access to his own timesheet file, as well as customers' credit reports files. But he should not have access to other employees' timesheets. He also should not have access to the credit policy file, as it contains sensitive information that Tom does not need for his job.

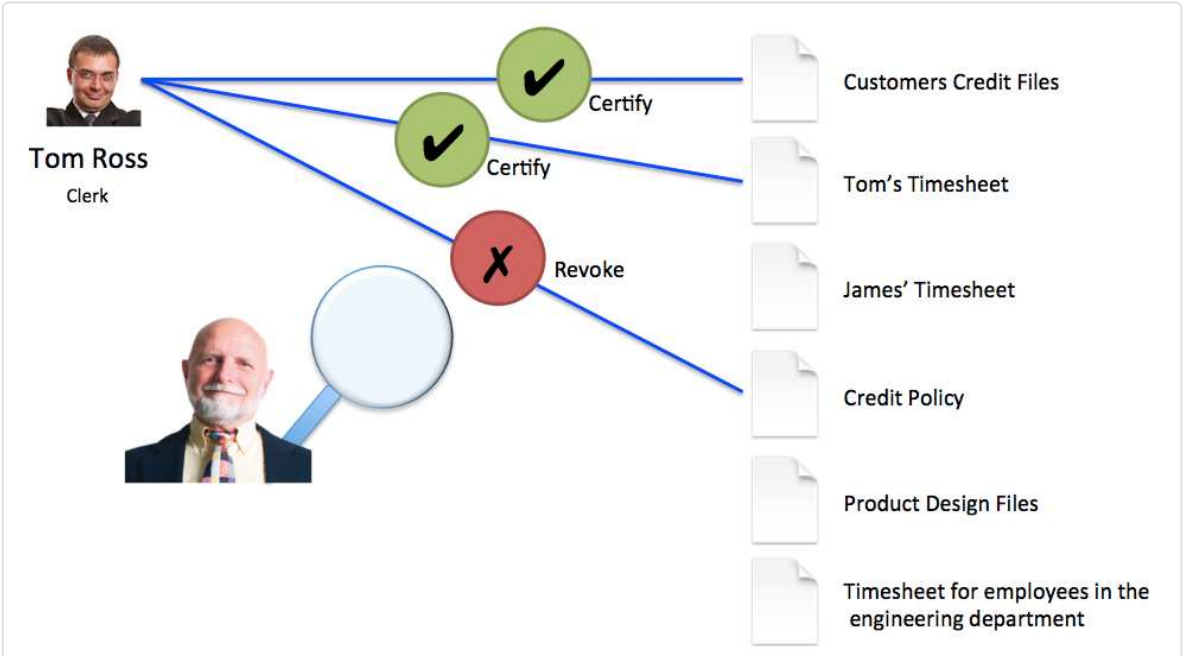
What is Access Review ?

As employees change job in the company or get promoted, the files they need for their job changes. To make sure that employees have access to a correct set of files, companies perform an activity called **Access Review**. In this activity, a manager goes through the list of his employees, and **reviews** the files each employee has access to. If the employee needs access to a file for doing his job, the manager **certifies** the access. If the employee does not need the access to the file for his job, the manager **revokes** the access to the file.



Example 3a

George, the chief sales officer of the company, reviews the list of files Tom has access to. He sees that Tom's job is a Clerk, and he needs access to his time sheets and customers' credit reports. But he does not need access to the policy file.



Example 3b

George certifies Tom's access to the customers' credit reports and Tom's own timesheet, but revokes the access to the credit policy file.

Proceed to Part II

© LERSSE
University of British Columbia

Version 1.0 -- January 15, 2013

Training Test

You need to complete the test to proceed to the study tasks. If you answer some questions incorrectly, please feel free to go back and review the training material.

Which one of the following statements is true about the employees' access to files in a company ?

- ◯ Employees should never be able to access files.
- ◯ None
- ◯ Always all employees can access all the files.
- ◯ Employees may have access to certain files.

What happens during access review ?

- ◯ Managers check the access of their employees
- ◯ Security admins check the access of the users in the organization
- ◯ Security admins review and assign users to roles
- ◯ Managers request access for new employees

When a user changes job from A to B, which one of the following events should happen ?

- ◯ User should be given access to those job B files that he does not already have access to.
- ◯ User should be given access to files related to job B.
- ◯ Access to the job A files should be revoked, and then user should be given access to files related to job B.
- ◯ None

What is the main factor that determines the files a user has access to?

- ◯ User's technical skills
- ◯ User's seniority in the company
- ◯ User's department
- ◯ User's job

When access to a file is revoked during access review ?

- ◯ When the file is not required for user to perform his job
- ◯ When the reviewer does not understand the purpose or content of the file

- When the file is misused by the user
- When the user has changed his job

Which one of these items describes access review the best ?

- Reviewing users and the files they have access to, certifying valid access, and revoking invalid access
- Removing access from users that left the company
- Reviewing files that users have access to, and providing users with the required access
- Reviewing users job, and providing them with access required for their job

Submit

© LERSSE
University of British Columbia

Version 1.0 -- January 15, 2013

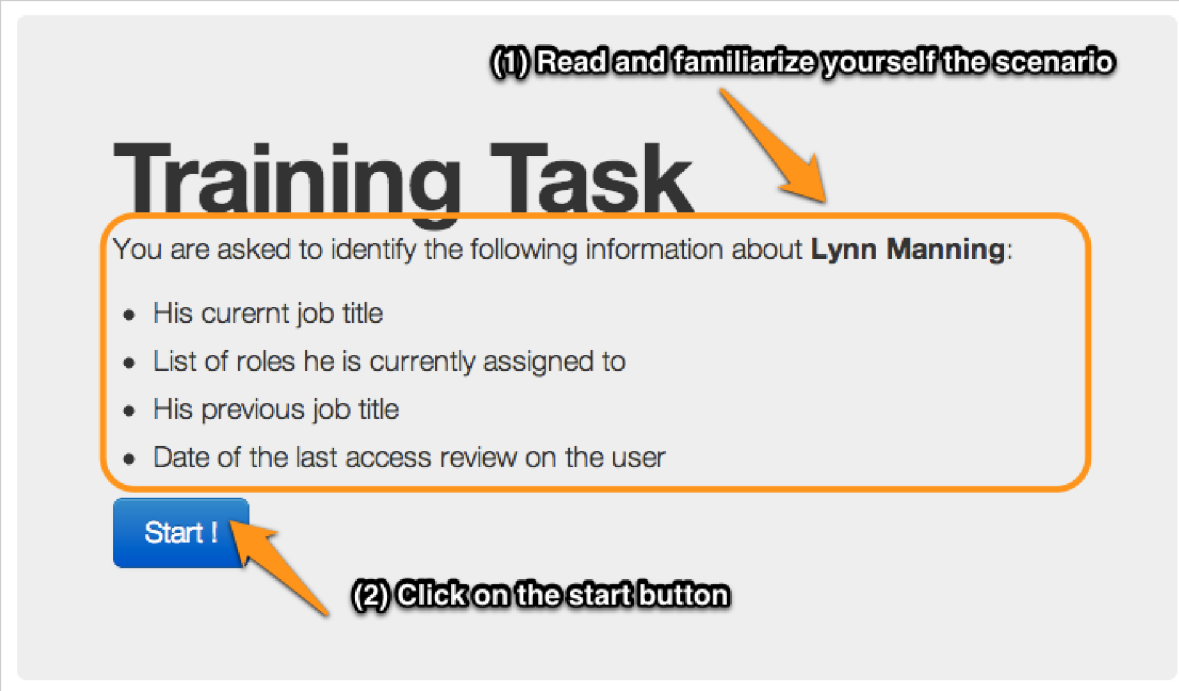
Study Guide

Now you are ready to go through the tasks and perform them on the provided system. Each task consists of a scenario that you should read first, and then try to perform it on the provided system. Please make sure you are following these steps:

Overview | Tasks

Starting Tasks

After reading this guide, you will be presented with a list of tasks to perform. You should perform tasks in the provided order, and you cannot proceed to the next task without finishing the current task you are working on. When you choose a task to perform, you will be presented with the task overview page (See the following Figure), which shows the scneario you are going to perform.

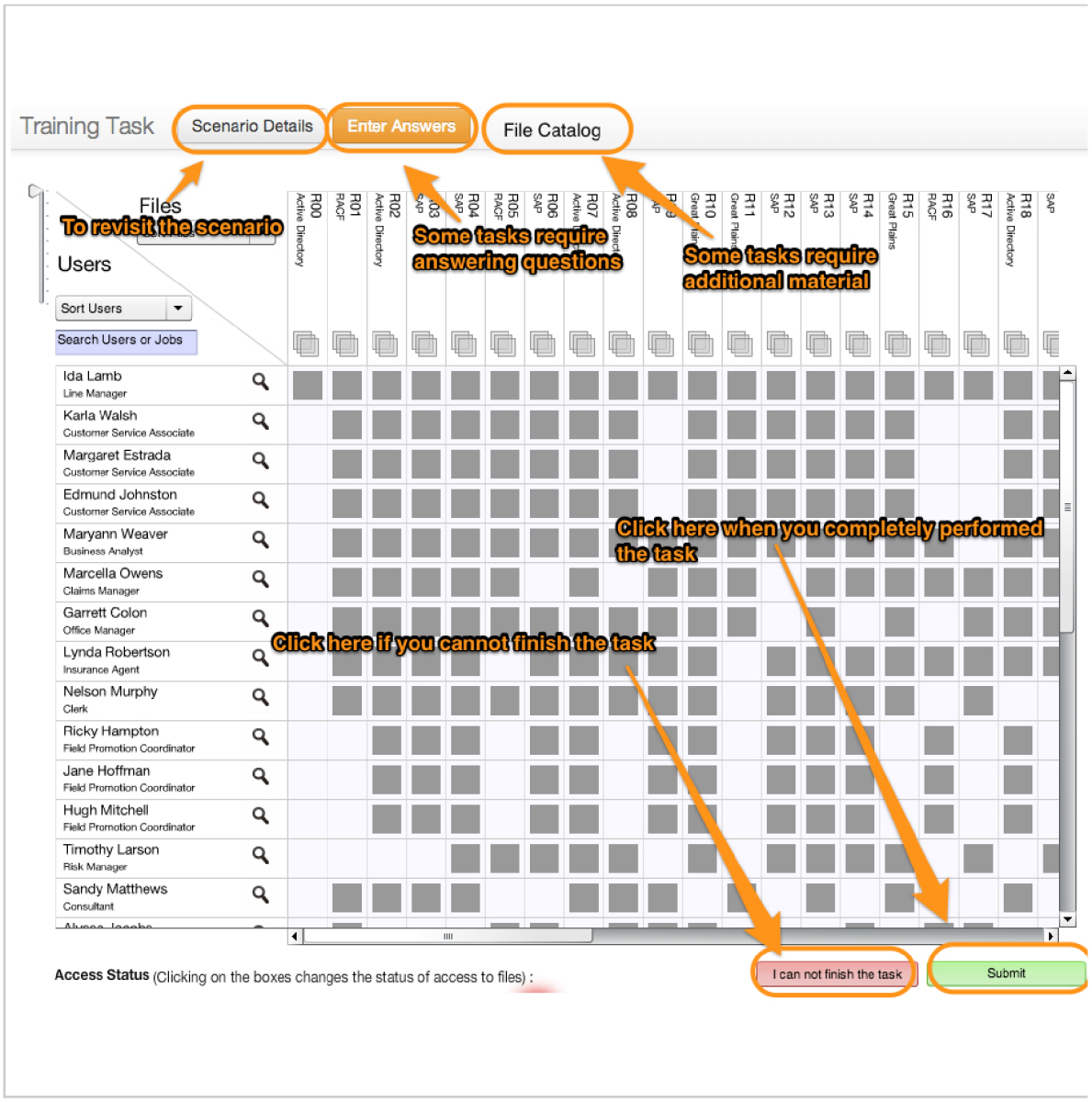


On the task overview page, follow these two steps:

- Read the scenario first
- Then click on the "Start!" button

Performing tasks

You will be then presented with the system, and required material for perfoming the tasks. For each task, we measure the time you spend on performing the task. Therefore, **make sure that you do not take breaks within a task.** Some tasks require certifying and revoking access using the interface, and some tasks require answering questions. When you complete the task, you will need to click on the **Submit** button on the interface to submit your work and proceed to the next task.



You may want to consider the following points when performing tasks:

- Make sure to check **the top menu items**. Some tasks require you to use additional material and some do not. If any additional tool is required for completing the task, **the link to the tool will be provided in the top menu.**
- If you see the *Enter Answers* button on the top of the screen, it means that to complete the task you need to answer questions. Click on the *Enter Answers* to answer questions.
- You can always check the scenario description using the **Scenario Details** button
- After you complete the task, click on the **Submit** button to record your results
- If you could not finish a task, but want to proceed to the next task, click on the **"I can not finish the task"** button. In this case, you **can not** return to this task later.

If you are ready to perform the tasks, please proceed: Proceed to tasks

© LERSSE
University of British Columbia

Version 1.0 -- January 15, 2013