

Exploring Twisted Paths: Analyzing Authorization Processes in Organizations

Steffen Bartsch
TZI, Universität Bremen
Bremen, Germany
Email: sbartsch@tzi.org

Abstract—Problems in organizational authorization result in productivity impacts and in security risks, for example, from over-entitlements and non-compliance. Many of the problems originate from organizational dynamics in combination with problematic authorization procedures for permission changes. To mitigate these problems and to improve the processes or craft supporting tools, a solid understanding of the processes and interactions between stakeholders is required. However, little prior empirical research covers authorization procedures. This paper presents an exploratory study of the procedures in organizational contexts. To enable a systematic analysis, an authorization process model is introduced that focuses on the interrelation of stakeholders and activities. The study discusses process characteristics, including the degree of centralization and the formality of interactions. Beyond this study, the model should serve as a basis for further research and support process designers to identify potentials of improvements.

Keywords—Access control, Authorization, Security management, Process management

I. INTRODUCTION

When authorization measures enforce overly restrictive or permissive authorization policies in information systems, organizational goals are impacted from productivity losses when employees cannot complete their tasks efficiently, or from ineffective security. The latter effect is either caused directly, from missing restrictions, or indirectly, when employees are forced to circumvent the system in dangerous ways. The dynamics of organizations make it difficult to maintain adequate policies to mitigate these effects, as, according to Schneier, “employees’ roles change all the time (...) and it’s often not obvious what information an employee needs until he actually needs it” [40].

These dynamics in authorization require adaption of the policy [44]. If the lead time or the effort for policy changes is too high, employees might not attempt to request changes, but rather fail to comply with the security policy [3]. Also, entitlements often remain assigned despite organizational changes [41]. Moreover, policy management can suffer from problems in the interaction of decision makers and policy authors [2]. Thus, the authorization procedures can be crucial for organizational productivity and security.

Prior work on authorization has primarily focused on the authoring and decision making for authorization policies, pointing out the usability problems with policy authoring interfaces [45] as well as with the comprehensibility and expressiveness of authorization models to formulate the policies [2], [7], [42].

However, security research also needs to consider the socio-organizational perspective [11]. In authorization, functional users provide valuable input on the necessary permissions for efficient work and their workflows need to be integrated with authorization [44]. Moreover, to implement adequate authorization models in the systems and integrate systems in the infrastructure, we also need to consider the perspective of systems development [24]. All these stakeholders need to contribute their tacit knowledge to achieve adequate measures, as Flechais and Sasse argue in case of security usability [13].

In order to improve the organizational productivity and security through adequate procedures and supporting tools, we first need to better understand the actual procedures today, particularly the interaction of the stakeholders and their activities. One aim of this paper is to explore authorization procedures in organizations to build a foundation for further research. However, prior process models for authorization do not offer the appropriate concreteness or neglect parts of the authorization ecosystem. This paper, thus, proposes the *Integrative Authorization Development Model for Usability* (INDUSE) for the systematical analysis. Beyond this study, the model provides a foundation for further research in the area and should also support process designers in improving authorization procedures.

II. BACKGROUND

A. Authorization process models

According to Dai and Alves-Foss, the policy life cycle encompasses the establishment, maintenance, and analysis on the general or tactical level as well as the specification, refinement, and integration of policies on the development or operational level [9]. Rees et al. propose a framework for the security policy life cycle on the tactical level, implementing a Plan–Do–Check–Act cycle [36]. On a similar tactical level, but from an IT operations perspective, the ITIL process “Access Management” includes receiving and validating change requests, providing permissions, and monitoring [32]. Closer to operation is ISO/IEC 29146 “A framework for access management”, which will focus on technical aspects of the policy management processes when finished [20]. However, those approaches are too narrowly focused on the policy management to capture the full breadth of authorization procedures in practice and lack an interrelation with end users and systems development. Kern et al. [24] define a life cycle for

role-based authorization with interrelating stakeholders from administration and development, albeit, on a technical level.

General security management processes have broader approaches, but consider authorization only abstractly. ISO/IEC 27002 [19] provides guidance on achieving security goals in several areas, including access control and human resources security. In IT management, ITIL [32] and CobiT [22] define processes and activities that can support authorization management, for example, request fulfillment, incident/problem management, and configuration management. In secure systems development, SSE-CMM only touches upon authorization as part of the controls to be implemented [21].

The discussed process models are either too abstract to adequately describe the actual interrelations in authorization procedures in practice, or they focus on specific areas or technologies and, thus, cannot support the analysis of the interrelation of different perspectives. Instead, an integrative approach is required.

B. Organizational authorization contexts

There has been little prior research on organizational authorization contexts. Authorization contexts are diverse and differ between the information systems within organizations, depending, amongst others, on the criticality of the contained data and the served business processes, and the implemented authorization model [19]. Information security is shaped by the organizational context [26]. Similarly, authorization context can be expected to be shaped by the organizational context, which is traditionally defined by its complexity as well as by the degree of formalization and centralization [28]. The complexity can be measured through the size and the organization's horizontal and vertical differentiation. The centralization can be determined on the *Decentralization Continuum*, between centralized hierarchies (military) and markets [27].

A further important factor of organizational context is the general risk level with respect to information security that the organization is operating under. This includes the business risks from information disclosure and data unavailability as well as the regulatory environment [19], [34]. Regulations may be market-specific, such as SOX and HIPAA, or depend on the type of data, such as privacy regulations [16].

III. INDUSE – AN INTEGRATIVE MODEL

INDUSE is a descriptive model of authorization procedures that focuses on the integration of and interrelations between stakeholders and activities. It is derived from literature on concrete authorization activities and high-level processes (cf. Section II-A). INDUSE goes beyond the existing process models by offering both a broad, integrative perspective on the activities and the level of detail to effectively describe real-world authorization procedures. INDUSE takes a software-engineering perspective on authorization: We consider the authorization policy an artifact that is adapted with the changing context, with defects affecting productivity and security. The model follows agile development models in that it emphasizes the integration of and communication between stakeholders,

TABLE I
INFORMATION FLOWS FROM THE FUNCTIONAL ACTIVITIES

F.1 → A.1	Task descriptions and security needs
F.2 → A.1	Policy decision problems
F.2 → A.4	Policy specification problems
F.2 → D.4	Defects related to decisioning and enforcement
F.3 → A.5	Indications of inadequate restrictions

and the context-specific tailorability of the process, covering light-weight as well as rather formal procedures [17].

In Figure 1, the INDUSE authorization development activities are shown for the stakeholder perspectives (see below), indicating the expected flows of information between perspectives as interrelations (cf. Tables I, II, and III). The activities can be assembled into processes, such as separate systems development and policy management processes.

A. Perspectives on authorization

Users are affected by authorization measures in very different roles and can be categorized by their respective perspective on authorization, similar to the ITIL concept of functions [33]. Functional staff and managers foremost regard authorization from the perspective of completing tasks, having a *functional perspective*. Policies defining authorization restrictions need to be authored (e.g. by security administrators), and decisions on changes to the policy need to be made (e.g. by functional managers) from an *administrative perspective*. Developers and administrators need to implement authorization in systems and integrate systems in provisioning infrastructures, having a *development perspective*. Outside of the scope of INDUSE, we find the *strategic* and *tactical security perspectives* that are concerned with high-level framing of authorization decisions and the process design, respectively.

B. Functional activities

1) *Define security needs (F.1)*: To enable well-founded decisions on what permissions to grant, functional stakeholders need to provide the security needs of resources, including who requires the availability of resources to carry out functional tasks and how critical the integrity and confidentiality of the resources is [21]. Eliciting the needs can be supported through task analysis [18] and should also take the system's purpose and the legal context into account [21]. According to ISO 27002 [19], business requirements for authorization include the risks that information is facing, relevant legislation, contractual obligations, and general compliance requirements.

2) *Functionally assure authorization (F.2)*: Functional stakeholders can be integrated into the assurance (in contrast to administrative assurance in A.4) and assess whether policy changes are adequate concerning their primary tasks, functionally assuring changes [19]. Typical means are acceptance tests [6] and reviews of authorization policies, either as a sign-off process step or in regular self-reviews [41].

3) *Functionally operate systems (F.3)*: The goal of functional stakeholders is to complete their primary tasks. This activity enables them to cope with interference from authorization measures, for example, when they are unable to complete

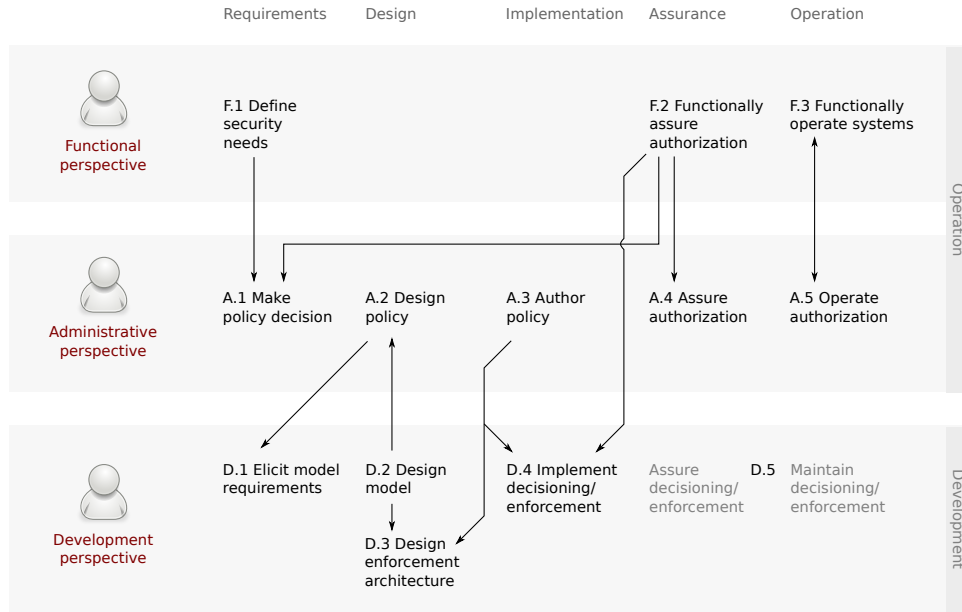


Fig. 1. INDUSE activities and interrelations

TABLE II
INFORMATION FLOWS FROM THE ADMINISTRATIVE ACTIVITIES

A.1 → A.2	Policy decisions
A.2 → A.3	Policy design
A.2 → D.1	Model expressiveness requirements for policy design
A.3 → A.2	Problems with policy design
A.3 → A.4	Policy specification
A.3 → D.3	Specific requirements on controls architecture
A.3 → D.4	Specific requirements on provisioning and controls
A.4 → A.1	Problems with decisions
A.4 → A.2	Policy design problems
A.4 → A.3	Policy specification problems
A.5 → A.4	General authorization problems
A.5 → F.3	Dissemination and disciplinary measures

a task because of missing permissions, and provides efficient and effective ways to mitigate the problems, for example, through change requests.

C. Administrative activities

1) *Make policy decisions (A.1)*: Policy makers need to make decisions on how the policy should govern the actions of users in systems, for instance, only allowing specific users to read sensitive data. The decisions may be taken on different levels of abstraction: high-level, business-focused (“enforce the need-to-know principle”) as well as low-level decisions focused on specific permissions and role assignments (“HR users may access personnel data”). The decisions are based on the functional security needs and adhere to the strategies on how to take policy decisions. Policy makers often apply the *need-to-know* and *least-privilege* principles. Decisions can also be based on organizational structures in top-down approaches [8] and often also account for the risks involved [21], [19].

2) *Design policy (A.2)*: The design of the policy defines the overall approach to authoring the policy, including, for

example, how the organizational structures should be mapped to the policy [8]. For role-based authorization, the design can be derived in role mining from the users’ tasks or permissions [4] as part of a role engineering process [30], [24].

3) *Author policy (A.3)*: In this activity, policy authors actually implement the policy and, for example, assign permissions to roles and roles to users. The policy specification occurs on multiple layers: close to the strategic layer, policies consist of abstract business rules, while, for example, role authoring and role assignment result in more concrete policies. Policy changes should be traceable, for instance, through configuration management [21].

4) *Assure authorization (A.4)*: To verify that authorization has the intended effect, policy authors and makers can conduct assurance activities to guarantee, for example, that an implemented policy complies with higher-level policies and concrete changes adhere to the decisions. Assurance can be conducted on the policy decisions, the policy design, and specification. Policy makers can conduct policy reviews, as sign-off of changes or at regular intervals [19]. Static or dynamic analysis can be conducted for implemented policies to assure specific characteristics. A validation and verification plan can define what artifacts should be assured by whom with what practices and at what points [21].

5) *Operate authorization (A.5)*: Operate authorization supports the authorization measure. On a rather technical level, monitoring of access violations, abnormal activities, and non-compliance with security policies can help to identify inadequate policies and other defects [19]. On a management level, incidents can require disciplinary measures as deterrent and additional containment activities [19]. Appropriate training and awareness campaigns can prevent incidents in the first place [37], [19].

TABLE III
INFORMATION FLOWS FROM THE DEVELOPMENT ACTIVITIES

D.1 → D.2	Authorization model requirements
D.2 → A.2	Available policy constructs
D.2 → D.3	Decisions to be enforced for authorization model
D.2 → D.4	Authorization model
D.3 → D.4	Enforcement architecture
D.4 → D.5	Decisioning and enforcement implementation
D.5 → D.2	Authorization model defects
D.5 → D.3	Enforcement architecture defects
D.5 → D.4	Decisioning and enforcement defects

D. Systems development activities

1) *Elicit authorization model requirements (D.1)*: Development stakeholders define the requirements that the authorization model needs to fulfill with respect to the targeted policy design, for instance, whether a role-based policy should be implemented or how dynamic the policies are.

2) *Select/design authorization model (D.2)*: Development stakeholders have to choose an adequate authorization model that addresses the requirements from D.1. Existing authorization models include discretionary (ACL) and mandatory (BLP) [35], role-based [12], attribute-based (XACML), and a broad range of other approaches [39].

3) *Design enforcement architecture (D.3)*: For the enforcement architecture, developers design how the authorization decisions are enforced in the program control flow, for example, through reference monitors for complete mediation [1], [38]. Enforcement approaches include runtime-system mechanisms, such as in Java [15], Aspect-oriented programming (AOP) [25], and enforcement hooks in the source code [15], [23]. Typically, the architecture defines a separation of concerns for the authorization decision and enforcement [5], [10].

4) *Implement decisioning and enforcement (D.4)*: Based on the algorithms defined by the authorization model and a given policy, the decisioning is implemented and derives whether to allow or deny an access request, for example, whether specific data may be displayed to a user. For the enforcement of the decisions, developers often insert enforcement controls in the program control flow [15], [23]. When integrating a system in an existing authorization infrastructure, the central policies need to be distributed to the system and mapped to the system's authorization model [29].

5) *Assure and maintain decisioning and enforcement (D.5)*: The authorization-specific assurance and maintenance practices follow the typical secure systems development and are thus consolidated in INDUSE. Assurance should be based organizationally on a verification and validation plan [21] and technically on dynamic (testing) and static analysis [14]. Maintenance similarly requires secure systems development practices, for example, vulnerability management [19].

E. Strategic and tactical perspectives

INDUSE supports the design and evaluation of authorization processes, including the definition of the environment for adequate policy decisions. The stakeholders from the tactical perspective compose processes from sequences of activities

with responsibilities, process inputs and outputs, and process performance indicators [31]. Process triggers act as the starting points for policy changes, for example, from software or organizational changes.

In most cases, multiple interrelated processes will be established in parallel. In small organizations with one systems development project, one operation and one development process may suffice. For larger organizations with several levels of hierarchy, layered processes are more appropriate. For example, high-level policies at a strategic level, close to business goals can be managed in a process independent from the changes to roles that follow organizational changes to accommodate the different change frequencies. On the development side, one development process could be instantiated per software development or integration project.

IV. STUDY DESIGN

Since there is little prior work on authorization processes in organizations, the primary purpose of the study is to systematically explore the processes in diverse environments and provide a basis for further research into authorization procedures. This includes studying (a) how policy changes occur in practice, and (b) how actors and activities interrelate. As secondary aims, the study evaluates the ability of INDUSE to describe diverse authorization processes and to support gap analyses by indicating potentials of improvement.

A. Research instruments

The primary source of the study are semi-structured interviews of between 30 and 90 minutes with stakeholders close to IT or organizational management. The interviews covered the authorization context (type of system, number of users), the policy change procedures (change activities, responsibilities), and the interrelation with systems development (systems integration, enforcement implementation). As secondary source, we reviewed process documents relating to authorization for two of the studied organizations.

B. Sampling

Organizations are generally reluctant to disclose information related to the sensitive subject of information security unless there is a trust relationship with the researching party, making broad empirical studies in this field difficult. Kotulic and Clark advise to focus on a small number of in-depth analyses with carefully selected organizations [26]. In this study, we likewise focus on a small number of cases, applying a careful sampling to cover a broad field of organizational contexts (see Section II-B). We operationalized the authorization contexts into two sampling dimensions, organizational complexity (number of users) and estimated risk-level (business risks, regulations). Eight organizations were individually contacted through personal contacts for this study (cf. Table IV), distributed according to the sampling dimensions as depicted in Figure 2.

TABLE IV
AUTHORIZATION CONTEXTS AND PROCESS CHARACTERISTICS

	Systems	Type of development	Users	Roles	Systems proc.	dev.	Operational process	Degree of formality	Process triggers
Large bank	Ca. 200 banking systems	Custom and COTS	50000	n/a	Per-system		Central and per-system	Formal	Software, organizational
Central uni. IT	E-learning, email, ...	Custom and COTS	37000	10	Local		Central and local	Informal	Software, organizational
Hospital	Numerous medical systems	COTS	3500	900	Per-system		Central and local	Formal	Organizational, review
Regional bank	Ca. 150 banking systems	Custom and COTS	3000	100	Per-system		Central	Formal	Software, organizational, review
Food industry	Navision-based ERP	Individualization	1000	200	Central		Central	Informal	Software, hindrances
University admin.	SAP-based ERP	Individualization	200	800	Central		Central	Formal	Software, organizational, hindrances
Charity org.	Business application	Web Custom	150	20	Central		Central	Informal	Software, hindrances
Quality assurance	Business application	Web Custom	100	16	Central (w/role modifications)		Central (role assignment)	Informal	Software, hindrances

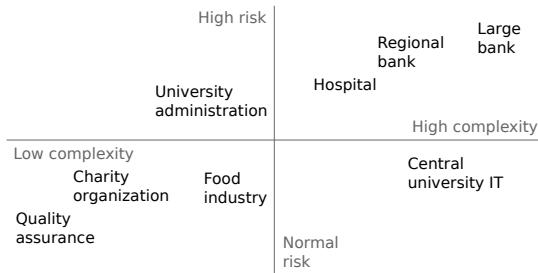


Fig. 2. Sampling of organizations in study

C. Analysis with INDUSE

The goal of the analysis is to understand the processes and interrelation of activities concerning authorization, both on the formal and informal level. From the detailed interview notes and the process documents, where available, we extracted data on the processes into a spreadsheet and graphically modeled the activities and interrelations of each case, as shown in one example for the regional bank in Figure 3. We systematically analyzed the processes in the following categories:

- *Process characteristics*: process formality, triggers, and instantiation,
- *Perspectives and stakeholders*: presence of perspectives, and the respective roles and tasks of stakeholders,
- *Activities and interrelations*: coverage of INDUSE activities, interrelations between activities.

V. FINDINGS ON AUTHORIZATION PROCEDURES

A. Process characteristics

Depending on the organizational context, authorization processes are defined in varying degrees of formality, from implicit, “lived” processes to those laid out in process documents. As shown in Table IV, larger organizations in this study are more likely to have a formally defined process. Of the four largest contexts, only one, the central university IT, has an informal process, largely because of the focus on the identity

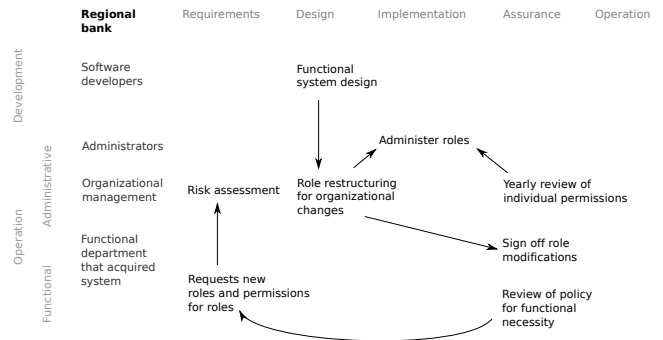


Fig. 3. Example of procedure visualization based on INDUSE activity model

management and infrastructure part of authorization. Likewise, three of the four smallest contexts only have informal processes.

Four different triggers for policy changes can be observed in the organizations (cf. Table IV):

- 1) Functional software changes (7 cases) that, for example, require new roles for added functionality or require role modifications to preserve behavior,
- 2) Organizational changes (5), for example, when processes are restructured or additional departments need to gain permissions to fulfill specific tasks,
- 3) Hindrances to functional stakeholders (4) in their primary work tasks,
- 4) Policy reviews (2) that are regularly conducted to identify inadequate and out-of-date permission assignments.

Organizations mostly establish separate processes for systems development and operational policy management (cf. Table IV). Only in the agile development case of the quality assurance company, the systems development process includes role modifications, although roles are assigned in an operational process. In the further three of four smallest cases, one central software development and one central policy management process are established. In larger cases, separate

processes for role modifications, role assignments, and user creation are instantiated, in part running locally, such as in the hospital case. In cases with numerous systems, several systems development processes are established in parallel.

B. Perspectives and stakeholders

As shown in Table V, in seven of the eight cases, all three INDUSE perspectives are present in the process descriptions. Only the central university IT lacks the functional perspective due to the focus on the infrastructure. Overall, the functional activities often relate to requesting authorization changes (5 cases), either directly or through discussions about functional requirements with development stakeholders. Other activities are quality assurance of changes (3), sign-off of requested changes (2) and reporting inadequate or defective restrictions (3). In two formal cases, technically-trained key users support policy changes and reviews. Only in the three smallest cases, functional users are explicitly incorporated in the process.

Stakeholders from the administrative perspective are either from the central IT departments or decentralized into local functional departments. In the case of the regional bank, organizational management staff is responsible for explicit requirement and design activities. In structured processes, such as the large bank and the hospital case, stakeholders from central IT and local administrators are both involved and interact in administrative activities, with more complex and critical activities centralized. For example, while role assignments are conducted locally in the hospital context by the associated role administrators, roles are only modified centrally. Explicitly assigning responsible stakeholders to roles is common for formal contexts (2 of 4 cases).

Authorization-related software development activities are mostly embedded in general software development activities. In three of the smaller cases, the administrative perspective and the development perspective are partly merged. This results either from in-house development (food industry) or from the policy management as part of the software development (quality assurance).

C. Activities and interrelations

Assessing which of the studied processes covers which INDUSE activity, as shown in Table VI, can provide insights into the process characteristics. Several processes lack or only partly cover the authorization activities in the areas of explicit decision-making (A.1; 3 full, 4 partly of 8), design (A.2; 3/2/8), and assurance (A.4; 4/0/8). None of the processes take the monitoring of authorization, disciplinary measures or awareness/training (A.5) into account. The analysis of the activities also indicate specific focuses of the processes. For example, the central university IT has only few functional and administrative activities due to their infrastructure focus, the hospital case few development activities, employing mainly COTS systems.

The formally defined and informal interrelations between activities, listed in Table VI, show how diverse the authorization procedures are in practice. In six processes, there are

formal or informal flows from security needs (F.1) to the administrative activities. In the other cases, the processes seem to be more development or administration-focused. While there are interrelations mentioned between systems development and administrative activities in all cases, for example, to reflect new or changed functionality in roles, eight of the nine mentions are informal ones. In the university administration, case the lack of a formal interrelation frequently causes problems of missing permission in operation after software updates.

VI. DISCUSSION

The study shows how broad the range of authorization development processes is in actual organizations. The limited number of cases in the study and the use of subjective data from interviews pose a threat to the study's validity. However, the careful sampling should result in a good spread of cases to provide a solid description of current authorization processes. While the subjectivity of interviews may cause in places imprecise individual descriptions, the larger picture as a hypothesis for more comprehensive research should be unaffected.

We employed the INDUSE analysis method laid out in Section IV-C to systematically describe and compare the processes. The *process characteristics* supported the categorization of processes (formality, instantiation) and indicated potential improvements (triggers). *Perspectives and stakeholders* showed how the processes are constructed in terms of centralization and how the perspectives are interrelated. *Activities and interrelations* provided details on the focus of the processes and the informal and formal interrelations, indicating missing formal relations. The informal relations between development and operation show that the integrative modeling approach, including development, is indeed necessary.

While several of the identified interrelations are not directly suggested by INDUSE, most unexpected ones represent short cuts of INDUSE interrelations, for example, F.2→A.3 for the hospital case where INDUSE suggests F.2→A.4→A.3. Unexpected interrelations are present between functional stakeholders (F.2→F.1) for reconsidering functional needs after reviews and from software requirements to policy decisions (D.1→A.1) when functional feature discussions lead to authorization changes. The expected INDUSE interrelations should thus be considered as lower bounds at the current stage and will be extended in further research.

From this study, the following recommendations on authorization process design can be derived:

- *Integrate functional staff*: Only half of the organizations in the study explicitly integrate functional users. The example of the charity organization indicates that it can be useful to motivate functional users to express their problems with authorization measures. In other cases, existing informal interactions could profit from a formalization. According to security usability models [3], acting early to increase the measure acceptability will not only improve the productivity, but also increase compliance and thus overall security.

TABLE V
STAKEHOLDER ROLES AND TASKS BY PERSPECTIVE

	Functional	Administrative	Development
Large bank	Functional dep.: provides functional role concept, requests change, QA	Functional admin.: role concept; Central admin.: implements changes	Developer: integrates enforcement
Central uni. IT		Admin.: assigns role to function/person; Local admin.: manage role subtree	Developers: design role subtree
Hospital	Line manager: requests role change; key user: signs off; requester: QA	Role admin: coordinates, changes roles; Local admin: assigns role	Developer: integrate external systems
Regional bank	Functional dep: requests role changes, signs off role changes, reviews policy	IT: manages role; Org. management: assesses risk, restructures, reviews	Developer: modifies system, requests new roles
Food industry	Functional department: informal requirements	Admin: manages roles	Developers: integrate enforcement with functional changes
Uni. admin.	Line manager and key user: request change; end user: reports defects	Admin: changes roles, validates requests, corrects roles after updates	Developer: implements system update
Charity org.	Functional dep: discusses policy change; end user: QA, reports defects	Functional consultant: design, implement policy; conduct QA	Developers: implement authorization model and enforcement
Quality assurance	App. owner: requests user story, discuss auth. reqs; end user: report problems	Developer/admin: modifies roles; App. owner: assigns roles	Developer: implement authorization with functionality

TABLE VI
ACTIVITIES COVERED BY THE STUDIED PROCESSES (O: PARTLY, X: MOSTLY/FULLY; INFORMAL INTERRELATIONS SET IN ITALICS)

	F.1	F.2	F.3	A.1	A.2	A.3	A.4	A.5	D.1	D.2	D.3	D.4	D.5	Interrelations
Large bank	x	x		x	x	x					o	x		F.1→A.1→A.2→A.3, A.2→D.4→A.3
Central uni. IT					o	x			o	x	x			<i>D.2→D.4, D.2→A.2, D.2→A.3</i>
Hospital	x	x		o		x	x					o	o	F.1→A.1→A.3, F.2→A.3, <i>D.4→A.3</i>
Regional bank	x	x		x	x	x	x			x				F.1→A.1→A.2→A.3, A.4→A.3, <i>D.2→A.2, F.2→F.1</i>
Food industry	x		x	o		x			o			o		<i>F.1→A.1→A.3, F.3→A.3, D.4→A.3</i>
Uni. admin.			o	x		x	x					o		<i>A.1→A.4→A.3, F.3→A.3, D.4→A.3</i>
Charity org.	o	x	o	o	x	x	x		o	o	o	x		<i>F.3→A.1, F.2→A.4→A.2, D.1→A.1</i>
Quality assurance	x		x	o	o	x			o	o	x			<i>F.1→A.1, F.3→A.3, F.3→A.1, D.2→A.2</i>

- *Establish operative activities*: None of the studied organizations has operative measures for authorization in place. However, monitoring authorization problems, such as circumventions, will help to react quickly and increasing awareness will further improve acceptability and thus overall security [43].
- *Integrate systems development*: Considering the full authorization life-cycle, including the systems development aspects [24], can reduce inadequacies of policies. One negative example is the university administration case, where productivity of functional users is impacted after software updates because permissions are missing. This could be mitigated by a formalization of these interactions.
- *Improve stakeholder interaction*: Interrelations are not only significant between perspectives, but also within perspectives, for example, when local and central administrators cooperate. As seen in the central university IT case, it can be problematic if there is no defined channel for interactions [2], [13].

While the study focused on authorization contexts in organizational environments today, INDUSE should also be applicable to future environments. One development in organizations is increasing its degree of distribution, requiring local decision making to lower security management costs [34]. As shown for the hospital case, where stakeholders from functional departments have administrative perspectives, this can be modeled well with INDUSE. Similarly, distributed

services (cloud computing) can be modeled as individual systems as in the case of the large bank. Processes in federated authorization architectures can be represented as distributed decision making and distributed systems development, and need to consider the provisioning and merging of policies for individual systems. Another development are privacy enhancing technologies (PETs), which cause additional high-level requirements, for example, from contracts with customers, to be incorporated in the policy decision and model requirement activities. Future technologies to guarantee these contracts can be integrated as part of the development activities.

VII. CONCLUSIONS

This paper studied the authorization contexts of eight organizations in order to improve our understanding of authorization processes, on which only little prior work exist. The model INDUSE is proposed for the systematic analysis of the processes. The findings on authorization processes offer a thorough description and a good initial hypothesis for further research in this area. The variety of processes and the differing degrees of formality show how challenging the systematic analysis of the “twisted paths” of authorization processes can be. Moreover, it demonstrates that researchers will often need to consider or even modify the existing processes in organizations to fundamentally address the authorization challenges.

The study indicates that INDUSE is well suited to describe authorization processes. Practitioners can profit from it in gap analyses and to understand existing processes. In research, INDUSE provides a solid basis for further, more focused

research into the nature and improvement of authorization processes by precisely describing and categorizing them.

VIII. ACKNOWLEDGMENTS

I would like to express my thanks to the interviewees for investing their time and placing the necessary trust in me. I would also like to thank Karsten Sohr for his valuable input on early drafts of this paper.

REFERENCES

- [1] J. P. Anderson. Computer security technology planning study. Technical Report ESD-TR-73-51, Deputy for Command and Management Systems, L.G. Hanscom Field, Bedford, MA, October 1972.
- [2] L. Bauer, L. F. Cranor, R. W. Reeder, M. K. Reiter, and K. Vanica. 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, 2009. ACM.
- [3] A. Beautement, M. A. Sasse, and M. Wonham. The compliance budget: Managing security behaviour in organisations. In *New Security Paradigms Workshop 2008*, 2008.
- [4] E. Bertino, S. Calo, H. Chen, N. Li, T. Li, J. Lobo, I. Molloy, and Q. Wang. Some usability considerations in access control systems. In *USM '08: Workshop on Usable IT Security Management*, 2008.
- [5] K. Beznosov, Y. Deng, B. Blakley, and J. Barkley. A resource access decision service for corba-based distributed systems. In *Annual Computer Security Applications Conference*, page 310, Los Alamitos, CA, USA, 1999. IEEE Computer Society.
- [6] B. W. Boehm. A spiral model of software development and enhancement. *IEEE Computer*, 21(5):61–72, 1988.
- [7] S. Brostoff, M. A. Sasse, D. W. Chadwick, J. Cunningham, U. M. Mbanaso, and S. Otenko. ‘R-What?’ development of a role-based access control policy-writing tool for e-scientists. *Softw., Pract. Exper.*, 35(9):835–856, 2005.
- [8] R. Crook, D. Ince, and B. Nuseibeh. Modelling access policies using roles in requirements engineering. *Information and Software Technology*, 45(14):979–991, 2003. Eighth International Workshop on Requirements Engineering: Foundation for Software Quality.
- [9] J. Dai and J. Alves-Foss. Logic based authorization policy engineering. In *The 6th World Multiconference on Systemics, Cybernetics and Informatics*, 2002.
- [10] B. De Win, F. Piessens, W. Joosen, and T. Verhanneman. On the importance of the separation-of-concerns principle in secure software engineering. In *ACSA Workshop on the Application of Engineering Principles to System Security Design*, 2003.
- [11] G. Dhillon and J. Backhouse. Current directions in is security research: towards socio-organizational perspectives. *Information Systems Journal*, 11:127–153, 2001.
- [12] D. Ferraiolo and R. Kuhn. Role-based access controls. In *15th NIST-NCSC National Computer Security Conference*, pages 554–563, 1992.
- [13] I. Flechais and M. A. Sasse. Stakeholder involvement, motivation, responsibility, communication: How to design usable security in e-science. *International Journal of Human-Computer Studies*, 67(4), 2009.
- [14] V. Ganapathy, T. Jaeger, and S. Jha. Retrofitting legacy code for authorization policy enforcement. In *IEEE Symposium on Security and Privacy*, pages 214–229, Los Alamitos, CA, USA, 2006. IEEE Computer Society.
- [15] L. Gong and G. Ellison. *Inside Java(TM) 2 Platform Security: Architecture, API Design, and Implementation*. Pearson Education, 2003.
- [16] D. S. Herrmann. *Complete Guide to Security and Privacy Metrics*. Auerbach Publications, Boston, MA, USA, 2007.
- [17] J. A. Highsmith. *Agile software development ecosystems*. Addison-Wesley, Boston, MA, USA, 2002.
- [18] E. Hollnagel. Task analysis: Why, what, and how. In *Handbook of Human Factors and Ergonomics*. Wiley, 2006.
- [19] ISO/IEC 27002:2005. *Information technology – Security techniques – Code of practice for information security management*. ISO, Geneva, Switzerland, 2005.
- [20] ISO/IEC JTC 1/SC 27 Secretariat. Standing document 7 (SD7): Catalogue of ISO/IEC JTC 1/SC 27 standards and projects. Online, retrieved 23 Sep 2010, 2009.
- [21] ISSEA. The systems security engineering capability maturity model (SSE-CMM), model document, 2003. Version 3.0.
- [22] IT Governance Institute. *CobiT 4.1*. ITIG, Rolling Meadows, IL, 2007.
- [23] T. Jaeger, A. Edwards, and X. Zhang. Consistency analysis of authorization hook placement in the Linux security modules framework. *ACM Trans. Inf. Syst. Secur.*, 7(2):175–205, 2004.
- [24] A. Kern, M. Kuhlmann, A. Schaad, and J. Moffett. Observations on the role life-cycle in the context of enterprise security management. In *SACMAT '02: Proceedings of the seventh ACM symposium on Access control models and technologies*, pages 43–51, New York, NY, USA, 2002. ACM.
- [25] G. Kiczales, J. Lamping, A. Mendhekar, C. Maeda, C. Lopes, J.-M. Loingtier, and J. Irwin. Aspect-oriented programming. In *ECOOP'97 Object-Oriented Programming*, pages 220–242, 1997.
- [26] A. G. Kotulic and J. G. Clark. Why there aren't more information security research studies. *Information & Management*, 41(5):597–607, 2004.
- [27] T. W. Malone. *The future of work: how the new order of business will shape your organization, your management style, and your life*. Harvard Business Press, 2004.
- [28] R. D. McPhee and M. S. Poole. Organizational structures and configurations. In F. M. Jablin and L. L. Putnam, editors, *The New Handbook of Organizational Communication*. SAGE, 2000.
- [29] A. Mönkeberg and R. Rakete. Three for one: role-based access-control management in rapidly changing heterogeneous environments. In *RBAC '00: Proceedings of the fifth ACM workshop on Role-based access control*, pages 83–88, New York, NY, USA, 2000. ACM.
- [30] G. Neumann and M. Strembeck. A scenario-driven role engineering process for functional RBAC roles. In *SACMAT '02: Proceedings of the seventh ACM symposium on Access control models and technologies*, pages 33–42, New York, NY, USA, 2002. ACM.
- [31] OGC. *ITILv3 – Continual Service Improvement*. TSO, London, UK, 2007.
- [32] OGC. *ITILv3 – Service Operation*. TSO, London, UK, 2007.
- [33] OGC. *ITILv3 – Service Strategy*. TSO, London, UK, 2007.
- [34] F. Pallas. *Information Security Inside Organizations – A Positive Model and Some Normative Arguments Based on New Institutional Economics*. PhD thesis, TU Berlin, 2009.
- [35] G. Pernul. Information systems security: Scope, state-of-the-art, and evaluation of techniques. *International Journal of Information Management*, 15(3):165–180, 1995.
- [36] J. Rees, S. Bandyopadhyay, and E. H. Spafford. PFIREs: a policy framework for information security. *Commun. ACM*, 46(7):101–106, 2003.
- [37] C. A. Roper, J. J. Grau, and L. F. Fischer. *Security Education, Awareness and Training: from Theory to Practice*. Butterworth-Heinemann, 2005.
- [38] J. H. Saltzer and M. D. Schroeder. The protection of information in computer systems. In *Proceedings of the IEEE 63-9*, 1975.
- [39] P. Samarati and S. de Vimercati di Vimercati. Access control: Policies, models, and mechanisms. *Foundations of Security Analysis and Design*, pages 137–196, 2001.
- [40] B. Schneier and M. Ranum. Schneier-Ranum face-off: Is perfect access control possible? *Information Security*, September 2009.
- [41] S. Sinclair, S. W. Smith, S. Trudeau, M. E. Johnson, and A. Portera. Information risk in the professional services – field study results from financial institutions and a roadmap for research. In *Proceedings for the 3rd International Workshop on Enterprise Applications and Services in the Finance Industry*, 2007.
- [42] D. K. Smetters and N. Good. 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, 2009. ACM.
- [43] R. West. The psychology of security. *Commun. ACM*, 51:34–40, April 2008.
- [44] T. Whalen, D. Smetters, and E. F. Churchill. User experiences with sharing and access control. In *CHI '06: CHI '06 extended abstracts on Human factors in computing systems*, pages 1517–1522, New York, NY, USA, 2006. ACM.
- [45] M. E. Zurko and R. T. Simon. User-centered security. In *NSPW '96: Proceedings of the 1996 workshop on New security paradigms*, pages 27–33, New York, NY, USA, 1996. ACM.