Automatically Extracting Business Level Access Control Requirements from BPMN Models to Align RBAC Policies

Roman Pilipchuk[1], Robert Heinrich[2] and Ralf Reussner[2]
[1]*FZI Research Center for Information Technology, 10117 Berlin, Germany*
[2]*Karlsruhe Institute of Technology, 76131 Karlsruhe, Germany*

Keywords: Business Processes, Access Control, RBAC.

Abstract: IT security becomes increasingly important due to the rise of cybercrime incidents but also obligatory security and privacy laws that include confidentiality regulations. To prevent cybercriminal attacks, the business level has to identify critical business data and introduce organization-wide security standards. A close cooperation with the IT level is crucial to avoid mistakes and misunderstandings of security requirements, both may cause severe security breaches. An important building block are access control requirements (ACRs). In a costly, complex and manual role engineering process, experts have to elicit appropriate role-based access control (RBAC) policies according to business security and confidentiality models. This paper makes a first step to close this gap with an approach that automatically extracts business level ACRs from BPMN business processes to build an initial RBAC role model and establish traceability from RBAC policies to business processes. Case study results indicate that the accuracy of extracted policies is appropriate, adaptations in evolution scenarios become faster and human errors are reduced during the engineering of RBAC policies.

1 INTRODUCTION

In times of obligatory security and privacy laws and a rising problem of cybercrime, IT security and data privacy are becoming crucial for organizations of all kind. To establish both appropriately, the business level of an organization (service design managers and compliance managers according to ITIL (AXELOS, 2011)) has three more goals to focus on: a) identify and protect critical business data, b) establish organization-wide IT security to prevent cybercriminal attacks and c) comply with the rising amount of security and privacy laws (Pilipchuk, 2018). These goals pertain access control requirements (ACRs). Guidelines like ISO 27000 (ISO/IEC, 2018), business process guidelines like ITIL and laws like the IT Security Act (Federal Republic of Germany, 2015) and the General Data Protection Regulation (European Union, 2016) impose requirements on access control. Thus, ACRs are fundamental to realize the three business level goals. They have to be incorporated by the IT level (enterprise architects and security experts). Different domain knowledge and domain-specific models widen a communication gap that leads to errors (Alpers et al., 2018). In addition, business and IT level affect each other in non-trivial ways (Aerts et al., 2004). For example, changes in business processes may require extensive adaptations in RBAC policies. So far, business and IT level are not well aligned (Wieringa et al., 2003), especially in terms of IT security and privacy (Alpers et al., 2019).

Role-based access control (RBAC) (INCITS, 2012) is widely used to restrict access in IT systems. It is beneficial in the management of access control and the provided degree of security (Ferraiolo et al., 2007). In 2010, NIST estimated that RBAC has saved the industry over $1.1 bil. over several years (O'Connor et al., 2010) by introducing roles comprising permissions. Users are assigned to roles (e.g. role *manager*). In RBAC, roles and permissions of an organization are gathered in a role model. However, a compliant incorporation of ACRs into the role model is challenging. Establishing RBAC is costly, complex and error prone (Ferraiolo et al., 2007). The challenge is to elicit appropriate roles and permissions matching the ACRs of the business level. Therefore, business processes have to be reviewed manually to understand the ACRs. Depending on the size of an organization business processes grow easily into hundreds resulting in a vast amount of complex and interrelated artifacts demanding specific business knowledge to understand them. The analysis of business processes is

Pilipchuk, R., Heinrich, R. and Reussner, R.
Automatically Extracting Business Level Access Control Requirements from BPMN Models to Align RBAC Policies.
DOI: 10.5220/0010184403000307
In *Proceedings of the 7th International Conference on Information Systems Security and Privacy (ICISSP 2021)*, pages 300-307
ISBN: 978-989-758-491-6

done manually by experts that may lead to errors inside the role model (Mitra et al., 2016). Each error can lead to a security breach undermining the aspired goals of the business level. Furthermore, continuous organizational evolution produces ACR changes that require repetitive role model adaptations.

In order to align the business level with the IT level in terms of ACRs, we introduce the approach *BPMN Access Permission Extractor* (BAcsTract). It extracts business level ACRs of role-permission type from business processes in XML (defined by BPMN (OMG, 2011)) automatically and transforms them to an initial role model for RBAC. BAcsTract reduces efforts in creating roles and access permissions of role models and helps security experts in aligning access permissions with business processes. This paper makes the following contributions: a) an ACR mapping model that interconnects elements of business processes specified in Business Process Model and Notation (BPMN) and access control elements of RBAC to establish traceability among the models, b) the approach BAcsTract and c) a case study evaluating BAcsTract by executing it with processes from a community driven case study of a supermarket chain.

The remainder of the paper is structured as follows. Sect. 2 presents state of the art. BAcsTract is introduced in Sect 3. In Sect. 4, an evaluation of the approach is done and Sect. 5 concludes the paper.

2 STATE OF THE ART

Role mining approaches analyze existing permissions of organizations with an algorithm to mine a role model. In (Mitra et al., 2016), a survey on role mining approaches was done. These bottom-up approaches provide roles from a technical perspective. Such roles only reflect the performed actions on data objects but not their business meaning. These approaches operate on the technical level. They cannot bridge the gap between the business and IT level nor can they analyze ACRs from the business point of view.

Role engineering approaches are carried out top-down. Experts decompose business artifacts like business processes manually into permissions that are required to carry out tasks (Ferraiolo et al., 2007). Then, these permissions are grouped into roles. Roles elicited with role engineering are business roles reflecting the hierarchy of an organization. Some role engineering approaches like (Coyne, 1996) describe the role engineering process from a high-level perspective and thus, lack details. Other approaches like (Crook et al., 2001) focus on IT level artifacts e.g. requirements engineering artifacts, rather than business level artifacts. In contrast to the approach in this paper, they cannot bridge the gap between the business level and IT level. Approaches like (Colantonio et al., 2009; Roeckle et al., 2000; Mark, 2010) lack scalability. They explain practices for experts on how to manually elicit a role model. However, the amount of business processes, which need to be analyzed, grows increasingly with the organizational size. The rigorous amount of human interventions required to analyze these business processes makes the proposed approaches error prone. Some approaches like (Narouei et al., 2015) try to automate role engineering but they do not focus on business processes. In the mentioned case natural language processing is used to analyze specifications in human language. While the approach of this paper is also a role engineering approach that focuses on business processes, the difference is that it automates the analysis of business processes and thus, tackles the above-mentioned problem of scalability and human errors. Another difference is that it generates an ACR mapping model that interconnects elements of business processes and RBAC, allowing to understand mutual dependencies and providing a documentation of design decisions.

Authors of (Fuchs et al., 2007) describe a structured process to introduce identity management in organizations. The approach focuses on proposing high-level manual steps for the business level regarding how to introduce and manage identities. In (Fuchs et al., 2008) they propose a combination of role engineering and role mining to mine roles that take business information into account. The approach of this paper might complement their role engineering part as they propose manual effort to extract relevant information from business artifacts. However, the authors do not provide detailed information on the concrete steps that are used to generate the role model. It seems that role mining is used to generate technical permission while business information is used to bundle them into roles. In contrast, BAcsTract extracts business permissions and provides a model that interconnects elements of business processes and RBAC.

Another approach (Ramadan et al., 2018) analyzes business processes to extract security requirements like confidentiality and integrity and transform them to UML diagrams. A major difference is that their approach requires the BPMN extension SecBPMN2 that introduces security related elements to BPMN. The approach of this paper is based on plain BPMN to allow organizations to utilize models that they design anyway. Hence, we do not consider BPMN extensions like SecBPMN2 (Brucker et al., 2012).

3 APPROACH

BAcsTract aligns business level and IT level artifacts of organizations in terms of ACRs. It extracts business level ACRs from business processes automatically and transforms them into an initial role model for RBAC. The approach can be used for example to establish RBAC, to check whether the role model is aligned with business processes or to adapt the role model in evolution scenarios of business processes.

We assume that ACRs incorporated in business processes by the business level are legally correct. The focus of this paper is not to identify falsely incorporated ACRs but to automate the transfer of ACRs from business processes to IT level artifacts.

BAcsTract interconnects elements of business processes and RBAC in an ACR mapping model to extract an initial role model and establish traceability and a documentation of design decisions. The ACR mapping model introduces two intermediate layers, in between roles and permissions that are specific for the context of business processes. In an organization the employee requires a set of permissions to fulfill the activities during his daily work. BPMN defines them as participants in lanes. RBAC assigns permissions to roles which are assigned to employees. Thus, lanes of business processes are conceptually similar to roles of RBAC and can be mapped on each other. They form the first layer (role) in the ACR mapping model. The work of an employee is about completing the set of activities in each of his business processes. Consequently, business processes (e.g. change price) are daily tasks of the employee, during which activities (e.g. change price of product) are fulfilled. Accordingly, the second and third layer of the ACR mapping model correspond to the business process itself and the activities of the employee's lane (process name and activity). In order to fulfill an activity, the employee requires the correct set of access permissions which are e.g. specified in RBAC. For example, to carry out the activity *change price of product*, the employee needs access permissions to the information system and to the service function for changing products. Some activities do not require permissions e.g. *sort product in shelf*. In business processes, activities can have associations with data inputs and outputs. Data input means that the activity requires an input in order to be carried out and thus, reads data. Data output means that the activity produces data and thus, writes data. Hence, the last layer (permission) specifies the required access to input and output data. By establishing the above-mentioned ACR mapping model (role, process, activity and permission), BAcsTract extracts ACRs from business processes and establishes a tracing between RBAC and business process elements. This model allows the business and IT level to understand the origin of each extracted access permission by tracing it back to their originating business process, lane, activity and input/output data. By doing so responsible employees, e.g. the business process owner, can be consulted if questions arise.

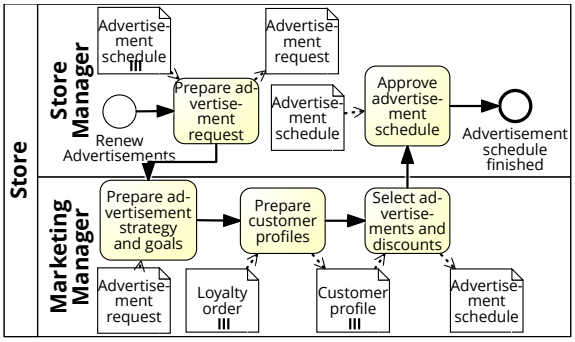

Figure 1: Business process of a supermarket chain for preparing advertisements and discounts.

BAcsTract is a role engineering approach that operates in six steps. Step one to four build the above-mentioned ACR mapping model by interconnecting roles, processes, activities and permissions. Step five and six form a simple hierarchy and extract the initial role model from the ACR mapping model. For better understanding we illustrate these steps with a simple running example shown in Fig. 1. However, the running example does not encompass all corner cases.

Fig. 1 shows a business process of a supermarket chain for the preparation of advertisements and discounts by the marketing manager and store manager. The store manager issues an advertisement request after exploring previous advertisement schedules. Upon receiving the advertisement request, the marketing manager prepares creates a new advertisement schedule, which comprises the advertisements and discounts. He creates customer profiles by the use of loyalty orders. These are orders from loyalty customers who agreed with a consent that their data is used for marketing purposes. Then, he selects advertisements and discounts depending on the customer profiles and finishes the advertisement schedule. Finally, the advertisement schedule is approved. Tab. 1 shows the ACR mapping model inside the database of BAcsTract after the running example has been processed. Hereafter, the extraction steps are explained.

Step 1: Roles are extracted from unique names of lanes with some exceptions. For example, closed lanes (like the customer in a sale process) or lanes without data association are not required. Pools represent organizational divisions of roles, making equal lanes distinguishable across organizational divisions.

The running example process in Fig. 1 has the lanes *Store Manager* and *Marketing Manager* that are part of the pool *Store*. They can be found in the column role of the ACR mapping model shown in Tab. 1.

Step 2: This step interconnects processes of roles, from step one. Each role is connected to only the processes it is part of. As our running example consists of one process, the process name is added to the ACR mapping model. See column process in Tab. 1.

Step 3: Each role's activities in a process are analyzed and added to the ACR mapping model. Subprocesses, are skipped to avoid duplicates, as they are analyzed as an own process in step 2. The activities of the running example can be found in the column activity of the ACR mapping model in Tab. 1. E.g. row one shows the *Store:Store Manager* with the activity *Prepare advertisement request* which is the first activity of the running example process in Fig. 1.

Step 4: This step extracts permissions from activities. Each activity is analyzed for data inputs and outputs. An association from the data object to the activity is a data input and thus, a read operation on the data object. An association from the activity to the data object is a data output and thus, a write operation on the data object. Data objects associated with a sequence flow have the same meaning as associations with activities. The problem of indistinct plural named data objects is avoided through the isCollection attribute of the BPMN Standard. This is depicted by the three dashes in the data object *Advertisement schedule* shown in the upper left corner of Fig. 1. The three dashes mean that the store manager has a collection of advertisement schedules as input and thus, requires a read permission on advertisement schedule. Row one of the ACR mapping model in Tab. 1 illustrates this with the permission *R Adv. schedule*.

Step 5: A simple hierarchy is elicited. Permissions of each role, e.g. employee, are inspected whether they are a subset of another role's permissions e.g. manager. If this is the case, role manager inherits from role employee. Virtual roles may be introduced according to (Lee et al., 2004) to reduce the amount of duplicate permissions and ease permission management. To find a place to introduce virtual roles, each role's activities are compared to activities of other roles. If any activities are similar, a virtual role is introduced. Virtual roles are never assigned to employees. They only serve for abstraction purposes.

Step 6: The initial role model is extracted from the ACR mapping model. Each role in the layer role that has permissions, becomes a role in the role model. Its permissions from the layer permission are extracted and only unique permissions are stored in the role model. For example, row one of Tab. 1 provides the entry *Store:Store Manager* and *R Adv. schedule*.

The role model serves security experts as an initial role model comprising business level ACRs in form of role-permission pairs. Complex ACRs, like separation of duty, and technical ACRs cannot be extracted as they cannot be modeled in plain BPMN. Nonetheless, BAcsTract makes role engineering less error prone, as parts are automated and the role model becomes better aligned with the business processes.

4 EVALUATION

4.1 Case Study

BAcsTract is executed with 17 processes and evaluated for: a) increasing efficiency by reducing complexity of the role engineering process, b) reducing human errors during the engineering of the role model and c) helping to faster adapt the role model during evolution scenarios. The Common Component Modeling Example (CoCoME) is used as a case study. It is a community driven case study for empirical research on software evolution approaches (Heinrich et al., 2016). It represents a comprehensive trading system of a supermarket chain. From the CoCoME

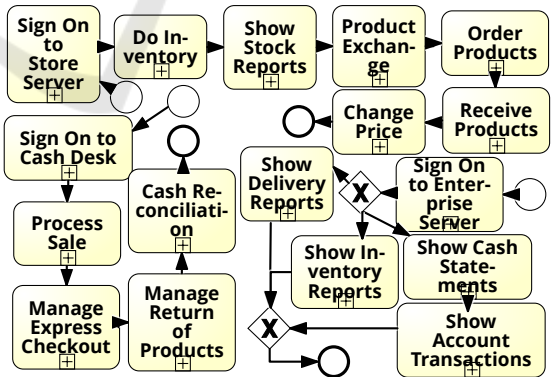

Figure 2: Overview of sub-processes for the case study with CoCoME.

Table 1: Excerpt of the ACR mapping model for the running example.

#	Role	Process	Activity	Permission
1	Store:Store Manager	Prepare adv. and dis.	Prepare adv. req.	R Adv. schedule
2	Store:Store Manager	Prepare adv. and dis.	Prepare adv. req.	W Adv. req.
3	Store:Marketing Manager	Prepare adv. and dis.	Prepare adv. strategy and goals	R Adv. req.

enterprise all supermarket stores are managed. The enterprise server connects to each store server which is managed by a store manager. Each store server connects a set of cash desks forming a cash desk line.

Table 2: Characteristics of the business processes.

Business Char.	#	Business Char.	#
Business process	17	Lane	48
Activity	166	Flow transition	294
ACR	112	ACR unique/role	81

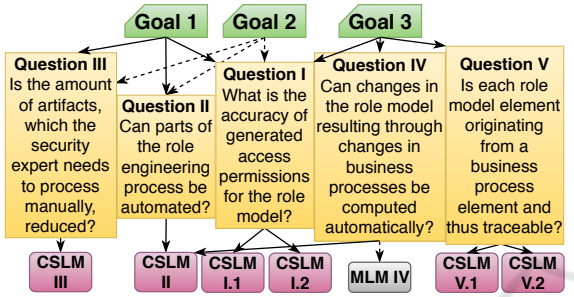

Figure 3: The GQM model.

Tab. 2 summarizes major characteristics of the CoCoME business processes. They demonstrate that the size of the case study is reasonable. Fig. 2 shows an overview of the 17 business processes. The lower left part shows the processes related to the cash desk management like the processing of purchased goods and cash reconciliation. The upper part shows the processes related to the inventory of the store like making inventory, ordering and receiving products. The right part shows the processes related to the enterprise incl. the access to delivery reports, cash statements, account transactions and inventory reports of each store. CoCoME is appropriate for examining ACRs as shown in (Pilipchuk et al., 2017). They show that CoCoME has to consider security requirements stemming from various sources, like laws and BPMN guidelines. ACRs were identified as an essential group of requirements. BAcsTract is implemented in Java and consumes BPMN processes in XML format (OMG, 2011). The evaluation is structured according to the GQM method (Basili et al., 1994) illustrated in Fig. 3. Goals (green boxes) represent evaluation objects that are desired to be achieved. Therefore, they are subdivided into research questions (yellow cards) that need to be answered in order to satisfy the goals. To confirm a hypothesis of a question, all corresponding metrics need to be reached.

Goal 1 - More Efficient Engineering of the Role Model. Typical role engineering consists of manual steps and thus, is slow and complex. Experts have to analyze a vast amount of business processes. The first goal is to validate that the engineering of the role model with BAcsTract reduces time and cost compared to the traditional role engineering process. Hence, the goal is to validate that BAcsTract can reduce complexity of the role engineering process by a) automating parts of the role engineering and b) transferring complete and semantically correct information about ACRs from business processes to RBAC.

Goal 2 - Reduction of Human Errors in the Engineering of the Role Model. The typical role engineering process requires analyzing a vast amount of business processes by hand which leads to errors. The second goal is to validate that BAcsTract reduces the amount of business processes that need to be processed by security experts manually and by doing so, reduces human errors (Haight, 2019).

Goal 3 – Faster Adaptation in Evolution Scenarios. Business processes have a lifecycle and thus, change constantly over time. The interrelations between business processes and access permissions are complex. Thus, evolution scenarios require changes of the RBAC role model. To identify and adapt these changes in the role model, security experts are required. While carrying out the traditional role engineering process for every evolution scenario repeatedly, they identify required changes. This is complex, cost intensive and slow. The third goal is to validate that BAcsTract allows faster adaptation of the role model during evolutional changes of processes.

Goals are connected to questions that need to be confirmed in order to satisfy the goal (see Fig. 3).

Question I is fundamental for all three goals as it examines whether BAcsTract extracts correct and complete ACRs. Therefore, the accuracy of the generated access permissions is measured. Hypothesis H I.1 claims that the transformation of ACRs from business processes into the role model is semantically correct. Hypothesis H I.2 claims that every ACR, of the type role-permission pair, is transformed from the business processes into the role model. To answer question I each generated access permission is classified based on a reference list of ACRs for the given business processes. An access permission is a true positive t_p if it has an exact counterpart in the reference list. An exact counterpart means that the role of the access permission corresponds to the lane and pool of an ACR, and that the data object and its read/write operation correspond to the data object and association of an ACR. It is a false positive f_p if there is no exact counterpart. A false negative f_n occurs if there is an ACR in the reference list for which no access permission is generated by BAcsTract. This classification is used to calculate the established metrics **CSLM I.1** precision $P = \frac{t_p}{t_p+f_p}$, to address H I.1

and **CSLM I.2** recall $R = \frac{t_p}{t_p+f_n}$, to address H I.2.

Question II proposes Hypothesis H II which claims that a part of the role engineering process can be automated. To answer question II the process during which BAcsTract is executed with the case study system is examined for human interventions. A human intervention i means that security experts have to conduct some task in order for BAcsTract to begin or continue its work. We exclude simple tasks like selecting the input models and pressing buttons. This classification is used to calculate: **CSLM II:** Number of Human Interventions $HI = \sum i$, to address H II.

Question III examines whether the engineering of the role model becomes more efficient, as the vast amount of business processes are analyzed automatically by BAcsTract. Hypothesis H III claims that the number of business processes that need to be processed manually by security experts is reduced. Therefore, CSML III counts the number of business process BP_A that are analyzed automatically.

Question IV examines whether changes in the role model resulting through changes in business processes can be computed automatically. Hypothesis H IV claims that BAcsTract can compute changes of the role model, resulting from evolution of business processes, automatically. Question IV is partly answered on the metamodel level. Therefore, two points need to be examined: a) does BAcsTract require any adaptation of input models after the evolution of business processes and b) are all transformation steps of BAcsTract to generate the new role model automatic. For b) we reuse metric CSLM II.

Question V examines the traceability of generated access permissions from BAcsTract. Therefore, the precision and recall of the generated ACR mapping model is examined. While hypothesis H V.1 claims that the traceability information is semantically correct, hypothesis H V.2 claims that it is complete. To answer question V each entry in the ACR mapping model is classified based on a reference list of ACR mappings for the given business processes. An entry is a true positive t_p if it has an exact counterpart (a tuple role, process, activity and permission) in the reference list. It is a false positive f_p if there is no exact counterpart. A false negative f_n occurs if there is an entry in the reference list for which BAcsTract did not generate an entry in the ACR mapping model. This classification is used to calculate the two established metrics **CSLM V.1** precision $P = \frac{t_p}{t_p+f_p}$, to address H V.1 and **CSLM V.2** recall $R = \frac{t_p}{t_p+f_n}$, to address H V.2.

4.2 Results & Discussion

We calculated metrics for the case study as described.

CSLM I.1 & CSLM I.2. We measured 81 true positives, zero false positives and false negatives for the accuracy of extracted access permissions. This brings us to a precision $P = \frac{81}{81+0} = 1.0$ and a recall $R = \frac{81}{81+0} = 1.0$. Results confirm H I.1 and H I.2. Tab. 3 shows an excerpt of the generated permissions.

Table 3: Excerpt of the generated role model.

Role	Permission
Store:Store Manager	Write Staff schedule
Store:Cashier	Read Cashier ID
Ent.:Ent. Manger	Read Inventory report

CSLM II. To identify whether parts of the role engineering process can be automated with BAcsTract we counted the number of human interventions which impose additional effort. Our baseline is that organizations have designed state of the art business processes. The 17 business processes are modeled according to the BPMN 2.0 standard. BAcsTract does not require any adaptation nor extension of the BPMN models. While preparing the input no additional human intervention is required. BAcsTract extracts the role model automatically according to the defined steps in Sect. 3. This brings us to a number of human interventions $HI = 0+0 = 0$. This confirms hypothesis H II that BAcsTract can extract the role model automatically and without any human intervention before and during the extraction process.

CSLM III. To measure the amount of artifacts which the security experts do not need to analyze manually, we counted the number of business process that are analyzed automatically by BAcsTract. It is $BP_A = 17$. The extracted role model comprises all ACRs from the business processes.

CSLM V.1 & CSLM V.2. Regarding the accuracy of the generated ACR mapping model we measured 112 true positives, zero false positives and zero false negatives. This brings us to a precision $P = \frac{112}{112+0} = 1.0$ and a recall $R = \frac{112}{112+0} = 1.0$ and confirms hypothesis H V.1 and H V.2. It means that BAcsTract generated a correct entry in the ACR mapping model for all business process ACRs.

MLM IV. Two points are examined: a) does BAcsTract require any adaptation of input models after the evolution of business processes and b) are all transformation steps of BAcsTract to generate the new role model automatic. During the evolution of business processes these business processes are modified within the scope of the BPMN standard. We do not count these modifications as they reflect the evolution scenario itself and are done anyway. As BAcsTract operates on plain BPMN models, no further modifications or extensions are required after the evolu-

tion. Thus, regarding a) no additional manual effort is required. Regarding b) results for CLSM II showed that BAcsTract extracts the role model from the input models automatically and without any significant human intervention. This confirms hypothesis H IV that changes in the role model resulting through changes of business processes can be computed automatically and without additional effort.

The case study confirms all hypotheses raised by the questions. Hence, goals are reached as follows.

Goal 1. Results for question I show that the extracted access permission have a high accuracy. BAcsTract identifies correctly all 81 unique access permission of the role model according to the scheme explained in Sect. 3. Results for question II and III demonstrate that parts of the role engineering process can be automated by using BAcsTract. In this case study BAcsTract analyzed 17 business processes with appropriate complexity on behalf of the security experts. Results for question II show that no modifications of input models are required in order for BAcsTract to work. Our research indicates that the vast amount of business processes an organization has can be analyzed automatically producing an initial role model that comprises business level ACRs of role-permission type. This relieves security experts from manually analyzing business processes making the role engineering process quicker and reducing complexity of the overall role engineering process.

Goal 2. Results for question I and II show that BAcsTract successfully extracts all 81 unique access permission of the role model correctly. Furthermore, the extraction process is fully automated and does not require any human intervention like the extension or modification of input models. In conjunction with the results for question III we show that the usage of BAcsTract reduces the amount of manual steps during the role engineering processes by analyzing the vast amount of business process on behalf of the security experts. This reduces human errors.

Goal 3. While the results for question I show that BAcsTract correctly extracts the access permissions for the role model, results for question V show that the ACR mapping model is built correctly and with high accuracy for all 112 ACRs of the business processes. Results for question IV show that BAcsTract can be utilized during evolutional changes of business processes without imposing additional effort as BAcsTract operates automatically and on de facto standard models. Due to these facts and the automation of BAcsTract the adaptation of the role model during the evolution of business processes becomes faster compared to the manual engineering by security experts. This enables the business level and IT level to better decide among various evolution scenarios, as their impact on the role model can be better understood. Furthermore, results for question V show that each generated element of the role model is traceable to its origin in the business processes. This automatic documentation of design decisions enables to understand why certain roles and permissions are inside the role model, what otherwise would not be easy to understand. Consequently, mutual dependencies between business processes and RBAC can be understood better due to traceability with the ACR mapping model.

4.3 Threats to Validity

We discuss the four aspects of validity for case study research based on (Runeson et al., 2012, pp. 71).

Construct validity is about the adequacy of taken measures. If possible, we used established metrics as precision and recall and provided a reasonable classification scheme. We explained how research questions and metrics are derived from evaluation goals and applied the GQM method (Basili et al., 1994).

Internal validity ensures that an expected influencing factor is not affected by other factors. We expect the input models, the BAcsTract algorithm and the result classifications to influence the results. We analyze the factor BAcsTract algorithm. Regarding the input models we relied on a community driven case study (Heinrich et al., 2016) of a realistic supermarket chain. It was not developed by the authors and thus, not tailored to our approach. Tab. 2 shows case study characteristics that undermine appropriateness of the case study size. (Pilipchuk et al., 2017) has shown that CoCoME is suitable for examining ACRs. For result classifications we provided and explained a classification scheme for each metric in Sect. 4.1. If possible, we relied on established metrics. Any reference lists were made manually by two postgraduates. Their versions were compared to avoid mistakes.

External validity is about generalizability of results. According to Runeson a general problem of case study results is that they cannot be generalized in a universal way as no statistically relevant sample has be drawn. Nevertheless, results can be generalized to cases with similar characteristics. The most relevant characteristic is the input language BPMN. BPMN is the de facto standard language for business processes. This makes the results at least meaningful for a broad amount of other cases. We already discussed the appropriateness of CoCoME for ACR research earlier.

Reliability ensures that results are not influenced by researchers. For conducting the evaluation was required: creating input models, running the analysis and classifying results. For input models we relied

on a community drive case study as mentioned earlier. We explained how the algorithm works in Sec. 3, which is fully automated. Hence, we could not influence results during these steps. For the last step, we explained our metrics and classifications in Sect. 4.1.

5 CONCLUSION

The business level increasingly focuses on IT security due to the rising threat of cybercrime and number of security and privacy laws. Incorporate correct and secure ACRs is challenging. There is a communication gap between the business level and IT level, giving potential for security breaches in access control.

This paper tries to overcome this gap by extracting business level ACRs from business processes to generate an initial role model for RBAC. A case study-based evaluation undermines that the proposed approach increases the efficiency of engineering the role model with an automated extraction of business level ACRs. Furthermore, this leads to a reduction human errors that, otherwise would lead to security breaches. This becomes especially crucial during evolution scenarios where the role model requires repetitive adaptations. In our future work, we will apply BAcsTract to a real-world case study to further assess its accuracy. Furthermore, we will extend BAcsTract to transfer the extracted business level ACRs to enterprise application architectures (EAA) to identify forbidden data flows in an early design phase and help the enterprise architect building a business level aligned EAA.

REFERENCES

Aerts, A. et al. (2004). Architectures in context: on the evolution of business and ICT platform architectures. *Information and Management*, pages 781–794.

Alpers, S. et al. (2018). Identifying needs for a holistic modelling approach to privacy aspects in enterprise software systems. In *the International Conference on Information Systems Security and Privacy*, pages 74–82.

Alpers, S. et al. (2019). The current state of the holistic privacy and security modelling approach in business process and software architecture modelling. *Information Systems Security and Privacy*, pages 109–124.

AXELOS (2011). ITIL. Accessed: February 25, 2019.

Basili, R. et al. (1994). The goal question metric approach. *Encyclopedia of Software Engineering*, 1.

Brucker, A. D. et al. (2012). SecureBPMN: Modeling and enforcing access control requirements in business processes. In *ACM symposium on access control models and technologies (SACMAT)*, pages 123–126.

Colantonio, A. et al. (2009). A formal framework to elicit roles with business meaning in rbac systems. In *ACM symposium on access control models and technologies (SACMAT)*, pages 85–94.

Coyne, E. J. (1996). Role engineering. In *Proceedings of the ACM Workshop on Role-based access control*.

Crook, R. et al. (2001). Modelling access policies using roles in requirements engineering. *Information and Software Technology*, 45:979–991.

European Union (2016). General data protection regulation.

Federal Republic of Germany (2015). IT Security Act.

Ferraiolo, D. et al. (2007). *Role-Based Access Control*. Artech House Publishers.

Fuchs, L. et al. (2007). Supporting compliant and secure user handling - a structured approach for in-house identity management. In *The Second International Conference on Availability, Reliability and Security (ARES'07)*, pages 374–384.

Fuchs, L. et al. (2008). Hydro hybrid development of roles. In *Information Systems Security*, pages 287–302.

Haight, J. (2019). Automated control systems do they reduce human error and incidents? *ASSE Professional Development Conference and Exposition*.

Heinrich, R. et al. (2016). The cocome platform for collaborative empirical research on information system evolution. Technical Report 2, Karlsruhe.

INCITS (2012). INCITS 359-2012 - role based access control standard.

ISO/IEC (2018). ISO 27000.

Lee, H. et al. (2004). A framework for modeling organization structure in role engineering. In *Applied Parallel Computing (PARA)*.

Mark, S. (2010). Scenario-driven role engineering. *IEEE Security and Privacy*.

Mitra, B. et al. (2016). A survey of role mining. *ACM Comput. Surv.*, 48(4):37.

Narouei, M. et al. (2015). Towards an automatic top-down role engineering approach using natural language processing techniques. In *ACM symposium on access control models and technologies*, pages 157–160.

O'Connor, A. et al. (2010). NIST economic analysis of role-based access control. Technical report.

OMG (2011). Business process model and notation v2.0.2.

Pilipchuk, R. (2018). Coping with access control requirements in the context of mutual dependencies between business and it. In *Proceedings of the Central European Cybersecurity Conference*, pages 16:1–16:4.

Pilipchuk, R. et al. (2017). Defining a security-oriented evolution scenario for the cocome case study. In *4nd Collaborative Workshop on Evolution and Maintenance of Long-Living Software Systems*, pages 60–77.

Ramadan, Q. et al. (2018). Integrating bpmn- and uml-based security engineering via model transformation. In *Proceedings of the SE 2018*, pages 63–64.

Roeckle, H. et al. (2000). Process-oriented approach for role-finding to implement role-based security administration in a large industrial organization. In *Workshop on Role-based access control*, pages 103–110.

Runeson, P. et al. (2012). *Case Study Research in Software Engineering: Guidelines*. John Wiley & Sons, Inc.

Wieringa, R. J. et al. (2003). *Aligning Application Architecture to the Business Context*, pages 209–225.