# Architectural Alignment of Access Control Requirements Extracted from Business Processes

Zur Erlangung des akademischen Grades eines

Doktors der Ingenieurwissenschaften

von der KIT-Fakultät für Informatik
des Karlsruher Instituts für Technologie (KIT)

genehmigte
Dissertation

von

Roman Pilipchuk

aus St. Petersburg

Tag der mündlichen Prüfung: 28. Oktober 2021
Erster Gutachter: Prof. Dr. Ralf H. Reussner
Zweiter Gutachter: Prof. Dr. Andreas Oberweis

# Abstract

Business processes and information systems evolve constantly over time and affect each other in non-trivial ways. Therefore, aligning security requirements between business processes and enterprise application architectures (EAAs) is a challenging task. This is especially true for access control requirements (ACRs) that are of great significance in IT security and privacy. The following three goals of the business level of an organization illustrate their importance:

1. Identifying and protecting critical assets and sensitive data.
2. Establishing appropriate organization-wide IT security and privacy strategies.
3. Complying with the rising amount of security and privacy laws.

ACRs are crucial to achieve these goals. However, the implementation of these goals requires knowledge to be transferred from the business level to the IT level. The size and complexity of organizations make a complete and correct implementation of ACRs a challenge for the IT level. The different terminologies within both domains additionally complicate this process. Furthermore, the size of organizations, the complexity of EAAs and the interconnection between EAAs and business processes affect the error rate during the design phase of ACRs and EAAs negatively. These interrelationships lead to misalignments between EAA, access permissions and business processes and they increase over time as adjustments are required due to evolutionary change of business processes and information systems.

Previous work relies heavily on extensions of business process and architecture modeling languages. This imposes serious effort for organizations to extend existing standard models and maintain these extensions over time. Other approaches rely on manual processes to solve the aforementioned problems. Such approaches require a lot of effort, do not scale, and are error-prone for complex systems.

The aim of this thesis is to research how ACRs can be aligned between the business level and IT level with minimal additional effort for organizations. Specifically, this thesis explores how business level ACRs can be extracted from business processes to be automatically transferred into access permissions for role-based access control (RBAC) and how the EAA can be analyzed for violations of these business level ACRs. These proposed approaches will assist security experts during the design of access permissions for RBAC and reduce complexity of this engineering process. They will also enable enterprise architects to inspect the EAA at design time for data flows that violate business level ACRs and help the enterprise architects in resolving the identified violations.

The main contributions of this thesis can be summarized as follows:

(I) An approach to automatically extract business level ACRs from business processes with a subsequent transformation into an initial role model for RBAC.

(II) An approach to automatically generate architectural data flow constraints from ACRs to identify data flows of services in EAAs that violate the ACRs.

(III) A high-level process for organization on how to use these approaches within different evolution scenarios.

(IV) A model for mapping relevant elements from business processes, RBAC, and EAAs with respect to ACRs together. This model is created automatically by the approaches and is used, among other things, to document design decisions, improve the mutual understanding of domain models and assist the enterprise architect in resolving errors within the EAA.

Within the scope of this thesis two case studies are conducted to validate the approaches and proposed contributions. The first case study is a real-world case study, resulting from a cooperation with a national art gallery that revises its information systems. Another case study is based on the Common Component Modeling Example (CoCoME). CoCoME is a case study of a realistic supermarket chain developed by the scientific community. It was developed to research software evolution and has several evolution scenarios extended by various research groups. Both case studies are suitable for researching ACRs as they are affected by substantial number of legal regulations pertaining IT security as well as data protection and have a multitude of sensitive data flows. For each case study a goal question metric (GQM) model is developed to systematically validate the contributions of this thesis. Therefore, validation goals are defined. Scientific questions are methodically derived from the validation goals. Afterwards, for each scientific question appropriate metrics are specified in order to investigate the scientific questions. The following aspects are examined throughout the case studies:

- Quality of generated access permissions.
- Quality of identified data flows in services of the EAA that violate ACRs.
- Completeness and correctness of the generated model for the traceability of ACRs across business and IT models.
- Applicability of approaches in evolution scenarios of business processes and EAAs.

At the end of this thesis I elaborate on the future work with regard to the approaches of this thesis. This encompasses how the model for mapping relevant elements from business processes, RBAC, and EAAs with respect to ACRs can be enriched with elements from other models of the business level and IT level, how the approaches of this thesis can benefit from additional input information and how other domain models can profit from the extracted information about business level ACRs.

# Zusammenfassung

Geschäftsprozesse und IT-Systeme sind einer ständigen Evolution unterworfen und beeinflussen sich in hohem Maße gegenseitig. Dies führt zu der Herausforderung, Sicherheitsaspekte innerhalb von Geschäftsprozessen und Enterprise Application Architectures (EAAs) in Einklang zu bringen. Im Besonderen gilt dies für Zugriffskontrollanforderungen, welche sowohl in der IT-Sicherheit als auch im Datenschutz einen hohen Stellenwert haben. Die folgenden drei Ziele der Geschäftsebene verdeutlichen die Bedeutung von Zugriffskontrollanforderungen:

1. Identifikation und Schutz von kritischen und schützenswerten Daten und Assets.
2. Einführung einer organisationsweiten IT-Sicherheit zum Schutz vor cyberkriminellen Attacken.
3. Einhaltung der zunehmenden Flut an Gesetzen, welche die IT-Sicherheit und den Datenschutz betreffen.

Alle drei Ziele sind in einem hohen Maß mit Zugriffskontrollanforderungen auf Seiten der Geschäftsebene verbunden. Aufgrund der Fülle und Komplexität stellt die vollständige und korrekte Umsetzung dieser Zugriffskontrollanforderungen eine Herausforderung für die IT dar. Hierfür muss das Wissen von der Geschäftsebene hin zur IT übertragen werden. Die unterschiedlichen Terminologien innerhalb der Fachdomänen erschweren diesen Prozess. Zusätzlich beeinflussen die Größe von Unternehmen, die Komplexität von EAAs sowie die Verflechtung zwischen EAAs und Geschäftsprozessen die Fehleranfälligkeit im Entwurfsprozess von Zugriffsberechtigungen und EAAs. Dieser Zusammenhang führt zu einer Diskrepanz zwischen ihnen und den Geschäftsprozessen und wird durch den Umstand der immer wiederkehrenden Anpassungen aufgrund von Evolutionen der Geschäftsprozesse und IT-Systeme verstärkt. Bisherige Arbeiten, die auf Erweiterungen von Modellierungssprachen setzen, fordern einen hohen Aufwand von Unternehmen, um vorhandene Modelle zu erweitern und die Erweiterungen zu pflegen. Andere Arbeiten setzen auf manuelle Prozesse. Diese erfordern viel Aufwand, skalieren nicht und sind bei komplexen Systemen fehleranfällig. Ziel meiner Arbeit ist es, zu untersuchen, wie Zugriffskontrollanforderungen zwischen der Geschäftsebene und der IT mit möglichst geringem Mehraufwand für Unternehmen angeglichen werden können. Im Speziellen erforsche ich, wie Zugriffskontrollanforderungen der Geschäftsebene, extrahiert aus Geschäftsprozessen, automatisiert in Zugriffsberechtigungen für Systeme der rollenbasierten Zugriffskontrolle (RBAC) überführt werden können und wie die EAA zur Entwurfszeit auf die Einhaltung der extrahierten Zugriffskontrollanforderungen überprüft werden kann. Hierdurch werden Sicherheitsexperten beim Entwerfen von Zugriffsberechtigungen für RBAC Systeme unterstützt und die Komplexität verringert. Weiterhin werden Enterprise-Architekten in die Lage versetzt, die EAA zur Entwurfszeit auf Datenflüsse von Services zu untersuchen, welche gegen die geschäftsseitige Zugriffskontrollanforderungen verstoßen und diese Fehler zu beheben.

Die Kernbeiträge meiner Arbeit lassen sich wie folgt zusammenfassen:

(I) Ein Ansatz zur automatisierten Extraktion von geschäftsseitigen Zugriffskontrollanforderungen aus Geschäftsprozessen mit anschließender Generierung eines initialen Rollenmodells für RBAC.

(II) Ein Ansatz zum automatisierten Erstellen von architekturellen Datenfluss-Bedingungen aus Zugriffskontrollanforderungen zur Identifikation von verbotenen Datenflüssen in Services von IT-Systemen der EAA.

(III) Eine Prozessmodell für Unternehmen über die Einsatzmöglichkeiten der Ansätze innerhalb verschiedener Evolutionsszenarien.

(IV) Ein Modell zur Verknüpfung relevanter Elemente aus Geschäftsprozessen, RBAC und EAAs im Hinblick auf die Zugriffskontrolle. Dieses wird automatisiert durch die Ansätze erstellt und dient unter anderem zur Dokumentation von Entwurfsentscheidungen, zur Verbesserung des Verständnisses von Modellen aus anderen Domänen und zur Unterstützung des Enterprise-Architekten bei der Auflösung von Fehlern innerhalb der EAA.

Die Anwendbarkeit der Ansätze wurden in zwei Fallstudien untersucht. Die erste Studie ist eine Real-Welt-Studie, entstanden durch eine Kooperation mit einer staatlichen Kunsthalle, welche ihre IT-Systeme überarbeitet. Eine weitere Fallstudie wurde auf Basis von Common Component Modeling Example (CoCoME) durchgeführt. CoCoME ist eine durch die Wissenschaftsgemeinde entwickelte Fallstudie einer realistischen Großmarkt-Handelskette, welche speziell für die Erforschung von Software-Modellierung entwickelt wurde und um Evolutinsszenarien ergänzt wurde. Aufgrund verschiedener gesetzlicher Regularien an die IT-Sicherheit und den Datenschutz sowie dem Fluss von sensiblen Daten eignen sich beide Fallstudien für die Untersuchung von Zugriffskontrollanforderungen. Beide Fallstudien wurden anhand der Goal Question Metric-Methode durchgeführt. Es wurden Validierungsziele definiert. Aus diesen wurden systematisch wissenschaftliche Fragen abgleitet, für welche anschließend Metriken aufgestellt wurden, um sie zu untersuchen. Die folgenden Aspekte wurden untersucht:

- Qualität der generierten Zugriffsberechtigungen.
- Qualität der Identifikation von fehlerhaften Datenflüssen in Services der EAA.
- Vollständigkeit und Korrektheit des generierten Modells zur Nachverfolgbarkeit von Zugriffskontrollanforderungen über Modelle hinweg.
- Eignung der Ansätze in Evolutionsszenarien von Geschäftsprozessen und EAAs.

Am Ende dieser Arbeit wird ein Ausblick gegeben, wie sich die vorgestellten Ansätze dieser Arbeit erweitern lassen. Dabei wird unter anderem darauf eingegangen, wie das Modell zur Verknüpfung relevanter Elemente aus Geschäftsprozessen, RBAC und EAAs im Hinblick auf die Zugriffskontrolle, um Elemente aus weiteren Modellen der IT und der Geschäftsebene, erweitert werden kann. Weiterhin wird erörtert wie die Ansätze der Arbeit mit zusätzlichen Eingabeinformationen angereichert werden können und wie die extrahierten Zugriffskontrollanforderungen in weiteren Domänenmodellen der IT und der Geschäftsebene eingesetzt werden können.

# Acknowledgements

On this page, I want to gratefully thank the numerous people who supported me over the last couple of years (some of them in the background) during my PhD research time at FZI Research Center for Information Technology and KIT Karlsruhe Institute of Technology and during the writing of this thesis.

First of all, I want to thank my supervisor Prof. Dr. Ralf Reussner for the warm welcome at his research group, for the guidance with insightful and intelligent comments throughout my research path and for keeping me always in mind and connected to the over PhD students in Karlsruhe, while I was living in Berlin. I am very grateful for your valuable guidance and support that helped me solving the large as well as the small problems that I encountered during my research. I appreciated our fruitful discussions in our video conference calls and I am deeply grateful to you for keeping me grounded the whole time. Our video conference calls always inspired me and gave me fresh energy to further deepen my research. Furthermore, this thesis was accompanied by several great professors and postdocs: Prof. Dr. Ralf Reussner, Prof. Dr. Andreas Oberweis, Dr. Robert Heinrich and Prof. Dr. Anne Koziolek. I want to thank Prof. Dr. Andreas Oberweis for being available from the beginning of my research for any of my questions and for the support and excitement regarding my research topic. I want to thank Dr. Robert Heinrich for the valuable support in my research, his constant availability for fruitful discussions and the patient support while working on papers. Prof. Dr. Anne Koziolek I want to thank for her helpful feedback at the research retreats and her openness for discussions.

During the research for my PhD, I worked with many great people at FZI and KIT and had the opportunity to find many new and dear friends. I want to thank all of you for the joint work and our collective achievements. I am very thankful for your thoughtfulness and warm welcomes that you gave me throughout all my visits and research retreats.

One of my greatest thanks belongs to my colleague Stephan Seifermann. Thank you for working so closely with me on our joint research topics and a lot of thanks for advising me in all of my organizational questions. I am very fortunate to have you as a friend and research colleague. I enjoyed working with you (and also in our joint sessions with Emre Taspolatoglu) and I had a lot of fun during our research retreats and business trips (especially the one in Hanover at the Software Engineering Conference in 2017). On the same note, I want to appreciate Emre Taspolatoglu for his friendly and enjoyable personality and for the constructive discussions on the various security topics. I really enjoyed and still enjoy working with you both. Moreover, I want to thank Dr. Sascha Alpers for our joint research. We had many productive and fascinating conversations while working together on our publications.

I want to express my deepest gratitude to Evelina Bobrow and Thorsten Lucke. I am very thankful for having you Evelina Bobrow and your epic support in all the linguistic questions and pitfalls you guided me through in the last years. You supported me tremendously and I learned a lot from you all that time. Much thanks to you, Thorsten Lucke, for supporting me in the formal description of my contributions. I enjoyed the intensive discussions we had at our meals. Furthermore, I got great support from my students Patrick Woche, Tobias Knopf and Stephan Nordheim. I had a good time with you three and I am happy for being the supervisor in your bachelor and master theses. Each of you contributed his piece to the puzzle of my research. Likewise, I want to acknowledge Georg Berns from Bayer who willingly supported me in understanding what Enterprise Application Architecture really means for the industry and also acknowledge Florian Trott and Philipp Stanehl from the national art gallery in Karlsruhe for our pleasant and joint work that allowed me to validate my approaches on a real-world case study.

I want to thank Rodger Burmeister, for being such a good office neighbor and colleague in all that time and supporting me with your wisdom on all possible questions regarding dissertational research and thousands of other topics. I learned a lot from you and I enjoyed it very much. Thank you. I also want to thank my division managers Luise Kranich, Jan Sürmeli and Jörg Henss in the continuous support of my research and in encouraging me to complete my thesis.

Just as importantly, I want to thank my parents and my deceased grandparents for supporting me at any time, in any point and with every decision of my life. You encouraged me at my work and gave me confidence to find my way and make my PhD. Thank you for your patience and personal guidance and in particular for having my back. At this point, I want to give a special and extraordinary big thanks to my wife Charlotta Pilipchuk who supported me morally on a daily basis and throughout this long journey of doing my PhD. You are incredible and I love you endlessly for your reassurance and for showing me what matters most in my life. Finally, a special thanks to Stanislaw Perogan, Turac Dursun, Jana Wollstein, Ksenya Pankova, Dennis Colman, Boris Reznikov, Arseni Bobrow and all the others who believed in me and extolled me for my personality and my being. You are all very precious to me.

# Contents

**Abstract** . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . **i**

**Zusammenfassung** . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . **iii**

**Acknowledgements** . . . . . . . . . . . . . . . . . . . . . . . . . . . . . **vii**

**List of Figures** . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . **xiii**

**List of Tables** . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . **xvii**

**1 Introduction** . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . **1**

1.1 Business Level Access Control Requirements and IT . . . . . . . . . . . . 1
1.2 Problem Statement . . . . . . . . . . . . . . . . . . . . . . . . . . . 5
1.3 Research Questions . . . . . . . . . . . . . . . . . . . . . . . . . . . 9
1.4 Approach . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 12
1.5 Scope and Assumptions . . . . . . . . . . . . . . . . . . . . . . . . . 15
1.6 Contributions . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 16
1.7 Outline . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 19
1.8 Previous Publications . . . . . . . . . . . . . . . . . . . . . . . . . . 21

**2 Foundations** . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . **23**

2.1 Business Process Model and Notation . . . . . . . . . . . . . . . . . . 23
2.1.1 Basic Model Elements . . . . . . . . . . . . . . . . . . . . . . . 24
2.1.2 Value and Usage . . . . . . . . . . . . . . . . . . . . . . . . . . 27
2.2 Role-Based Access Control . . . . . . . . . . . . . . . . . . . . . . . . 27
2.2.1 Role-Based Access Control Concept . . . . . . . . . . . . . . . . 28
2.2.2 Role Engineering . . . . . . . . . . . . . . . . . . . . . . . . . . 30
2.2.3 Role Mining . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 31
2.2.4 Value and Adoption . . . . . . . . . . . . . . . . . . . . . . . . 32
2.2.5 Summary . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 32
2.3 Enterprise Application Architecture . . . . . . . . . . . . . . . . . . . 33
2.4 Palladio Component Model . . . . . . . . . . . . . . . . . . . . . . . . 34
2.4.1 Development Roles . . . . . . . . . . . . . . . . . . . . . . . . . 34
2.4.2 PCM Metamodel . . . . . . . . . . . . . . . . . . . . . . . . . . 35
2.4.3 Data Flow Extension Data Centric Palladio . . . . . . . . . . . . 40
2.4.4 Summary . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 43
2.5 Integrated Business IT Impact Simulation . . . . . . . . . . . . . . . . 43

**3 Approach** . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . **47**
3.1 Concept . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 47
3.1.1 Concept for Role Model Extraction from Business Processes . . . 48
3.1.2 Concept for Identification of Access Control Requirement Breaches in Enterprise Application Architectures . . . . . . . . . . . . . 59
3.2 Refinement for BPMN and PCM . . . . . . . . . . . . . . . . . . . 63
3.2.1 Knowledge Base . . . . . . . . . . . . . . . . . . . . . . . . 64
3.2.2 Model Overview, Responsibilities and Assumptions . . . . . . . . 74
3.2.3 Role Model Extraction from Business Processes . . . . . . . . . . 78
3.2.4 Identification of Access Control Requirement Breaches in Enterprise Application Architectures . . . . . . . . . . . . . . . . . 108
3.3 Discussion of BPMN Access Permission Extractor and Palladio Access Permission Extractor . . . . . . . . . . . . . . . . . . . . . . . . . 121
3.3.1 Discussion of Problem Statements and Contributions . . . . . . . 121
3.3.2 Discussion of Limitations . . . . . . . . . . . . . . . . . . . . 124
3.4 Discussion of Access Permission Architecture Aligner . . . . . . . . . . 125
3.4.1 Discussion of Problem Statements and Contributions . . . . . . . 125
3.4.2 Discussion of Limitations . . . . . . . . . . . . . . . . . . . . 127

**4 Process for Utilizing Approaches in Organizations** . . . . . . . . . . . . . . . . **131**
4.1 Process Typical in Organizations . . . . . . . . . . . . . . . . . . . 131
4.2 Process when Utilizing Approaches . . . . . . . . . . . . . . . . . . 132
4.2.1 Problem Statement . . . . . . . . . . . . . . . . . . . . . . . 132
4.2.2 High-Level Process . . . . . . . . . . . . . . . . . . . . . . . 133
4.3 Discussion of the Process when Utilizing Approaches . . . . . . . . . . 142

**5 Validation** . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . **147**
5.1 Case Study I . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 149
5.1.1 Validation Goals and Research Questions . . . . . . . . . . . . . 149
5.1.2 Case Study System . . . . . . . . . . . . . . . . . . . . . . . 156
5.1.3 Results and Discussion . . . . . . . . . . . . . . . . . . . . . 159
5.1.4 Threats to Validity . . . . . . . . . . . . . . . . . . . . . . . 166
5.1.5 Summary . . . . . . . . . . . . . . . . . . . . . . . . . . . . 167
5.2 Case Study II . . . . . . . . . . . . . . . . . . . . . . . . . . . . 168
5.2.1 Validation Goals and Research Questions . . . . . . . . . . . . . 168
5.2.2 Case Study System . . . . . . . . . . . . . . . . . . . . . . . 175
5.2.3 Results and Discussion . . . . . . . . . . . . . . . . . . . . . 177
5.2.4 Threats to Validity . . . . . . . . . . . . . . . . . . . . . . . 184
5.2.5 Summary . . . . . . . . . . . . . . . . . . . . . . . . . . . . 186

**6 Related Work** . . . . . . . . . . . . . . . . . . . . . . . . . . . . . **189**
6.1 Security Extensions for Business Process and Architecture Languages . . 189
6.1.1 Business Process Security Extensions . . . . . . . . . . . . . . . 190
6.1.2 Architecture Security Extensions . . . . . . . . . . . . . . . . . 193
6.1.3 Transformation on Language Extensions . . . . . . . . . . . . . 198

6.2 Access Control . . . . . 202
6.2.1 Role Engineering Approaches . . . . . 202
6.2.2 Role Mining Approaches . . . . . 206
6.2.3 Role Mining for Hierarchy Optimization . . . . . 207
6.2.4 Hybrid Approaches . . . . . 208
6.2.5 Hybrid RBAC Concepts . . . . . 210
6.3 IT Architecture Approaches . . . . . 213
6.3.1 Architecture Alignment Approaches . . . . . 213
6.3.2 Security Analysis on Architecture Models . . . . . 217

**7 Conclusion . . . . . 221**
7.1 Summary . . . . . 221
7.2 Benefits . . . . . 225
7.3 Assumptions and Limitations . . . . . 229
7.3.1 Assumptions and Limitations of BAcsTract and PAcsTract . . . . 230
7.3.2 Assumptions and Limitations of AcsALign . . . . . 231
7.4 Future Work . . . . . 233

**Bibliography . . . . . 237**

# List of Figures

2.1 The flow objects of BPMN. . . . . . . . . . . . . . . . . . . . . . . . . . . 24

2.2 The connecting objects of BPMN. . . . . . . . . . . . . . . . . . . . . . . . 25

2.3 The swimlanes of BPMN. . . . . . . . . . . . . . . . . . . . . . . . . . . . 25

2.4 The artifacts of BPMN. . . . . . . . . . . . . . . . . . . . . . . . . . . . . 26

2.5 Shows the business process *Prepare Advertisements and Discounts* of a supermarket store. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 26

2.6 The reference model of the core RBAC [2]. . . . . . . . . . . . . . . . . . . 29

2.7 The reference model of the hierarchical RBAC [2]. . . . . . . . . . . . . . . 29

2.8 Shows the metamodel for the data type of PCM [189]. . . . . . . . . . . . . 36

2.9 Shows the metamodel for the interface of PCM [189]. . . . . . . . . . . . . 37

2.10 Shows the metamodel for the component and system of PCM [189]. . . . . . 38

2.11 Shows an exemplarily excerpt of a repository in PCM. . . . . . . . . . . . . 39

2.12 Shows an example of a SEFF in PCM. . . . . . . . . . . . . . . . . . . . . 40

2.13 Shows the metamodel for assembly models in PCM [189]. . . . . . . . . . . 41

2.14 Shows an example of an assembly model in PCM. . . . . . . . . . . . . . . 41

2.15 Shows an example for a characterized SEFF in PCM. . . . . . . . . . . . . 42

2.16 Shows an excerpt of the metamodel of IntBIIS regarding the integration of business processes into PCM [105]. . . . . . . . . . . . . . . . . . . . . . . 44

2.17 Shows the metamodel of IntBIIS regarding the integration of actor resources into PCM [105]. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 45

2.18 Shows the metamodel of IntBIIS regarding the integration of data objects into PCM [105]. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 45

3.1 Metamodel of RBAC. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 49

3.2 Simplified metamodel of business processes that is inspired by BPMN [5]. . . 53

3.3 Simplified metamodel of an enterprise application architecture inspired by the UML metamodel [6]. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 60

3.4 Extended metamodel of IntBIIS. The blue highlighted arrows and black highlighted classes are introduced by IntBIIS_LP. Other elements are part of the IntBIIS metamodel. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 70

3.5 An example for the model that interconnects *EntryLevelSystemCalls*, *AcquireDeviceResourceActions* and *ReleaseDeviceResourceActions* with *ActorSteps*. 71

3.6 Extended metamodel of IntBIIS. The blue highlighted arrows and black highlighted classes are introduced by IntBIIS_LP. Other elements are part of the IntBIIS metamodel. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 72

3.7 An example of the model that extends the organizational unit for roles. . . . 73

3.8 Illustrates the extensions provided by IntBIIS_LP. . . . . . . . . . . . . . . 73

3.9 Shows the ACR mapping model filled by BAcsTract and PAcsTract during the extraction of ACRs from business processes. . . . . . . . . . . . . . . . . . . 79
3.10 Illustrates an example of an entry in the ACR mapping model. . . . . . . . . 81
3.11 Shows the business process *Prepare Advertisements and Discounts* of a supermarket store. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 83
3.12 Explains how pools distinguish organizational divisions. . . . . . . . . . . . 84
3.13 Shows an example of a closed lane in BPMN. . . . . . . . . . . . . . . . . . . 84
3.14 Shows different ways to model associations of data objects in BPMN. . . . . 86
3.15 Shows a simple hierarchy. . . . . . . . . . . . . . . . . . . . . . . . . . . . . 89
3.16 Shows an excerpt of the HTML output of the ACR mapping model. . . . . . 90
3.17 Shows an excerpt of the HTML output of the role model. . . . . . . . . . . . 91
3.18 Shows a HTML output of the unique permissions. . . . . . . . . . . . . . . . 91
3.19 Shows a HTML output of the processes and roles. . . . . . . . . . . . . . . . 91
3.20 Shows the system diagram. . . . . . . . . . . . . . . . . . . . . . . . . . . . 93
3.21 Shows the IntBIIS_LP process Prepare Advertisements and Discounts of the supermarket store CoCoME. . . . . . . . . . . . . . . . . . . . . . . . . . . 93
3.22 Shows the organization environment model. . . . . . . . . . . . . . . . . . . 94
3.23 Shows the *Pool Ext Container* that defines the organizational units for roles of a process. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 94
3.24 Shows the data model. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 94
3.25 Shows the IntBIIS_LP process Prepare Advertisements and Discounts of a supermarket store and the properties for the first actor step. . . . . . . . . 95
3.26 Shows the data model with properties. . . . . . . . . . . . . . . . . . . . . . 98
3.27 Shows an excerpt of the HTML output of the ACR mapping model. . . . . . 101
3.28 Shows an excerpt of the HTML output of the role model. . . . . . . . . . . . 101
3.29 Shows a HTML output of the unique permissions. . . . . . . . . . . . . . . . 102
3.30 Shows a HTML output of the processes and roles. . . . . . . . . . . . . . . . 102
3.31 Shows an IntBIIS_LP process of a supermarket store and the properties for the first acquire device resource action. . . . . . . . . . . . . . . . . . . . . 103
3.32 Shows an excerpt of the HTML output about the overview of roles and their acquire and release device resource actions. . . . . . . . . . . . . . . . . . . 105
3.33 Shows a HTML output of the role model with ACRs for acquire device resource actions. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 105
3.34 Shows a HTML output of the acquire device resource actions and their surrounding actor steps and *EntryLevelSystemCalls*. . . . . . . . . . . . . . . . 107
3.35 Shows the ACR mapping model extended with EAA elements by AcsALign. 109
3.36 Illustrates an example of an extended entry in the ACR mapping model. . . . 109
3.37 Shows the system diagram of the supermarket enterprise CoCoME. . . . . . 111
3.38 Shows the components of the system *Store*. . . . . . . . . . . . . . . . . . . 112
3.39 Shows the IntBIIS_LP process *Prepare Advertisements and Discounts* of the supermarket enterprise CoCoME. . . . . . . . . . . . . . . . . . . . . . . . 113
3.40 Shows the *ELSC Match AS Ext Container* that interconnects actor steps with their EntryLevelSystemCalls for the business process *Prepare Advertisements and Discounts*. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 114
3.41 Shows the data model of the supermarket enterprise CoCoME. . . . . . . . . 115

3.42 Shows an excerpt of the HTML output of the results from AcsALign for the running example. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 120

4.1 Shows the high-level process for utilizing the approaches of this thesis. . . . 134
4.2 Shows the business process *Prepare Advertisements and Discounts* of the CoCoME supermarket store. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 135
4.3 Shows the business process *Solve Customer Problem* of the CoCoME supermarket store. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 136
4.4 Shows the business process *Solve Customer Problem* of the CoCoME supermarket store. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 136
4.5 Simplified enterprise application architecture of the CoCoME supermarket [178]. 137

5.1 Shows the overall GQM model for both case studies. . . . . . . . . . . . . . . 148
5.2 The GQM model for the case study regarding the extraction of a role model from business processes [177]. . . . . . . . . . . . . . . . . . . . . . . . . . . 151
5.3 Simplified architecture of the CoCoME supermarket chain. . . . . . . . . . . 157
5.4 Overview of sub-processes for the CoCoME supermarket chain. . . . . . . . 157
5.5 The GQM model for the identification of ACR breaches in EAAs. . . . . . . . 169
5.6 Excerpt of the EAA from the national art gallery. . . . . . . . . . . . . . . . . . 175
5.7 Overview of sub-processes for the exhibition preparation. . . . . . . . . . . . 176

# List of Tables

3.1 Comparison of elements in IntBIIS required for the concept of role model extraction. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 66
3.2 Comparative analysis of elements in BPMN and IntBIIS. . . . . . . . . . . . 67
3.3 Elements that need to be extended in IntBIIS. . . . . . . . . . . . . . . . . . . 69
3.4 Business process model input for BAcsTract. . . . . . . . . . . . . . . . . . . 75
3.5 Business process models and EAA models that are input for PAcsTract and AcsALign. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 76
3.6 Shows the model responsibilities in case of PAcsTract and AcsALign. . . . . 77
3.7 Comparative elements of BPMN and IntBIIS_LP. . . . . . . . . . . . . . . . 80
3.8 ACR mapping model after step one. . . . . . . . . . . . . . . . . . . . . . . . . 83
3.9 ACR mapping model after step two. . . . . . . . . . . . . . . . . . . . . . . . . 84
3.10 Excerpt of the ACR mapping model after step three. . . . . . . . . . . . . . . 85
3.11 Excerpt of the ACR mapping model after step four. . . . . . . . . . . . . . . . 87
3.12 Role model for the excerpt of the ACR mapping model from Table 3.11. . . . 88
3.13 Excerpt of the ACR mapping model after step one. . . . . . . . . . . . . . . . 96
3.14 ACR mapping model after step two. . . . . . . . . . . . . . . . . . . . . . . . . 96
3.15 Excerpt of the ACR mapping model after step three. . . . . . . . . . . . . . . 97
3.16 Excerpt of the ACR mapping model after step four. . . . . . . . . . . . . . . . 99
3.17 Role model for the excerpt of the ACR mapping model from Table 3.16. . . . 100
3.18 Excerpt of the ACR mapping model with acquire device resource actions. . . 104
3.19 Data input for AcsALign. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 110
3.20 Excerpt of the ACR mapping model for the business process *Prepare Advertisements and Discounts*. . . . . . . . . . . . . . . . . . . . . . . . . . . . 114
3.21 Excerpt of the extended ACR mapping model for the business process *Prepare Advertisements and Discounts*. . . . . . . . . . . . . . . . . . . . . . . . . 115
3.22 Excerpt of the generated data flow constraints for the business process *Prepare Advertisements and Discounts*. . . . . . . . . . . . . . . . . . . . . . . . . 116
3.23 Excerpt of the read and written data types for the invoked EntryLevelSystemCalls of the business process *Prepare Advertisements and Discounts*. . . . . . 116
3.24 Results of the AcsALign comparison algorithm for detecting ACR breaches. . 118

4.1 Process with and without the approaches of this thesis. . . . . . . . . . . . . 132
4.2 Extracted role model from the business process in Figure 4.2 and Figure 4.4. . 138

5.1 Characteristics of the business processes of the CoCoME supermarket chain [177]. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 158
5.2 Summarizes results for the CSLMs. . . . . . . . . . . . . . . . . . . . . . . . . 159

5.3 Excerpt of the generated role model [177]. . . . . . . . . . . . . . . . . . . . . 160
5.4 Characteristics of the business processes and the EAA of the case study [179]. 177
5.5 Summarized results for the CSLMs. . . . . . . . . . . . . . . . . . . . . . . . 178
5.6 Results for the identification of ACR breaches with AcsALign (MT means mistake type and mistakes are set in italic) [179]. . . . . . . . . . . . . . . . . . 178
5.7 Excerpt of the generated role model. . . . . . . . . . . . . . . . . . . . . . . . 183

# 1 Introduction

This thesis introduces an approach to align access control requirements (ACRs) from business processes with role-based access control (RBAC) and enterprise application architectures (EAA). The approach reduces needed time, costs and errors of security experts during the role engineering process. It enables enterprise architects to check whether the designed EAA is in line with business level ACRs and allows service design managers to understand how access control policies, systems and service calls are coupled with the business processes. These benefits become crucial especially in evolution scenarios of business processes, RBAC and EAAs. Section 1.1 motivates why aligning business processes with RBAC and EAA is essential from the business point of view of an organization. Section 1.2 points out the current problems with regard to the alignment. Afterwards, Section 1.3 presents the scientific research questions that are derived from the problems. Section 1.4 summarizes the approach of this thesis and Section 1.5 explains the scope and assumptions. Then Section 1.6 lays down the corresponding contributions of this thesis. Finally, Section 1.7 outlines the structure of this thesis and Section 1.8 concludes by presenting the parts of the thesis that were published in scientific publications.

## 1.1 Business Level Access Control Requirements and IT

The world is in an age when data has become the most valuable asset for organizations [70]. Unfortunately, that data needs to be protected as the interconnected digital world paves the way for cybercriminal attacks. Its protection depends on an appropriate management as well as the confidentiality of data. Nonetheless, the problem of cybercrime becomes increasingly alarming and the amount of obligatory IT security and privacy laws is growing [1]. The costs that organizations lose due to cybercriminal incidents is growing year by year [12]. One major reason for that is that criminals have begun to organize themselves in big and professional groups [116]. Thus, adhering to security and privacy requirements given by corporate risk management, law and customers is becoming more essential for organizations of all kinds. Both, IT security and privacy are non-functional requirements affecting business processes as well as IT systems. Access control is a fundamental building block of IT security and privacy that has to be implemented accurately. Defining and enforcing ACRs is a challenging task. To cope with the rising amount of functional and non-functional requirements stemming from various stakeholders, organizations model business processes and EAAs.

On that account, the *business level* of an organization has several more goals to focus on [25, 176]. Within the scope of this thesis, *business level* refers to service design managers and compliance managers which are defined by the Information Technology Infrastructure Library (ITIL) [34]. The **first goal** is to identify critical business assets, such as sensitive data and business secrets, classify them according to their protection needs and establish appropriate organizational rules and standards to protect them. As only the business level knows which assets are critical for the organization, they are in the sole position to define appropriate IT security and privacy requirements. Defining and enforcing these requirements is a challenging task [230].

To prevent cybercriminal attacks and resulting reputational damage, both leading to a loss of monetary income, the business level has a **second goal**: establishing organization-wide IT security and privacy strategies to enforce requirements. Many organizations operate worldwide having branches and subsidiaries all over the world, that need to be protected. This complex management task involves numerous departments, thousands of employees and dozens of heterogeneous IT landscapes with various processes and architectural models. There are guidelines like the ISO/IEC 27000-series [210] or the IT Baseline Protection [117] from the German Federal Office for Security in Information Technology (BSI) describing how to establish, manage and maintain information security effectively in organizations. Among others, these guidelines describe ACRs from the business level perspective. However, not only technical guidelines exist describing how and where to establish security in an organization. Business process guidelines like ITIL [34] and COBIT [32] that comprise sets of practices for IT service management have also dedicated parts for governance and management of IT security and especially access control. ITIL and COBIT focus on the alignment of IT services with business needs by describing necessary services and their interactions. Best practice business processes, tasks and checklists are provided to help describing the services in a way that organizations can properly integrate these services.

Establishing organization-wide IT security and privacy is not enough. The business level is increasingly stressed by governments to comply with a rising amount of laws regulating IT security and data privacy. This is a challenging task, especially when operating in different countries all over the world. Organizations have to adhere to each country's laws. Thus, IT compliance is the **third goal** of the business level. Laws pertaining IT security exist for various sectors. The Basel Accords [36] and the Minimum Requirements for Risk Management (MaRisk) [88] regulate bank capital adequacy, risk management and market liquidity in the finance sector. For the sector of critical infrastructures, Germany enacted, for example, the IT Security Act [89]. It seeks to establish state-of-the-art security measures and an uniformly way to report security incidents to the national agency. This affects facilities for electricity generation, gas/oil production, telecommunication, water supply, agriculture, heating, public health, transportation systems, financial services and security services, as they all belong to critical infrastructures. Another example from the United States of America is the Health Insurance Portability and Accountability Act (HIPAA) [103], that regulates the healthcare information flow through different healthcare organizations. The General Data Protection Regulation (GDPR) [222] is another example. It is a regulation of the European Union (EU) to strengthen the data protection of individuals in EU countries. It prescribes security and privacy requirements and thus, ACRs for the collection, processing and use of personal data in any organization. If an organization violates a law it is prosecuted. To avoid prosecution by the state and severe monetary penalties, for example, up to four percent of the organization's worldwide turnover according to the GDPR, it becomes important for the business level to establish organization-wide IT security and privacy strategies that comply with the laws. To make matters worse the aforementioned laws are only a small part of the progressively growing amount of regulations that pertain IT security and privacy. In [180] a security review of a supermarket information system exemplifies the numerous security and privacy requirements stemming from various laws.

IT security and privacy defined at the business level have several advantages. The business level knows best which assets are critical for the organization. They have a holistic view of all organizational locations as well as communications and information flows between all departments and branches. This holistic view comprises also knowledge about communications with third parties. They are critical for the organization, as they provide weak spots for attacks and information leakage. Furthermore, the business level is the responsible organizational part to identify and incorporate regulations from laws, obviously with the support of the legal department. Due to these facts, the business level is most suited to define appropriate security and privacy requirements.

For the business level to fulfill the three aforementioned goals of a) defining critical assets and their protection degree, b) establishing organization-wide IT security and privacy strategies and c) enforcing IT compliance, support from the IT level is required. Within the scope of this thesis, IT level refers to enterprise architects and security experts. They are responsible to implement IT security and privacy appropriately. As access control is a fundamental building block of both, IT security and privacy, this thesis focuses on access control requirements (ACRs). All of the above-mentioned goals demand appropriate establishment and enforcement of access control for organizational data because IT security as well as privacy prescribe various access control restrictions.

While only the business level knows which assets need to be protected, only the IT level is able to realize technical solutions. For example, access control is a part of information systems and thus, implemented by the IT level. Several problems, on which I elaborate in [25], such as different terminology, domain knowledge, domain-specific models and modeling tools of the IT level and business level, widen a communication gap that may lead to errors and security breaches. For example, the business level holds information in laws and business processes, while the IT level works with EAAs and use cases. Gartner, an American research and advisory firm for information technology insights, identifies several enterprise architecture pitfalls [115] undermining the gap between the business level and the IT level. Among others, they identify the following enterprise architecture pitfalls: doing only technical domain-level architecture, insufficient stakeholder understanding, not spending enough time for communications and not engaging businesspersons. The authors conclude that there is a demand for a holistic modeling approach that includes business and IT. I elaborate on the demand for a holistic modeling approach for IT security and privacy in [24] and [25]. Gartner also identifies the complexity in understanding correct requirements due to different knowledge and models of business and IT and states this as a severe challenge of the business level [115]. These facts underline the existence of a communication gap that may lead to serious security breaches, as one error is enough to undermine the whole organizational protection.

While an alignment between business and IT models is beneficial [90], it is hard to realize due to various reasons [140, 230]. As a consequence, the business level and IT level are not well aligned [37, 231] undermining the above-mentioned goals of the business level. Additionally, the fact that organizations are evolving is often neglected making the matter worse. Business processes, EAAs, systems, and requirements evolve steadily over time and each has its own lifecycle. So far, these complex interrelations are not completely understood and are not adequately researched. Consequently, the IT and business level affect each other in non-trivial ways [9]. For example, the business level defines requirements implicitly in business processes, which are not part of the IT terminology. However, the enterprise architect designs the EAA based on the information from the business processes and this is done continuously as the whole organization evolves over time. As a result, the different models are not well aligned with each other, representing not the requested outcome [24, 25]. Another consequence is that decisions concerning the IT level cannot be made reliably, since important business level information might not be considered [9]. As I described in [25] there are security analysis approaches for business models and also for IT models. However, organizations face the problem to model IT security and privacy holistically across models of business and IT and organizations have an increasing need to design non-functional requirements uniformly to reduce design flaws and human errors during the implementation of requirements. So far, there is no approach that solves this problem holistically satisfying the three mentioned goals of the business level [25]. Hence, there is a need to align business and IT models with regard to ACRs especially during evolution scenarios.

Eliminating potential software faults and mistakes in an early development lifecycle is important as it helps cutting down development costs significantly [43]. The later a fault is identified the more complex and cost-intensive it is to repair it [40, 123]. A mistake that is introduced during the requirements phase costs five times more to correct them during the design phase, ten times more than during the code phase and 368 times more than during the operation phase. Identifying mistakes in an early design phase is essential as otherwise they lead to security breaches and information leakage undermining the aspired goals of the business level. Besides the potential for security breaches, the costs to repair a mistake is a key business driver for organizations to identify and prevent mistakes early on during the realization of business requirements.

In order to align the models of the business level and the IT level with regard to ACRs and identify mistakes and misalignments in an early design phase, it is necessary to understand the mutual dependencies between the core models of the business level and IT level. On the one hand, there is the lifecycle of business processes and on the other hand, the lifecycle of EAAs and access control policies. Changes in one of them may need adjustments in the others [9, 25]. For the purpose of supporting enterprise architects and security experts to incorporate ACRs of the business level correctly, it is beneficial to:

a) help security experts to transfer ACRs from the business level correctly into access control policies of the access control system.

b) help enterprise architects to align the EAA with ACRs from the business level.

In this thesis, I propose an approach to automatically extract business level ACRs from the business level artifact, namely business processes and transfer them to the design phase of the IT level with the intention to tackle the above-mentioned problems (see also problem statements in Section 1.2). In essence, the extracted ACRs are used to:

a) form an initial role model with access control policies for RBAC.

b) transform them into architectural data flow constraints in order to identify ACR breaches in the EAA.

The approaches to extract ACRs from business processes is called BPMN Access Permission Extractor (BAcsTract) and Palladio Access Permission Extractor (PAcsTract). While BAcsTract extracts ACRs from the de facto standard business process language Business Process Model and Notation (BPMN) [5], PAcsTract extracts ACRs from IntBIIS_LP, a business process language in the Palladio Component Model (PCM) [189]. On top of PAcsTract a transformation algorithm called Access Permission Architecture Aligner (AcsALign) transforms the extracted ACRs into architectural data flow constraints to identify ACR breaches in the EAA.

As mentioned earlier mistakes in the RBAC role model and the EAA happen due to misalignments during the design phase and especially during evolution scenarios. While there are security analysis approaches on the business level side or on the IT level side, the approaches presented in this thesis specifically help to identify and resolve misalignments across models on both sides. Thus, the approaches facilitate an alignment of ACRs between the core models of the business level and the IT level.

## 1.2 Problem Statement

To achieve the three goals of the business level a) Identifying assets to be protected, b) an organization-wide IT security and privacy strategy, and c) compliance with laws, described in the beginning of Chapter 1, the need arises for:

- an appropriate and compliant establishment of access control policies in the access control system.
- an alignment between the EAA and the ACRs from the business level.

There are several key problems concerning this need, which are addressed as part of this thesis.

**P1 Missing knowledge on IT level:** Enterprise architects and security experts have typically not enough knowledge about which business assets are critical and the required protection degree [25]. The overall view of communication to third parties and between organizational departments is missing. Thus, the IT level is not able to define appropriate ACRs. This means that essential knowledge about which systems are allowed to access which assets and how to design the access control policies is missing on the IT level. Only the holistic view of the business level has the required knowledge. This problem corresponds to one of the aspired goals of the business level and is described in more detail in Section 1.1.

**P2 Different terminology between business and IT level:** Typically, the business level is concerned with regulations and business goals. Their way to express their knowledge is, for example, in business processes. This thesis focuses on the Business Process Model and Notation (BPMN) [5] as a representative language for business processes. While enterprise architects design EAAs with prominently the Unified Modeling Language (UML) [6, 212], security experts configure their access control systems for instance with access control policies. However, different terminology, domain knowledge, domain-specific models and modeling tools of the business and IT level widen a communication gap [115]. This may lead to errors and security breaches. I elaborate on this in Section 1.1 and in more detail in [25]. However, the business level is in the need of expressing ACRs in a way that the IT level properly understands them because one mistake may undermine the whole security of the organization. Due to these problems, EAAs and access control systems are not well aligned with laws and business level ACRs [37, 231]. This is especially true for IT security and privacy requirements [25]. The need of organizations to cope with these problems is growing, such as shown in my systematic literature review [24]. Consequently, decisions of the IT level cannot be made reliably, since important business level information may not be considered [9]. All of these reasons may lead to serious security breaches undermining the aspired goals of the business level.

**P3 Experts needed to understand business level:** A direct consequence from problem P2 is the need for experts who understand the terminology and models of both, the business level and IT level. This requirement slows down the overall design process and makes it more expensive. Especially in evolution scenarios, where the whole process of aligning artifacts has to be repeated constantly the reliability on the expert's skills is a factor that brings in an error rate, as the experts have to work through a vast amount of artifacts (see Section 2.2). There is no possibility to automatically check whether the produced results from the experts are correct.

With regard to access control systems, organizations, especially bigger organizations, are interested in role-based access control (RBAC) [2], as it has its advantages [8]. An example is the medical industry where RBAC is common and widely deployed [65, 211]. Compared to other access control systems it is beneficial in the management and expression of access control policies as well as in the provided security degree [8, 7]. The greater the number of employees in an organization the more RBAC simplifies complexity in managing permission assignment and individual user permissions compared to other access control concepts. In 2010 a NIST economic report estimated that RBAC research has saved industry 1.1 billion dollar over multiple years [7]. Their conclusion was that more than 80 percent of the analyzed organizations with an employee size more than 500 realized a better security strategy through the usage of roles and reduced administrative costs. These are some reasons why RBAC is a widely used concept [7, 65, 91]. Its model consists of roles containing the actual permissions. An additional hierarchy reduces the amount of duplicate permissions and eases the management and assignment of employees to permissions. The user is not assigned to permissions directly, but rather to roles. A role can express a job of an employee, from a business point of view, for example, *manager* and comprises all permissions needed by the manager. The resulting benefit for the human resources department is a comprehensible assignment between permissions and employees that makes the assignment process less error-prone, as the roles reflect the jobs of employees [8]. In addition, the complexity in handling evolution of permissions in large companies is reduced. Other benefits arise from the integrated separation of duties concept, which allows restricting the power of individual employees [8]. This increases the protection against internal fraud. As stated above, organizations have a need for a compliant incorporation of ACRs into RBAC, but RBAC has its own challenges. More information on RBAC is provided in Section 2.2.

**P4 Costly and error-prone engineering of the RBAC role model:** Due to the complexity of RBAC systems, many organizations fear the change from their legacy access control systems to RBAC, even though it brings a higher security degree [8]. Establishing RBAC is both, costly [151] and error-prone [8, 151, 91]. Experts have to manually engineer a role model containing roles, permissions and a hierarchy matching the needs of the organization. This is a complex task. The challenge is to elicit appropriate roles, their permissions and hierarchy matching the ACRs of the business level [91]. These circumstances make the overall engineering process slow, as the experts have to work through a vast amount of documents to understand the ACRs of the organization [8]. Business processes, process documentations and organizational charts are evaluated to develop an appropriate role model. Depending on the size of the organization, business level artifacts like business processes grow easily into hundreds, resulting in a vast amount of complex and interrelated artifacts demanding a specific business knowledge to understand them. This is a complicated and tedious task, which may end with errors inside the role model [8, 35]. However, each error is a potential security threat to the organization, as it may result in vulnerabilities and data leakage. While organizations continuously evolve, ACRs change and demand adjustments. This increases the problem of errors due to the manual adaptation process and increases the overall costs for RBAC.

**P5 Missing alignment between RBAC and business level access control requirements:** After security experts develop the role model (roles, their permissions and hierarchy) by manually going through the vast amount of business processes, neither the security experts nor the business level can check if the resulting role model reflects the business level ACRs correctly. The whole process of building the role model is manual and depends heavily on experts. Furthermore, there is no traceability between the resulting role model and the business artifacts, e.g., business processes. Due to missing traceability and the absent possibility to check the role model automatically against ACRs from the business level, RBAC is not well aligned with the business level ACRs. There is a lack of tool support for security experts during the manual engineering of the role model.

Organizations aim for their business goals by executing business processes. Often software systems are required to support these business processes. The EAA links business processes and software systems by organizing the system landscape and services in an architecture. It is an IT level artifact and designed by enterprise architects. As explained in Section 1.1, the alignment of business processes and EAAs provides considerable benefits such as efficiency, resource savings and increased performance [90] but this alignment is hard to achieve due to various reasons [90, 140]. More information on EAAs is provided in Section 2.3 and Section 6.3.1.

**P6 Complex and error-prone designing of the enterprise application architecture:** To cope with the rising number of functional and non-functional requirements stemming from different stakeholders, the enterprise architect designs the EAA. According to Gartner [115], EAA requires a holistic approach that does not only focus on technical solutions but involves stakeholders and businesspersons making it complex to design a correct EAA [230]. In particular, modeling security and privacy requirements is complex [25]. Gartner also identifies the complexity in understanding correct requirements due to different knowledge and models of business and IT and states this as a severe challenge of the business level. As previously described in Section 1.1, a communication gap exists that may lead to design errors, undermining the whole organizational security. Such mistakes are very cost-intensive if not found during the design phase [40, 123]. The most prominent mistakes are logical and design mistakes. While logical mistakes simply arise from faults and false solution approaches, design mistakes arise from unclear, false interpretation and misunderstanding of requirements. All in all, designing a correct EAA with regard to the business level ACRs is complex and error-prone.

**P7 Missing alignment between enterprise application architecture and business level access control requirements:** The enterprise architect has to consider many functional and non-functional requirements. Two of them are IT security and privacy requirements. A fundamental building block of both are the ACRs stemming from laws and the business level. The knowledge about critical business assets lies on the business level. This widens a communication gap due to different terminology and domain specific models, making it difficult for the enterprise architect to understand business level's needs appropriately [140] (see Section 1.1 for more details). Consequently, it is challenging to align the EAA with business level ACRs correctly [231, 140]. Business processes and the EAA are mutually dependent. They affect each other in non-trivial ways [9]. As a result, the models are not well aligned with each other, representing not the requested outcome [37]. They are developed separately and without an appropriate and automatic transfer of ACRs from the business level [25]. At this point, the enterprise architect lacks tool support to check the EAA for violations of business level ACRs.

**P8 Missing support of evolution scenarios for RBAC and enterprise application architecture:** Evolution in organizations is a fact. Not only departments change but also business processes, ACRs and EAAs change steadily. In reality, the business level and IT level are tightly coupled. However, this is not the case for their models [37]. The models affect each other in non-trivial ways [9]. Changes in business processes impose access control changes and thus, require changes in RBAC and EAAs. To this point, different employees in the organization are responsible for these artifacts and the evolutionary change is not well studied and understood so far, especially for ACRs [25]. However, because of the evolution of all the artifacts, an adaptation and alignment of these artifacts is crucial. This adaptation process is complex and manual, resulting in mistakes endangering the overall security of the organization as well as the aspired goals of the business level. Furthermore, each mistake that is not identified during the design phase is very expensive later on [40, 123]. Overall, neither the IT level nor the business level knows or have tool support to check if RBAC or the EAA are aligned correctly with the business level ACRs.

To cope with the problems illustrated in this section there is the need to help security experts in building the role model in line with the ACRs from the business level and furthermore, to help the enterprise architect to analyze the EAA for compliance with business level ACRs.

## 1.3 Research Questions

Two needs arise from the three goals of the business level that were described in the beginning of Chapter 1: an appropriate and compliant establishment of access control policies in the access control system and an alignment between the EAA and the ACRs from the business level. These needs lead to several key problems that were presented in Section 1.2. In addition to these needs, the appropriate and compliant establishment of RBAC and EAA has its own challenges. These challenges are also reflected in the context of organizational evolution and are formulated as key problems in Section 1.2. This thesis formulates research questions based on the problems described in Section 1.2 that arise from the needs of the business level and IT level to align business level ACRs from business processes with RBAC and the EAA. The research questions are structured in two layers. The first layer, RQ1 to RQ8, state an overarching research question for each problem. Afterwards, the second layer subdivides these overarching research questions into more specific sub-questions, which are addressed in this thesis.

**RQ1 What kind of business knowledge can be extracted from business processes about access control requirements?** RQ1 arise from the problem that the IT level has missing knowledge about which business assets are critical and their required protection degree (**P1**). The essential knowledge about which systems are allowed to access which assets and how to design appropriate access control policies resides at the business level. In order to transfer the required business knowledge to the IT level, this thesis addresses the following research questions that subdivide RQ1:

RQ1.1 Is the information about access control requirements from business processes correctly and completely transferred into the role model?

RQ1.2 Are all extracted access control requirements from business processes analyzed in the EAA?

In the case studies conducted in the course of this thesis, RQ1.1 and RQ1.2 will be addressed with accuracy metrics.

**RQ2 How can an alignment of business processes, RBAC and the enterprise application architecture help the business level and IT level to better understand mutual dependencies stemming from access control requirements?** The different terminology, domain knowledge, domain-specific models and modeling tools between the business level and IT level widen a communication gap (**P2**). This may lead to errors and security breaches undermining the aspired goals of the business level. However, the business level is in need to express ACRs in a way that the IT level understands properly. To address this problem RQ2 needs to be answered. In order to close the communication gap between the business level and IT level the sub-questions RQ1.1 and RQ1.2 of RQ1 need to be answered.

**RQ3 What kind of business knowledge is no longer needed on the IT level when RBAC and the enterprise application architecture are automatically aligned with business level access control requirements?** Experts are required on the IT level who understand different terminologies and domain-specific models of the business level and IT level (**P3**), so that business level ACRs can be correctly transferred to models of the IT level. This need slows down the overall process of alignment, especially in evolution scenarios. Furthermore, these experts bring in an additional error rate during the complex alignment process. To address these problems RQ3 needs to be answered. In order to reduce complexity in the alignment process, this thesis addresses the following research questions that subdivide RQ3:

RQ3.1 Is the number of artifacts, which the security experts need to process manually, reduced?

RQ3.2 Are any extensions or modifications of EAA or business process models required for the architectural analysis?

**RQ4 To what extent can an automatic extraction of business level access control requirements make role engineering more efficient?** RQ4 arise from the complexity of engineering an appropriate role model for RBAC (**P4**). Experts are required who have business- and IT-specific knowledge. During the role engineering, they have to manually work through a vast amount of business processes. Furthermore, the role model requires continuous adjustment due to organizational evolution. Both problems make the overall role engineering process slow, costly and error-prone. In order to make the role engineering more efficient and reduce the errors through the use of BAcsTract and PAcsTract, the sub-question RQ3.1 of RQ3 needs to be answered and additionally, this thesis addresses the following research question with regard to RQ4:

RQ4.1 Can parts of the role engineering process be automated?

**RQ5 How can RBAC be aligned with business level access control requirements?** The missing alignment between RBAC and business level ACRs arise from the fact that security experts have to manually work through a vast amount of business processes to elicit ACRs (**P5**). This is a manual task that is tedious and complex and does not provide enough traceability between the resulting role model and the business processes. To support this process by establishing traceability between the role model and the business processes with regard to ACRs, this thesis addresses the following research question with regard to RQ5:

RQ5.1 Is a generated role model element always originating from a business process element and thus traceable?

**RQ6 To what extent can an identification of access control requirement breaches in the enterprise application architecture make error resolution more efficient?** The enterprise architect designs the EAA to cope with the rising number of functional and non-functional requirements of the business level, for example, IT security, privacy and ACRs that stem from various stakeholders. In particular modeling security and privacy requirements is complex. Another challenge lies in the complexity of correctly understanding requirements due to different terminology and domain-specific models between the business level and IT level. Consequences are logical and design mistakes. Logical mistakes simply happen from faults and false solution approaches. Design mistakes happen due to misunderstood requirements. Hence, designing an appropriate EAA is complex and error-prone (**P6**). To tackle this problem RQ6 needs to be addressed. In order to reduce complexity as well as logical and design mistakes, this thesis addresses the following research questions that subdivide RQ6:

RQ6.1 Are logical and design mistakes of service call input/output parameters identified?

RQ6.2 What is the accuracy of identified ACR breaches in the EAA?

**RQ7 How can the enterprise application architecture be aligned with business level access control requirements?** RQ7 arises from the missing alignment between the EAA and business level ACRs (**P7**). While the enterprise architect has to design an enterprise application in compliance with ACRs, the knowledge about critical business assets and ACRs lies on the business level. This and other reasons make it difficult for the enterprise architect to correctly understand and implement business level needs [230, 90]. Additionally, business processes and the EAA are mutually dependent. This means they affect each other in non-trivial ways making it difficult to align these models with each other. To support the enterprise architect in the establishing of a traceability between the EAA and the business processes and thus, to align both in terms of ACRs, this thesis addresses the following research question with regard to RQ7:

RQ7.1 What is the accuracy of generated traceability information in the ACR mapping model?

**RQ8 How can an alignment of business processes, RBAC and the enterprise application architecture support evolution scenarios of business processes, RBAC and the enterprise application architecture?** Organizations and their domain-specific models evolve constantly over time. However, models of the business level and IT level are coupled tightly in practice. Changes in business processes may require changes in RBAC or the EAA. Thus, during evolution scenarios adjustments in each model are inevitable. This evolutionary change is not well studied and understood so far, especially for ACRs (**P8**). The alignment process itself is manual, complex, error-prone and lacks appropriate tool support. To tackle this problem RQ8 needs to be addressed. In order to align business processes, RBAC and the EAA during evolution scenarios, this thesis addresses the following research questions that subdivide RQ8:

RQ8.1 Can changes of the role model resulting through changes in business processes be computed automatically?

RQ8.2 Can the architectural analysis be automatically computed in evolution scenarios?

## 1.4 Approach

To tackle the problems introduced in Section 1.2, this thesis proposes an approach to align models of the business level with models of the IT level in terms of ACRs. The need for an alignment arises from several business level goals, whose importance grows increasingly (see Section 1.1). Business Process Model and Notation (BPMN) [5] is a semi-formal notation and the most prominent modeling language for business processes [23, 213]. Another language is Petri nets [170] that provide a formalized view of business processes. Transformations between both exists. A major aim of this thesis is to provide approaches that need minimal to no adjustments or extensions to the domain specific models that are prominently used in organizations. The approaches of this thesis are designed in a way to impose nearly no additional overhead and require no additional expertise in order to be utilized. This is achieved by reusing already existing models for business processes and IT architecture, that organizations have to design anyway. Due to this reason, this thesis focuses on BPMN, the de facto standard modeling language for business processes [23, 213], as the main business level artifact.

The idea of this thesis is to reuse implicit knowledge about ACRs that lies in business processes (business level artifact) and to transfer this knowledge to RBAC and EAAs (IT level artifacts). Many organizations, especially big organizations, model business processes to automate and improve the quality of their organizational processes. There are many laws and security guidelines like the IT Security Act [89], the ISO/IEC 27000-series [210] and the IT Baseline Protection [117] prescribing or recommending security mechanisms, including ACRs. Many organizations implement them either by prescription or due to business level obligations. Another kind of guidelines, like ITIL [34], details practices for IT service management. ITIL improves quality of IT services in a standardized manner by using best practices to establish effective and appropriate business processes. Therefore, ITIL proposes a set of business processes that help to align IT with business level needs. A specific part of ITIL focuses on access control, providing various business processes for access control and best practices on how to incorporate access control into all organizational business processes. These facts make business processes a rich reservoir for business level ACRs, as not only laws are reflected in them, but also business level needs and IT security and privacy demands (see further explanation in Section 1.1 and Section 1.2).

In addition to BPMN, another business process language is chosen, IntBIIS_LP, which is coupled closer with IT architectures and EAAs. This allows to research how information about ACRs can be transformed to and used in EAAs. IntBIIS_LP is an extension that introduces business processes to PCM, which is an architecture modeling and state-of-the-art performance prediction approach [189]. Conceptually IntBIIS_LP is based on the BPMN standard and introduces a minimal required set of BPMN elements [5].

The approaches to align business processes with domain models of the IT level tackle the problems introduced in Section 1.2. Their purpose is to enable security experts to engineer a more correct role model for RBAC more quickly and to help enterprise architects in designing a more aligned and less flawed EAA. The idea intends to impose only little additional effort to utilize the approaches, by reusing already existing models of business processes and architectures that have to be designed anyway. This makes the approaches especially useful during evolution scenarios, where many misalignments of models are produced due to human errors (see further explanations in Section 1.1). In summary, business level ACRs are extracted from already existing business processes automatically to form:

a) an initial role model with access control policies for RBAC.

b) architectural data flow constraints to analyze the EAA for data flows that violate business level ACRs.

In case a) this thesis proposes the approaches BPMN Access Permission Extractor (BAcsTract) and Palladio Access Permission Extractor (PAcsTract). While BAcsTract extracts implicit ACRs from business processes defined in the de facto standard business process language BPMN, PAcsTract extracts them from IntBIIS_LP. In both cases, the automatically extracted business level ACRs are used to form an initial role model for RBAC. BAcsTract and PAcsTract help security experts during the role engineering process to build an initial role model automatically and to incorporate all ACRs from the business level correctly and completely. During the role engineering process, security experts only have to extend the initial role model with technical ACRs making the overall role engineering process quicker, more cost effective and more importantly, less error-prone. Additionally, an ACR mapping model is computed automatically that links elements of business processes with elements of RBAC. This establishes an alignment that helps the business level as well as security experts to better understand design decisions and mutual dependencies of the models. More details on BAcsTract and PAcsTract are presented in Chapter 3.

In case b) this thesis proposes the approach Access Permission Architecture Aligner (AcsALign). AcsALign builds on top of the extraction of business level ACRs and transforms the extracted ACRs into architectural data flow constraints. Then these constraints are used in an architectural data flow analysis on the EAA to identify forbidden data flows. Forbidden data flows represent violations of the ACRs, i.e., that the EAAs has logical or design mistakes. While logical mistakes simply arise from faults and false solution approaches, design mistakes arise from unclear, false interpretation and misunderstanding of requirements. They are the most prominent mistakes made in the EAA, which happen, for example, due to the complexity of building EAAs (see Section 1.2 for more details). The identification of logical and design mistakes in the EAA helps the enterprise architect to resolve these mistakes and to provide a correct and aligned EAA with regard to ACRs. Furthermore, AcsALign extends the previously built ACR mapping model by linking elements of the EAA with elements of RBAC and business processes together. This establishes a better alignment and comprehensibility of design decisions. Morover, the ACR mapping model helps the enterprise architect throughout the resolution of identified mistakes by providing helpful information that foster the comprehensibility about the mistakes. Further details on AcsALign are presented in Chapter 3.

There are manifold scenarios in which BAcsTract, PAcsTract and AcsALign can be utilized throughout the organizational processes. The engineering and establishment of RBAC and the EAA, are only the obvious ones. As organizations and their domain-specific models constantly evolve, a periodic adaptation of the models and systems is required. This creates a wide scope of evolution scenarios where the approaches of this thesis can be utilized to either improve the adaptation processes or predict required changes and misalignments in one of the three aforementioned models that are stemming from changes in other models. A detailed explanation of how the approaches of this thesis can be utilized in organizations is presented in Chapter 4.

## 1.5 Scope and Assumptions

To explain under which circumstances the proposed approaches show most effects the scope of this work and the assumptions are discussed in this section. The various laws that regulate IT security and privacy [222, 89, 88, 36] as well as the different IT security guidelines [210, 117] build the fundamentals for the business level to identify critical business assets and establish organization-wide IT security as well as privacy strategies and adhere to regulations. These guidelines and regulations are a key business driver for the business level to tailor organization specific ACRs and incorporate them into their information systems.

The business level considers ACRs during the design of business processes. There are various guidelines [34, 32] supporting this design phase. In the scope of this thesis, I assume that ACRs incorporated into business processes by the business level are legally correct and in line with the business goals introduced in Chapter 1. The reason is that this thesis focuses not on identifying erroneously defined ACRs at the business level, but on defining an automated transformation of ACRs from business processes to IT level artifacts. For the same reason I assume that business processes are designed syntactically and semantically correct.

Organizations that define business processes can benefit most from the approaches of this thesis, as these business processes serve as input for the approaches. Especially during evolution scenarios the approaches can directly propagate changes in business processes with regard to ACRs to the IT. As this thesis focuses on the alignment of ACRs between the business level and IT level, organizations with many or complex access control policies or organizations with high-security requirements can benefit most from the approaches presented in this thesis.

In terms of access control, organizations utilizing RBAC [2] or hybrid RBAC concepts (see Section 6.2.5) can profit the most. The extracted ACRs can be used with various access control concepts, but this thesis focuses on RBAC, as it is widely used among organizations of all kind [7] (see Section 2.2 for more explanations regarding RBAC). The healthcare sector is a prominent example [211]. Furthermore, RBAC is also used in combination with other access control concepts (hybrid RBAC concepts are presented in Section 6.2.5). An example is the combination of attribute-based access control (ABAC) with RBAC. Nonetheless, in hybrid RBAC concepts roles and permissions are still required. Hence, the approaches of this thesis can provide considerable benefit by generating them. Based on the generated information security experts can either manually complete the required access control policies or it is possible to extend the approaches in order to extract the required information from other sources. Another reason why RBAC was taken as the access control concept is that the granularity of RBAC policies is well suited in order to extract appropriate access control policies from business processes in order to align the RBAC with the business processes.

Among RBAC there are two disciplines to engineer the role model and its access permissions. Role engineering elicits role models from business artifacts in a manual process, while role mining elicits role models from already existing access control policies in IT systems. Both are described in Section 2.2 in more detail. The approaches proposed in this thesis are role engineering approaches, as they extract ACRs from business processes. However, a major difference to other role engineering approaches is that the extraction of the role model from the business processes is automated and does not require any significant human intervention.

In the area of enterprise architecture this thesis focuses on aligning the EAA with business processes. Therefore, I assume that business processes and the EAA are built in a top-down manner, meaning that the IT level has to meet the requirements of the business level (i.e. business processes). Organizations with such scenarios can benefit most from the approaches of this thesis, but other evolution scenarios of organizations, where changes in RBAC or the EAA impacts the business processes, are also supported. Still Chapter 4 elaborates on the spectrum of scenarios in which organizations may utilize the approaches of this thesis. In particular, the chapter states how organizations may utilize the approaches throughout different evolution scenarios.

The approaches of this thesis have three fundamental objectives which were considered during their design. The first and most significant objective is, that the approaches aim for imposing least possible effort for organizations in order to utilize them. This reflects, for example, in the input models of the approaches. The concepts of the approaches are build on de facto standard languages as BPMN [23, 213] and UML [212] and on models as business processes and EAAs that most organizations have to design anyway. Second, the approaches transfer implicitly modeled business knowledge about ACRs to the IT level. This objective is tightly coupled with the first objective and is reflected in the fact that the approaches do not build upon extended modeling languages that introduce and thus, require specific modeling of security related information as part of the input models. However, the approaches presented in this thesis are compared and delineated from approaches that utilize modeling extensions as part of the related work in Chapter 6. The third objective is that the approaches of this thesis aim to align ACRs across models of the business level (business processes) and IT level (RBAC and EAAs) during evolution scenarios, because throughout evolution scenarios most misalignments and mistakes are introduced.

## 1.6 Contributions

The contributions of this thesis consist of the approaches BAcsTract, PAcsTract and AcsALign. They extract business level ACRs from business processes, to align them with role models of RBAC and EAAs. Thereby, the approaches address the problems described in Section 1.2 with the following contributions.

**C1 Extract business level access control requirements from business processes:** In many organizations, especially organizations dealing with critical infrastructure, business processes are full of ACRs from the business level. By extracting these ACRs the knowledge about critical assets and their protection degrees is transferred from the business level to the IT level (**P1**). This bridges the communication gap between the business level and IT level that widens due to different terminology and domain-specific models in terms of ACRs. Furthermore, this helps the IT level to better understand the demands of the business level (**P2**). After the knowledge about ACRs is extracted, it is transformed into domain-specific models of the IT level. Thus, the knowledge is transferred automatically and the necessity for experts and dependencies on their skills is reduced (**P3**).

**C2 Establish an ACR mapping model for access control requirements between business processes, RBAC and the enterprise application architecture:** The different terminology and domain-specific models between the business level and the IT level widen a communication gap (**P2**). Consequently, understanding correct ACRs is challenging and requires experts having expertise in both domains (**P3**). Both problems are tackled by interconnecting access control relevant elements of business processes with elements of RBAC and the EAA. This establishes an ACR mapping model that allows to track design decisions regarding ACRs across the three mentioned models and thus, couples the domain-specific models together. Experts, but also the business level and IT level, are supported in understanding design decisions in models outside of their expertise. As elements of business processes are connected to elements of RBAC and the EAA, an alignment between these models is established (**P5** and **P7**). This is also the case for evolution scenarios of each of these models (**P8**).

**C3 Build initial RBAC role model from extracted access control requirements:** ACRs from the business level, which are extracted from business processes, are transformed to an initial role model for RBAC. Theoretically, the ACRs can be transformed into any model or policy required by a particular access control system. This thesis focuses on RBAC due to the enormous potential for organizations as explained in Section 1.2 and Section 2.2. The generated initial RBAC role model possesses ACRs form the business level, making the error-prone and cumbersome process of manually extracting ACRs from business processes by security experts, unnecessary. The resulting role model has a complete set of ACRs that resided in the business processes. After the generation of the initial role model it can be extended by security experts with technical ACRs. Altogether, the automatically generated initial role model eases the cumbersome and error-prone work of the security experts of going through the vast amount of business processes (**P3**). This fact also eases the difficulties for organizations to migrate to RBAC systems, as the overall process is accelerated as well as less error-prone due to automation and simultaneously does not impose additional effort on organizations due to the usage of de facto standard models that are designed anyway (**P4**). The traceability between the role model elements of RBAC and business process elements, which is established by the ACR mapping model, allows an alignment of RBAC with business level ACRs (**P5**). Due to the automated generation of the initial role model and the established traceability of elements by the ACR mapping model, a faster adaptation and better support is facilitated during evolution scenarios (**P8**). This provides the opportunity to react faster to changes of evolution scenarios.

**C4 Generate architectural data flow constraints for data flow analysis in the enterprise application architecture:** The extracted ACRs from business processes are transformed into architectural data flow constraints. These constraints are verified by using a data flow analysis on the EAA. This reflects whether the designed architecture is aligned with the ACRs from the business level (**P7**). Any identified forbidden data flow indicates that the designed architecture has logical or design mistakes which violate ACRs. Furthermore, the identification of forbidden data flows enables the enterprise architect to identify the security breach and resolve it at design time (**P6**). This reduces the requirements to rely on expert knowledge in order to identify security breaches resulting from ACR breaches and increases the comprehensibility of identified security breaches and related design decision through the provided results of the architectural data flow analysis (**P3**). Moreover, ACR breaches are resolved before the implementation phase, after which the change of software systems is costly, complicated and sometimes even impossible [40, 123]. The automated analysis that checks whether the EAA complies with the ACRs from the business level, also supports evolution scenarios of business processes, RBAC and EAAs. It means the analysis enables to check whether a change in the role model or business processes or the EAA leads to an ACR breach and thus, requires a change in the EAA (**P8**).

**C5 A high-level process to align RBAC and the enterprise application architecture with business level access control requirements:** Models of the IT level and the business level affect each other in non-trivial ways. Neither role models of RBAC nor EAAs are well aligned with business processes, which are designed by the business level. This is especially true for ACRs (**P5** and **P7**). Therefore, this thesis proposes a high-level process for organizations that explains how to utilize BAcsTract, PAcsTract and AcsALign in order to align RBAC and EAAs with ACRs from business processes.

**C6 A high-level process to identify inconsistencies between models in evolution scenarios of business processes, RBAC and the enterprise application architecture:** Business processes, RBAC and the EAA affect each other in non-trivial ways and are not well aligned. This is especially problematic as business processes, RBAC and EAAs evolve constantly over time. An evolution of one model may require adaptations in the others. This adaptation process is complex and is done manually. Consequently, this process produces errors endangering the overall security of the organization (**P8**). This thesis proposes a high-level process for organizations that explains how to utilize BAcsTract, PAcsTract and AcsALign throughout different evolution scenarios to understand mutual dependencies and compare model alternatives with each other.

Within the scope of this thesis I conducted several case studies to validate the approaches and proposed contributions (see Chapter 5). For each case study a goal question metric

(GQM) model [181] is developed to systematically conduct the case study and measure results. The goals of the GQM model are derived from the problems introduced in Section 1.2. The proposed contributions of this section are related to the goals of the GQM model, as they address the aforementioned problems. Research questions stated in Section 1.3 match with the questions of the GQM model. For each question metrics are introduced to measure the results and to finally derive whether the goals are achieved. The case studies use two different case study systems to validate the approaches of this thesis. First, the community-driven case study system Common Component Modeling Example (CoCoME) is used. It covers a real-world supermarket chain with several evolution scenarios. Second, a collaboration with a national art gallery was done in order to elicit business processes, the EAA and build a role model. Both case study systems are explained in more detail in Section 5.1.2 and Section 5.2.2 of Chapter 5.

## 1.7 Outline

The remainder of this thesis is structured as follows:

- **Chapter 2:** introduces the foundations concerning business processes, RBAC, EAAs, Palladio and IntBIIS. Section 2.1 introduces BPMN, the de facto standard modeling language for business processes, followed by the access control method RBAC in Section 2.2. Afterwards, Section 2.3 introduces EAAs. The last two sections present PCM, a modeling language for IT architectures and IntBIIS, which introduces a modeling language for business processes as part of PCM.

- **Chapter 3:** presents the approaches BAcsTract and PAcstract for extracting ACRs from business processes to form a RBAC role model and the approach AcsALign to identify ACR breaches in EAAs. Section 3.1 begins with the formal concepts of the approaches and lays down which characteristics modeling languages of business processes and EAAs have to fulfill in order to utilize the approaches. Afterwards, Section 3.2 describes the realization of the concepts from the previous section for the modeling languages BPMN and PCM. Therefore, Section 3.2.1 defines the input for BAcsTract, PAcsTract and AcsALign along with relevant boundary conditions. Section 3.2.2 provides an overview over the input models for the approaches, responsible roles and assumptions of this thesis. Afterwards, Section 3.2.3 explains in detail the approaches BAcsTract and PAcsTract and how they extract ACRs from business processes to form a RBAC role model, followed by an explanation of AcsALign and how it uses the output of PAcsTract to identify ACR breaches in EAAs in Section 3.2.4. This chapter concludes with a discussion about BAcsTract and PAcsTract in Section 3.3 and a discussion about AcsALign in Section 3.4.

- **Chapter 4:** elaborates on how organizations can utilize the approaches from the previous chapter throughout different evolution scenarios of business processes and EAAs. It describes how organizations can align their business processes with RABC and EAA when using the approaches. Therefore, Section 4.1 briefly discusses the phases that organizations typically undergo when establishing business processes, as well as an EAA and a role model. Afterwards, Section 4.2 outlines the phases when the approaches of this thesis are utilized. Section 4.3 concludes this chapter with a discussion about benefits and limitations resulting from the utilization of the approaches in evolution scenarios.

- **Chapter 5:** describes the experimental validation of the approaches. Section 5.1 presents the first case study based on the Common Component Modeling Example (CoCoME). While Section 5.1.1 derives research questions and metrics from validation goals according to the GQM method [181], Section 5.1.2 introduces the case study system CoCoME. CoCoME is a community driven case study for collaborative empirical research on software evolution approaches that illustrates a comprehensive supermarket chain. Section 5.1.3 discusses the results and findings of the case study and Section 5.1.4 discusses four aspects of validity with regard to the case study research. The first case study is concluded with a summary in Section 5.1.5. The second case study in Section 5.2 describes the real-world case study of a national art gallery which evolves due to digitalization. While Section 5.2.1 derives research questions and metrics from validation goals according to the GQM method, Section 5.2.2 introduces business processes and the EAA of the national art gallery. Afterwards, Section 5.2.3 discusses the findings of the case study and Section 5.2.4 elaborates on the four aspects of validity with regard to the second case study. The second case study concludes with a summary in Section 5.2.5.
- **Chapter 6:** discusses the related work concerning the contributions of this thesis. Section 6.1 begins with a discussion of IT security and privacy extensions for business process languages and architecture languages as well as transformation approaches for IT security and privacy attributes based on these language extensions. Section 6.2 surveys related approaches with regard to RBAC. Therefore, the contributions of this thesis are contrasted to existing role engineering, role mining and hybrid approaches. Role mining approaches that optimize the role model hierarchy and hybrid access control concepts that include or extend RBAC are discussed, too. Finally, Section 6.3 elaborates on the relation of enterprise architecture management to the approaches of this thesis and on differences to security analysis approaches that are based on de facto standard IT architecture models.
- **Chapter 7:** concludes the thesis by summarizing the scientific contributions and research findings in Section 7.1, laying down how security experts, enterprise architects, the business level and the organization as a whole benefit from utilizing the approaches presented inSection 7.2, recapitulating assumptions and limitation in Section 7.3 and outlining future work on the alignment of the business level and IT level with regard to ACRs in Section 7.4 of this thesis.

## 1.8 Previous Publications

The approaches, concepts and experimental results presented in this thesis are published in scientific publications. In the following, these scientific publications are briefly introduced.

> **1:** Sascha Alpers, Roman Pilipchuk, Andreas Oberweis, and Ralf Reussner. "Identifying Needs for a Holistic Modelling Approach to Privacy Aspects in Enterprise Software Systems". In: *Proceedings of the 4th International Conference on Information Systems Security and Privacy (ICISSP).* vol. 1. 2018, pp. 74–82. DOI: `10.5220/0006606200740082`

A systematic literature review to compare modeling approaches of business and IT was done in the first publication. It identifies the need for a holistic modeling approach for IT security and privacy beginning at the business level models and continuing to the IT level models.

> **2:** Roman Pilipchuk, Stephan Seifermann, and Emre Taspolatoglu. "Defining a Security-Oriented Evolution Scenario for the CoCoME Case Study". In: *4nd Collaborative Workshop on Evolution and Maintenance of Long-Living Software Systems (EMLS'17).* Vol. 37. Softwaretechnik Trends 2. 2017, pp. 70–73

A security review of an information system exemplifying the numerous IT security and privacy requirements stemming from an increasing number of laws is demonstrated in the second publication.

> **3:** Roman Pilipchuk, Stephan Seifermann, and Robert Heinrich. "Aligning Business Process Access Control Policies with Enterprise Architecture". In: *Proceedings of the ACM Central European Cybersecurity Conference 2018.* CECC'18. ACM Association for Computing Machinery, 2018, 17:1–17:4. DOI: `10.1145/3277570.3277588`

The general approach of extracting business level ACRs (BAcsTract) and transforming them to RBAC and EAAs (AcsALign) is illustrated with a real-world running example in the third publication.

> **4:** Roman Pilipchuk. "Coping with Access Control Requirements in the Context of Mutual Dependencies between Business and IT". in: *Proceedings of the ACM Central European Cybersecurity Conference 2018.* CECC'18. ACM Association for Computing Machinery, 2018, 16:1–16:4. DOI: `10.1145/3277570.3277587`

Problems and soaring needs concerning role engineering and how BAcsTract and AcsALign may be utilized by organizations to cope with them in various evolution scenarios are presented in the fourth publication.

**5:** Sascha Alpers, Roman Pilipchuk, Andreas Oberweis, and Ralf Reussner. “The Current State of the Holistic Privacy and Security Modelling Approach in Business Process and Software Architecture Modelling”. In: *Information Systems Security and Privacy* (2019). Ed. by Paolo Mori, Steven Furnell, and Olivier Camp, pp. 109–124. DOI: 10.1007/978-3-030-25109-3

The fifth publication is a systematic literature review and elaborates on the communication gap between the business level and the IT level arising from different terminology, domain knowledge, domain-specific models and modeling tools. It discusses the three aspired goals of the business level (identifying critical business assets, establishing organization-wide IT security and privacy strategies and complying with IT security and privacy laws) and that ACRs are a fundamental building block to achieve all the three goals.

**6:** Roman Pilipchuk, Robert Heinrich, and Ralf Reussner. “Automatically Extracting Business Level Access Control Requirements from BPMN Models to Align RBAC Policies”. In: *Proceedings of the 7th International Conference on Information Systems Security and Privacy (ICISSP).* vol. 1. ScitePress, 2021, pp. 300–307. DOI: 10.5220/0010184403000307

A detailed explanation and discussion of BAcsTract alongside with a case study and the GQM model is presented in the sixth publication.

**7:** Roman Pilipchuk, Stephan Seifermann, Robert Heinrich, and Ralf Reussner. “Challenges in Aligning Enterprise Application Architectures to Business Process Access Control Requirements in Evolutional Changes”. In: *Proceedings of the 18th International Conference on e-Business (ICE-B).* ScitePress, 2021, pp. 13–24. DOI: 10.5220/0010511800130024

The seventh publication contains a detailed presentation of PAcsTract and AcsALign including a case study with the corresponding GQM model.

# 2 Foundations

This chapter introduces the foundations on which the approaches of this thesis, BAcsTract, PAcsTract and AcsALign, are based on. Section 2.1 describes the business process modeling language Business Process Model and Notation (BPMN) along with its basic modeling elements. Business process models in BPMN are consumed by BAcsTract as input. BAcsTract extracts ACRs from the BPMN models to generate a role model for RBAC. Section 2.2 introduces the concept of role-based access control (RBAC), its value and adaptation in the industry as well as role engineering and role mining, two types of methods on how to elicit access permissions for RBAC. Section 2.3 presents briefly the required fundamentals in the area of enterprise application architectures (EAA). The final two sections introduce the Palladio Component Model (PCM) and the Palladio extension Integrated Business IT Impact Simulation (IntBIIS), which introduces business processes in PCM. IntBIIS is extended in this thesis with IntBIIS_LP to introduce a basic set of business process elements into PCM. The approaches BAcsTract and PAcstract of this thesis generate role models for RBAC. In contrast to BAcsTract, PAcsTract does this by consuming business process models in IntBIIS_LP. The approach AcsALign of this thesis, that uses PAcsTract to extract ACRs from business processes, aligns EAAs modeled in PCM with business process ACRs.

## 2.1 Business Process Model and Notation

Business processes reflect the business activities of an organization by specifying sequences of tasks that are done by employees and information systems. [158] categorizes business process languages into traditional, object-oriented, dynamic, and integration languages. Traditional languages such as Event Process Chains focus on understandability. An exception are Petri nets that focus on analyzability. These languages focus on describing the behavior within business processes. While object-oriented languages such as UML focus on defining structures, integration languages focus on exchange formats to orchestrate services. The language Business Process Model and Notation (BPMN) is part of the dynamic languages. They also focus on behavior description. In 2004 the Business Process Management Initiative developed and specified the BPMN standard [5]. They invented it to provide a notation to design business processes that are easy to design and understandable by all stakeholders. Today, BPMN achieved mass adoption and is the de facto standard modeling language for business processes in all kinds of organizations [23, 213].

### 2.1.1 Basic Model Elements

BPMN uses graphical notations to visualize the flow of activities for interacting participants. It has four basic element categories:

- **Flow objects:** are the core describing elements in BPMN and consist of events, activities and gateways.
- **Connecting objects:** are the connectors between the various flow objects.
- **Swimlanes:** organize activities into different responsibilities.
- **Artifacts:** are elements that allow to provide additional context within the other business process elements.

The flow objects events, activities and gateways are the three core elements in BPMN. Figure 2.1 visualizes them.

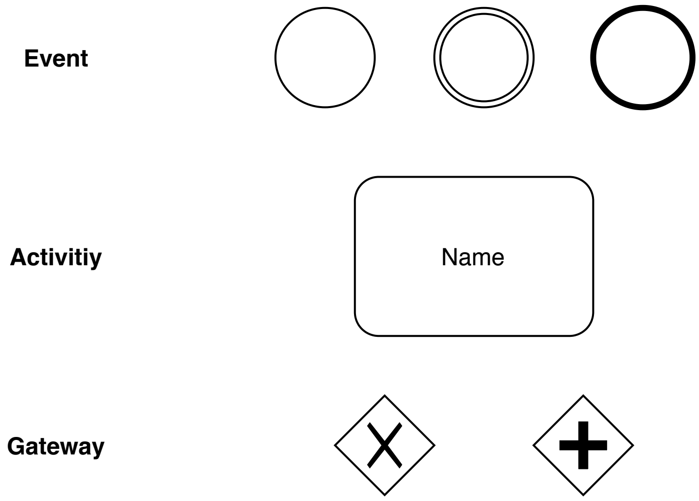


**Figure 2.1:** The flow objects of BPMN.

Events represent something that happens during a business process, for example, an indicator light that goes on. Such an event affects the flow in a process by triggering a sequence of activities or indicating a result that finishes the business process. Events are represented by circles. There are three event types. The circle on the left of Figure 2.1 indicates a start event. It marks the beginning of a business process. The circle in the middle of Figure 2.1 is an intermediate event that is triggered during a business process. Finally, the circle on the right of Figure 2.1 marks an end event that indicates that the business process is finished. The rounded box in the middle of Figure 2.1 denotes an activity. It represents a task that an employee or an information system does in order to achieve the goal of the business processes. An activity can also represent a sub-process, if it has a small plus sign in the bottom. Sub-processes are used to hide a process in an activate of a superior business process. Gateways are represented as diamonds and are used to control the divergence and convergence of sequence flows. This is, for example, used to illustrate different cases of decisions. The gateway on the left of Figure 2.1 represents a split of alternative paths based on a decision, while the gateway on the right represents a fork for parallel paths without any further conditions.

Connecting objects interconnect the various flow objects to form a control flow. Figure 2.2 visualizes them.

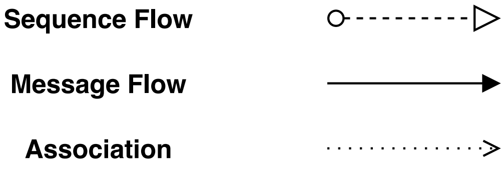


**Figure 2.2:** The connecting objects of BPMN.

A sequence flow is used to define the control flow of a business process by defining the sequence in which the activities have to be performed. When an activity is finished, the sequence flow shows the next activity that has to be done. A massage flow is used to define the flow of messages that are exchanged between the boundaries of pools. Associations are used to interconnect artifacts with other flow objects. For example, in the context of data objects the association indicates whether the data object is required as input for an activity or is produced as an output of an activity.

Swimlanes are elements to organize activities into separate visual parts to illustrate different responsibilities. Figure 2.3 visualizes them.

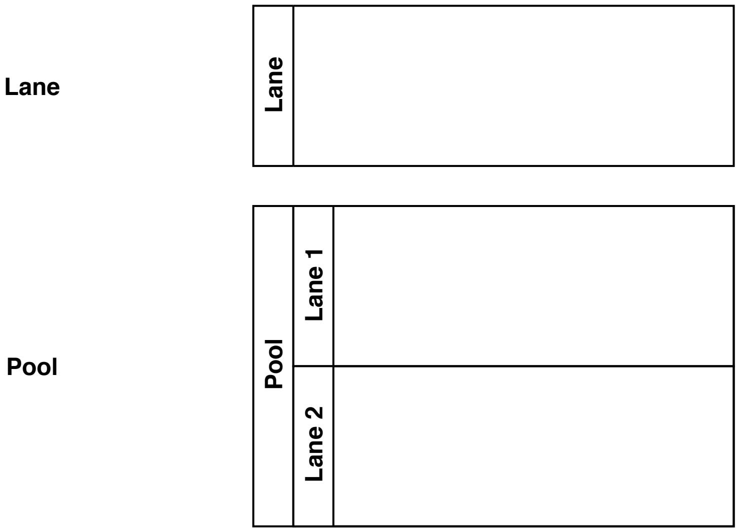


**Figure 2.3:** The swimlanes of BPMN.

The top of Figure 2.3 shows a lane. The lane groups activities into a single responsibility together. The lower part of Figure 2.3 illustrates a pool with two lanes. The pool is used to group one or more lanes together, which is used to illustrate departments or different organizations.

Artifacts are elements in BPMN that allow to add specific information and context to a business process. There are three different elements that belong to the category of artifacts that are shown in Figure 2.4.

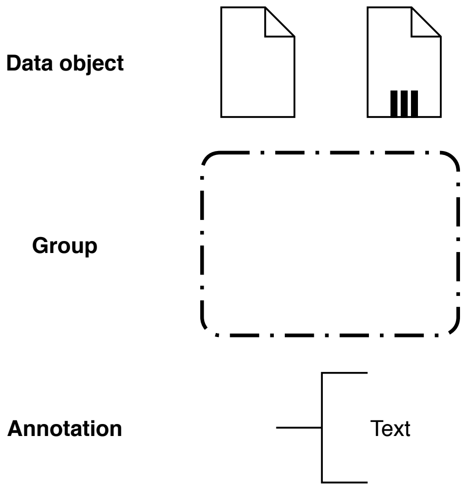


**Figure 2.4:** The artifacts of BPMN.

Data objects represent data that is required or produced in an activity. Data objects with three dashed lines inside represent a collection. A group is used to add another layer of activity grouping. In contrast to lanes they does not affect the sequence flow in a business process. Annotations provide the possibility to add further textual information.

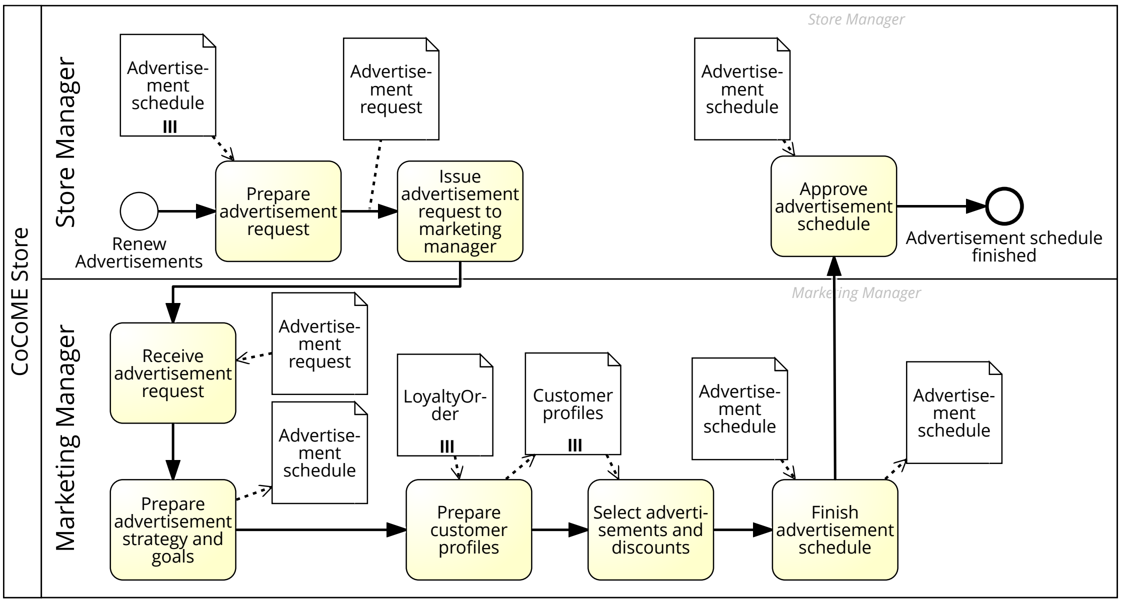

CoCoME Store
Store Manager
Marketing Manager
Store Manager
Marketing Manager
Advertise-ment schedule
Advertise-ment request
Advertise-ment schedule
Renew Advertisements
Prepare advertisement request
Issue advertisement request to marketing manager
Approve advertisement schedule
Advertisement schedule finished
Receive advertisement request
Advertise-ment request
Advertise-ment schedule
LoyaltyOr-der
Customer profiles
Advertise-ment schedule
Advertise-ment schedule
Prepare advertisement strategy and goals
Prepare customer profiles
Select adverti-sements and discounts
Finish advertisement schedule

**Figure 2.5:** Shows the business process *Prepare Advertisements and Discounts* of a supermarket store.

Figure 2.5 shows an example process in BPMN. It depicts the process *Prepare Advertisements and Discounts* of a supermarket store and has a pool named *CoCoME Store* and two lanes named *Store Manager* and *Marketing Manager*. The process is triggered by the store manager, who decides to renew advertisements and discounts of the store. This is shown by the start event in the lane store manager in the top left corner of Figure 2.5. Therefore, the store manager reviews the previously issued advertisement schedules and prepares a new advertisement request for the marketing manager in the activity *Prepare advertisement request*. This activity requires the input data collection *Advertisement schedule* and produces the data object *Advertisement request*. Afterwards, he sends the advertisement request to the marketing manager which is represented by the activity *Issue advertisement request to marketing manager*. The marketing manager receives this advertisement request and begins the preparation of a new advertisement schedule according to the advertisement request of the store manager (activities *Receive advertisement request* and *Prepare advertisement strategy and goals*). In order to select proper advertisements and discounts he analyzes customer profiles of loyalty customers (activity *Prepare customer profiles*). Finally, he selects advertisements and discounts according to the needs of the customers and finishes the advertisement schedule (activity *Select advertisements and discounts* and *Finish advertisement schedule*). In the last activity *Approve advertisement schedule*, the store manager receives the data object *Advertisement schedule* as input and approves the proposed advertisement schedule. Afterwards, the business process finishes with an end event.

### 2.1.2 Value and Usage

In 2014 the BPMN version 2.0.2 was released. BPMN is now specified by the Object Management Group (OMG), which is a leading consortium for IT industry standards. Under the ISO 19510 BPMN is ratified as a world-wide standard. Business process guidelines such as ITIL [34] provide sample processes in BPMN. Today the majority of organizations with business processes rely on BPMN. Two key factors have led to this mass adoption. First, the graphical notation of BPMN is easy to understand for business experts as well as for stakeholders from other areas like law and IT. Second, BPMN has historically helped to close the gap between modeling business processes and executing business processes by providing an internal mapping of the graphical BPMN notation to the core process execution language Business Process Execution Language (BPEL). Consequently, by taking BPMN as the language of the business level to express their requirements, additional effort for organizations to use the approaches is reduced.

## 2.2 Role-Based Access Control

This section briefly surveys the foundations of role-based access control (RBAC). Section 2.2.1 lays down the core concepts of RBAC. Sections 2.2.2 and 2.2.3 introduces the two method types of eliciting access permissions for RBAC. Afterwards, Section 2.2.4 briefly states the value of utilizing RBAC in organizations. Finally, Section 2.2.5 concludes with a summary.

### 2.2.1 Role-Based Access Control Concept

RBAC is a widely used access control concept to manage and restrict access in information systems [7]. It uses roles to organize access permissions and a hierarchy to ease the management of roles. Typically, non-IT employees are responsible to assign roles to employees, while IT employees are responsible to establish correct roles and functionalities for the purpose of assignment. The introduction of RBAC made a major step in easing this complex process [66]. In 1992 a first systematic definition of RBAC was published by David Ferraiolo and Richard Kuhn [67]. After, several more publications about RBAC the National Institute of Standards and Technology (NIST) adopted in 2004 RBAC as a standard. In 2012 it was revised forming the current NIST RBAC standard [2].

An access permission states which operation is allowed on which object. It describes an object-operation pair, where object is defined as any system resource, for example, a file, printer or service and operation is defined as an execution of some function on this object. RBAC introduces roles between the user and the access permissions. Instead of assigning access permissions directly to the user they are assigned to roles. A user is in possession of one or more roles which can be activated during a session to take over the access permissions of the activated role. Roles are distinguished between business roles and technical roles. While business roles describe jobs and business functions of employees in the organizational context, technical roles reflect the usage of objects or services and have no organizational meaning [66].

RBAC uses hierarchies to work effectively with roles and define relationships between roles. Therefore, roles may inherit permissions from other roles building a role hierarchy [66]. There are different types of hierarchies which all reflect organizational aspects of employees. For example, the role head of division may inherit from the role accounting clerk as the role head of division is the manager and thus, can do everything what the accounting clerk is allowed to do. This reflects the organizational hierarchy within departments. In RBAC a role model is used to define the required set of roles, their permissions, a hierarchy and the assignment of users to roles. The elicitation of a role model tailored to the needs of the organization is a challenging task [66].

There are several types of RBAC published first by Ravi Sandhu et al. in [183]. Later, NIST refined these types and proposed a standard including reference models for each type of RBAC [2]. Each RBAC reference model defines a taxonomy of RBAC functionalities that are bundled into a package. In the following paragraphs, the types core RBAC and hierarchical RBAC are introduced.

Core RBAC defines the fundamental parts of RBAC that are bundled into a basic package. The defined element sets and relations are shown in Figure 2.6. Core RBAC defines six basic element sets: the users, roles, sessions, operations, objects and permissions [2].

On the right side of Figure 2.6, the set of permissions is shown as a set of object-operation pairs. The other ovals define the sets for roles, users and sessions. RBAC defines several relations which are used to formalize policies for RBAC. In the *Permission Assignment* the

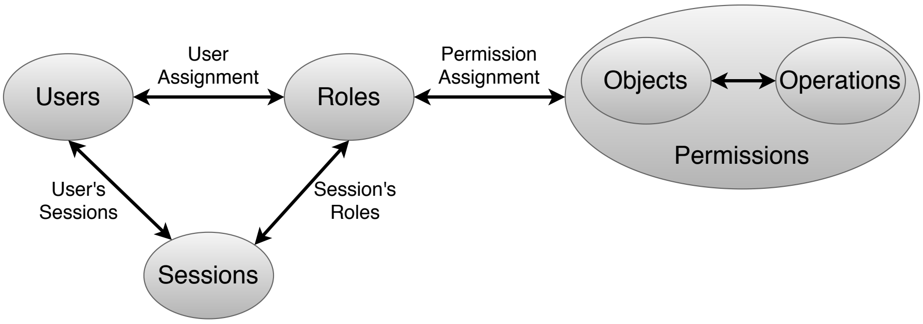


**Figure 2.6:** The reference model of the core RBAC [2].

role is assigned to its set of permissions. The bi-directional arrow indicates a many-to-many relation, meaning that a role can contain one or more permissions and a permission can be assigned to one or more roles. In the *User Assignment* the user is assigned to its roles. This relation is also a many-to-many relation. During a session a user activates one or more roles, usually by using some kind of authentication, for example, username and password, to obtain the permissions of the role. A session always belongs to exactly one user. The relations *User's Sessions* and *Session's Roles* provide the sessions of the user and respectively the roles activated in a session.

The core RBAC does not define a hierarchy for roles. This is done in the hierarchical RBAC. This type extends the core RBAC with role hierarchies [2]. The reference model is shown in Figure 2.7.

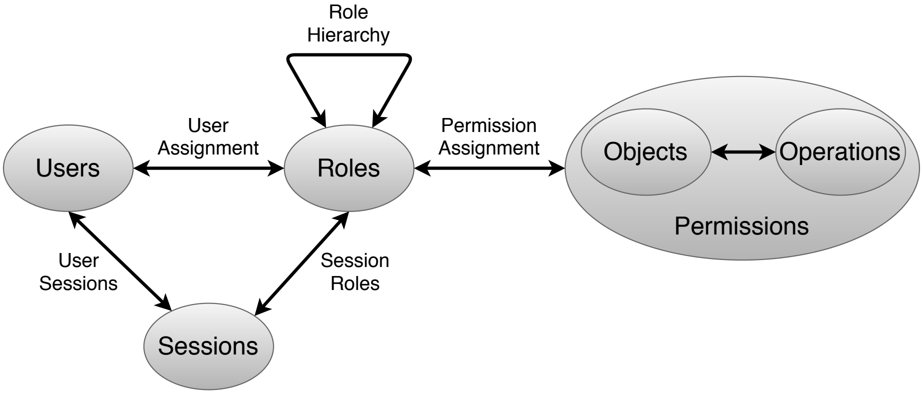


**Figure 2.7:** The reference model of the hierarchical RBAC [2].

In the top of Figure 2.7 a bi-directional arrow on the set roles defines a relation for the *Role Hierarchy*. The role hierarchy is an inheritance that is defined as a subset between roles. If role A has at least all permissions of role B then role A inherits from role B in the role hierarchy.

The central part of RBAC is the role model. It bundles all roles, their permissions and the role hierarchy. There are two major method types to elicit a role model according to the needs of an organization. The following two sections will describe them.

### 2.2.2 Role Engineering

Edward Coyne et al. [72] were the first to propose a systematic top-down scheme to elicit role models. They introduced the term role engineering as an essentially requirements engineering process. During the role engineering the role model is engineered manually by security experts. Therefore, they analyze various business artifacts manually to understand the ACRs of the organization [66]. Such business artifacts can be business processes, process descriptions, organizational charts, job instructions, etc. Even interviews with employees might be carried out, reflecting their daily work. The challenge is to engineer appropriate and accurate roles, their permissions and a hierarchy, so that they comply with the ACRs, which are mostly implicitly hidden across various types of documents of the organization.

Various role engineering approaches are proposed in [60, 172, 98, 57, 219, 218] on how to systematically engineer a role model by hand, taking different business artifacts into account. They all have in common that they define roles from a business point of view. Such business roles express the employee's daily work and comprise all permissions required to fulfill this work. They are structured along the business value of an organization by looking at the various tasks done by employees across their business processes. Business roles can express organizational functions, for example, developer and project manager, they can express organizational responsibilities, for example, head of department and facility manager or they can express parts of processes such as privacy audits and production monitoring. These definitions of roles align with the organizational structure and reflect identities inside organizations and business functions. When compared to technical roles that are elicited with role mining approaches (explained in Section 2.2.3), business roles have several benefits [66]. They are easier to understand and thus, easier to manage as the permissions of roles, for example, developer or project manager are deducible from their name. Non-IT employees are able to understand the meaning behind such roles which becomes crucial as they are often human resources employees or managers of various domains that are responsible in correctly assigning roles to employees. The alignment along the organizational structure also helps in defining an appropriate and meaningful role hierarchy [66]. Hierarchies of business roles often reflect the organizational chart.

Besides the many benefits of RBAC, a correct and compliant incorporation of ACRs into the role model is challenging [66, 151, 35]. The challenge is to elicit appropriate and accurate roles, their permissions and a hierarchy matching the ACRs of the business level [151]. Engineering a role model for RBAC is costly as several security experts are required to undergo a manual process of engineering the role model from the vast amount of business artifacts. As explained in the Chapter 1, depending on the size of an organization business processes of an organization can grow easily into hundreds, resulting in a vast amount of complex and interrelated processes demanding specific business knowledge to understand them. Consequently, role engineering approaches are not scalable. Due to the required knowledge in different artifacts of business and IT, experts are required to conduct the role engineering process. Furthermore, the engineering of the role model is error-prone [66, 91]. Role engineering is a complex and tedious process in which a vast number of interrelated artifacts have to be understood correctly. Human errors may occur leading to a potential security threat for the organization. Each error may evolve to a severe security breach that may lead to a leakage of sensitive information or business secrets (see also Chapter 1).

Furthermore, each organization and their artifacts evolve continuously over time resulting in changes of ACRs. Hence, security experts are constantly required to adjust the role model to the changes of ACRs. For example, this is the case for the lifecycle of business processes and EAAs, meaning that changes of business processes or the EAA may require adjustments in the role model. This continuous adaptation of the role model has all the drawbacks of the role engineering process. It also increases the problem of human errors. Altogether, the manual engineering of the role model is slow, costly and error-prone.

### 2.2.3 Role Mining

Role mining is a bottom-up approach to elicit a role model from existing user-permission assignments as access-control lists. In comparison to role engineering it works in reverse order because it starts at the resources and analyzes the structures of user-permission assignments. The basic idea is that user-permission assignments of organizations already possess all the required information to form a role model. While this seems to be a trivial problem, it is often required to meet further conditions to get an optimal set of roles.

In [35] and [83] role mining approaches were analyzed in a meta-study. Many different role mining approaches exist that are based on different types of user-permission assignments or aim to optimize different characteristics in the resulting role model. Based on the same set of user-permission assignments two role mining approaches may result in completely different role models. This is the case as the approaches differ in their optimization goal. The optimization goal might be, for example, to minimize the amount of roles, to minimize the number of assignments, to optimize the size of the hierarchy, etc. [83].

Role mining has several assumptions and limitations [35]. While it is scalable in comparison to role engineering approaches, the major drawback is that it cannot provide a role model from the business point of view. It means that role mining approaches provide technical roles and not business roles as role engineering approaches. The problem with technical roles is that they do not reflect the business function of an employee nor any other organizational structure, making it hard to comprehend the meaning of the resulting roles [66]. This contradicts the goal of RBAC to provide a meaningful set of roles to ease the management and role-user assignment. Besides, an assumption is the existence of a complete and correct set of user-permission assignments. While this might be the case for existing organizational departments, it is not the case for newly build departments and evolution scenarios of organizations and business processes. Furthermore, it is hard to assume that existing user-permission assignments are correct, as it often occur that permissions of users are not revoked and grow with the time of employment. Hence, building upon a set of existing user-permission assignments might result in an erroneous role model that does not reflect the ACRs of the business level. In addition, a role model that is based on role mining does not take business artifacts, for example, business processes into account. Thus, it cannot be told whether it complies with the ACRs from the business processes or not. Due to these reasons, role mining approaches are used as a complementary approach to role engineering [66].

### 2.2.4 Value and Adoption

RBAC is a widely used access control concept to restrict access to information systems [7]. Especially bigger organizations are interested in RBAC as it brings several advantages. Compared to other access control concepts such as mandatory access control and discretionary access control, it is beneficial in the management and expression of access control as well as in the provided degree of security [66]. The greater the number of employees in an organization, the more RBAC simplifies complexity in managing permission assignments and individual user permissions. RBAC is developed to simplify authorization management and thus, reduces administrative costs [66]. The concept of role hierarchies reduces the amount of duplicate permissions and eases the overall management and assignment of employees to permissions [66]. Hence, RBAC not only enhances security and integrity of IT systems but also the organizational productivity. Due to these advantages, many organizations and industrial sectors benefit from RBAC. There are many case studies in the healthcare sector that undermine these benefits [211].

In 2010, NIST estimated that RBAC research has saved the industry over 1.1 billion dollars over multiple years [7]. Over time the overall adoption of RBAC has grown across organizations. The economic benefits were based on three achievements. RBAC allowed a "more efficient provisioning by network and systems administrators", a "reduced employee downtime from more efficient provisioning" and a "more efficient access control policy maintenance and certification" [7]. NIST concluded that more than 80 percent of the analyzed organizations with an employee size of more than 500 employees realized a better security strategy through the usage of roles.

### 2.2.5 Summary

This section briefly outlined the development of RBAC and summarized most important definitions of the RBAC types core RBAC and hierarchical RBAC in Section 2.2.1. In Section 2.2.2 role engineering was introduced explaining its assumptions, concepts and challenges. Afterwards, Section 2.2.3 differentiated role mining approaches from role engineering approaches and explained their concepts, assumptions and limitations. Finally, Section 2.2.4 presented scientific findings on the value and adoption of RBAC in the industry.

## 2.3 Enterprise Application Architecture

Organizations aim for their business goals by executing business processes. Often software systems are required to support business processes. The enterprise application architecture (EAA) links business processes and software systems by organizing the system landscape and their services in an architecture. EAA is a view in the much broader perspective of the enterprise architecture. Enterprise architecture are frameworks and practices on how to develop and organize various important information of the organization, for example, business processes, data, applications and infrastructure to achieve business goals and realize business strategies. Such frameworks aim to enhance effectiveness and efficiency in the organization. There are various frameworks, for example, TOGAF [96], FEAF [94] and EAF Gartner [114], that propose ways on how to establish and maintain enterprise architectures. All of them define various layers with different views that reflect certain perspectives on the information of the organization. One of these views is the EAA.

In IT EAAs describe the interaction of systems and the behavior of their services. It focuses on defining how the numerous services work with each other. This is done by organizing the various systems of an organization, for example, databases, applications servers and middleware systems and by defining their interfaces. The interfaces contain service functions that are provided by a service running on a system. By wiring interfaces the various systems are connected with each other. Calls to other systems are termed external calls and can be required by a service for its computation. For example, when a marketing manager finishes an advertisement schedule it has to be persisted in a database. Therefore, an external call to the database is made. The enterprise architects are responsible to design the EAA according to the needs of the business so that systems and services support business processes efficiently and reliably. Therefore, managing the dynamics of services and the composite architecture of systems as well as their components is essential. Understanding the different technologies and how they work and affect each other is important to shape a set of technologies that do not jeopardize each other.

The alignment of business and IT artifacts provides considerable benefits such as efficiency, resource savings and increased performance [90]. But enterprise architecture approaches are hard to apply due to their method complexity, the complexity and dynamic of organizations and the number of involved stakeholders [90, 140]. Furthermore, different roles in an organization are responsible for the different views [230]. For example, business analysts are responsible for designing business processes and enterprise architects are responsible for designing the EAA. Both have their own practices, techniques and domain terminology that lead to models that are designed separately from each other. Hence, designing the EAA and aligning it to the business level needs is desirable but hard to achieve in general [149].

## 2.4 Palladio Component Model

The Palladio Component Model (PCM) [189] is a software component model to design component-based architectures, define model transformations and predict various software quality attributes like performance and reliability. PCM allows to describe different components, their connections, interfaces, services, data types, hardware and software resources, the allocation of system and components to hardware resources and the usage of systems and components. It is suited to design various levels of IT architectures, one of which are EAAs. Fundamentally, the concrete syntax of PCM is based on the syntax of UML. Section 2.4.1 introduces the different roles in the development process of PCM. Afterwards, Section 2.4.2 explains the PCM metamodel and Section 2.4.3 summarizes a data flow extension in PCM, which is used in this thesis to identify read and written data types of service calls. Finally, Section 2.4.4 concludes with a summary.

### 2.4.1 Development Roles

The architecture development process in PCM proposes four roles that work on four views of the software architecture [189]. Each role groups tasks that have to be done in order to reflect a view of the software architecture. One or more persons are assigned to a role and are responsible for their tasks. It is possible that one person is assigned to more than one role and thus, works on several views of the software architecture. For example, a component developer can be also a software architect. The following paragraphs discusses the responsibilities of the four roles.

- **Software architect:** In the Palladio development process the software architect leads the development process of an application architecture. He is responsible for designing the architecture by planning the required components of a system, wiring them, defining interfaces and specifying the available services for other applications and users. The bigger picture of the application architecture, which defines how the various parts of the application are connected with each other, is defined in the assembly model. Furthermore, the software architect delegates task to other involved roles.
- **Component developer:** The component developer is responsible for the more fine-grained parts of the application by designing the behavioral specification of components. Therefore, he specifies and implements the components of an application architecture. This is done by specifying the architecture of components as well as the wiring inside components which may also consist of components. He defines interfaces, data types and describes the behavior of services by defining the service effect specification. Service effect specifications describe an abstraction of a component’s behavior as well as the interaction with other components.
- **Deployer:** Specifying software and hardware resources, their characteristics and planning the application resources lies in the responsibility of the deployer. For the resource environment specification he specifies, for example, which CPUs, hard

disk drives, memory and network connections are available. In the allocation model he allocates the various application parts and instances to the previously defined resources.

- **Domain expert:** The domain expert is responsible for specifying the usage of the application. He is an expert in the application's domain and defines the user behavior. This is done by defining usage models that describe how users interact with the components and include a sequence of service calls.

The previously defined roles reflect the typical Palladio roles that are required during the design of the application architecture. Application architectures are a subpart of EAAs that are a subpart of enterprise architectures. When considering the IT architecture from the business perspective of an organization two more roles need to be considered.

- **Enterprise architect:** The enterprise architect is responsible for designing the system and application landscape. This is done in the EAA. Therefore, he is responsible for defining systems, their wiring, interfaces and how they interact with each other from the perspective of the organization as a whole. Software architects are responsible for specifying applications that reside on certain systems that were defined by the enterprise architect.
- **Business expert:** The business expert is responsible for defining the business processes of an organization. He specifies the roles of the organization in the organization environment model and the data objects that are exchanged in the business processes in the data model. The definition of business processes including activities is done in the business process usage model. Each business process is defined in a separate model. With support of the enterprise architect he defines the technical aspects of business processes. These are the service calls done during activities of the business processes. When considering the typical Palladio roles the work of the domain expert is replaced by the business expert as the business expert defines the user behavior as well es the user interaction with the available application services provided by the systems of the organization.

### 2.4.2 PCM Metamodel

PCM has six models that can be specified in a development process of an architecture:

- **Repository:** defines the available data types, interfaces, their operation signatures, components and systems.
- **Assembly:** denotes models that compose components into composite components, composite components into systems and systems into EAAs.
- **Service effect specification (SEFF):** specifies the behavior of a service provided by a component including internal actions and external calls.
- **Allocation:** denotes the deployment of applications, components and systems to hardware resources.

- **Resource environment:** defines the software and hardware resources that are available in an organization.
- **Usage:** denotes models hat specify the user interaction with available services of the architecture.

The following paragraphs present a subset of the aforementioned PCM models that are relevant for the understanding of the research done in this thesis. These are the repository, assembly and SEFF models.

The repository defines the basic elements of the architecture. It contains all available data types, interfaces with operation signatures, components and systems. These elements are used throughout the other models.

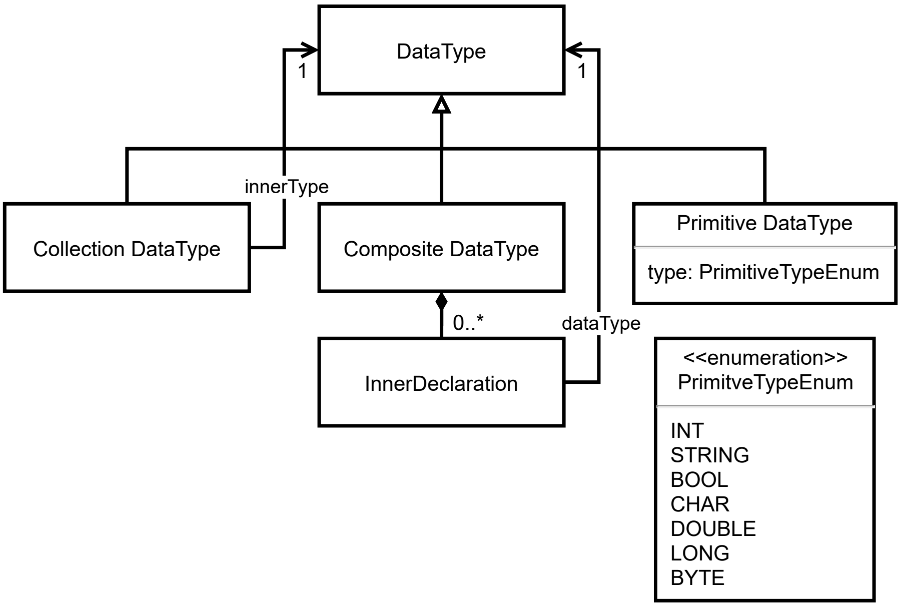


**Figure 2.8:** Shows the metamodel for the data type of PCM [189].

Figure 2.8 shows the metamodel for the data type of PCM. A data type in PCM can be a primitive data type, a composite data type or a collection data type. It is defined and stored in the repository and can be used, for example, in parameter definitions or return types of operation signatures. PCM provides several fixed primitive data types as *INT* and *STRING*. They are shown in the enumeration in the lower right part of Figure 2.8. More complex data types are composite and collection data types. A collection data type represents a data structure, for example, an array or a list. It defines a set of one of the other data types. The *innerType* reference points to the specific data type which the collection represents. A composite data type groups one or more data types together. Thus, it contains an *innerDeclaration* for each data type it contains. The data type order in a sale process is an example for a composite data type as an order itself consists of a list of ordered products, an order number, date and so on.

Figure 2.9 shows the metamodel for the interface of PCM. An interface is a set of services that a component or system provides which it uses to communicate with other components

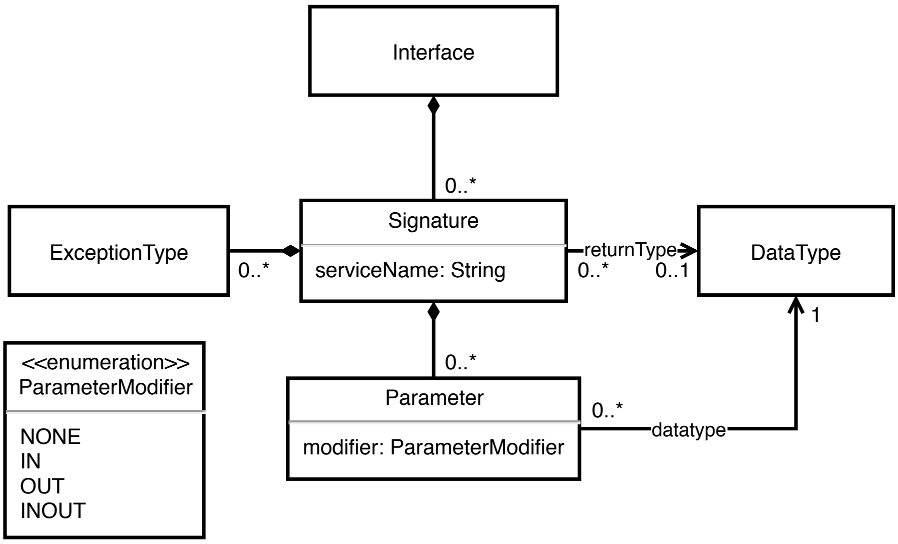


**Figure 2.9:** Shows the metamodel for the interface of PCM [189].

and services. Figure 2.9 shows that the interface contains zero to infinite *Signatures* with a *serviceName*. They represent an operation of a service that is provided to other components and systems. A signature has zero to infinite *Parameters* and zero or one *returnType*. While parameters are data types that need to be passed to the service in order for the service to process the request, the return type is a data type that represents the computed result of the service that is returned to the caller of the signature. A signature can define zero to infinite *ExceptionTypes* which represent the possibly thrown exceptions if an error occurs.

Components and systems bundle services and provide interfaces for communication. Figure 2.10 shows their metamodel. Basic components, composite components and systems (denoted as *SubSystem* in the metamodel) are an *InterfaceProvidingRequiringEntity* that can refer to interfaces that are provided, the *ProvidedRole*, and interfaces that are required, the *RequiredRole*. A basic component is atomic and cannot be further decomposed. Composite components are built by composing several basic components or other composite components. Systems represent a higher level of abstraction as they bundle composite components of similar purpose together. From the perspective of an architect EAAs represent a landscape of systems that bundle certain services together in each system. A system can be further decomposed into components. Structurally they decompose the services provided by the system into smaller parts as well as internal services.

Figure 2.11 shows an excerpt from a PCM repository with the name *org.cocome.cloud*. The first element in the list is a composite component named *org.cocome.web*. The following three elements are basic components as denoted in the end of each line with *<Basic Component>*. The basic component *org.cocome.cloud.web.enterprise* provides the interface *IEnterpriseInformation* and requires the interfaces *storeDAO* and *IEnterpriseDAO* as denoted by the three elements beneath the component in Figure 2.11. Beneath these elements six interfaces are shown. The interface *IEnterpriseInformation* is expanded. Inside it has several operation signatures that contain parameters and a return type. The actual data

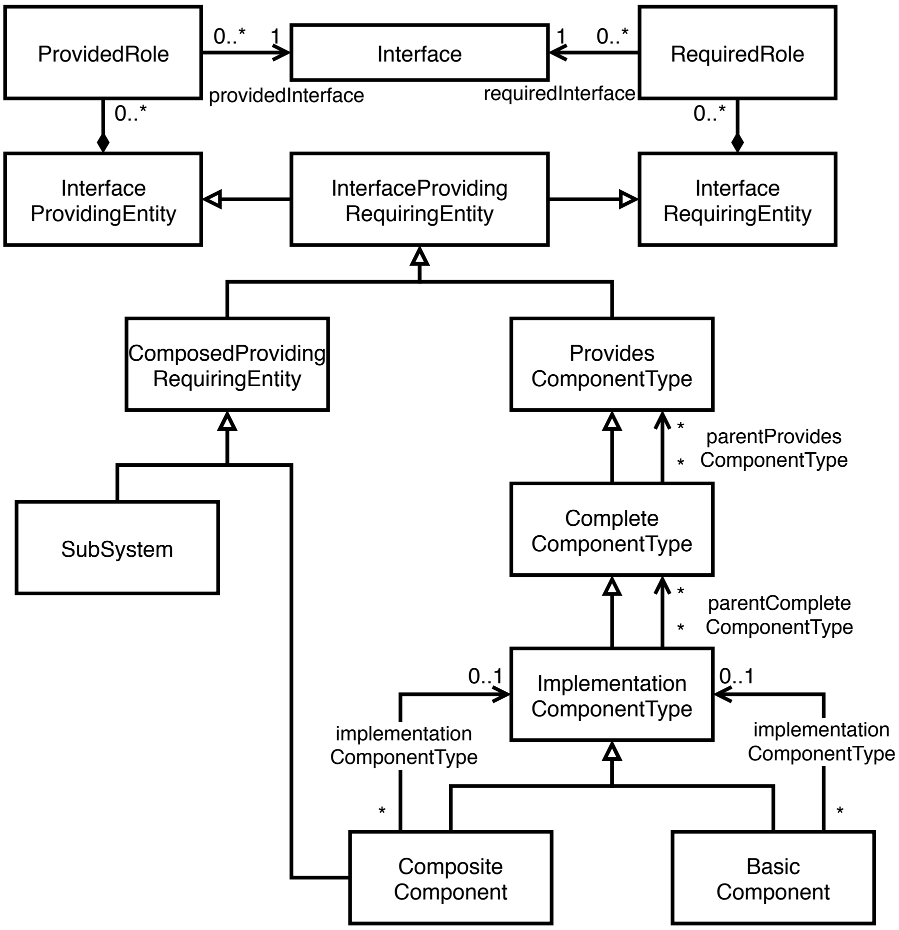
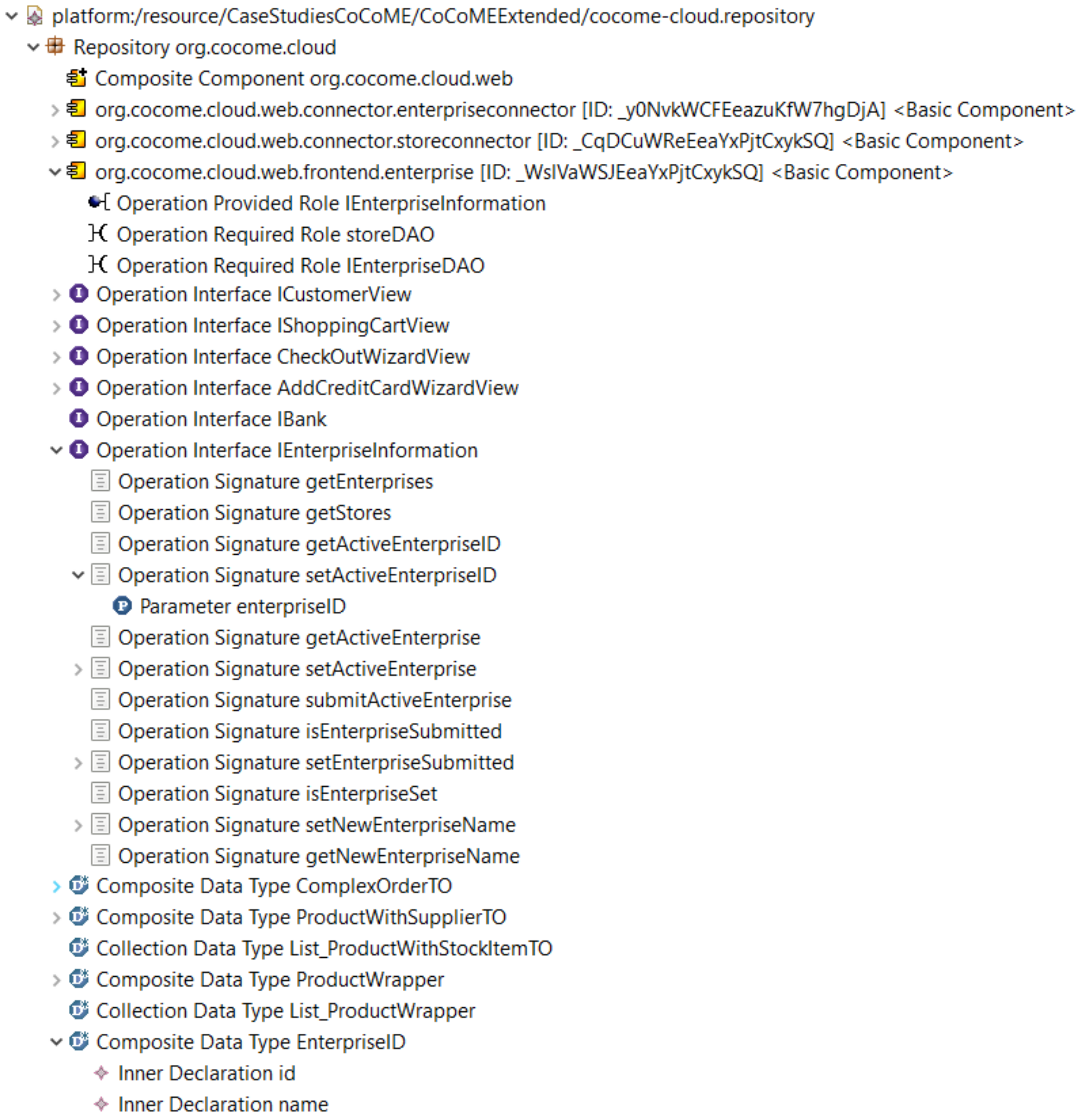


**Figure 2.10:** Shows the metamodel for the component and system of PCM [189].

types are shown in the lower part of Figure 2.11. There are four collection data types, for example, *LIST_ComplexOrderTO* and two composite data types. The composite data type *EnterpriseID* has two inner declarations pointing to other data types.

The behavior of a component is specified in SEFFs. A SEFF is specified for a certain service signature of a component's interface. It always starts with a start action and ends with a stop action. There are several other actions that allow to define the control flow inside the signature. Examples are the branch action and abstract loop action. The internal action and external call action provide the possibility to define computation steps. While the internal action denotes an internal computation in the component, the external call action denotes a call to another component. The external call action calls a signature of a service of another component. Data is transmitted to the other component if the signature specifies parameters and data is returned to the invoking component if a return type is specified in the signature. Figure 2.12 shows an example of a SEFF in PCM.

The SEFF in Figure 2.12 describes the behavior of the operation signature *updateInventory* of the inventory component. While the first circle in Figure 2.12 denotes the start action of the SEFF, the last circle denotes the stop action of the SEFF. The first box in Figure 2.12



```
platform:/resource/CaseStudiesCoCoME/CoCoMEExtended/cocome-cloud.repository
  Repository org.cocome.cloud
    Composite Component org.cocome.cloud.web
    org.cocome.cloud.web.connector.enterpriseconnector [ID: _y0NvkWCFEeazuKfW7hgDjA] <Basic Component>
    org.cocome.cloud.web.connector.storeconnector [ID: _CqDCuWReEeaYxPjtCxykSQ] <Basic Component>
    org.cocome.cloud.web.frontend.enterprise [ID: _WslVaWSJEeaYxPjtCxykSQ] <Basic Component>
      Operation Provided Role IEnterpriseInformation
      Operation Required Role storeDAO
      Operation Required Role IEnterpriseDAO
    Operation Interface ICustomerView
    Operation Interface IShoppingCartView
    Operation Interface CheckOutWizardView
    Operation Interface AddCreditCardWizardView
    Operation Interface IBank
    Operation Interface IEnterpriseInformation
      Operation Signature getEnterprises
      Operation Signature getStores
      Operation Signature getActiveEnterpriseID
      Operation Signature setActiveEnterpriseID
        Parameter enterpriseID
      Operation Signature getActiveEnterprise
      Operation Signature setActiveEnterprise
      Operation Signature submitActiveEnterprise
      Operation Signature isEnterpriseSubmitted
      Operation Signature setEnterpriseSubmitted
      Operation Signature isEnterpriseSet
      Operation Signature setNewEnterpriseName
      Operation Signature getNewEnterpriseName
    Composite Data Type ComplexOrderTO
    Composite Data Type ProductWithSupplierTO
    Collection Data Type List_ProductWithStockItemTO
    Composite Data Type ProductWrapper
    Collection Data Type List_ProductWrapper
    Composite Data Type EnterpriseID
      Inner Declaration id
      Inner Declaration name
```

**Figure 2.11:** Shows an exemplarily excerpt of a repository in PCM.

shows the internal action *calculateNewInventory* that calculates the amount of goods in the actual inventory. Afterwards, the new inventory is stored in the database with the external call action *updateDBInventoryLogic.required.IInventoryDB.add.*

Figure 2.13 shows the metamodel for the assembly models in PCM. An assembly model defines the architecture of composed structures such as composite components and systems. Instead of containing the actual components and systems an assembly model contains *AssemblyContexts* that represent instances of the original components and systems. An assembly model describes one composite component or one system. Therefore, the inner components are defined as assembly contexts. Their provided and required interfaces are either connected through assembly connectors with each other or with the provided and required interfaces of the composite component or system that is specified in the current

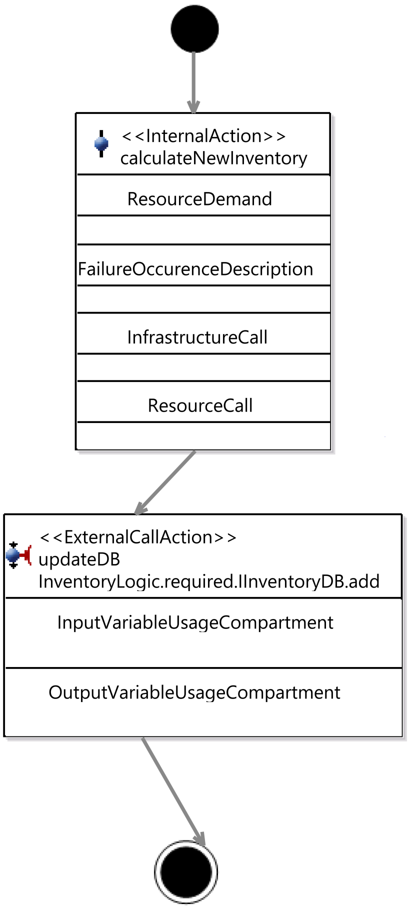

<<InternalAction>>
calculateNewInventory
ResourceDemand
FailureOccurenceDescription
InfrastructureCall
ResourceCall
<<ExternalCallAction>>
updateDB
InventoryLogic.required.IInventoryDB.add
InputVariableUsageCompartment
OutputVariableUsageCompartment

**Figure 2.12:** Shows an example of a SEFF in PCM.

assembly model. The latter wires the interfaces of the specified composite component or system to the outside. The assembly model defines the inner architecture of composite components and systems. An example of an assembly model is shown in Figure 2.14.

Figure 2.14 shows the composition of the system *CustomerOrderManagement* denoted in the higher part of the figure. The system has a provided interface that is shown by the circle in the upper part of Figure 2.14. The system itself consists of the three basic components denoted by the *«AssemblyContext»* inside the system. Their interfaces are connected with arrows which visualize assembly connectors.

### 2.4.3 Data Flow Extension Data Centric Palladio

The authors of [202] describe a data flow extension for PCM called Data Centric Palladio (DC-PCM). In the course of this thesis, this data flow extension is used to derive the read and written data types of service calls of the EAA. Therefore, it extends the PCM external call metamodel element of SEFFs to specify the actual data flows between services and identify database components as stores. By doing so, internal operations on data types such

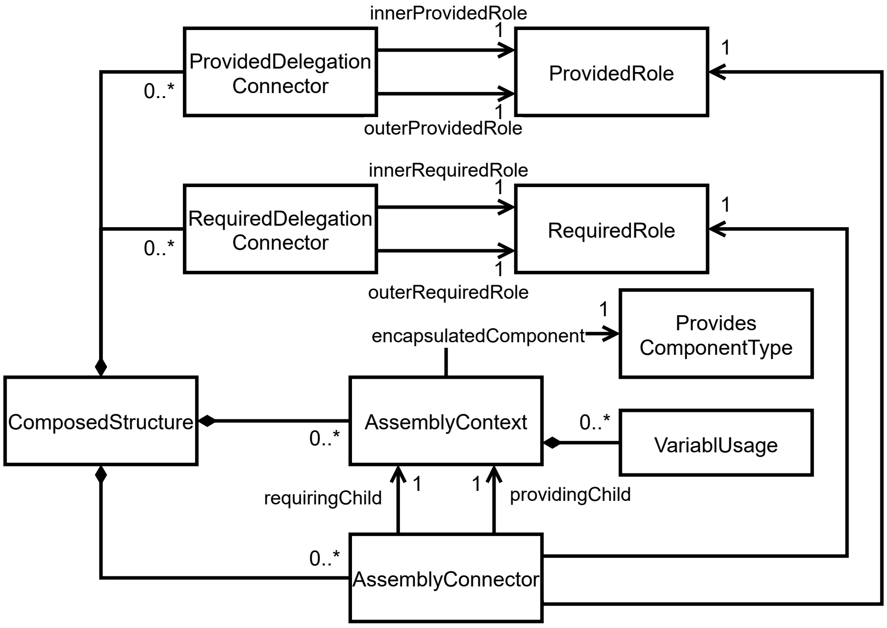


**Figure 2.13:** Shows the metamodel for assembly models in PCM [189].

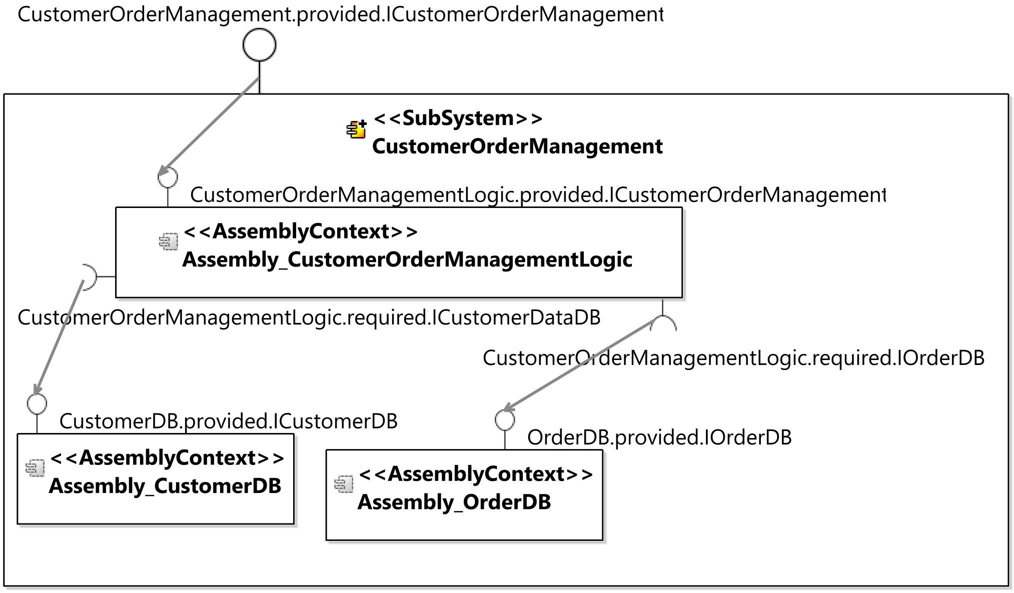
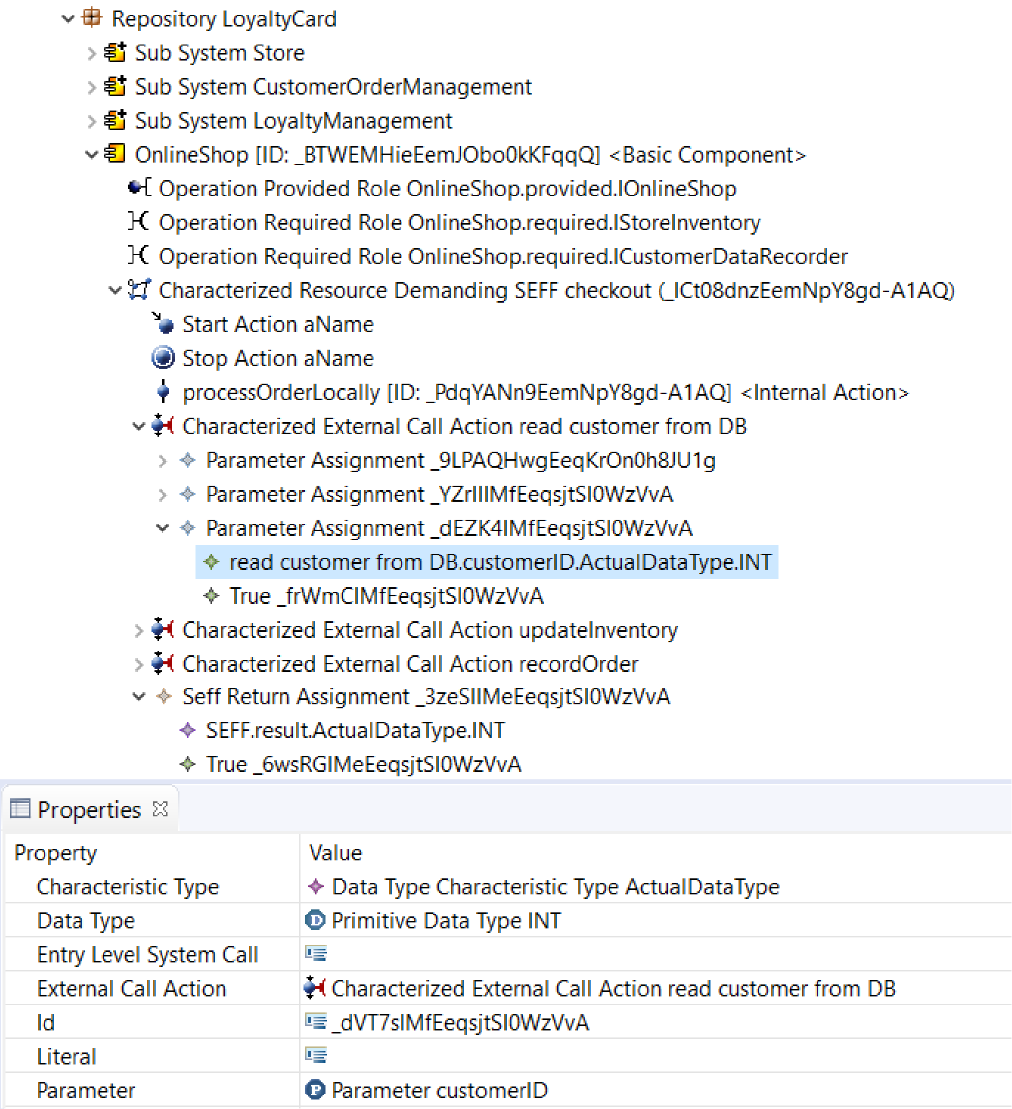

CustomerOrderManagement.provided.ICustomerOrderManagement
<<SubSystem>>
CustomerOrderManagement
CustomerOrderManagementLogic.provided.ICustomerOrderManagement
<<AssemblyContext>>
Assembly_CustomerOrderManagementLogic
CustomerOrderManagementLogic.required.ICustomerDataDB
CustomerOrderManagementLogic.required.IOrderDB
CustomerDB.provided.ICustomerDB
OrderDB.provided.IOrderDB
<<AssemblyContext>>
Assembly_CustomerDB
<<AssemblyContext>>
Assembly_OrderDB

**Figure 2.14:** Shows an example of an assembly model in PCM.

as compositions and decompositions can be reflected in the data flow by specifying the concrete data types that are exchanged between service calls. With the use of the extended information, the signatures of services and the wiring of components and interfaces the actual data flow analysis is conducted. During the data flow analysis external call actions of invoked SEFFs are analyzed. These can be resolved by looking into the assembly models, that define the wiring of systems and components and match them with the called SEFF. By traversing all of these paths a data flow graph is build. To identify the read data types of an invoked service its data flow is analyzed to identify data types that flow during the last step back into the invoked service and that additionally are returned by the service itself. The identification of written data types of an invoked service is computed by analyzing the data flow for data types that flow to an interface of a database and then are stored inside the database with a store operation. The following paragraph explains how the characterized SEFF extends the PCM SEFF.



```
Repository LoyaltyCard
  Sub System Store
  Sub System CustomerOrderManagement
  Sub System LoyaltyManagement
  OnlineShop [ID: _BTWEMHieEemJObo0kKFqqQ] <Basic Component>
    Operation Provided Role OnlineShop.provided.IOnlineShop
    Operation Required Role OnlineShop.required.IStoreInventory
    Operation Required Role OnlineShop.required.ICustomerDataRecorder
    Characterized Resource Demanding SEFF checkout (_ICt08dnzEemNpY8gd-A1AQ)
      Start Action aName
      Stop Action aName
      processOrderLocally [ID: _PdqYANn9EemNpY8gd-A1AQ] <Internal Action>
      Characterized External Call Action read customer from DB
        Parameter Assignment _9LPAQHwgEeqKrOn0h8JU1g
        Parameter Assignment _YZrIIIMfEeqsjtSI0WzVvA
        Parameter Assignment _dEZK4IMfEeqsjtSI0WzVvA
          read customer from DB.customerID.ActualDataType.INT
          True _frWmCIMfEeqsjtSI0WzVvA
      Characterized External Call Action updateInventory
      Characterized External Call Action recordOrder
      Seff Return Assignment _3zeSIIMeEeqsjtSI0WzVvA
        SEFF.result.ActualDataType.INT
        True _6wsRGIMeEeqsjtSI0WzVvA
```

Properties

| Property | Value |
|---|---|
| Characteristic Type | Data Type Characteristic Type ActualDataType |
| Data Type | Primitive Data Type INT |
| Entry Level System Call | |
| External Call Action | Characterized External Call Action read customer from DB |
| Id | _dVT7sIMfEeqsjtSI0WzVvA |
| Literal | |
| Parameter | Parameter customerID |

**Figure 2.15:** Shows an example for a characterized SEFF in PCM.

Figure 2.15 shows the expanded characterized SEFF *checkout* of the *OnlineShop* component. Inside the SEFF there are a start action, a stop action, an internal action, three *Characterized External Call Actions* and three *Seff Return Assignments*. The data flow extension allows to append parameter assignments to characterized external call actions as shown in the expanded *read customer from DB* action. These parameter assignments are used to define which data types flow as input from the SEFF into the called service. Therefore, each parameter assignment has a left- and right-hand side. While the left-hand side (the first child value in a parameter assignment) defines which available parameter from the SEFF flows into the parameter of the called service, the right-hand side (the second child value in a parameter assignment) defines an additional predicate that is applied to the left-hand side. This can be, for example, a *true*, an *and* or an *or*. A true simply supports the definition from the left-hand side. An example for this is shown in the highlighted parameter assignment in Figure 2.15. It shows that for the parameter *customerID* of the service call *read customer from DB* a primitive data type INT is passed. An *and* defines a compositional flow of data meaning that several data types from the SEFF are combined and then passed to the called service. An *or* defines an exclusive flow meaning that depending on a condition one or another data type is passed to the called service. With these variations of the right-hand side complex internal computations of data can be represented which cannot be reflected by a simple analysis of invoked signatures. The *Seff Return Assignment* defines for the current SEFF the data type that flows out as a return type. Again, the left-hand side defines the data type that flows for the return type and the right-hand side applies additional predicates to it. An example is shown in the lower part of Figure 2.15. The left-hand side defines the primitive data type INT as the return type of the SEFF and the right-hand side supports this flow with a *true*. To sum up, by defining the concrete data types that are passed by SEFFs during external calls to services and provided by SEFFs as return types complex internal data transformations such as composition and decomposition can be represented to reflect the real data flow between components and systems.

### 2.4.4 Summary

This section introduced PCM, an architectural description language for EAAs and software systems that allows to predict various software quality attributes like performance and reliability. PCM defines different roles that design four views of PCM on the IT architecture. Afterwards, an extension to PCM that allows the analysis of data flows in PCM models was introduced. The approaches of this thesis use PCM as a modeling language for EAAs. AcsALign uses ACRs provided by PAcsTract to analyze the EAA in PCM for ACR breaches. The presented data flow extension in PCM supports this analysis by identifying read and written data types of service calls of the EAA.

## 2.5 Integrated Business IT Impact Simulation

Integrated Business IT Impact Simulation (IntBIIS) [105] is an approach for integrated performance simulation of business processes and information systems. During the performance simulation IntBIIS considers the mutual impact of business processes and information systems. IntBIIS is an extension to PCM. PCM allows to design domain specific models of software architecture and is explained in Section 2.4. The core of PCM does not consider business processes. Thus, IntBIIS introduces formalisms to design parts of business processes that are relevant for performance and enrich them with IT-specific information. Both is done in the context of PCM. IntBIIS extends the event-based simulator

EventSim that is integrated in Palladio with specific properties of business processes. This allows to perform a more precise performance prediction of information systems by taking business processes and especially workload burstiness into account. In the context of this thesis, IntBIIS is extended to provide a complete set of standard business process elements. Therefore, the following paragraphs focus on the explanation of the business process elements integrated by IntBIIS into PCM.

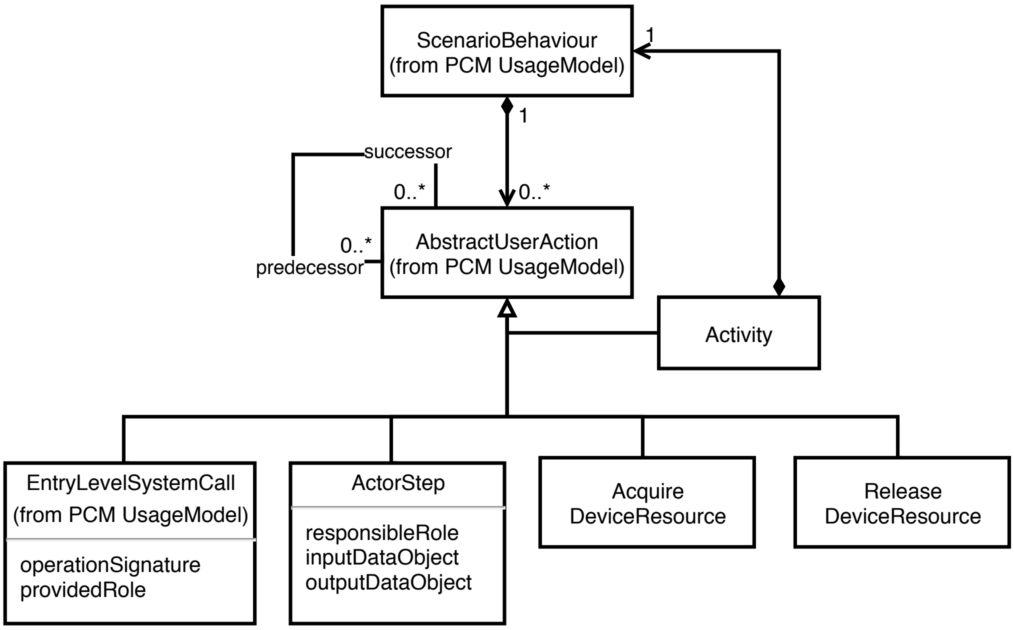


**Figure 2.16:** Shows an excerpt of the metamodel of IntBIIS regarding the integration of business processes into PCM [105].

Figure 2.16 shows the main elements of the business process model. Classes that are originally from PCM have an additional parentheses stating *from PCM UsageModel*. The *ScenarioBehaviour* bundles elements that belong to a business process. It consists of zero to infinite amount of *AbstractUserActions*. An *AbstractUserAction* can be an *EntryLevelSystemCall*, *ActorStep*, *AcquireDeviceResource*, *ReleaseDeviceResource* or an *Activity*.

- **EntryLevelSystemCall:** denotes a step in the business process that is performed by an information system. The attribute *operationSignature* specifies the service that is invoked and the attribute *providedRole* specifies the system's interface of the invoked service. Depending on the service's signature parameters may be passed and results may be returned.
- **ActorStep:** denotes a step in the business process that is performed by an employee of the organization. It has several attributes. The *responsibleRole* represents the role that fulfills the actor step. The attributes *inputDataObject* and *outputDataObject* represent the required and produced data objects during the course of the actor step.
- **AcquireDeviceResource:** Defines when a device resource is acquired by an employee or an information system.

- **ReleaseDeviceResource:** Defines when a device resource is released by an employee or an information system.
- **Activity:** is a container that allows to model nested business processes. This is required, for example, when defining branches of control flow.

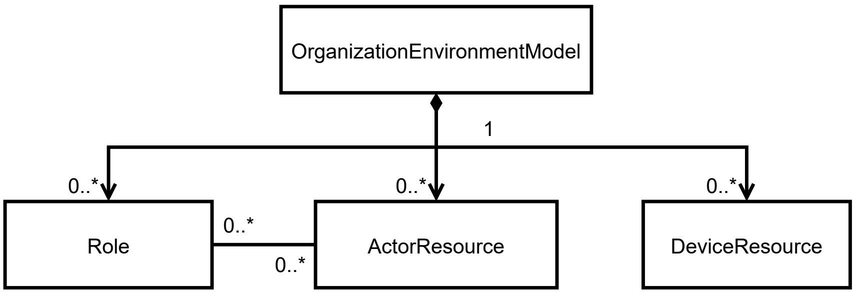


**Figure 2.17:** Shows the metamodel of IntBIIS regarding the integration of actor resources into PCM [105].

Figure 2.17 shows the metamodel for the actor resources that are used in the business processes. The organization environment model represents the organizational context for the business processes. Therefore, it defines *ActorResources*, *Roles* and *DeviceResources*.

- **ActorResource:** denotes a human resource in an organization. It provides the possibility to define a concrete employee that can be assigned to one or more roles.
- **Role:** denotes a role that bundles actor resources and is assigned to activities of business processes.
- **DeviceResource:** denotes a device or machine in the organization that can be acquired in a business process in order to fulfill some activities.

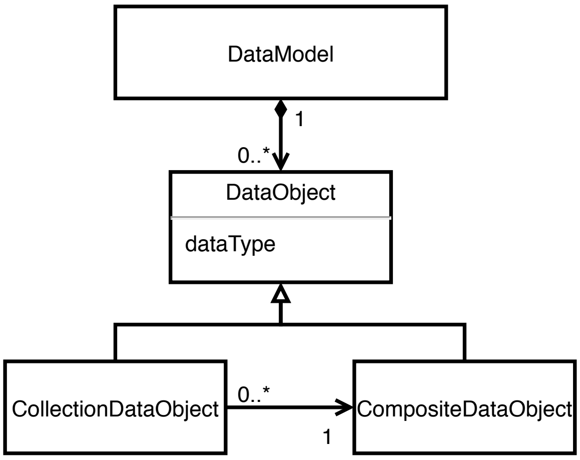


**Figure 2.18:** Shows the metamodel of IntBIIS regarding the integration of data objects into PCM [105].

Figure 2.18 shows the metamodel for the data objects that are specified in the business processes. The data model represents the data context of the business processes and defines composite and collection data objects that are used in the context of actor steps in order to denote the required and produced data objects during the course of an activity. Therefore, the data model consists of zero to infinite amount of *DataObjects*. A *DataObject* has the attribute *datatype* that connects it with a data type in the IT architecture and can be a *CollectionDataObject* or an *CompositeDataObject*.

- **CompositeDataObject:** is a single data object.
- **CollectionDataObject:** is a collection of a single data object and has a reference to it.

# 3 Approach

On the one hand, this chapter explains the automatic approach of extracting ACRs from business processes to form an initial role model for RBAC, henceforth referred to as BPMN Access Permission Extractor (BAcsTract) and Palladio Access Permission Extractor (PAcsTract). On the other hand, this chapter explains the approach to identify ACR breaches in enterprise application architectures (EAAs) by using an architectural data flow analysis, hereinafter referred to as Access Permission Architecture Aligner (AcsALign). Section 3.1 introduces the formal concepts of the approaches and shows that they are independent of the chosen language for business processes and EAAs. Section 3.2.1 defines the business process languages and boundary conditions for the input to BAcsTract, PAcsTract and AcsALign. Business processes comprise ACRs from laws and corporate regulations that are incorporated by the service design managers and compliance managers [34], as argued in Chapter 1. They are the business level of an organization. In the remainder of this thesis, I assume that ACRs incorporated in business processes by the business level are legally correct and in line with the business goals introduced in Chapter 1, since the focus of this thesis is not to identify erroneous ACRs, but to define an automated transformation of ACRs from business processes to IT level artifacts. Section 3.2.3 explains the approaches itself and thus, how ACRs are extracted from business processes to form an initial role model for RBAC and how architectural data flow constraints are generated from ACRs to identify ACR breaches in EAAs. For the extraction of ACRs from business processes there are two different implementations. BAcsTract extracts ACRs from BPMN, while PAcsTract extracts ACRs from the BPMN equivalent in PCM called IntBIIS_LP. PAcsTract is capable of extracting more information from IntBIIS_LP compared to BAcsTract, as IntBIIS_LP models is tightly coupled with EAAs (see Section 2.5 for detailed explanation). Section 3.2.4.2, describes how the extracted ACRs are transformed into architectural data flow constraints for EAAs to conduct an architectural data flow analysis and identify violated ACRs. This chapter concludes with a discussion of BAcsTract, PAcsTract and AcsALign in Section 3.3 and Section 3.4.

## 3.1 Concept

In this section, the underlying concepts for the approaches are introduced. The concepts are formal definitions. Based on this, Section 1.4 defines the approaches for specific languages. In Section 3.1.1 the concept for the extraction of ACRs from business processes is explained. It comprises the extraction of ACRs from business processes, their transformation to access permission for RBAC and the formation of a simple role hierarchy. During the extraction process an ACR mapping model is built, which is also formally described. In Section 3.1.2 the concept for identifying ACR breaches in EAA is explained. Accordingly, the ACRs are transformed into data flow constraints that are afterwards used in an architectural data flow analysis. In addition, this section explains how the ACR mapping model is extended with IT specific elements to interconnect elements of business processes with elements of RBAC and EAA.

### 3.1.1 Concept for Role Model Extraction from Business Processes

This section introduces the formal concept of extracting ACRs from business processes, their transformation into a role model for RBAC and the elevation of an ACR mapping model for ACRs. The ACR mapping model interconnects elements of business processes and RBAC. This documents design decisions regarding ACRs automatically and allows to trace them among models of business and IT. The concept of this section tackles the following problems of Section 1.2:

- **P1 Missing knowledge on IT level:** Knowledge about which business assets are critical and the required protection degree lies on the business level and thus, is missing on the IT level.
- **P2 Different terminology between business and IT level:** Several discrepancies, e.g., different terminology, domain knowledge, domain-specific models and modeling tools of the business level and IT level widen a communication gap that may lead to errors and security breaches.
- **P3 Experts needed to understand business level:** Experts are required who know the terminology and models of both, business level and IT level.
- **P4 Costly and error-prone engineering of the RBAC role model:** Role engineering is a manual, slow and complex task making it costly and error-prone.
- **P5 Missing alignment between RBAC and business level access control requirements:** Due to the manual and complex engineering of the role model, it is not well aligned with business level ACRs.
- **P8 Missing support of evolution scenarios for RBAC and enterprise application architectures:** Especially during evolution scenarios the role model is not well aligned with business level ACRs.

The knowledge about which assets are critical for business and which protection degree is appropriate lies on the business level. The business level, consisting of service design managers and compliance managers, has to incorporate IT security requirements and also privacy requirements stemming from various laws, for example, [88] and [222]. Both, IT security and privacy requirements are non-functional requirements affecting business processes as well as IT systems. A fundamental building block of both are the ACRs that have to be implemented accurately. Defining and enforcing correct ACRs is a challenging task. Furthermore, the business level uses IT security and privacy guidelines during the development of business processes [210], [117]. These business processes describe, among others, ACRs form the business level perspective. Business process guidelines as ITIL [34] and COBIT [32] that are used during business process development, also have dedicated sets of practices for the governance and management of access control. Consequently, business processes are rich of implicitly modeled ACRs. They are modeled by the business level while incorporating the various laws and guidelines. More details on this topic were illustrated in Section 1.1. During the course of this thesis, I assume that the ACRs incorporated by the business level are legally correct, as the focus of this thesis is not to identify erroneous ACRs, but the automated transformation of ACRs from business processes onto IT level artifacts.

To extract ACRs from business processes and form a role model, ACRs have to be defined from the perspective of RBAC. Therefore, RBAC is described formally at first. As explained in Section 2.2, RBAC is a prominently used access control approach due to its manifold benefits for organizations. RBAC enforces access control using roles, permissions, objects and operations comprised in a role model. The metamodel of RBAC is shown in the following Figure.

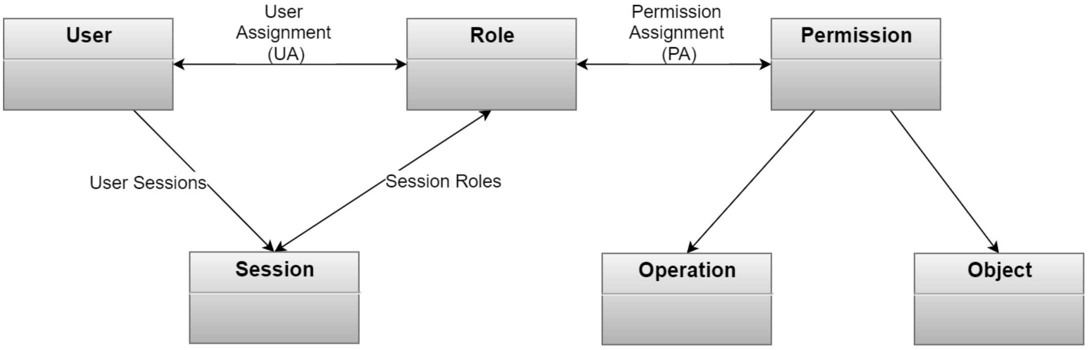


**Figure 3.1:** Metamodel of RBAC.

Figure 3.1 shows all parts of RBAC and how they are interconnected. Users are assigned to roles during the user assignment (UA). Roles comprise the actual permissions that are defined during the permissions assignment (PA). Permissions are operation and object pairs. Roles and their permissions form a role model that is developed during the role engineering. If a user wants to execute a permission in one of his roles, the user has to log in with that role into a session. Further details on RBAC can be found in Section 2.2.

In the following, the formal definition of the hierarchical RBAC is introduced which is partially based on the definitions of the NIST Standard for RBAC [2] and the first proposal of RBAC [67]. First, the six basic data elements shown in Figure 3.1 have to be defined:

- *USERS*: is the set of users.
- *ROLES*: is the set of roles.
- *PERMISSIONS*: is the set of permissions.
- *OBS*: is the set of objects.

- $OPS = \{read, write\}$: is the set of operations, that can be *read* or *write*.
- $SESSIONS \subseteq USERS \times ROLES^n$: is the set of sessions, in which a user can have activated roles.

(3.1)

The relations, where the basic data elements are semantic constructs for formulating policies, are the main concept of RBAC. According to Figure 3.1 a permission consists of operation and object pairs. Therefore, it can be defined as follows:

$$PERMISSIONS \subseteq OPS \times OBS. \tag{3.2}$$

The user assignment (*UA*) is a many-to-many mapping of users and roles indicated by the arrows in the upper left part of Figure 3.1. Each user can be assigned to one or more roles and vice versa, i.e.,

$$UA \subseteq USERS \times ROLES. \tag{3.3}$$

The permission assignment (*PA*) is a many-to-many mapping of permissions and roles indicated by the arrows in the upper right part of Figure 3.1. This relation forms the actual role model that is the core of RBAC defining the existing roles and their permissions:

$$ROLEMODEL = PA \subseteq PERMISSIONS \times ROLES. \tag{3.4}$$

Several functions can be used to get associations for the basic data elements within the relations. In order to get the users associated with a certain role, the function *assigned_users*() is used. It maps a role onto a set of assigned users, i.e.,

$$assigned\_users(r : ROLES) \subseteq \{u \in USERS \mid (u, r) \in UA\}. \tag{3.5}$$

The function *assigned_roles*() returns the roles assigned to a certain user:

$$assigned\_roles(u : USERS) \subseteq \{r \in ROLES \mid (u, r) \in UA\}. \tag{3.6}$$

Permissions assigned to a role are provided by the function *assigned_permissions*(). It maps a role onto a set of permissions, i.e.,

$$assigned\_permissions(r : ROLES) \quad := \quad \{perm \in PERMISSIONS \mid (perm, r) \in PA\}. \tag{3.7}$$

The function *session_user*() returns the user of a session:

$$session\_user(s : SESSIONS) \in USERS, \tag{3.8}$$

according to the rule:

$$\forall s \in SESSIONS, \exists u \in USERS, \exists r \in ROLES^n : \ (session\_user(s) = u \Rightarrow (u, r) = s), \tag{3.9}$$

The function *session_user* is used by the function *session_roles*() to return the activated roles of a user during a session:

$$session\_roles(s : SESSIONS) \subseteq \{r \in ROLES \mid (session\_user(s), r) \in UA\}. \tag{3.10}$$

The function *avail_session_perm*() returns the permissions available for a user in a session. These are the permissions assigned to the roles that are activated in the user's session. Formally:

$$avail\_session\_perms(s : SESSIONS) \quad := \bigcup_{r \in session\_roles(s)} assigned\_permissions(r). \tag{3.11}$$

Users may execute a permission of an activated role in a session. The function *exec*() is true if the user can execute the permission, otherwise it is false:

$$\begin{aligned} &exec(u : USERS, perm : PERMISSIONS) := \\ &\qquad \begin{cases} true & \text{if user can execute permission p,} \\ false & \text{otherwise} \end{cases} \end{aligned} \tag{3.12}$$

Three rules are required to define the *exec*() function:

1. Role assignment: For a user to execute a permission, the user needs to have at least one assigned role. Formally:

$$\begin{aligned} &\forall u \in USERS, \forall perm \in PERMISSIONS, \forall s \in SESSIONS : (exec(u, perm) = true \Rightarrow \\ &\qquad \{r \in ROLES \mid session\_user(s) = u \wedge session\_roles(s) = r\} \neq \emptyset). \end{aligned}$$

2. Role authorization: This rule ensures that the user can only activate roles to which he is assigned to. Formally:

$$\begin{aligned} &\forall u \in USERS, \forall s \in SESSIONS : \\ &\quad \{r \in ROLES \mid session\_user(s) = u \wedge session\_roles(s) = r\} \subseteq assigned\_roles(u). \end{aligned}$$

3. Permission authorization: This rule claims that a user can only execute a permission, if this permission is assigned to one of the user's activated session roles. Formally:

$$\begin{aligned} &\forall u \in USERS, \forall perm \in PERMISSIONS, \forall s \in SESSIONS : \\ &(exec(u, perm) = true \Rightarrow session\_user(s) = u \wedge perm \in avail\_session\_perms(s)). \end{aligned}$$

Finally, hierarchical RBAC has a hierarchy for roles. The hierarchy defines that roles inherit permissions from their ancestor role. Therefore, a binary relation ≥ is defined on

*ROLES*. The inheritance relation $r_1 \succeq r_2$ on role $r_1$ and $r_2$ is true, if all permissions of $r_2$ are permissions of $r_1$, and all users of $r_1$ are users of $r_2$. For this, Equation (3.11) is used, i.e.,

$$r_1 \succeq r_2 :\Leftrightarrow assigned\_permissions(r_2) \subseteq assigned\_permissions(r_1). \qquad (3.13)$$

The three previously defined functions need to be extended to comply with the role hierarchy: Equation (3.5) *authorized_users*() that maps a role onto a set of assigned users, Equation (3.6) *assigned_roles*() that maps a user onto a set of roles and Equation (3.7) *assigned_permissions*() that maps a role onto a set of permissions. Formally:

$$assigned\_users(r : ROLES) \subseteq \{u \in USERS \mid (u, r) \in UA \wedge r' \succeq r\}, \qquad (3.14)$$

and

$$assigned\_roles(u : USERS) \subseteq \{r \in ROLES \mid (u, r) \in UA \wedge r' \succeq r\}, \qquad (3.15)$$

and

$$assigned\_permissions(r : ROLES) := \\ \{perm \in PERMISSIONS \mid (perm, r) \in PA \wedge (r' \succeq r)\}. \qquad (3.16)$$

Now, that the formal definition of RBAC is complete, ACRs can be defined from the perspective of RBAC. ACRs are tuples of roles and permissions. They define which role is allowed to access which object. In conjunction with Equation (3.2), they are triples of roles, objects and operations, i.e.,

$$ACR \subseteq ROLES \times PERMISSIONS = ROLES \times (OBS \times OPS). \qquad (3.17)$$

Until now, this section defined the basic data elements of RBAC (eq. (3.1)) and their relations (eqs. (3.2) to (3.4)) shown in the metamodel of Figure 3.1. Furthermore, several functions were defined in Equations (3.8), (3.10), (3.11) and (3.14) to (3.16) to get the associations from within the relations in presence of the role hierarchy in Equation (3.13). This allowed the definition of the *exec*() function in Equation (3.12) with its three rules enabling legitimate users to execute an authorized permission. Finally, ACRs could be defined from the perspective of RBAC (eq. (3.17)).

First, to define the extraction algorithm, business processes need to be formally defined. The following definition is weak, as it only defines a rudimentary set of elements. The corresponding metamodel is shown in the following figure.

In the top of Figure 3.2 the process consisting of lanes is shown. Lanes group responsibilities together for an employee. The middle of Figure 3.2 illustrates the main parts of a lane: pools, activities, events, gateways. The pool is the organizational department to which the lane as well as the employee belongs. Activities are the daily duties that need to be fulfilled by an employee during the process of a lane. Events denote something that happens, after that an activity is triggered. Gateways allow to fork and merge flows depending on a condition. The last three elements are connected by flow transitions representing the flow

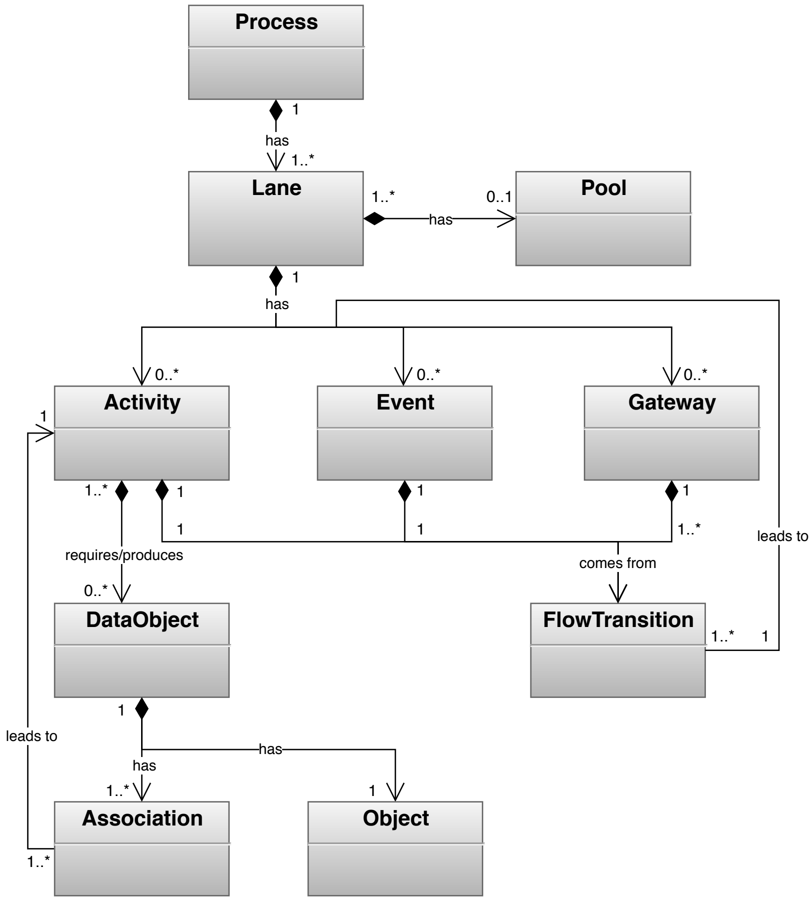


**Figure 3.2:** Simplified metamodel of business processes that is inspired by BPMN [5].

of actions. A flow transitions have start and end points. In the lower left part of Figure 3.2 the data object is shown with its association. It is connected to an activity. Depending on the association (in or out) the data object is an input data object or an output data object. Input data objects are required during an activity to complete it. Output data objects are produced during the activity and can be inputs to other activities. Details on the business process language BPMN that may help to understand this section can be found in Section 2.1.

Out of the presented elements of the metamodel in Figure 3.2, two elements are necessary for the concept of the extraction algorithm: The concept of lanes, that groups responsibilities together and the concept of data objects, that represents information that flows in the business process.

In the following, a simplified formal definition of business processes is introduced according to the metamodel of Figure 3.2:

- *POOLS*: is the set of pools.
- *FLOWTRANSITIONS*: is the set of flow transitions.
- *ACTIVITIES*: is the set of activities.

- *EVENTS*: is the set of events.
- *GATEWAYS*: is the set of gateways.
- *DATAOBJECTS* ⊆ *OBJECTS* × *ASSOCIATIONS*: is the set of data objects consisting of object and association pairs.

(3.18)

On the basis of the previously defined elements, lanes and processes are defined as follows: *LANES* is a set of lanes consisting of a pool, a set of activities, events and Gateways, *PROCESS* is a process, consisting of one or more lanes, and *PROCESSES* is a set of processes. Formally:

$$\begin{aligned}&LANES \subseteq POOLS \times \{ACT \mid ACT \subseteq ACTIVITIES\} \times \\ &\qquad \{EV \mid EV \subseteq EVENTS\} \times \{GAT \mid GAT \subseteq GATEWAYS\},\end{aligned} \tag{3.19}$$

and

$$PROCESS \subseteq LANES, \tag{3.20}$$

and

$$PROCESSES \subseteq \{PR \mid PR \subseteq PROCESS\}. \tag{3.21}$$

There are several special cases that define certain sets: *ACTIVITIESWITHDATAOBJECTS* defines a set of activities that have data objects, *LANESWITHDATAOBJECTS* defines a set of lanes in which each lane has at least one activity with a data object and *CLOSEDLANES* defines a set of lanes without any activities. The latter one, can be specified in some business process languages to express external organizations, which are relevant for the process, but which's activities are not known. Formally:

$$ACTIVITIESWITHDATAOBJECTS \subseteq \{a \in ACTIVITIES \mid a \text{ has an DATAOBJECT}\}, \tag{3.22}$$

and

$$\begin{aligned}&LANESWITHDATAOBJECTS \subseteq \{l \in LANES \mid \\ &\qquad \exists l.a \in ACTIVITIES \Rightarrow l.a \in ACTIVITIESWITHDATAOBJECTS\},\end{aligned} \tag{3.23}$$

and

$$CLOSEDLANES \subseteq \{l \in LANES \mid l \text{ has no activities}\}. \tag{3.24}$$

Along with these definitions two helper functions will be required to get activities and data objects from a lanes:

$$get\_lane\_activities(pr : PROCESS) \subseteq ACTIVITIES, \tag{3.25}$$

and

$$get\_lane\_dataobjects(pr : PROCESS) \subseteq DATAOBJECTS. \tag{3.26}$$

Now that the business process, RBAC and ACR are formally defined, the extraction algorithm of the role model can be formalized. The extraction algorithm consumes a set of processes and transforms them into a set of ACRs, as defined in Equation (3.17). As ACRs are defined from the RBAC perspective, they are in the form of tuples of roles, objects and operations. These tuples can be grouped together to form a role model containing roles and their permissions as defined in Equation (3.4). In the following, these transformations are formally described.

During the transformation of processes to a role model an ACR mapping model is build containing the actual ACRs. The ACR mapping model aligns elements of business processes with elements of RBAC. Besides, it is beneficial for the understanding of design decisions. For example, to understand how a certain access permission in RBAC is emerged, it is possible to trace the corresponding business process, role and activity in the ACR mapping model from which the access permission is originating. This allows to understand the border conditions from which the access permission was derived and thus, to follow the design decisions. The ACR mapping model is similar to a documentation of design decisions regarding ACRs that is built along the way. In a later step the actual role model is derived from the ACR mapping model. First, to better understand the ACR mapping model and how it is built, it will be explained in more detail.

For the beginning, a mapping between relevant elements of business processes and RBAC is defined that serves the purpose of extracting ACRs:

- **Lanes and pools**: Lanes and pools of business processes (eqs. (3.18) and (3.19)) can be mapped to roles of RBAC (eq. (3.1)), as both lanes and roles group responsibilities together for an employee. While lanes group activities for which an employee needs a certain amount of access permissions, roles group directly the access permissions together. The pool describes the organizational department and thus, specializes roles so that same roles in different departments can be differentiated from each other. Function *String*() extracts the name of an element. The formalized function *convertLaneToRole*() converts a lane into a role and is taking Equations (3.1), (3.18), (3.19), (3.23) and (3.24) into account, i.e.,

$$convertLaneToRole(l : LANES) \in ROLES, \tag{3.27}$$

  and the rule:

$$\begin{gathered}\forall l \in LANES, \forall l.p \in POOLS, \exists! r_l \in ROLES : \\ l \notin CLOSEDLANES \wedge l \in LANESWITHDATAOBJECT \Rightarrow r_l = l.p \circ String(l).\end{gathered}$$

- **Data objects**: Data objects of business processes (eq. (3.18)) can be mapped to permissions of RBAC (eq. (3.2)). While data objects of business processes consist of object and association pairs, permissions of RBAC consist of object and operation pairs. Obviously, the object is equivalent in both. The association of the business process data object describes an input or output, which can be seen as a read and write operation. For the input of an object to an activity the object needs to be read, while for the output of an object from an activity the object needs to be written. The formalized function *convertDataObjectToPermission*() taking Equations (3.2) and (3.18) into account, looks as follows:

$$convertDataObjectToPermission(do : DATAOBJECTS) \in PERMISSIONS, \quad (3.28)$$

and the rule:

$$\forall(do.ob, do.ass) \in DATAOBJECTS, \exists!(perm.op, perm.ob) \in PERMISSIONS : \\ perm.ob = do.ob \wedge perm.op = do.ass.$$

- **Processes and activities**: The process and activities of business processes are the holders of the actual lanes and data objects. They do not have an equivalent in RBAC but can be seen as jobs and tasks that need to be done in order to fulfill the daily duty of an employee. Thus, they serve as an intermediate level between the lanes and data objects, interconnecting all together.

On the basis of the introduced mapping in Equations (3.27) and (3.28) the ACR mapping model is defined as a tuple of role, process, activity and permission. Formally:

$$ACRMAPPINGMODEL \subseteq ROLES \times PROCESSES \times ACTIVITIES \times PERMISSIONS. \quad (3.29)$$

Several functions are needed to fill the ACR mapping model. First, roles are extracted from business processes with the help of Equations (3.1), (3.19), (3.20) and (3.27):

$$extract\_roles(pr : PROCESS) := \\ \{r \in ROLES \mid \{l \in LANES\} = pr \wedge convertLaneToRole(l) = r\}. \quad (3.30)$$

In the second step, the function *extract_process*() returns true if the process is extracted. Only those processes, that have roles inside, are relevant. Therefore, again Equation (3.27) is used, i.e.,

$$extract\_process(pr : PROCESS) := \begin{cases} true & \text{if } extract\_roles(pr) \neq \emptyset, \\ false & \text{otherwise} \end{cases} \quad (3.31)$$

As a third step, the activities are extracted. Only those activities need to be extracted that have an associated data object. Obviously, lanes without data objects can be skipped, as their activities do not have any associated data objects. Formally:

$$extract\_activities(pr : PROCESS) := \\ \{a \in ACTIVITIES \mid \{l \in LANES\} = pr \wedge convertLaneToRole(l) \in extract\_roles(pr) \wedge \\ a \in ACTIVITIESWITHDATAOBJECT \wedge a \in get\_lane\_activities(l)\} \quad (3.32)$$

Fourth, the permissions are extracted. The previously defined mapping function in Equation (3.28) is used. Formally:

$$\begin{aligned}&\mathit{extract\_permissions}(pr : \mathit{PROCESS}) :=\\ &\qquad \{perm \in \mathit{PERMISSIONS} \mid \{l \in \mathit{LANES}\} = pr \land do \in \mathit{DATAOBJECTS} \land\\ &\quad do \in \mathit{get\_lane\_dataobjects}(l) \land perm = \mathit{convertDataObjectToPermission}(do)\}. \end{aligned} \tag{3.33}$$

Finally, a rule is required to define when a tuple of role, process, activity and permission is put into the ACR mapping model. The four previously defined functions in Equations (3.30) to (3.33) are required. Formally:

$$\begin{aligned}&\forall r_{pr} \in \mathit{ROLES}, \forall a_{pr} \in \mathit{ACTIVITIES}, \forall perm_{pr} \in \mathit{PERMISSIONS}, \forall pr \in \mathit{PROCESSES} :\\ &\quad (r_{pr} \in \mathit{extract\_roles}(pr) \land \mathit{extract\_process}(pr) = \mathit{true} \land a_{pr} \in \mathit{extract\_activities}(pr) \land\\ &\quad perm_{pr} \in \mathit{extract\_permissions}(pr)) \Rightarrow (r_{pr}, pr, a_{pr}, perm_{pr}) \in \mathit{ACRMAPPINGMODEL}.\end{aligned} \tag{3.34}$$

Now that the mapping between business process elements and RBAC elements as well as the ACR mapping model is defined, the ACRs (eq. (3.17)) need to be extended by a rule. This rule establishes an alignment of ACRs between business processes and RBAC. Therefore, the extraction functions of Equations (3.30) and (3.33) are reused. Formally:

$$\begin{aligned}&\forall r_{pr} \in \mathit{ROLES}, \forall pr \in \mathit{PROCESSES}, \forall perm_{pr} \in \mathit{PERMISSIONS} :\\ &\qquad r_{pr} \in \mathit{extract\_roles}(pr) \land perm_{pr} \in \mathit{extract\_permissions}(pr) \Rightarrow\\ &\qquad\qquad (r_{pr}, perm_{pr}) \in \mathit{ACR}.\end{aligned} \tag{3.35}$$

Furthermore, after the alignment of ACRs (eq. (3.35)), the role model (eq. (3.4)) can be specialized, establishing an alignment between business processes and RBAC, i.e.,

$$\forall r \in \mathit{ROLES}, \forall perm \in \mathit{PERMISSIONS} : (r, perm) \in \mathit{ACR} \Rightarrow (r, perm) \in \mathit{PA}. \tag{3.36}$$

In Equations (3.1) to (3.12) RBAC was formally defined with its required functions. Then in Equations (3.17) to (3.16) a hierarchy for RBAC was specified. Afterwards, in Equations (3.18) to (3.28) BPMN was formalized as well as the mapping between business process and RBAC elements that are relevant for the role model extraction. At the same time, the Equations (3.29) to (3.34) introduced the ACR mapping model along with its extraction functions. Their purpose is to extract information about ACRs from business processes and store them in the ACR mapping model. Later, this alignment allows to track design decisions across both models of business and IT. ACRs were first defined from the perspective of RBAC in Equation (3.17) and afterwards, they were aligned with business processes in Equation (3.35). Finally, the extraction of the role model from the information in the ACR mapping model was defined in Equation (3.36).

As a last step, a simple hierarchy needs to be established on top of the extracted role model. Therefore, the binary relation $\succeq$, introduced in Equation (3.13), is used along with the definition of the role model in Equation (3.36). Formally:

$$\begin{aligned}\forall r_1, r_2 \in ROLES, \forall perm_{r_1}, perm_{r_2} \in PERMISSIONS :&\\ (r_1, perm_{r_1}), (r_2, perm_{r_2}) \in PA \land&\\ assigned\_permissions(r_2) \subseteq assigned\_permissions(r_1) \Rightarrow r_1 \succeq r_2.&\end{aligned} \tag{3.37}$$

The Equations (3.1) to (3.37) formally define the extraction of ACRs from business processes and the transformation to an ACR mapping model, from which the actual role model is extracted. In the beginning of this section, the problems from Section 1.2 were introduced, that are tackled by this part of the approach. The presented concept provides the following contributions (see also Section 1.6) to solve the previously mentioned problems:

- **C1 Extract business level access control requirements from business processes:** With the extraction of ACRs from business processes, the information about critical assets and access permissions is transferred from the business level to the IT level (tackles problem **P1**). Additionally, this helps closing the gap between the business level and IT level, as the demands of the business level are better understood (tackles problem **P2**). Through the automatic extraction of business knowledge, the necessity for experts and dependencies on their skills is reduced (tackles problem **P3**).
- **C2 Establish an ACR mapping model for access control requirements between business processes, RBAC and the enterprise application architecture:** By interconnecting access control relevant elements of business processes with elements of RBAC in the ACR mapping model a confirmability of design decisions is established. This closes the communication gap between the business level and IT level and supports the business and IT level in understanding design decisions in models outside of their expertise (tackles problem **P2** and **P3**). As elements of business processes are interconnected with elements of RBAC, an alignment between these models is established (tackles problem **P5**). This is also the case in evolution scenarios of either of each of models, as the approach can be applied during the evolution scenario without noticeable effort (tackles problem **P8**).
- **C3 Build initial RBAC role model from extracted access control requirements:** The extraction of the role model from business processes is automatic. This reduces the needed effort of experts to go through the vast amount of business processes during the role engineering processes (tackles problem **P3**). Additionally, this eases the difficulties for organizations to migrate to RBAC systems, as the overall role engineering process is accelerated and less error-prone due to the automation (tackles problem **P4**). The extraction processes allow an alignment of the role model and the business level ACRs (tackles problem **P5**). Considering evolution scenarios, a faster and better support is provided due to the quick applicable extraction, providing the opportunity to react immediately to changes (tackles problem **P8**).

### 3.1.2 Concept for Identification of Access Control Requirement Breaches in Enterprise Application Architectures

This section introduces the formal concept for identifying ACR breaches in EAAs and the extension of the ACR mapping model with EAA elements. The extended ACR mapping model interconnects elements of business processes, RBAC and EAA with regard to ACRs. The concept of this section tackles the following problems from Section 1.2:

- **P2 Different terminology between business and IT level:** Several discrepancies like different terminology, domain knowledge, domain-specific models and modeling tools of the business level and IT level widen a communication gap that may lead to errors and security breaches.
- **P3 Experts needed to understand business level:** To analyze the EAA for correctness experts are required who understand terminology and models of both, business level and IT level.
- **P6 Complex and error-prone designing of the enterprise application architecture:** Logical and design mistakes are done during the design of the EAA for various reasons as misunderstanding correct requirements, complexity of interrelating models and a widening communication gap due to different terminology.
- **P7 Missing alignment between enterprise application architecture and business level access control requirements:** Due to the complexity of engineering the EAA it is not well aligned with business level ACRs.
- **P8 Missing support of evolution scenarios for RBAC and enterprise application architectures:** Especially during evolution scenarios the EAA becomes misaligned with business level ACRs, as new logical and design mistakes may be introduced.

To define the algorithm for the identification of ACR breaches the EAA needs to be defined formally. The following definition of an EAA is weak, as it only defines a rudimentary set of elements. The corresponding metamodel is shown in Figure 3.3.

The EAA consists of one or more systems. Each system provides several interfaces. An interface itself consists of services that can have data types as parameters and return values. Void is not considered as a data type.

In the following, the basic elements of the EAA metamodel shown in Figure 3.3 are defined:

- *SYSTEMS*: is the set of systems.
- *INTERFACES*: is the set of interfaces.
- *SERVICES*: is the set of services.
- *DATATYPES*: is the set of data types.

(3.38)

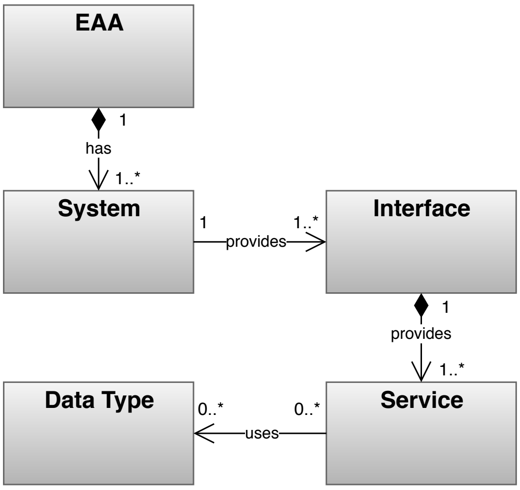


**Figure 3.3:** Simplified metamodel of an enterprise application architecture inspired by the UML metamodel [6].

Furthermore, two functions are required to get associations between the basic elements. The function *interface_of_service*() is used to get the interface of a service. The function *system_of_interface*() is used to get the system of an interface. Formally:

$$interface\_of\_service(sc : SERVICES) \in INTERFACES, \tag{3.39}$$

and

$$system\_of\_interface(i : INTERFACES) \in SYSTEMS. \tag{3.40}$$

I assume that two functions are available that provide data types that are read and written by the actual data flows of a service. The function *service_read*() provides the data types that are read during a service invocation and the function *service_write*() provides the data types that are written during a service invocation.

$$service\_read(sc : SERVICES) \subseteq DATATYPES, \tag{3.41}$$

and

$$service\_write(sc : SERVICES) \subseteq DATATYPES. \tag{3.42}$$

In order to identify ACR breaches in the EAA a set of ACRs is required as input according to the definition in Equation (3.17):

$$ACR \subseteq ROLES \times PERMISSIONS = ROLES \times (OBS \times OPS). \tag{3.43}$$

In the following we assume that ACRs are provided, for example, by one of the following alternatives: manually defined, extracted from an access control system, e.g., RBAC, extracted from an access policy document or by the previously introduced concept in Section 3.1.1 that extracts ACRs from business processes.

Furthermore, two mapping functions link a) data objects from business processes to data types of the EAA and b) activities of business processes to services of the EAA. Therefore, the formal definition of business processes from Equation (3.18) is used. The function *mapObject*() maps each data object of a business processes to strictly one data type, i.e.,

$$mapObject(ob : OBS) \in DATATYPES, \tag{3.44}$$

and the corresponding rule:

$$\forall r \in ROLES, \forall ob \in OBS, \forall op \in OPS, \exists dt \in DATATYPES : \\ ((r, ob, op) \in ACR \Rightarrow dt = mapObject(ob)). \tag{3.45}$$

The function *mapActivity*() maps each activity of a business processes to a set of services that are invoked during the completion of the activity, i.e.,

$$mapActivity(a : ACTIVITIES) \subseteq SERVICES. \tag{3.46}$$

So far, the basic elements of the EAA were defined along with two helper functions providing the actual read and written data types of service invocations. Afterwards, the input for the analysis in form of ACRs as well as a mapping between certain elements of business processes and the EAA were formalized. In the following, the data flow constraints and the analysis itself will be defined.

To generate data flow constraints required by the analysis to identify data flow breaches the formal definition of business processes from Equation (3.18), the *convertLaneToRole*() function from Equation (3.27) to convert lanes into roles and the mapping functions from Equations (3.44) to (3.46) are used. In addition, the inverse image function $^{-1}$ is used. The formal function *generate_dfconstraints*() generates data flow constraints in form of:

$$DFCONSTRAINTS \subseteq \{SC \mid SC \subseteq SERVICES\} \times ROLES \times DATATYPES \times OPS, \tag{3.47}$$

by consuming ACRs as follows

$$generate\_dfconstraints((r_{acr}, ob_{acr}, op_{acr}) : ACR) := \\ \{SC, r_{acr}, dt, op_{acr}) \in DFCONSTRAINTS \mid \exists l \in convertLaneToRole(^{-1}\{r_{acr}\})\wedge \\ \exists l.a \in get\_lane\_activities(l) \wedge SC = mapActivity(l.a) \wedge dt = mapObject(ob_{acr}). \tag{3.48}$$

The analysis consumes a generated data flow constraint along with the corresponding service. The function is true if the service satisfies the data flow constraint, otherwise it is false. False means that the service produces a data flow that breaches the data flow constraint and thus, indicates an ACR breach. Formally:

$$analyse\_eaa((SC_{dfc}, r_{dfc}, dt_{dfc}, op_{dfc}) : DFCONSTRAINTS, sc : SERVICES) := \\ \begin{cases} true & \text{if } (sc \in SC_{dfc} \wedge op_{dfc} = read \wedge dt_{dfc} \subseteq service\_read(sc))\vee \\ & \qquad (sc \in SC_{dfc} \wedge op_{dfc} = write \wedge dt_{dfc} \subseteq service\_write(sc), \\ false & \text{otherwise.} \end{cases} \tag{3.49}$$

On the basis of the mapping function in Equations (3.44) to (3.46), the ACR mapping model can be extended with elements from the EAA. The ACR mapping model defined in Equation (3.29) looked as follows:

$$\mathit{ACRMAPPINGMODEL} \subseteq \mathit{ROLES} \times \mathit{PROCESSES} \times \mathit{ACTIVITIES} \times \mathit{PERMISSIONS}.$$

By extending it with EAA elements it becomes a tuple of role, process, activity, permission, system, interface and service:

$$\begin{aligned} \mathit{ACRMAPPINGMODEL} \subseteq \mathit{ROLES} \times \mathit{PROCESSES} \times \mathit{ACTIVITIES} \times \mathit{PERMISSIONS} \times \\ \mathit{SYSTEMS} \times \mathit{INTERFACES} \times \mathit{SERVICES}. \end{aligned} \tag{3.50}$$

The previously defined functions to fill the ACR mapping model from Equations (3.30) to (3.33) need to be extended by an extraction function for the system interface and service, i.e.,

$$\begin{aligned} &\mathit{extract\_eaaTrace}((\mathit{SC}_{dfc}, r_{dfc}, dt_{dfc}, op_{dfc}) : \mathit{DFCONSTRAINTS}) := \\ &\quad \{(sys, i, sc) \in \mathit{SYSTEMS} \times \mathit{INTERFACES} \times \mathit{SERVICES} \mid \\ &\quad sc \in \mathit{SC}_{dfc} \land i = \mathit{interface\_of\_service}(sc) \land sys = \mathit{system\_of\_interface}(i)\}. \end{aligned} \tag{3.51}$$

Finally, the rule from Equation (3.34) that defines when a tuple is put into the ACR mapping model needs to be redefined with the use of Equations (3.46) and (3.51). The ACR mapping model then consists of elements from business processes (lane, process, activity and permission), RBAC (role and permission) and the EAA (system, interface and service) interconnecting the three models with regard to ACRs. Formally:

$$\begin{gathered} \forall r_{pr} \in \mathit{ROLES}, \forall a_{pr} \in \mathit{ACTIVITIES}, \forall perm_{pr} \in \mathit{PERMISSIONS}, \forall sys_{dfc} \in \mathit{SYSTEMS}, \\ \forall i_{dfc} \in \mathit{INTERFACES}, \forall sc_{dfc} \in \mathit{SERVICES}, \forall pr \in \mathit{PROCESSES}, \forall dfc \in \mathit{DFCONSTRAINTS} : \\ (r_{pr} \in \mathit{extract\_roles}(pr) \land \mathit{extract\_process}(pr) = \mathit{true} \land a_{pr} \in \mathit{extract\_activities}(pr) \land \\ perm_{pr} \in \mathit{extract\_permissions}(pr) \land (sys_{dfc}, i_{dfc}, sc_{dfc}) \in \mathit{extract\_eaaTrace}(dfc) \land \\ \mathit{mapActivity}(a_{pr}) = sc_{dfc}) \Rightarrow (r_{pr}, pr, a_{pr}, perm_{pr}, sys_{dfc}, i_{dfc}, sc_{dfc}) \in \mathit{ACRMAPPINGMODEL}. \end{gathered} \tag{3.52}$$

The innovation regarding the presented concept is that it can be combined with the concept from Section 3.1.1 in order to use the extracted ACRs from business processes for generating data flow constraints to identify ACR breaches in the EAA. To sum up, Equations (3.38) to (3.39) formally define the EAA, Equations (3.41) to (3.42) define the data types that are read and written during a service, Equations (3.44) to (3.46) provide a mapping between business processes and the EAA, Equations (3.47) to (3.49) define the data flow constraints and the actual analysis to identify ACR breaches in the EAA and finally, Equations (3.51) to (3.52) formally extend the ACR mapping model. In the beginning of this section, the problems from Section 1.2 tackled by this part of the approach were introduced. The presented concept provides the following contributions (see also Section 1.6) to solve the previously mentioned problems:

- **C2 Establish an ACR mapping model for access control requirements between business processes, RBAC and the enterprise application architecture:** By extending the previously defined ACR mapping model to interconnect RBAC and business processes with elements from the EAA, the enterprise architect can comprehend previously made design decisions. The ACR mapping model enables him to trace ACR breaches of an invoked service back to the violated RBAC permission and the violated activity of a business process. This closes the communication gap between the business level and IT level and supports both in understanding design decisions in models outside of their expertise (tackles problem **P2** and **P3**). As elements of the EAA are interconnected with elements of business processes and RBAC, an alignment between these models is established (tackles problem **P7**). This is also the case during evolution scenarios of any of the models, as the approach can be applied during the evolution scenario to align the models without noticeable effort (tackles problem **P8**).
- **C4 Generate architectural data flow constraints for data flow analysis in the enterprise application architecture:** Information about ACRs from the business level is transformed to data flow constraints on the IT level closing the gap between both (tackles problem **P2**). With the generation of data flow constraints and the subsequent analysis on the EAA to identify ACR breaches an alignment of the EAA with business processes and RBAC, with regard to ACRs is established (tackles problem **P7**). After the enterprise architect has resolved the mistakes, which had led to ACR breaches, the EAA becomes more secure (tackles problem **P6**). Due to the identification of ACR breaches and the provided traceability information for the resolution of the breaches, no experts are required to manually check the EAA for alignment with the ACRs (tackles problem **P3**).

## 3.2 Refinement for BPMN and PCM

This section describes the realization of the concepts presented in Section 3.1. It explains the approaches BAcsTract, PAcsTract and AcsALign in detail. The first section, Section 3.2.1, defines the business process languages, EAA language and boundary conditions for the input data of each approach. Afterwards, Section 3.2.3 explains how the concept of Section 3.1.1 is realized for the business process language BPMN and for IntBIIS_LP, the BPMN equivalent in PCM. First, this section introduces the approach BAcsTract, which extracts ACRs from BPMN. Second, this section introduces PAcsTract, which extracts ACRs from IntBIIS_LP. In addition, the ACR mapping model is introduced. It is a fundamental building block of the extraction process and it is used to track and document design decisions as well as understand mutual dependencies between business and IT models. The last section, Section 3.2.4.2, explains how the concept of Section 3.1.2 is realized for the EAA language PCM. The approach generates architectural data flow constraints from the previously extracted ACRs to identify ACR breaches in the EAA. The section also describes how the ACR mapping model from Section 3.2.3.1 is extended to interconnect it with IT specific elements from the EAA. This extended ACR mapping model allows additionally to track design decisions over EAA models, to document these interconnections and to establish a better comprehensibility about mutual dependencies between business processes, RBAC and EAA in terms of ACRs.

With the extraction of ACRs from business processes the approaches align artifacts of business level and IT level in terms of ACRs. Organizations can utilize the approach, for example, to establishing RBAC, to check whether their RBAC role model is compliant with the business processes or to update their role model in evolution scenarios of business processes and access permissions. Various evolution scenarios in terms of EAA and business processes are supported. More details on this high-level process of utilizing BAcsTract, PAcsTract and AcsALign will be introduced in Chapter 4.

### 3.2.1 Knowledge Base

Knowledge base is a circumscription for the input data for an approach. From the perspective of the approach this input data is the initial knowledge it gets as input. Using the input data, the approach computes a certain output. This section defines the business process and EAA languages, as well as the boundary conditions for the input data on which the approaches BAcsTract, PAcsTract and AcsALign operate. Section 3.2.1.1 describes how the business process language BPMN serves BAcsTract as input. InSection 3.2.1.2 the BPMN equivalent IntBIIS_LP of PCM is introduced as input data for PAcsTract. Finally, Section 3.2.1.3 elaborates on the various inputs for AcsALign.

#### 3.2.1.1 Input for BPMN Access Permission Extractor

Section 3.1.1 introduced the formal concept of the extraction of ACRs from business processes. BAcsTract realizes this concept. This section introduces details about the input data to BAcsTract.

The business process language BPMN [5], explained in more detail in Section 2.1, is a semi-formal notation and is the de facto standard language for business processes [23, 213]. Due to this reason it is used widely across organizations of all kind. This drove the decision to choose BPMN as the input language for business processes. There are numerous of other business process languages [158], for example, Petri nets [170, 187]. Petri nets provide a formalized view of processes and focus on analyzability. This business process language could also serve as the business process language for the input to BAcsTract. Besides, a transformation between both languages exist. This allows to transform Petri net models to BPMN models. A major focus of this thesis is to provide approaches that need minimal to no adjustments or extensions to the domain specific models used in organizations. Thus, the proposed approaches should impose little to no overhead and additional expertise to use them. This makes BPMN well suited, as it is prominently used across organizations.

There are many approaches extending BPMN. These extensions enrich BPMN with additional elements, for example, to model security aspects. I conducted a systematic literature review and an analysis of such approaches in [24]. After extensive research for BPMN extensions, some examples are [49, 146, 120, 169, 168, 120, 194, 28, 161, 29, 195, 190, 24], the decision was to stay with plain BPMN for several reasons:

1. **Utilization of extensions:** None of the BPMN extensions are broadly accepted and thus, far less organizations use them compared to plain BPMN. If the approach would operate on a specific BPMN extension a far smaller number of organizations could use the approach without additional effort. This is also the reason why the development of a new BPMN extension was discarded because the acceptance of the extension among organizations would be even smaller.

2. **Modeling effort:** If a BPMN extension is used, organizations which do not use the specific extension on a normal base would have to invest additional effort to extend their models. This modeling effort poses an undesired hurdle that needs to be overcome. Due to the steady evolution of business processes, this additional effort may become very costly and time consuming and thus, should be avoided if possible.

3. **Usability of approach:** If the proposed approaches work on plain BPMN, a major benefit would be that the approaches also work on all BPMN models with extensions. This makes the approach usable for a wider range of organizations. Another benefit is that many organizations, especially big organizations, already have their business processes modeled in BPMN. Hence, they can use the approach directly without additional overhead and expertise.

In Section 3.1.1 a formal definition of business processes was given. Along with this weak definition, the necessary elements for the concept of the extraction algorithm were explained. Consequently, for the extraction of ACRs by BAcsTract the BPMN models require the appropriate lanes for the organization and their respective data objects. In order to build the ACR mapping model, the data objects need to be connected with their activities. All these elements are fundamental in BPMN.

The previous paragraphs explained that the approach operates on plain BPMN business processes in order to impose only little additional effort for organizations and allow them to use the approach directly. This is achieved by reusing already existing models of business processes that have to be defined anyway. Hence, the input for BAcsTract consists of all BPMNs of an organization. Ideally, the processes encompass all the various departments in the organization. This provides a comprehensive picture of the ongoing work done by employees on a daily basis. Operating on this knowledge base BAcsTract will extract the role model and establish the ACR mapping model fitting the needs of the organization.

#### 3.2.1.2 Input for Palladio Access Permission Extractor

The formal concept of the extraction of ACRs from business processes was introduced in Section 3.1.1. Along with BPMRME, PAcsTract realizes this concept. This section introduces details about the input data for PAcsTract.

PCM (Section 2.4) and IntBIIS (Section 2.5) are chosen as the EAA and business process languages because they tightly interconnect business processes and the EAA. In addition, the realization of the concept with IntBIIS shows that the concept of Section 3.1.1 is realizable with various business process languages. With PCM IT architectures and in particular EAA are modeled. IntBIIS, which is part of PCM, is a language to model business processes. This tight interconnection of the business and IT sector allows to make deeper research in the field of model alignment of business and IT. A major difference between BPMN and IntBIIS is that IntBIIS models have an additional technical aspect of business processes. Here, service calls are modeled that are triggered by employees during their activities in a process. From a research point of view these facts make PCM and IntBIIS interesting for the particular research topics of this thesis.

Section 3.1.1 described formal elements that are required in business processes to apply the concept of role model extraction. These elements are lanes, data objects and activities and are shown in Table 3.1.

**Table 3.1:** Comparison of elements in IntBIIS required for the concept of role model extraction.

| Concept Role Model Extraction | IntBIIS |
|---|---|
| Lane | Responsible Role |
| Data Object | Composite/Collection Data Object |
| Activity | Actor Step |

In IntBIIS lanes are modeled by *responsible roles*. *Responsible roles* are defined in the organization environment model and are attributes of *actor steps*. *Actor steps* are the actual activities. Data objects are defined in the data model as *composite/collection data objects* and are specified as input data object and output data object attributes in *actor steps*. Hence, data objects are interconnected with their activities. The requirements to apply the concept of role model extraction are fulfilled by IntBIIS. Further details on IntBIIS can be found in Section 2.5.

IntBIIS was initially intended for performance prediction and thus, models only the required parts of business process elements. Due to this reason IntBIIS is extended during the course of this thesis to model a minimum amount of business process elements. Therefore, the following subsection analyzes IntBIIS for integrated business process elements to identify which elements are missing. Afterwards, Section 3.2.1.2 introduces the extension IntBIIS_LP. It extends IntBIIS by the missing elements in order to make it complete in terms of business process modeling.

### Integrated Business IT Impact Simulation (IntBIIS)

IntBIIS is a part of PCM and was designed to reflect mutual impact of performance between business processes and IT architecture. Section 2.5 introduces IntBIIS in more detail. In this section, IntBIIS is analyzed in terms of business process elements that can and cannot be modeled.

Here a comparative analysis of BPMN standard elements and IntBIIS elements is done. A detailed analysis of this was done in the thesis of Tobias Knopf [137], that was supervised by me. Table 3.2 summarizes the results.

**Table 3.2:** Comparative analysis of elements in BPMN and IntBIIS.

| BPMN | IntBIIS |
|---|---|
| Business Process | Usage Scenario |
| **Lane** | Role, Attribute: Responsible Role, - |
| **Pool** | - |
| Subprocess | Scenario Behavior |
| Start Event | Start |
| Stop Event | Stop |
| Activity | Actor Step, System Step |
| Gateways | Branch |
| Sequence flow | Attribute: Successor, Predecessor |
| Data Object | Composite/Collection Data Object |
| **Association** | Attribute: Input/Output Data Object, - |

The elements of Table 3.2 correspond with the business process metamodel elements shown in Figure 3.2 of Section 3.2.1.1. As indicated by the hyphen not all business process elements are expressible in IntBIIS. Lanes and pools are highlighted in bold, as they cannot be fully described in IntBIIS. This means that not every activity in IntBIIS can be assigned to an executing employee. A corresponding element for lanes exists and is modeled by the *role* in the organization environment model. *Actor steps* have the attribute *responsible role* that links it to its executing *role*, but *system steps* misses this attribute. The pool, describing the organizational unit, is also missing in IntBIIS.

The overall business process is defined by a *usage scenario* in IntBIIS and can be found alongside with lane and pool in the top of Table 3.2. In the middle of Table 3.2 the subprocess, start event, stop event and activity are shown. The corresponding element for subprocess is the *scenario behavior* that is defined inside the *usage scenario.* Start and stop events are defined by *start* and *stop* elements. The activity has two expressions in IntBIIS. On the one hand, the *actor step* denotes the part performed by a human actor, and on the other hand, the *system step* denotes the part performed by the information system. The lower part of Table 3.2 shows the corresponding elements for gateways, sequence flow, data object and association. Gateways are expressed by *branches.* The sequence flow is defined by the *predecessor* and *successor* attributes in the *actor step* and *system step.* The data object is modeled in the data model as a *composite data object* or a *collection data object* and finally, the association is realized by the attributes *input data object* and *output data object* inside of the actor step. However, the corresponding part for the *system step* is missing.

IntBIIS_LP extends IntBIIS to model the previously described missing elements. Consequently, the input for PAcsTract are all processes of an organization modeled in IntBIIS_LP. Ideally, the processes encompass all the various departments in the organization. This provides a comprehensive picture of the ongoing work done by employees on a daily basis. Operating on this knowledge base allows PAcsTract to extract the role model and establish an ACR mapping model fitting the needs of the organization. The following section describes IntBIIS_LP and the metamodel extension in more detail.

#### Integrated Business IT Impact Simulation_Lanes and Pools (IntBIIS_LP)

IntBIIS_LP extends the metamodel of IntBIIS to complete the set of business process elements. Table 3.3 shows the elements that need to be extended.

To complete the element lane the attribute *responsible role* needs to be extended in the *system steps.* In IntBIIS the system step is represented by an *EntryLevelSystemCall* that refers to an interface of the system. To establish the element pool (see second row in Table 3.3), an organizational unit needs to be associated with a *role* of a process. Finally, the associations of data objects need to be extended in *system steps* (see last row in Table 3.3). These associations are already established within the *actor steps* by the attribute *input/output data object.* Both, *actor steps* and *system steps* form the activity of a business process, but the *system steps* need also a link to the data objects of the *actor step.*

**Table 3.3:** Elements that need to be extended in IntBIIS.

| BPMN | IntBIIS |
|---|---|
| Lane | Role, Attribute: Responsible Role, -<br>⬡ |
| Pool | - |
| Association | Attribute: Input/Output Data Object, - |

The optimal solution to complete the lane element, illustrated by the first row of Table 3.3, is to connect the *EntryLevelSystemCalls* with their *actor step*. This solves additionally the completion of the data object associations illustrated in the last row of Table 3.3. During this extension IntBIIS_LP additionally interconnects the *AcquireDeviceResourceActions* and *ReleaseDeviceResourceActions* with the corresponding *ActorStep*. As *AcquireDeviceResourceActions* and *ReleaseDeviceResourceActions* are triggered during an activity of an employee, these actions also belong to a certain *role* and thus, are realized the same way as with the *EntryLevelSystemCalls*. This interconnection allows to track which role acquires and releases which device.

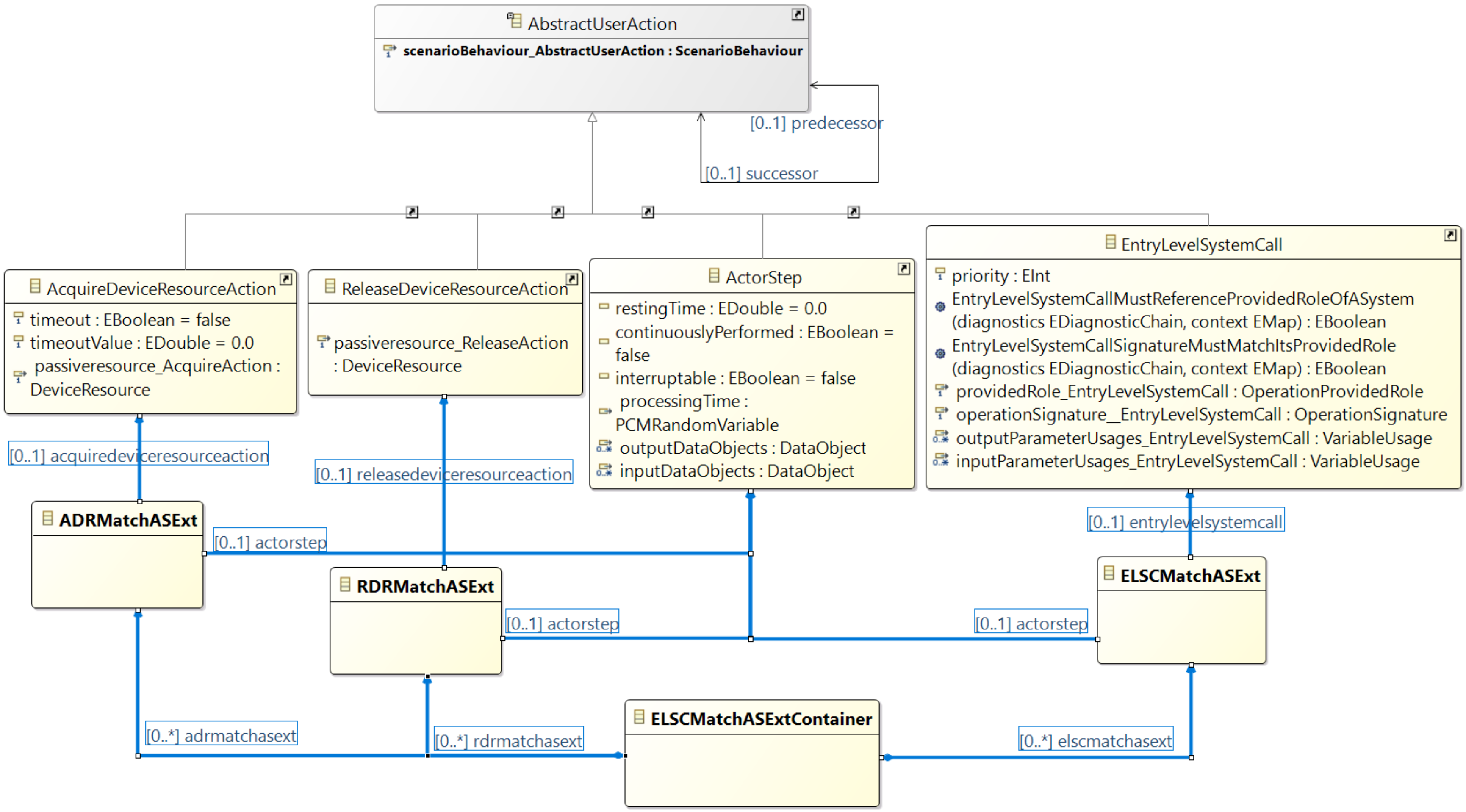
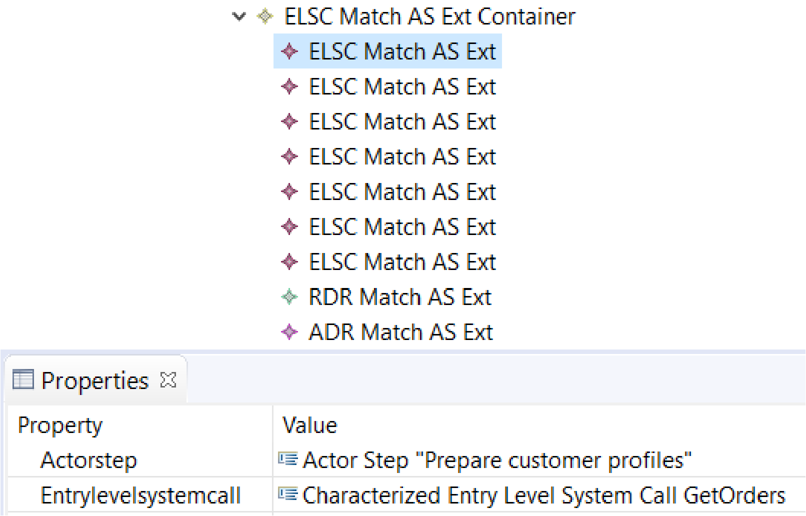

AbstractUserAction
scenarioBehaviour_AbstractUserAction : ScenarioBehaviour
[0..1] predecessor
[0..1] successor
AcquireDeviceResourceAction
timeout : EBoolean = false
timeoutValue : EDouble = 0.0
passiveresource_AcquireAction : DeviceResource
ReleaseDeviceResourceAction
passiveresource_ReleaseAction : DeviceResource
ActorStep
restingTime : EDouble = 0.0
continuouslyPerformed : EBoolean = false
interruptable : EBoolean = false
processingTime : PCMRandomVariable
outputDataObjects : DataObject
inputDataObjects : DataObject
EntryLevelSystemCall
priority : EInt
EntryLevelSystemCallMustReferenceProvidedRoleOfASystem (diagnostics EDiagnosticChain, context EMap) : EBoolean
EntryLevelSystemCallSignatureMustMatchItsProvidedRole (diagnostics EDiagnosticChain, context EMap) : EBoolean
providedRole_EntryLevelSystemCall : OperationProvidedRole
operationSignature__EntryLevelSystemCall : OperationSignature
outputParameterUsages_EntryLevelSystemCall : VariableUsage
inputParameterUsages_EntryLevelSystemCall : VariableUsage
[0..1] acquiredeviceresourceaction
[0..1] releasedeviceresourceaction
[0..1] entrylevelsystemcall
ADRMatchASExt
[0..1] actorstep
RDRMatchASExt
[0..1] actorstep
ELSCMatchASExt
[0..1] actorstep
ELSCMatchASExtContainer
[0..*] adrmatchasext
[0..*] rdrmatchasext
[0..*] elscmatchasext

**Figure 3.4:** Extended metamodel of IntBIIS. The blue highlighted arrows and black highlighted classes are introduced by IntBIIS_LP. Other elements are part of the IntBIIS metamodel.

Figure 3.4 shows the extended metamodel of IntBIIS for the interconnection of *EntryLevelSystemCalls*, *AcquireDeviceResourceActions* and *ReleaseDeviceResourceActions* with *ActorSteps*. The blue highlighted arrows and black highlighted classes are introduced by IntBIIS_LP. Other elements are part of the IntBIIS metamodel. To realize a non-intrusive metamodel, extension classes are provided (*ADRMatchASExt*, *RDRMatchAsExt* and *ELSCMatchASExt*) for each class that needs to be extended (*EntryLevelSystemCalls*, *AcquireDeviceResourceAction* and *ReleaseDeviceResourceAction*). Each extension class has a reference to the class that it extends and a reference to the class *ActorStep*. For example, in Figure 3.4 the class *ELSCMatchAsExt* has a reference to the class *EntryLevelSystemCall* and to the class *ActorStep*. Through this connection the class *EntryLevelSystemCall* is interconnected with its actor step. An additional container is needed to establish a model in IntBIIS_LP. In Figure 3.4 this container class is *ELSCMatchASExtContainer*.

The extension described above allows to interconnect *EntryLevelSystemCalls*, *AcquireDeviceResourceActions* and *ReleaseDeviceResourceActions* with *ActorSteps* for a certain process. Therefore, a new model is established for every process in IntBIIS_LP. An example is given in Figure 3.5.



| Property | Value |
|---|---|
| Actorstep | Actor Step "Prepare customer profiles" |
| Entrylevelsystemcall | Characterized Entry Level System Call GetOrders |

**Figure 3.5:** An example for the model that interconnects *EntryLevelSystemCalls*, *AcquireDeviceResourceActions* and *ReleaseDeviceResourceActions* with *ActorSteps*.

Figure 3.5 shows the interconnection of *EntryLevelSystemCalls* and actor steps for a process in IntBIIS_LP. In the top of Figure 3.5 the *ELSC Match AS Ext Container* establishes the required interconnection in its child elements. In the lower part of Figure 3.5 the properties of the blue highlighted child element *ELSC Match AS Ext* are shown. It has two properties: *Actorstep* and *EntryLevelSystemCall*. They allow to select the *EntryLevelSystemCall* and the actor step that should be interconnected. The equivalent is possible for the interconnection of *AcquireDeviceResourceActions* and *ReleaseDeviceResourceActions* by the child elements *RDR Math AS Ext* and *ADR Match As Ext*. This interconnection allows the matching with a certain actor step of a process. As the actor step has a distinct responsible role, this responsible role is connected to the interconnected *EntryLevelSystemCall*, *AcquireDeviceResourceAction* or *ReleaseDeviceResourceAction*. This completes the lane element for both parts of the activity (actor step and system step) in IntBIIS and additionally adds this property to the acquire device resource actions and release device resource actions. As the actor step has also distinct data objects that are associated by the *Input/Output Data Objects*, the same solution applies to the realization of the association for data objects.

To establish the pool element of Table 3.3, the *role* element modeled in the organization environment model is extended by an organizational unit. The metamodel extension is shown in Figure 3.6.

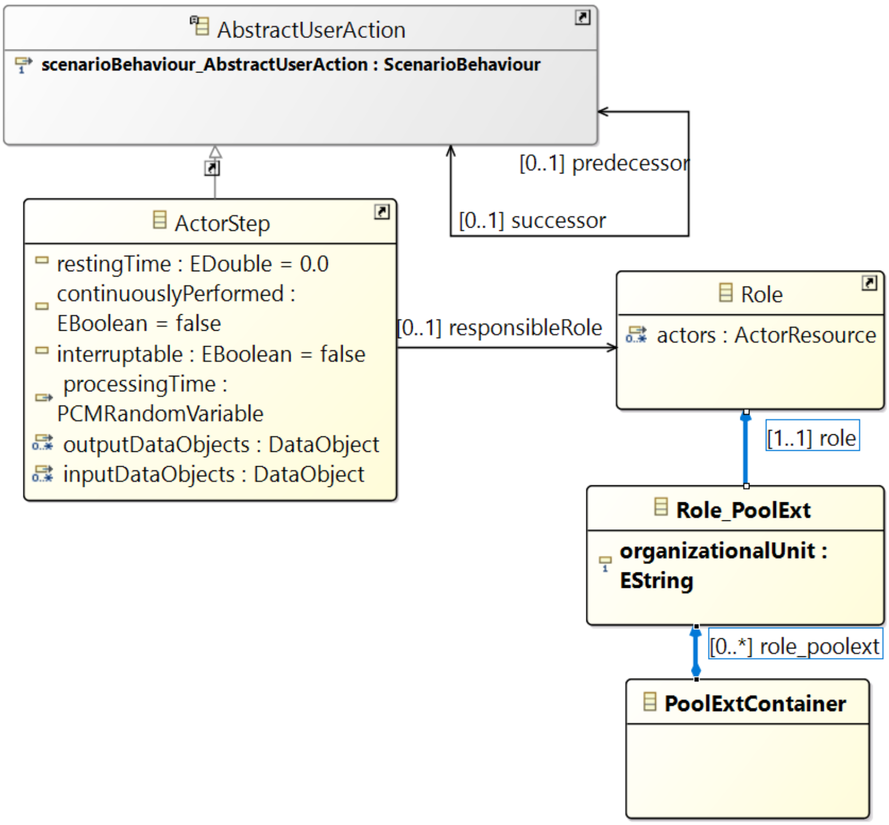
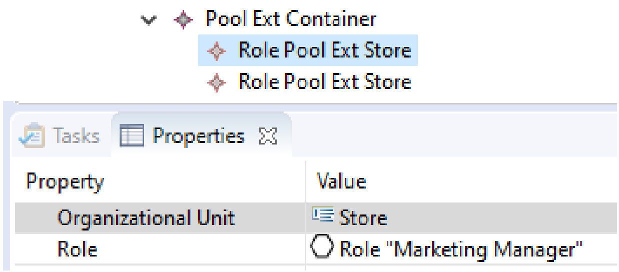

AbstractUserAction
scenarioBehaviour_AbstractUserAction : ScenarioBehaviour
[0..1] predecessor
[0..1] successor
ActorStep
restingTime : EDouble = 0.0
continuouslyPerformed : EBoolean = false
interruptable : EBoolean = false
processingTime : PCMRandomVariable
outputDataObjects : DataObject
inputDataObjects : DataObject
[0..1] responsibleRole
Role
actors : ActorResource
[1..1] role
Role_PoolExt
organizationalUnit : EString
[0..*] role_poolext
PoolExtContainer

**Figure 3.6:** Extended metamodel of IntBIIS. The blue highlighted arrows and black highlighted classes are introduced by IntBIIS_LP. Other elements are part of the IntBIIS metamodel.

The parts of Figure 3.6 that are not highlighted belong to the metamodel of IntBIIS. The blue highlighted arrows and black highlighted classes are extended by IntBIIS_LP. The class *Role_PoolExt* refers to the class *role* that is linked in the responsible role of the class *ActorStep*. It also defines the attribute *organizaitonalUnit* of type string. The class *PoolExtContainer* in the lower part of Figure 3.6 refers to the class *Role_PoolExt* and is required to establish a model.

Figure 3.7 shows an example model that extends the organizational unit in a process of IntBIIS_LP. In the top of the Figure 3.7 the *Pool Ext Container* allows to define child elements for the definition of the organizational unit. In the lower part of Figure 3.7 the properties of the blue highlighted child element *Role Pool Ext Store* are shown. It has two properties: *Role* and *Orgaizational Unit*. In the first property the role of a process is selected



Pool Ext Container
- Role Pool Ext Store
- Role Pool Ext Store

| Property | Value |
|---|---|
| Organizational Unit | Store |
| Role | Role "Marketing Manager" |

**Figure 3.7:** An example of the model that extends the organizational unit for roles.

and in the second property the organizational unit is defined for this role. By doing so, each role in a process in IntBIIS_LP has a distinct organizational unit.

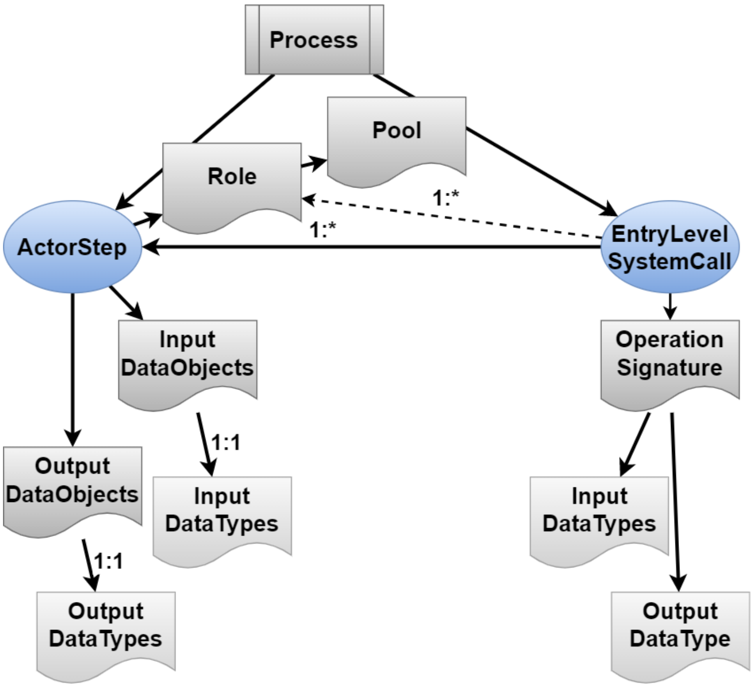


**Figure 3.8:** Illustrates the extensions provided by IntBIIS_LP.

Figure 3.8 illustrates the extensions provided by IntBIIS_LP on the basis of the interconnection between *EntryLevelSystemCalls* and actor steps. In the process shown in the top of Figure 3.8 the actor steps and *EntryLevelSystemCalls* are defined. The role of the actor step is linked to an organizational unit. Several *EntryLevelSystemCalls* can be linked to an actor step. This establishes the implicit connection to the role and the input/output data objects.

To sum up, PAcsTract consumes all processes of an organization modeled in IntBIIS_LP. Each process is modeled in a separate business process model. For each business process model there are two models providing the missing business process elements. The *ELSC Match AS Ext Container* connects the *EntryLevelSystemCalls* with their actor step and the *Pool Ext Container* defines the organizational unit for the role of a process.

#### 3.2.1.3 Input for Access Permission Architecture Aligner

In Section 3.2.1.3 the formal concept for the identification of ACR breaches in EAAs was introduced. AcsALign realizes this concept. This section elaborates on the input data required for AcsALign. As the approach is built on top of PAcsTract, the approach does also operate on EAAs defined in PCM (Section 2.4) and business processes defined in IntBIIS_LP (Section 3.2.1.2).

The underlying concept of AcsALign requires the following information: 1) an EAA, 2) a set of ACRs for the given EAA, 3) a mapping for data objects from business processes to data types of the EAA, 4) a mapping for activities from business processes to service calls of the EAA and 5) the actual read and written data types of the data flows of invoked services.

Hence, the following input data is required for AcsALign:

- The EAA is provided in form of PCM models.
- A set of ACRs is consumed from the output of PAcsTract. There are two possibilities in PAcsTract from where the ACRs might be derived. On the one hand, PAcsTract can extract the ACRs from business processes. On the other hand, a security expert can extend the extracted ACRs with technical ACRs. Both are possible sources of ACRs for AcsALign. AcsALign uses the ACR mapping model established by PAcsTract to consume the required information (further information on the ACR mapping model of PAcsTract will be provided in Section 3.2.3.1 and Section 3.2.3.3).
- The mapping from data objects to data types and the mapping from activities to service calls are extracted from business processes modeled in IntBIIS_LP. Business processes in IntBIIS_LP are tightly coupled with the EAA in PCM and thus, already comprise the required information.
- The read and written data types of service calls are provided by the palladio extension Data Centric Palladio (DC-PCM) [202]. Among others it provides a simple data flow analysis for PCM models that outputs a list of data types that are read and written during the invocation of services. Read means that data types are provided to the user or system that has invoked the service and thus, are read by them. Written means that data types are persisted in a database of the EAA. By operating on this knowledge base AcsALign provides a list of service calls that violates the ACRs.

### 3.2.2 Model Overview, Responsibilities and Assumptions

This section puts the assumptions of this thesis into a bigger frame. Afterwards, the various models used as input by the approaches of this thesis are divided into models of business processes and models of EAA. This is done to provide a better overview before beginning with the detailed sections about the approaches. At the end, employees of an organization are assigned to models they are responsible for.

Organizations that explicitly model and use business processes can benefit more from the approaches in this thesis, as they do not have to model business processes that serve as input for the approaches anymore. This is the case for many large organizations. Nevertheless, organizations that do not use business processes can model them in order to align their processes with ACRs and EAA. Clearly, organizations with high-security requirements have to comply with more ACRs and thus, benefit more from the alignment established by the approaches of this thesis.

In this thesis, I assume that business processes and IT architecture are built in a top-down manner, meaning that the IT level has to meet requirements of the business level. Business experts do not choose from ready-to-use IT modules, but rather the enterprise architect models the EAA according to the requirements of the business processes designed by the business expert. However, this assumption is only made to explain the approaches in a systematic way. How organizations can utilize the approaches that already have an EAA or business processes or an evolution scenario, is detailed in Chapter 4.

Within the scope of this thesis I assume that ACRs incorporated into business processes by the business level are legally correct and in line with the business goals introduced in Chapter 1. This thesis does not focus on identifying erroneous ACRs, but on defining an automated transformation of ACRs from business processes to IT level artifacts. For the same reason, I assume that business processes modeled by the business level are syntactically correct and do express the intended matters.

The following two tables summarize the models that are required as input for BAcsTract and PAcsTract. Table 3.4 specifies the business process models that are required as input for BAcsTract.

**Table 3.4:** Business process model input for BAcsTract.

| Model Name | Business Process Model Description |
|---|---|
| BPMN | Model for business processes with all the required information as lanes, pools, activities and data objects. |

In the case of BAcsTract, it is simple. Table 3.4 shows that only BPMN models are required. The business expert is responsible for all parts of the BPMN models. No other models are required.

Table 3.5 specifies the business process models and EAA models that are required as input for PAcsTract and AcsALign and provides a short explanation of each model. The first five models of Table 3.4 specify business processes in IntBIIS_LP. The last five models specify the EAA in PCM.

Table 3.6 summarizes which employees are in charge of the different models of Table 3.5. The responsibility of a particular employee means that this employee has the final decision on certain elements of the model.

Table 3.6 shows a top-down specification of models in an organization and their responsible employees. Certainly, most organizations do already have some models and undergo

**Table 3.5:** Business process models and EAA models that are input for PAcsTract and AcsALign.

| Model Name | Business Process Model Description | EAA Model Description |
|---|---|---|
| Business Process Model (Usage Scenario, BP Usage Model) | Model for the flow of actor steps and system steps and the interconnection of elements modeled in other IntBIIS_LP models. | x |
| Data Model | Model for data objects. | x |
| Organization Environment Model | Model for roles and devices. | x |
| ELSC Match AS Ext Container | Model for the interconnection of system steps and acquire/release device resource actions with actor steps. | x |
| Pool Ext Container | Model for the organizational unit of roles. | x |
| System | x | Model for the interconnection of systems, components and interfaces. |
| Repository | x | Model for systems, components, interfaces and data types. |
| Service Effect Specification | x | Model for the behavioral description of components. |

evolution scenarios where parts of the models are changing. Here, a top-down specification is considered which pertains organizations that start from scratch and don't have any models. Regarding BAcsTract, PAcsTract and AcsALign it does not matter if organizations start from scratch or undergo evolution scenarios. In both cases they can utilize the approaches equally. Chapter 4 will discuss in detail how organizations may use the approaches when they undergo different evolution scenarios. For the sake of clarity, in the following the top-down specification of models is discussed. Fundamentally, all responsible employees cannot change elements ultimately that lie in the responsibility of other employees. It is possible that employees propose dummies or suggestions for elements outside of their responsibilities, but the responsible employee takes always the final decision on the model elements he is responsible for. Exemplarily, the enterprise architect cannot change any elements that the business expert is responsible for and has finally decided on, but the business expert may propose system steps.

**Table 3.6:** Shows the model responsibilities in case of PAcsTract and AcsALign.

| Nr. | Employee in Charge | Model Name | Model Elements |
|---|---|---|---|
| 1 | Business Expert | Business Process Model<br>Data Model<br>Organization Environment Model<br>Pool Ext Container<br>Repository | Actor Steps, ADR/RDR Actions<br>All<br>All<br>All<br>Data Types (for Data Objects) |
| 2 | Enterprise Architect | Business Process Model<br>ELSC Match AS Ext Container<br>System<br>Repository | System Steps<br>All<br>Systems with Interfaces<br>Interfaces (for Systems),<br>Data Types (extends) |
| 3 | Software Architect | System<br>Repository | Components with Interfaces<br>Interfaces (for Components),<br>Data Types (extends) |
| 4 | Component Developer | Service Effect Specification | All |
| 5 | System Deployer | Resource Environment<br>Allocation | All<br>All |

1. **Business Expert:** is responsible for modeling most of the IntBIIS_LP models. Data objects are specified in the data model. Roles and devices are specified in the organization environment model. The actual business processes are modeled in the business process models. In the business process model the business expert is responsible for the acquire/release device resource actions and the actor steps. Inside the actor steps he interconnects data objects and responsible roles. He may also propose system step dummies for the enterprise architect. The business expert specifies the organizational units for the roles in the pool ext container. With these steps he completes all the elements that are part of business processes. Finally, for each data object from the data model an equivalent data type is modeled in the repository. This step can be automated, as it is an one-to-one mapping.

2. **Enterprise Architect:** takes models from the business expert. He specifies the required interfaces and extends the current data types in the repository. In the same way, he specifies systems and their interfaces in the system diagram. By doing so, he specifies the system landscape, meaning the EAA. The enterprise architect is also responsible for the technical part of the IntBIIS_LP models. In the business process models, he specifies or completes the system steps. Afterwards, he interconnects system steps with their actor steps in the *ELSC Match AS Ext Container*.

3. **Software Architect:** uses the models of the enterprise architect to model the architecture inside the systems. He extends datatypes and specifies interfaces in the repository. Additionally, he specifies components with interfaces in the system diagram.

4. **Component Developer:** is responsible for the behavioral specification of components and uses models of the software architect and enterprise architect. Therefore, he models the service effect specification for each component. By this he describes the behavior of components and their interactions with other components.

5. **System Deployer:** uses the models of the previous roles to specify the available resources, for example, CPU, HD, network and memory and to allocate components to resources.

### 3.2.3 Role Model Extraction from Business Processes

The two approaches described in this section realize the concept introduced in Section 3.1.1, namely to align business level and IT level artifacts of organizations in terms of ACRs. Both approaches extract implicitly modeled ACRs from business processes that are designed by service design managers and compliance managers [34] (formerly introduced as the business level), automatically. During the extraction participants, their activities and associated data objects are analyzed to build an ACR mapping model that interconnects elements of access control and business processes. Section 3.2.3.1 explains the ACR mapping model that is part of the concept of Section 3.1.1. Afterwards, Section 3.2.3.2 explains the approach BAcsTract and how it extracts ACRs from BPMN to form a role model for RBAC. Section 3.2.3.3 and Section 3.2.3.4 explain the approach PAcsTract that extracts ACRs from IntBIIS_LP, a BPMN pendant in PCM.

#### 3.2.3.1 Access Control Requirements Mapping Model

Section 3.1.1 explained the concept for extracting ACRs from business processes by elevating an ACR mapping model that aligns business processes with RBAC. This section introduces the ACR mapping model that is built during the extraction process of BAcsTract and PAcsTract. Throughout the extraction the ACR mapping model is built by interconnecting elements of business processes with elements of RBAC. Afterwards, it contains the implicitly modeled ACRs of the business processes. In a final step BAcsTract and PAcsTract extract the role model from the ACR mapping model. Apart from the usage during the extraction, the ACR mapping model provides a documentation of design decisions and allows to trace access permissions. This enables the business level as well as the IT level to better understand mutual dependencies between models of business and IT, especially during evolution scenarios.

This thesis proposes a concept for decomposition and aggregation of roles and permissions (cf. [76]). Roles and permissions that are elements of access control are interconnected with intermediate layers. These newly introduced intermediate layers are specific for the context of business processes [177]. Altogether, they form the layers of the ACR mapping model. In addition, this thesis proposes a new process in which roles, permissions and the newly introduced intermediate layers are not engineered manually but extracted from business processes automatically [177]. By doing so, the approaches BAcsTract and PAcsTract extract business level ACRs automatically.

Figure 3.9 shows the layers of the ACR mapping model according to the definition of Equation (3.29) in Section 3.1.1. It is implemented as a database, where each column of the database represents a layer. Altogether, there are four layers: role, process, activity, permission.

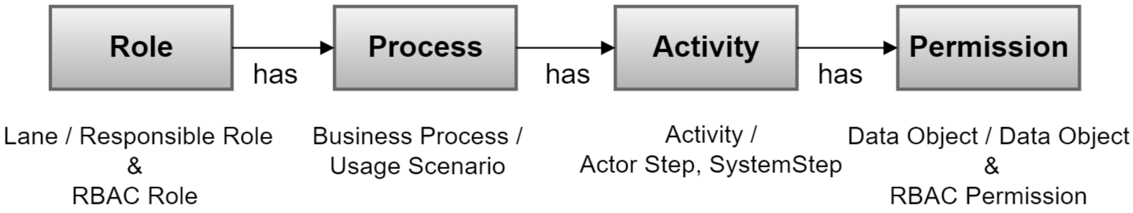

Role
has
Process
has
Activity
has
Permission
Lane / Responsible Role & RBAC Role
Business Process / Usage Scenario
Activity / Actor Step, SystemStep
Data Object / Data Object & RBAC Permission

**Figure 3.9:** Shows the ACR mapping model filled by BAcsTract and PAcsTract during the extraction of ACRs from business processes.

Each layer of the ACR mapping model in Figure 3.9 represents either an element of a business process, RBAC or of both together. The mapping of these elements was formally defined in Equations (3.27) to (3.28) of Section 3.1.1. Table 3.7 lines up the elements of BPMN and IntBIIS_LP and should help to understand the mapping of these elements to the layers of the ACR mapping model.

- **Role**: The first layer is clearly part of RBAC and represents the RBAC role. In RBAC the role comprises a set of permissions in order to fulfill the tasks that are the duty of that role. The employee responsible for these tasks is assigned to that role. In business processes, as defined in Equations (3.17) to (3.24) of Section 3.1.1, a set of activities is composed into lanes. The lane stands typically for one or more employees that are responsible to fulfill these activities during their daily work. This makes the concept of the lane similar to that of the RBAC role and thus both elements are mapped in layer role. The corresponding element for the lane in IntBIIS_LP is the responsible role of the actor step (see Table 3.7). As system steps are connected to actor steps, they also belong to the responsible role of the actor step. Thus, in case of PAcsTract the responsible role is mapped to the layer role.
- **Process**: This layer represents only elements of business processes. Namely, the business process itself. The work of an employee (role) is organized across several business processes, during which he has to fulfill the activities in his lane. Thus, each role has a number of processes it is connected with. The corresponding element for the business process in IntBIIS_LP is the usage scenario (see Table 3.7). So, in case of BAcsTract the process name is mapped to the layer process. In case of PAcsTract the name of the usage scenario is mapped.

**Table 3.7:** Comparative elements of BPMN and IntBIIS_LP.

| BPMN | IntBIIS |
|---|---|
| Business Process | Usage Scenario |
| Lane | Role, Actor Step Attribute: Responsible Role, ELSC Match As Ext<br>⬡ |
| Pool | Role Pool Ext Store Attribute: Organizational Unit |
| Subprocess | Scenario Behavior |
| Start Event | Start<br>↘● |
| Stop Event | Stop<br>◉ |
| Activity | Actor Step, System Step<br>✦ |
| Gateways | Branch |
| Sequence flow | Actor Step Attribute: Successor, Predecessor |
| Data Object | Composite/Collection Data Object<br>✦ ✦ |
| Association | Actor Step Attribute: Input/Output Data Object, ELSC Match As Ext |

- **Activity**: This layer also represents only elements of business processes. An employee (role) has to fulfill during his work different tasks (business processes). During each task (business process) he has to complete a set of activities, which are the duty of his work. By completing these activities, he fulfills the overall business process. Hence, the third layer represents the activities a role has to fulfill during its processes. The corresponding elements for the activity in IntBIIS_LP are the actor step and system step (see Table 3.7). Thus, in case of PAcsTract the actor steps and system steps of the role's processes are mapped to the layer activity.

- **Permission**: The last layer represents elements of business processes and RBAC. In order to complete an activity, the employee needs the correct amount of access permissions to the required devices, systems and folders. These access permissions are specified in an access control system, such as RBAC. For example, to carry out the activity *change price of product* shown in Figure 3.10, the employee needs access permissions to the information system and to the file of the product or to the service function for changing product price. There might be activities where no access permissions are needed, for example, *sort product in shelf.* In business processes, activities have associations with data objects. Depending on the association the activity needs a data object as input in order to be carried out and thus, reads data, or produces a data object as a result of the activity and thus, writes data. Consequently, input and output data objects of activities represent permissions. Thus, the last layer represents the input and output data objects that a role requires to fulfill the activities during its processes. The corresponding elements for the data objects and their associations in IntBIIS_LP are the data object modeled in the data model and its binding in the actor step attribute input/output data object (see Table 3.7). As system steps are connected to actor steps, they are also associated with the input/output data object attribute of the actor step. So, in case of PAcsTract the input/output data object attribute is mapped to the layer permission.

An example of an entry in the ACR mapping model is illustrated in Figure 3.10. A part of the store manager's duty is the business process *change price.* In this process his lane is called *store manager* and he has the activity *change price of product.* This activity has the output data object *product*, meaning that the store manager does a write operation to that product during the change of the product's price. In this example the row in the ACR mapping model would look like as follows: role would be *store manager*, process would be *change price*, activity would be *change prices of product* and permission would be *write product* (see Figure 3.10).

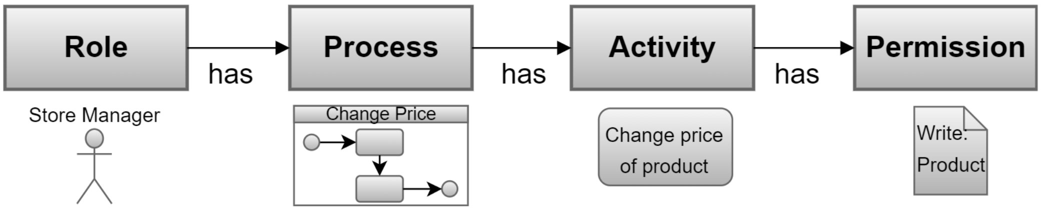


**Figure 3.10:** Illustrates an example of an entry in the ACR mapping model.

To sum up, a role in the RBAC role model can be seen as a lane in the business process. Both are represented in the first layer role and are assigned to an employee. Each role has a set of business processes which it needs to fulfill as part of its daily work duties. The business processes of a role are represented in the second layer process. In each business process, the role completes a certain amount of activities, each of which requires a definite set of permissions in order to fulfill the activity. The activities of all role's processes and their permissions are represented by the third and fourth layer. The mapping of business process and RBAC elements to the layers of the ACR mapping model is also summarized in Figure 3.9.

By establishing the above-mentioned ACR mapping model, BAcsTract and PAcsTract extract ACRs from business processes and establish a traceability between RBAC and business process elements. This allows to understand the origin of each extracted business level ACR by tracing it back to their originating business process, lane, activity and associated data object.

#### 3.2.3.2 BPMN Access Permission Extractor

Section 3.1.1 explained the formal concept of extracting ACRs from business processes on which BAcsTract is based. BAcsTract is a top down approach to elicit role models for RBAC automatically by analyzing the business processes of an organization. ACRs are extracted from business processes by establishing an ACR mapping model. The ACR mapping model was introduced in the previous section. During the final steps of BAcsTract a role model is extracted from the ACR mapping model together with a hierarchy. The resulting RBAC role model is an initial role model comprising the business level ACRs. Technical ACRs has to be extended by security experts. Section 3.2.1.1 introduced the input data for BAcsTract. This section will elaborate on the main logic of the approach BAcsTract. It realizes the concept introduced in Section 3.1.1 for the de facto standard business process language BPMN [5]. Therefore, BAcsTract is provided with a set of BPMN business processes of an organization. Ideally, these processes encompass all the various departments of the organization and provide a comprehensive picture of the ongoing work done by the employees on a daily basis.

Figure 3.11 shows an example process which is further on used to explain how BAcsTract is working. It depicts the process *Prepare Advertisements and Discounts* of a supermarket store called CoCoME. The process is triggered by the store manager, who decides that advertisements and discounts of the store have to be renewed. Therefore, he reviews the previously issued advertisement schedules and prepares a new advertisement request for the marketing manager. The marketing manager receives this advertisement request and begins the preparation of a new advertisement schedule according to the advertisement request of the store manager. In order to select proper advertisements and discounts he analyzes customer profiles of the loyalty customers of the supermarket store. Finally, he selects advertisements and discounts according to the needs of the customers and finishes the advertisement schedule. In the last step, the store manager approves the proposed advertisement schedule. Based on this example process, BAcsTract will be explained further on.

BAcsTract extracts ACRs from business processes by building an ACR mapping model from which afterwards the initial role model is formed. This is done in six steps. Step one to four build the ACR mapping model that was explained in Section 3.2.3.1. Step five and six create a hierarchy and form the initial role model.

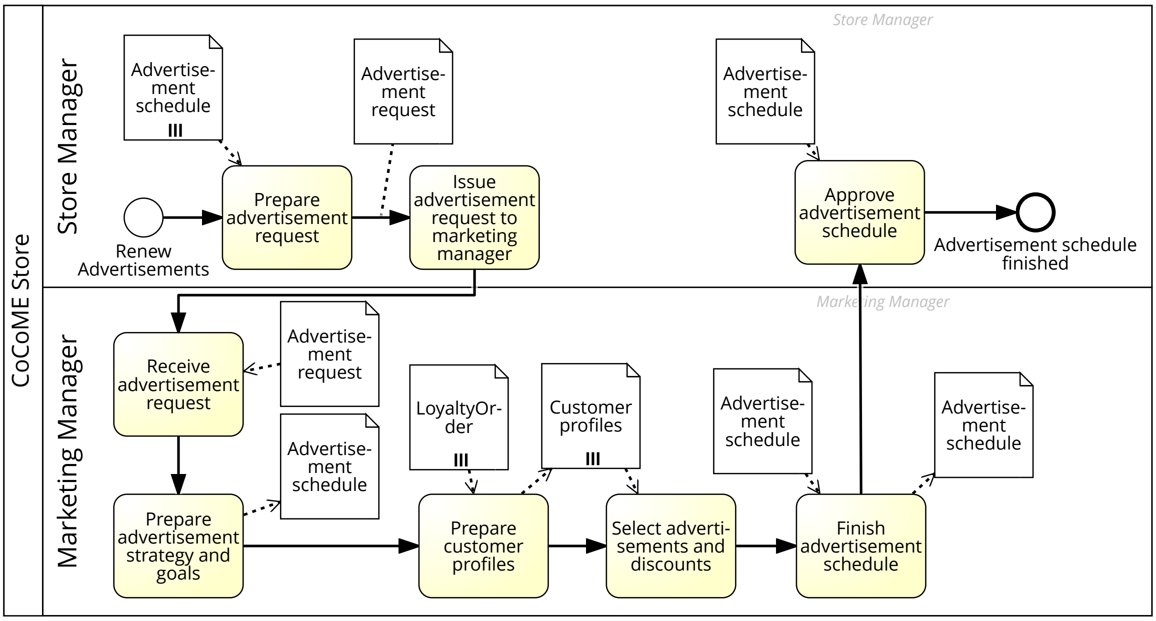

Store Manager
CoCoME Store
Store Manager
Marketing Manager
Advertise-ment schedule
Advertise-ment request
Advertise-ment schedule
Renew Advertisements
Prepare advertisement request
Issue advertisement request to marketing manager
Approve advertisement schedule
Advertisement schedule finished
Marketing Manager
Receive advertisement request
Advertise-ment request
Advertise-ment schedule
LoyaltyOr-der
Customer profiles
Advertise-ment schedule
Advertise-ment schedule
Prepare advertisement strategy and goals
Prepare customer profiles
Select adverti-sements and discounts
Finish advertisement schedule

**Figure 3.11:** Shows the business process *Prepare Advertisements and Discounts* of a supermarket store.

*Step 1:* During the first step, roles are extracted from business processes. This was formalized in Equation (3.30) of Section 3.1.1. In this step, unique names are extracted from lanes and their organizational divisions are extracted from pools. Pools represent organizational divisions of roles, making equal roles distinguishable across processes. Figure 3.12 shows this. Both lanes are named *Director*, but they represent different roles. By considering the organizational division inside the pool (*marketing* and *sales*), it is possible to distinguish the roles. Not every lane transforms to a role inside the role model. For example, closed lanes are used in BPMN to model external participants that are interacting in a process (like a customer in a sale process). Figure 3.13 shows a closed lane in BPMN. It does not have any activities nor data objects, as it does not belong to the organization itself and has no access rights within the organization. Thus, such lanes have no entry in the ACR mapping model. Another example for roles that are not transformed into the role model, are lanes without a single data object in any of their activities. Roles without any data object associations have no defined access rights in the business processes and thus, do not require an entry in the role model. Nevertheless, these roles have entries in the ACR mapping model in order to be complete. Table 3.8 shows the ACR mapping model for the example process in Figure 3.11 after step one.

**Table 3.8:** ACR mapping model after step one.

| Nr. | Role | Process | Activity | Permission |
|---|---|---|---|---|
| 1 | CoCoME Store: Store Manager | | | |
| 2 | CoCoME Store: Marketing Manager | | | |

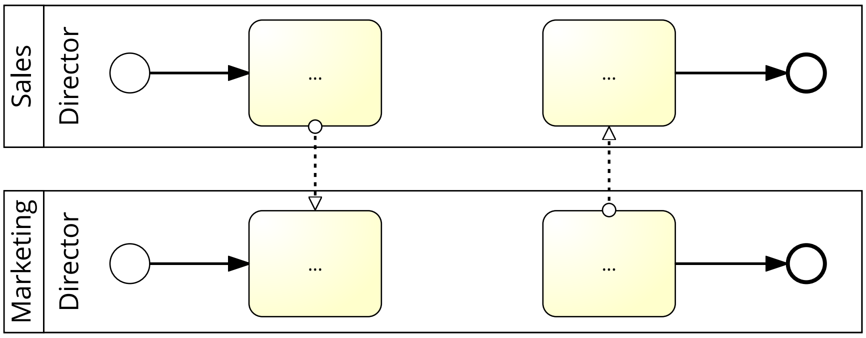


**Figure 3.12:** Explains how pools distinguish organizational divisions.

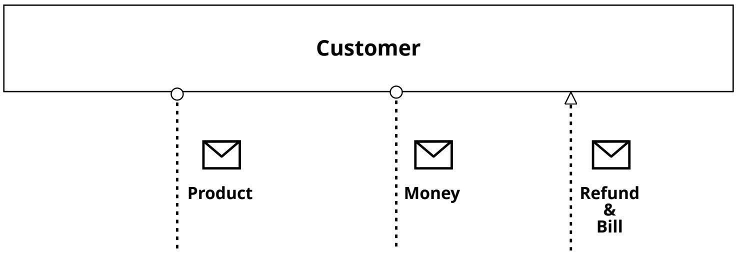


**Figure 3.13:** Shows an example of a closed lane in BPMN.

Table 3.8 has two entries, one for each lane of the process shown in Figure 3.11. The first row corresponds with the first lane store manager. The second row corresponds with the second lane marketing manager. Both role entries in the ACR mapping model have their organizational unit *CoCoME Store* beforehand.

*Step 2:* In this step, the process name is extracted for each role that participates in the process. This was formalized in Equation (3.31) of Section 3.1.1. During this step, processes and roles are interconnected with each other. A role gets an entry with the process name in the ACR mapping model for each process it is participating in. Table 3.9 shows the ACR mapping model for the example process in Figure 3.11 after step two. As there is only one process in this example each role has exactly one entry.

**Table 3.9:** ACR mapping model after step two.

| Nr. | Role | Process | Activity | Permission |
|---|---|---|---|---|
| 1 | CoCoME Store: Store Manager | Prepare Advertisements and Discounts | | |
| 2 | CoCoME Store: Marketing Manager | Prepare Advertisements and Discounts | | |

*Step 3:* In step three, activities are extracted according to the formalization in Equation (3.32) of Section 3.1.1. Each role's activities in a process are analyzed and added to the ACR mapping model. By doing so, each role and process is interconnected with the role's activities. Sub-processes, that are visualized as activities with an extra plus, are forming an exception. As they link to another process that is analyzed in any event as an own process, they produce no entries in the ACR mapping model to avoid duplicates. Table 3.10 shows an excerpt of the ACR mapping model for the example process in Figure 3.11 after step three.

**Table 3.10:** Excerpt of the ACR mapping model after step three.

| Nr. | Role | Process | Activity | Permission |
|---|---|---|---|---|
| 1 | CoCoME Store: Store Manager | Prepare Advertisements and Discounts | Prepare advertisement request | |
| 2 | CoCoME Store: Store Manager | Prepare Advertisements and Discounts | Issue advertisement request to ... | |
| 3 | CoCoME Store: Store Manager | Prepare Advertisements and Discounts | Approve advertisement schedule | |
| 4 | CoCoME Store: Marketing Manager | Prepare Advertisements and Discounts | Receive advertisement request | |
| | | ... | | |

As can be seen in Table 3.10, there are more entries in the ACR mapping model than after step two. This is the case, because in step three each row identifies a certain activity of a role in a process. The store manager has three activities in the process of Figure 3.11. These three activities can be found in row one to three of the ACR mapping model shown in Table 3.10. *Prepare advertisement request* is interconnected with the process *Prepare Advertisement and Discounts* and role *CoCoME Store: Store Manager*. The same is valid for the other two activities as well as for the activities of the marketing manager.

*Step 4:* This step extracts permissions from activities of business processes, according to the formalization in Equation (3.33) of Section 3.1.1. Therefore, activities of all processes of a role are analyzed for data objects and their input/output associations. There are two different possibilities to associate data objects. The left part of Figure 3.14 (1. and 2.) shows the association with an activity, while the right part (3.) shows the association with a flow transition.

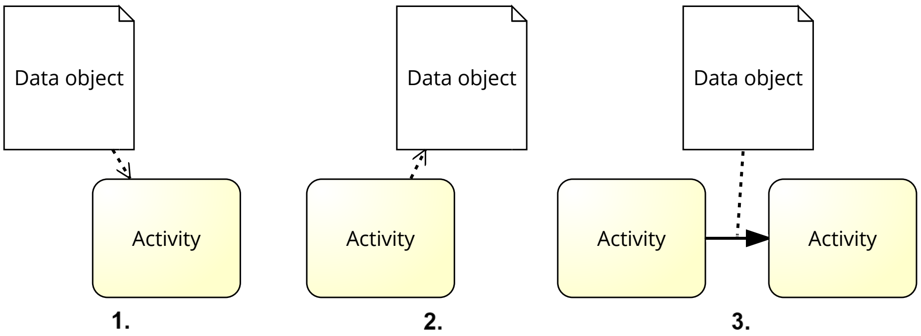


**Figure 3.14:** Shows different ways to model associations of data objects in BPMN.

1. In (1.) of Figure 3.14, the association depicted by a dashed arrow, is pointing to the activity. This means that the data object is required as an input to the activity in order to fulfill it. An input data object is synonymous with a read operation on this data object. For example, in the last activity of the store manager in Figure 3.11, *Approve advertisement schedule*, he has to approve the *advertisement schedule* that was prepared by the marketing manager. To do this, he requires read access to the advertisement schedule. This is depicted by the input data association. In order to open the file of the advertisement schedule he makes a read operation. In terms of access control, he requires a read permission.

2. In (2.) of Figure 3.14, the association is the other way around, pointing from the activity to the data object. This means that during the activity a data object is produced as an output. This is synonymous with a write operation and thus, requires a write permission. For example, the marketing manager prepares the advertisement strategy and goals for the *advertisement schedule* in the activity *Prepare advertisement strategy and goals* in the lower left part of Figure 3.11. To do this, he has to write the advertisement strategy and goals to the advertisement schedule and therefore needs a write permission for the file. This is depicted by the output association in Figure 3.11. The activity produces the output data object advertisement schedule, meaning that this data object is created or modified.

3. In (3.) of Figure 3.14, the data object is connected to the flow transition between two activities. This notation can be decomposed into two associations as follows: in the first activity the data object has an output association and in the second activity an input association. Hence, the first activity requires a write permission while the second requires a read permission for the data object. An example is shown in between the first and second activity of the store manager in Figure 3.11. The data object *advertisement request* is associated to the flow transition between the activities *Prepare advertisement request* and *Issue advertisement request to marketing manager*. This means that the first activity produces the advertisement request as an output, therefore requiring a write permission and afterwards this advertisement request serves as an input to the second activity, therefore requiring a read permission.

By analyzing associated data objects this way, read and write permission are extracted from the business processes and are interconnected to the activities of a role's process in the ACR mapping model. Attention has to be payed to the isCollection attribute of data objects defined by the BPMN standard [5]. It allows to define the multiplicity of the data object. A collection data object means that the data object has the multiplicity of n. For example, the input data object advertisement schedule in the first activity of the store manager in Figure 3.11 is a collection. This is depicted by the three dashes in the lower part of the data object. In order to prepare the advertisement request, the store manager has to look at all the previous advertisement schedules. Thus, the input data object *advertisement schedule* is denoted as a collection, meaning that there are several advertisement schedules read by the store manager. For better understanding, the name of the data object can be interpreted as the data type, meaning that there are several input files of type advertisement schedule. This is crucial, because indistinction of data object names due to multiplicities are avoided. In this case, the advertisement schedule and the collection advertisement schedule have the same name. Hence, the approach can identify that both are of the same data type, but the first is a single object and the latter a collection of this object. This allows to distinguish in the permissions whether the data object is a collection or not. It is important as information systems work with this concept. Table 3.11 shows an excerpt of the ACR mapping model for the example process in Figure 3.11 after step four.

**Table 3.11:** Excerpt of the ACR mapping model after step four.

| Nr. | Role | Process | Activity | Permission |
|---|---|---|---|---|
| 1 | CoCoME Store: Store Manager | Prepare Advertisements and Discounts | Prepare advertisement request | READ Advertisement schedule (Coll.) |
| 2 | CoCoME Store: Store Manager | Prepare Advertisements and Discounts | Prepare advertisement request | WRITE Advertisement request |
| 3 | CoCoME Store: Store Manager | Prepare Advertisements and Discounts | Issue advertisement request to ... | READ Advertisement request |
| 4 | CoCoME Store: Store Manager | Prepare Advertisements and Discounts | Approve advertisement schedule | READ Advertisement schedule |
| 5 | CoCoME Store: Marketing Manager | Prepare Advertisements and Discounts | Receive advertisement request | READ Advertisement request |
| | | ... | | |

Table 3.11 shows that in each row the permission is associated with its activity, process and role. The permission itself consists of a READ, WRITE or READ/WRITE and the

corresponding data object. The first row shows the required read permission for the advertisement schedule collection during the activity *Prepare advertisement request.* Row two depicts the required write permission for the advertisement request during the same activity, as the data object is associated with the flow transition and thus, row three depicts the required read permission for the following activity *Issue advertisement request to marketing manager.* After step four, all business level ACRs from business processes are extracted and stored in the ACR mapping model in form of interconnected tuples of role, process, activity and permission.

*Step 5:* During this step, a simple hierarchy is elicited on the basis of the ACRs in the ACR mapping model. This is done according to the formalization in Equation (3.37) of Section 3.1.1. Therefore, the permissions of each role, e.g., cashier, are inspected whether they are a subset of another role, e.g., manager. If this is the case, the role manager inherits from role cashier. An example is given in Figure 3.15. In the upper part of Figure 3.15, the role cashier and the role manager are represented with their permissions. Both roles have the permission *READ Inventory list*, but only the manager has the permission *READ/WRITE Financial data.* The permissions of the cashier are a subset of the permissions of the manager. Thus, a hierarchy is built as shown in the lower part of Figure 3.15, where the role manager inherits from the role cashier. There is the possibility to introduce virtual roles according to [145] to optimize the hierarchy. This may help to reduce the amount of duplicate permissions and ease permission management. Virtual roles combine a subset of permissions from which other roles can inherit. The only difference between the normal hierarchy and virtual roles is that virtual roles are technically never assigned to an employee [145]. They only serve for abstraction purposes. 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 may be introduced.

*Step 6:* During the last step, the initial RBAC role model is extracted from the ACR mapping model. The formal definition was given in Equations (3.35) and (3.36) of Section 3.1.1. For each unique role, which has permissions, all unique permissions are extracted. Afterwards, they are combined with the information from step five resulting in a role model with a hierarchy. Roles without any permissions are ignored. The role model for the provided example according to the excerpt of the ACR mapping model from Table 3.11 is shown in Table 3.12.

**Table 3.12:** Role model for the excerpt of the ACR mapping model from Table 3.11.

| Nr. | Role | Permission |
|---|---|---|
| 1 | CoCoME Store: Store Manager | READ Advertisement schedule (Coll.) |
| 2 | CoCoME Store: Store Manager | READ/WRITE Advertisement request |
| 3 | CoCoME Store: Store Manager | READ Advertisement schedule |
| 4 | CoCoME Store: Marketing Manager | READ Advertisement request |
| | | ... |

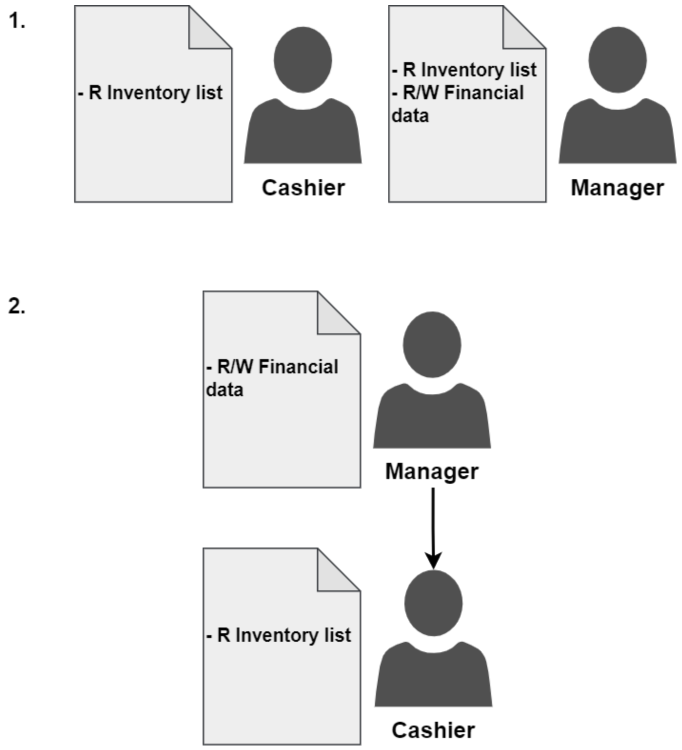
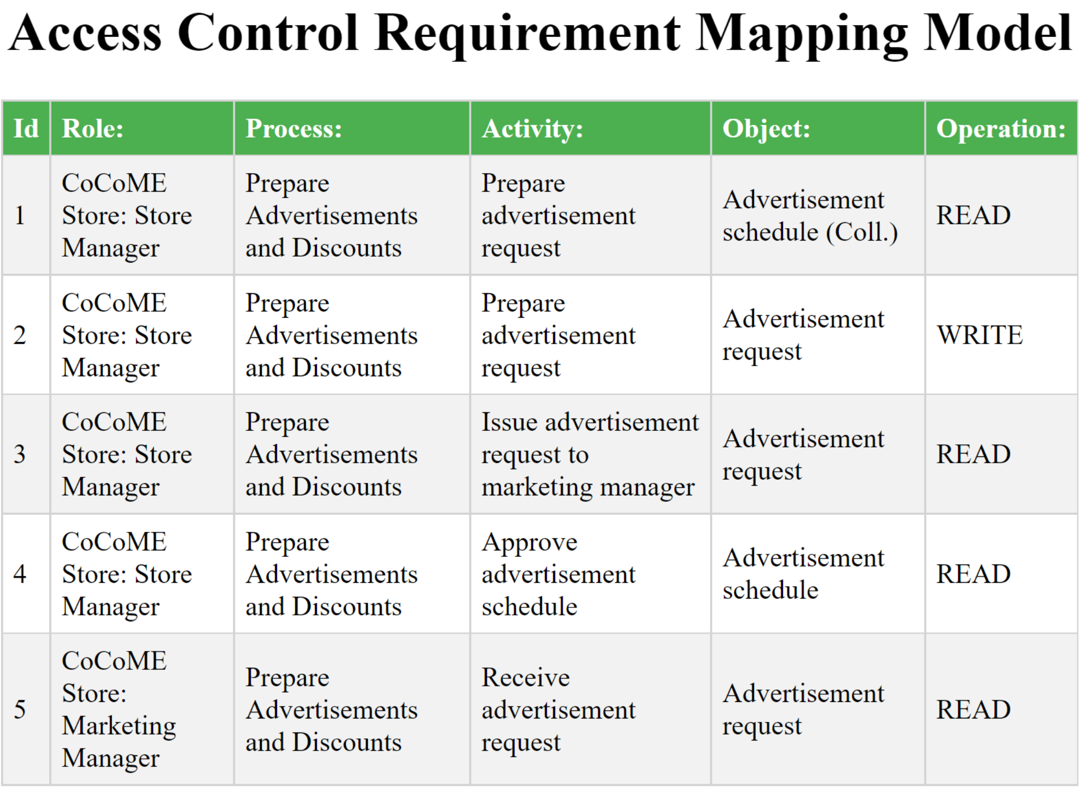


**Figure 3.15:** Shows a simple hierarchy.

The fields role and permission of the first two rows of Table 3.11 can be found identically in the role model in Table 3.12. Row three of Table 3.11 is merged into the second row of Table 3.12, as both data objects are the same. It results in the permission *READ/WRITE Advertisement request*. If the first permission of the role model in Table 3.12 would not point to the collection data object advertisement schedule but to the single data object, then the fourth row of Table 3.11 would impose the same permission as the first row of Table 3.11, *CoCoME Store: Store Manager* and *READ Advertisement schedule*. As the role model would already has this permission and the role model possesses only unique permissions, the permission from row four of Table 3.11 would be ignored. This example should illustrate how rows from the ACR mapping model are treated if they have an identical counterpart in the role model. As the first row points to the collection data object of advertisement schedule, it is not identical with the fourth row of Table 3.11 and thus, the fourth row of Table 3.11 is extracted into the third row of the role model in Table 3.12. Row five of Table 3.11 is extracted into the role model and shown in row four of Table 3.12.

The resulting role model serves security experts as an initial role model. It comprises business level ACRs extracted from business processes. Technical ACRs are missing, as they are not part of the business level knowledge. Nonetheless, the initial role model eases the role engineering process for security experts who are more technical-oriented and hence, can focus on technical parts. As a result, the overall role engineering process is less error-prone, as parts are automated and the resulting role model is better aligned with business level ACRs, allowing to backtrack design decisions by using the ACR mapping model.

There are several models BAcsTract outputs in form of HTML tables. They visualize results and help security experts during their work. An example of the ACR mapping model and the role model is shown in Figure 3.16 and Figure 3.17. While the ACR mapping model is identical with the ACR mapping model in Table 3.11, the role model shows the complete role model for the example process.

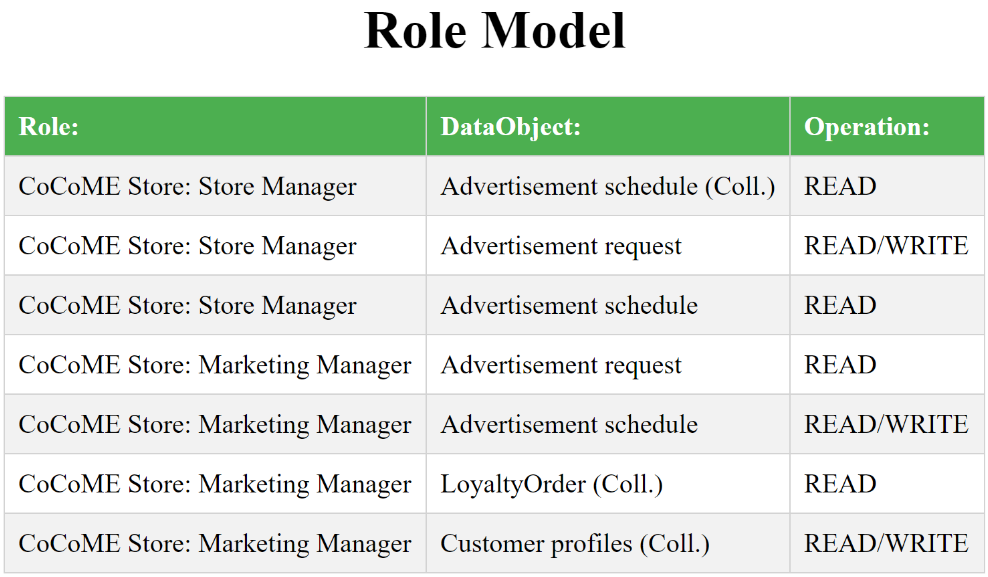


## Access Control Requirement Mapping Model

| Id | Role: | Process: | Activity: | Object: | Operation: |
|---|---|---|---|---|---|
| 1 | CoCoME Store: Store Manager | Prepare Advertisements and Discounts | Prepare advertisement request | Advertisement schedule (Coll.) | READ |
| 2 | CoCoME Store: Store Manager | Prepare Advertisements and Discounts | Prepare advertisement request | Advertisement request | WRITE |
| 3 | CoCoME Store: Store Manager | Prepare Advertisements and Discounts | Issue advertisement request to marketing manager | Advertisement request | READ |
| 4 | CoCoME Store: Store Manager | Prepare Advertisements and Discounts | Approve advertisement schedule | Advertisement schedule | READ |
| 5 | CoCoME Store: Marketing Manager | Prepare Advertisements and Discounts | Receive advertisement request | Advertisement request | READ |

**Figure 3.16:** Shows an excerpt of the HTML output of the ACR mapping model.

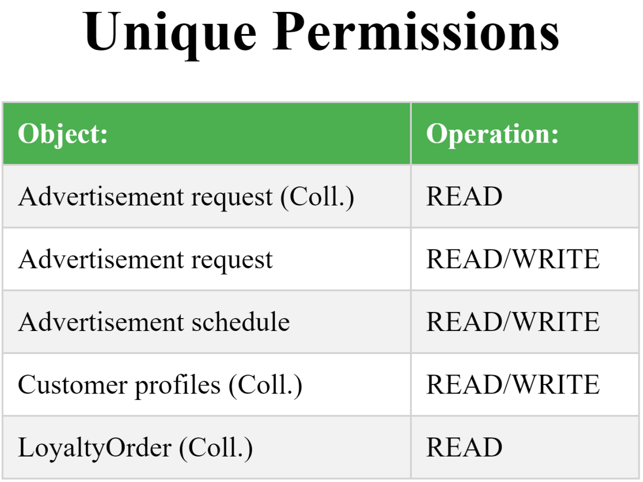
Beside these tables, BAcsTract outputs a table with unique permissions and a table summarizing processes and roles. Typically, the amount of business processes is very large. Thus, the number of roles and permissions is even larger. This makes it difficult to get an overview of roles and permissions or find a particular one. Hence, the aforementioned tables help in providing a comprehensive overview. An example for the unique permission output is shown in Figure 3.18. It provides an overview for security experts and the business level over all unique permissions, making it easier to track whether a certain permission is existing or not. Figure 3.19 provides an example for the output about processes and roles. It provides an overview over all processes and their participating roles. This helps to get an overview over the vast amount of processes and their participating roles in an organization. Both outputs help security experts and the business level to get a comprehensive overview and to find particular permissions.



## Role Model

| Role: | DataObject: | Operation: |
|---|---|---|
| CoCoME Store: Store Manager | Advertisement schedule (Coll.) | READ |
| CoCoME Store: Store Manager | Advertisement request | READ/WRITE |
| CoCoME Store: Store Manager | Advertisement schedule | READ |
| CoCoME Store: Marketing Manager | Advertisement request | READ |
| CoCoME Store: Marketing Manager | Advertisement schedule | READ/WRITE |
| CoCoME Store: Marketing Manager | LoyaltyOrder (Coll.) | READ |
| CoCoME Store: Marketing Manager | Customer profiles (Coll.) | READ/WRITE |

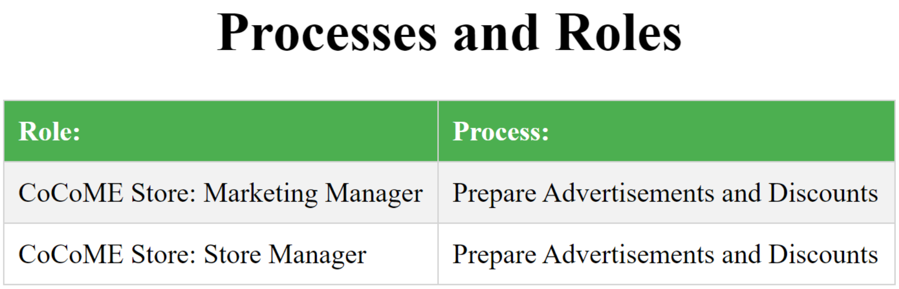
**Figure 3.17:** Shows an excerpt of the HTML output of the role model.



## Unique Permissions

| Object: | Operation: |
|---|---|
| Advertisement request (Coll.) | READ |
| Advertisement request | READ/WRITE |
| Advertisement schedule | READ/WRITE |
| Customer profiles (Coll.) | READ/WRITE |
| LoyaltyOrder (Coll.) | READ |

**Figure 3.18:** Shows a HTML output of the unique permissions.



## Processes and Roles

| Role: | Process: |
|---|---|
| CoCoME Store: Marketing Manager | Prepare Advertisements and Discounts |
| CoCoME Store: Store Manager | Prepare Advertisements and Discounts |

**Figure 3.19:** Shows a HTML output of the processes and roles.

To sum up, BAcsTract operates on business processes of an organization modeled in BPMN. They are analyzed in six steps implementing the formalized concept from Section 3.1.1.

During the six steps, relevant parts of the business processes are extracted to build an ACR mapping model interconnecting elements of business processes and RBAC. Therefore, tuples of role, process, activity and permission are built. During the final step, the role model is extracted out of the ACR mapping model. Lastly, BAcsTract creates several outputs besides the ACR mapping model and the role model that help security experts and the business level to get a comprehensive view of the results.

#### 3.2.3.3 Palladio Access Permission Extractor

PAcsTract basis on the formal concept for extracting ACRs from business processes explained in Section 3.1.1. It is a top down approach to elicit role models for RBAC automatically by analyzing business processes of an organization modeled in the PCM [189] extension IntBIIS_LP. During the extraction, an ACR mapping model is established, which was introduced in Section 3.2.3.1. Section 3.2.1.2 explained what kind of data PAcsTract consumes as input. There are several models designed in IntBIIS_LP representing the business processes in PCM: business process model, data model, organizational environment model, ELSC match AS ext container, Pool ext container. Additionally, models representing the EAA are consumed: system, repository and service effect specification. An overview over the input for PAcsTract was provided in Table 3.5. This section will introduce how PAcsTract extracts ACRs and the role model automatically. Ideally, the input processes encompass all the various departments of the organization and provide a comprehensive picture of the ongoing work done by the employees on a daily basis. In the following, the PCM and IntBIIS_LP models are explained, which are part of the running example introduced in the previous section.

Figure 3.20 shows the PCM system diagram. The round connectors on the left side and at the top of the surrounding system represent interfaces that serve service calls to the employees of the supermarket store. Rectangular boxes depict subsystems operated in the supermarket store. The system *Store* is responsible for the processes around the cash desks, the online shop and the inventory. It is connected to the *CustomerDataStore* via the interface ICustomerDataRecorder, that is used to store the orders done by customers in the supermarket. The *Store* is also connected to the *LoyaltyManagement* via the interface *ILoyaltyManagementOrderProcessing*, that is responsible for the loyalty program of the store. It is in turn connected to the *CustomerDataStore* to store orders from loyalty customers. The system *Marketing* is connected to the *CustomerDataStore* to get information about processed orders, that are required to build customer profiles from which advertisements and discounts are selected.

Figure 3.21 depicts the process *Prepare Advertisements and Discounts* modeled in IntBIIS_LP. It is equivalent to the BPMN process in Figure 3.11 and has been explained in Section 3.2.3.2. The corresponding roles are defined in the organization environment model shown in Figure 3.22.

Organizational divisions of roles are defined in Figure 3.23. For example, the role marketing manager belongs to the organizational division store.

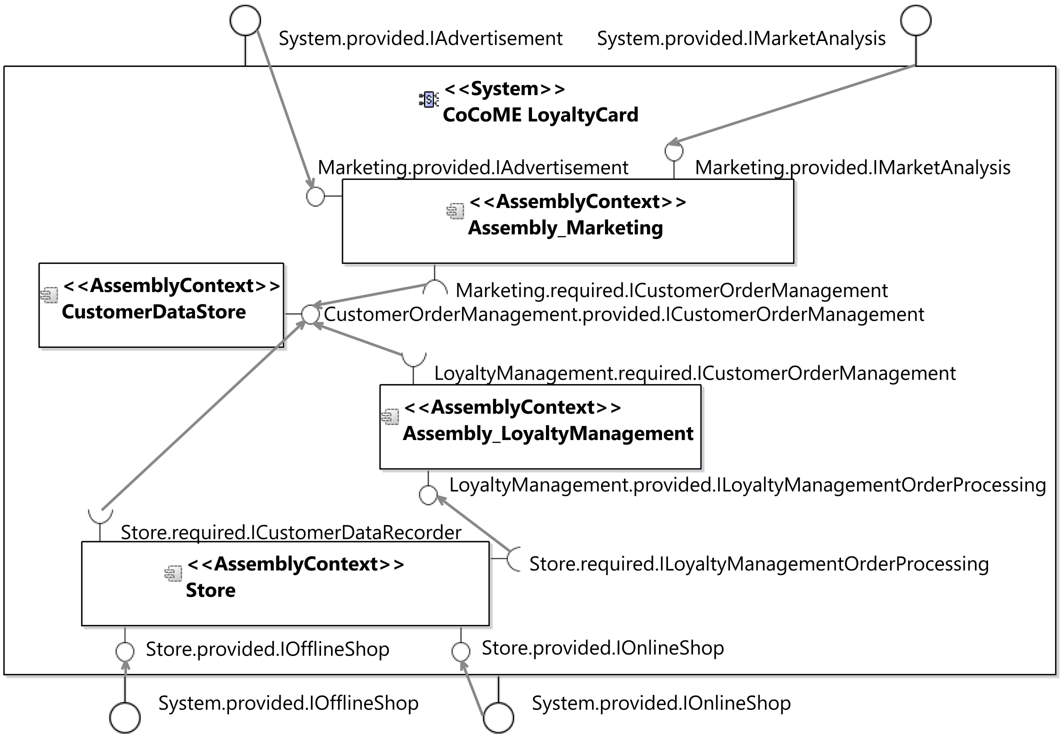
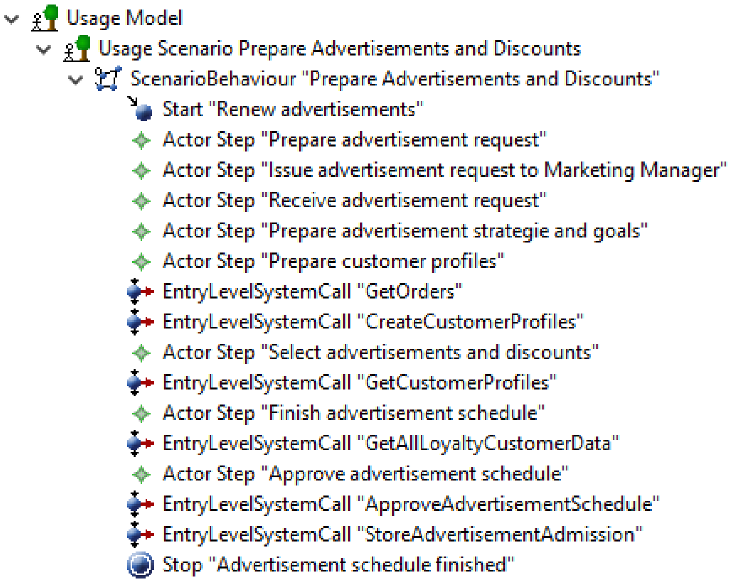

System.provided.IAdvertisement
System.provided.IMarketAnalysis
<<System>>
CoCoME LoyaltyCard
Marketing.provided.IAdvertisement
Marketing.provided.IMarketAnalysis
<<AssemblyContext>>
Assembly_Marketing
<<AssemblyContext>>
CustomerDataStore
Marketing.required.ICustomerOrderManagement
CustomerOrderManagement.provided.ICustomerOrderManagement
LoyaltyManagement.required.ICustomerOrderManagement
<<AssemblyContext>>
Assembly_LoyaltyManagement
LoyaltyManagement.provided.ILoyaltyManagementOrderProcessing
Store.required.ICustomerDataRecorder
Store.required.ILoyaltyManagementOrderProcessing
<<AssemblyContext>>
Store
Store.provided.IOfflineShop
Store.provided.IOnlineShop
System.provided.IOfflineShop
System.provided.IOnlineShop

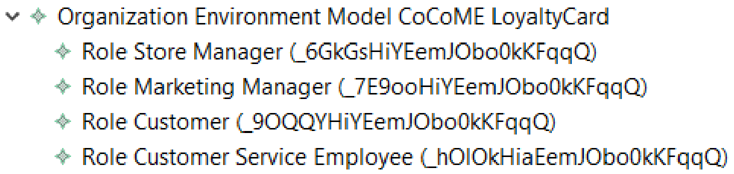
**Figure 3.20:** Shows the system diagram.



Usage Model
Usage Scenario Prepare Advertisements and Discounts
ScenarioBehaviour "Prepare Advertisements and Discounts"
Start "Renew advertisements"
Actor Step "Prepare advertisement request"
Actor Step "Issue advertisement request to Marketing Manager"
Actor Step "Receive advertisement request"
Actor Step "Prepare advertisement strategie and goals"
Actor Step "Prepare customer profiles"
EntryLevelSystemCall "GetOrders"
EntryLevelSystemCall "CreateCustomerProfiles"
Actor Step "Select advertisements and discounts"
EntryLevelSystemCall "GetCustomerProfiles"
Actor Step "Finish advertisement schedule"
EntryLevelSystemCall "GetAllLoyaltyCustomerData"
Actor Step "Approve advertisement schedule"
EntryLevelSystemCall "ApproveAdvertisementSchedule"
EntryLevelSystemCall "StoreAdvertisementAdmission"
Stop "Advertisement schedule finished"

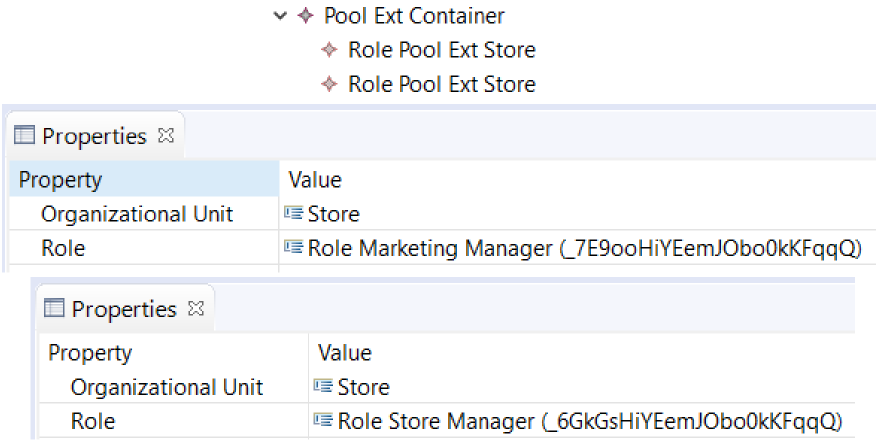
**Figure 3.21:** Shows the IntBIIS_LP process Prepare Advertisements and Discounts of the supermarket store CoCoME.

the IntBIIS_LP process in Figure 3.21 has eight actor steps that are equivalent to the activities of the BPMN process in Figure 3.11 and six EntryLevelSystemCalls. The data



Organization Environment Model CoCoME LoyaltyCard
- Role Store Manager (_6GkGsHiYEemJObo0kKFqqQ)
- Role Marketing Manager (_7E9ooHiYEemJObo0kKFqqQ)
- Role Customer (_9OQQYHiYEemJObo0kKFqqQ)
- Role Customer Service Employee (_hOlOkHiaEemJObo0kKFqqQ)

**Figure 3.22:** Shows the organization environment model.



Pool Ext Container
- Role Pool Ext Store
- Role Pool Ext Store

Properties

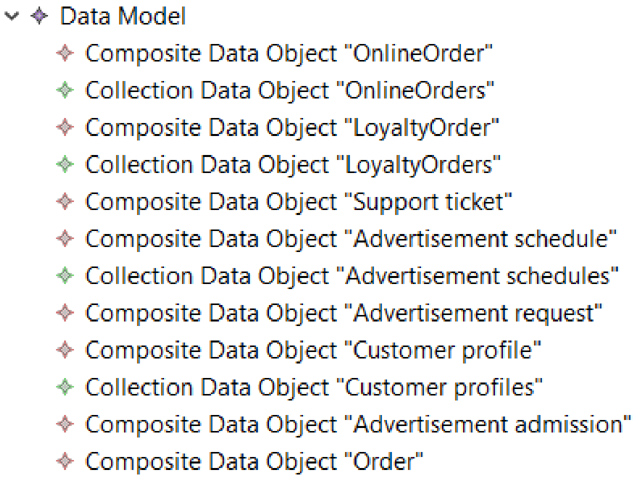
| Property | Value |
|---|---|
| Organizational Unit | Store |
| Role | Role Marketing Manager (_7E9ooHiYEemJObo0kKFqqQ) |

Properties

| Property | Value |
|---|---|
| Organizational Unit | Store |
| Role | Role Store Manager (_6GkGsHiYEemJObo0kKFqqQ) |

**Figure 3.23:** Shows the *Pool Ext Container* that defines the organizational units for roles of a process.

objects used by the actor steps and EntryLevelSystemCalls are modeled in the data model shown in Figure 3.24.

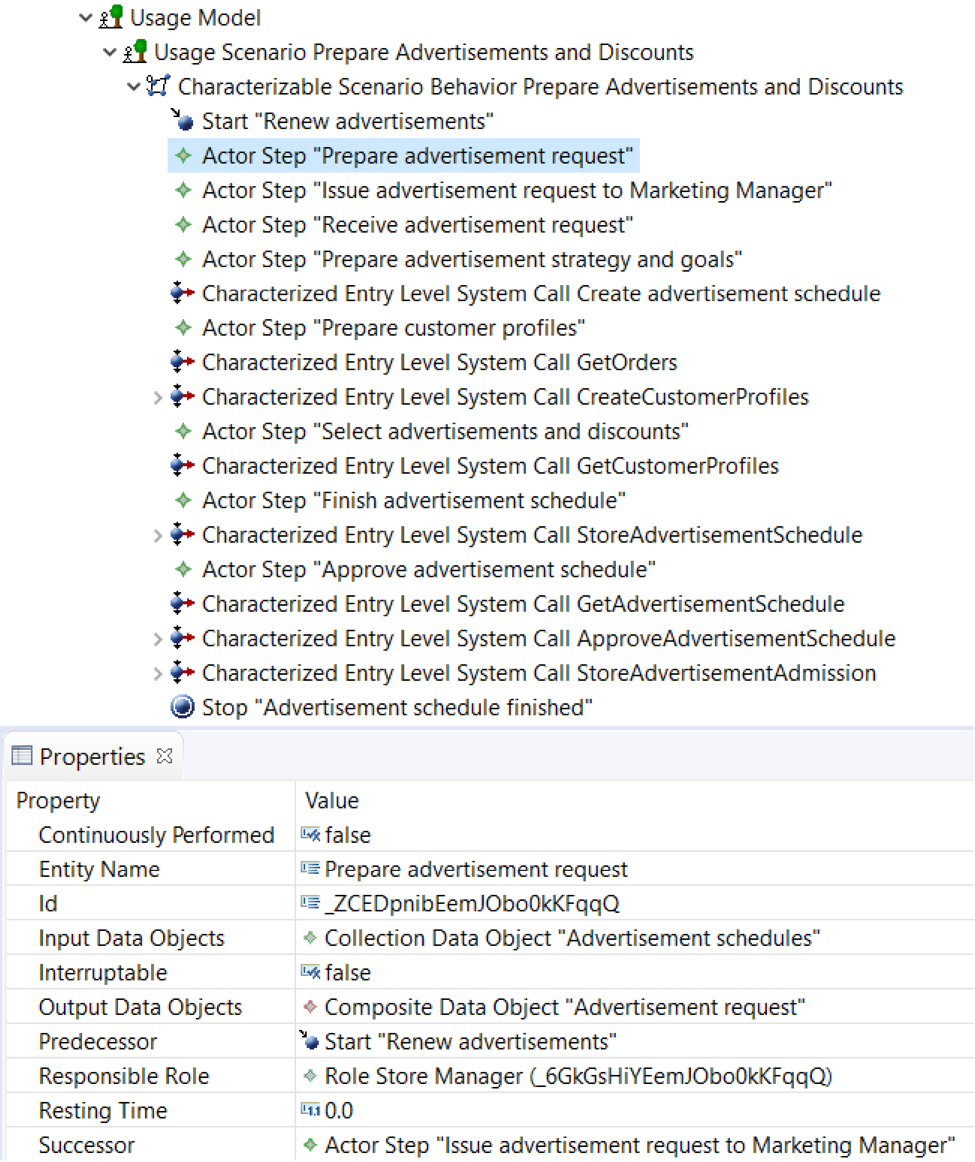


Data Model
- Composite Data Object "OnlineOrder"
- Collection Data Object "OnlineOrders"
- Composite Data Object "LoyaltyOrder"
- Collection Data Object "LoyaltyOrders"
- Composite Data Object "Support ticket"
- Composite Data Object "Advertisement schedule"
- Collection Data Object "Advertisement schedules"
- Composite Data Object "Advertisement request"
- Composite Data Object "Customer profile"
- Collection Data Object "Customer profiles"
- Composite Data Object "Advertisement admission"
- Composite Data Object "Order"

**Figure 3.24:** Shows the data model.

PAcsTract extracts ACRs from business processes by building the ACR mapping model explained in Section 3.2.3.1. Afterwards, the initial role model is extracted from the ACR mapping model. The extraction is done in six automatic steps which do not require any human interaction. Step one to four build the ACR mapping model. Step five and six create a hierarchy and form the initial role model.

*Step 1:* During the first step, roles are extracted from the IntBIIS_LP processes according to the formalization in Equation (3.30) of Section 3.1.1. Therefore, the properties of each actor step of an IntBIIS_LP process are analyzed for its responsible role. An example for the first actor step is shown in Figure 3.25.



- Usage Model
  - Usage Scenario Prepare Advertisements and Discounts
    - Characterizable Scenario Behavior Prepare Advertisements and Discounts
      - Start "Renew advertisements"
      - Actor Step "Prepare advertisement request"
      - Actor Step "Issue advertisement request to Marketing Manager"
      - Actor Step "Receive advertisement request"
      - Actor Step "Prepare advertisement strategy and goals"
      - Characterized Entry Level System Call Create advertisement schedule
      - Actor Step "Prepare customer profiles"
      - Characterized Entry Level System Call GetOrders
      - Characterized Entry Level System Call CreateCustomerProfiles
      - Actor Step "Select advertisements and discounts"
      - Characterized Entry Level System Call GetCustomerProfiles
      - Actor Step "Finish advertisement schedule"
      - Characterized Entry Level System Call StoreAdvertisementSchedule
      - Actor Step "Approve advertisement schedule"
      - Characterized Entry Level System Call GetAdvertisementSchedule
      - Characterized Entry Level System Call ApproveAdvertisementSchedule
      - Characterized Entry Level System Call StoreAdvertisementAdmission
      - Stop "Advertisement schedule finished"

Properties ☒

| Property | Value |
|---|---|
| Continuously Performed | false |
| Entity Name | Prepare advertisement request |
| Id | _ZCEDpnibEemJObo0kKFqqQ |
| Input Data Objects | Collection Data Object "Advertisement schedules" |
| Interruptable | false |
| Output Data Objects | Composite Data Object "Advertisement request" |
| Predecessor | Start "Renew advertisements" |
| Responsible Role | Role Store Manager (_6GkGsHiYEemJObo0kKFqqQ) |
| Resting Time | 0.0 |
| Successor | Actor Step "Issue advertisement request to Marketing Manager" |

**Figure 3.25:** Shows the IntBIIS_LP process Prepare Advertisements and Discounts of a supermarket store and the properties for the first actor step.

The properties for the actor step *Prepare advertisement request* in the lower part of Figure 3.25 show the responsible role *Store Manager*. For each role, the organizational division is extracted from the property *Organizational Unit* of the corresponding pool ext container. In the example shown in Figure 3.23, the organizational division for the store manager is *Store*. By considering the organizational division it is possible to distinguish roles of different organizational divisions. Table 3.13 shows an excerpt of the ACR mapping model for the example process in Figure 3.25 after step one.

**Table 3.13:** Excerpt of the ACR mapping model after step one.

| Nr. | Role | Process | Activity | Permission | Data Type |
|---|---|---|---|---|---|
| 1 | Store: Store Manager | | | | |
| 2 | Store: Store Manager | | | | |
| 3 | Store: Marketing Manager | | | | |
| 4 | Store: Marketing Manager | | | | |
| | | ... | | | |

Table 3.13 shows four of the eight entries in the ACR mapping model after step one. There is one row for each actor step containing the role and its organizational division. The extracted role *store manager* with its organizational division *store* of the first actor step is in the first row of the ACR mapping model shown in Table 3.13.

*Step 2:* In this step, the process names are extracted in which the roles participate. It formalized in Equation (3.31) of Section 3.1.1. This step is responsible for interconnecting processes and roles with each other. Each entry of the ACR mapping model is extended by the corresponding process name. The process name is found in the properties of the usage scenario shown in the top of Figure 3.25. Table 3.14 shows an excerpt of the resulting ACR mapping model after step two.

**Table 3.14:** ACR mapping model after step two.

| Nr. | Role | Process | Activity | Permission | Data Type |
|---|---|---|---|---|---|
| 1 | Store: Store Manager | Prepare Advertisements and Discounts | | | |
| 2 | Store: Store Manager | Prepare Advertisements and Discounts | | | |
| 3 | Store: Marketing Manager | Prepare Advertisements and Discounts | | | |
| 4 | Store: Marketing Manager | Prepare Advertisements and Discounts | | | |
| | | ... | | | |

*Step 3:* In step three, actor steps are extracted according to the formalization in Equation (3.32) of Section 3.1.1. Each role's actor steps in a process are analyzed and added to the ACR mapping model. The name can be found in the property *Entity Name* of each actor step and is shown in the lower part of Figure 3.25. By doing so, each role and process

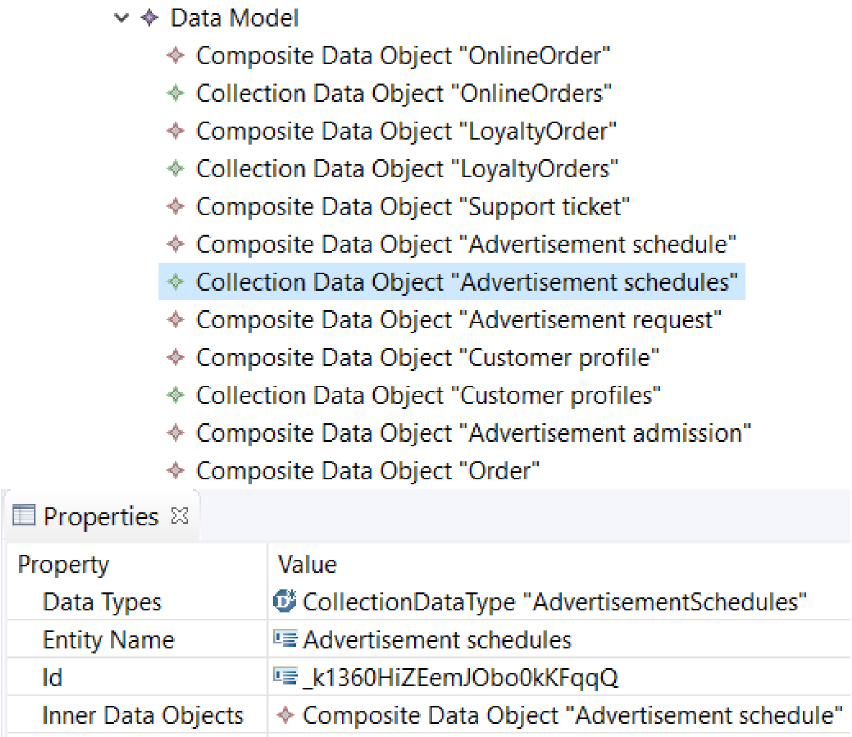
is interconnected with the role's actor steps. Table 3.15 shows an excerpt of the ACR mapping model for the example process in Figure 3.25 after step three. To each row the related actor step is added. For example, the first row belongs to the first actor step, thus *Prepare advertisement request* is added to this row.

**Table 3.15:** Excerpt of the ACR mapping model after step three.

| Nr. | Role | Process | Activity | Permission | Data Type |
|---|---|---|---|---|---|
| 1 | Store: Store Manager | Prepare Advertisements and Discounts | Prepare advertisement request | | |
| 2 | Store: Store Manager | Prepare Advertisements and Discounts | Issue advertisement request to ... | | |
| 3 | Store: Marketing Manager | Prepare Advertisements and Discounts | Receive advertisement request | | |
| 4 | Store: Marketing Manager | Prepare Advertisements and Discounts | Prepare advertisement strategy … | | |
| | | ... | | | |

*Step 4:* This step extracts permissions from actor steps of IntBIIS_LP processes, according to the formalization in Equation (3.33) of Section 3.1.1. Therefore, the properties *Input Data Objects* and *Output Data Objects* are analyzed. These properties associate data objects, that are modeled in the data model, with actor steps. In addition to the data object, IntBIIS_LP also models the data type of a data object. The data type is also extracted in this step, as it is the representation of the data object in the IT architecture. The isCollection attribute of data objects, defined by the BPMN standard, is realized through a specific collection data object. This means that the data object has the multiplicity of n. For example, the input data object *Advertisement schedules* of the first actor step in Figure 3.25 is a collection data object. In order to prepare the advertisement request the store manager has to look at all the previous advertisement schedules. Thus, the input data object *Advertisement schedules* is denoted as a collection, meaning that there are several advertisement schedules read by the store manager. The collection data object *Advertisement schedules* itself points to the inner data object of which the collection is implemented. Figure 3.26 shows the property *Inner Data Object* that points to the data object *Advertisement schedule*, meaning that the collection advertisement schedules implements a collection of the composite data object advertisement schedule.

The property *Data Types* of the data object *Advertisement schedules* in Figure 3.26 points to the data type *advertisementSchedules*. In the EAA modeled in PCM the communication between systems and components is realized via data types. Thus, signatures of service



- Data Model
  - Composite Data Object "OnlineOrder"
  - Collection Data Object "OnlineOrders"
  - Composite Data Object "LoyaltyOrder"
  - Collection Data Object "LoyaltyOrders"
  - Composite Data Object "Support ticket"
  - Composite Data Object "Advertisement schedule"
  - Collection Data Object "Advertisement schedules"
  - Composite Data Object "Advertisement request"
  - Composite Data Object "Customer profile"
  - Collection Data Object "Customer profiles"
  - Composite Data Object "Advertisement admission"
  - Composite Data Object "Order"

Properties

| Property | Value |
|---|---|
| Data Types | CollectionDataType "AdvertisementSchedules" |
| Entity Name | Advertisement schedules |
| Id | _k1360HiZEemJObo0kKFqqQ |
| Inner Data Objects | Composite Data Object "Advertisement schedule" |

**Figure 3.26:** Shows the data model with properties.

calls specify data types, e.g., getAdvertisements(date): advertisementSchedules. Data types are representations of the data objects from business processes in the EAA. That is why each data object in the data model is connected to its corresponding data type in the repository. As this data type is the representation of the business process data object, it is extracted and saved alongside with the permission in the ACR mapping model. This makes the resulting role model more aligned, as the specific IT object for the data object of the business process is the data type and is known. This means, that the extracted role model already points to the concrete IT object in its permission. The input/output associations data objects to an actor step is comparable to a read and write permission. An input data object in an actor step requires a read permission for that object and an output data object in an actor step requires a write permission for that object. For example, the first actor step in Figure 3.25 has the collection data object *Advertisement schedules* as input and the composite data object *Advertisement request* as output. As a result, the ACR mapping model will have two entries with a read to the collection advertisement schedules and a write to the composite advertisement request. Table 3.16 shows an excerpt of the ACR mapping model for the example process in Figure 3.25 after step four.

In Table 3.15 each row is extended by the extracted permissions and the data type. If there are more than one data object associated to an actor step, resulting in more than one permission, the row is duplicated for each other permission. An example for this is the first actor step of Figure 3.25. It has the collection data object *Advertisement schedules* as input and the composite data object *Advertisement request* as output. They can be found in row one and two of the ACR mapping model shown in Table 3.16. The first row

**Table 3.16:** Excerpt of the ACR mapping model after step four.

| Nr. | Role | Process | Activity | Permission | Data Type |
|---|---|---|---|---|---|
| 1 | Store: Store Manager | Prepare Advertisements and Discounts | Prepare advertisement request | READ Advertisement schedules (Coll.) | Advertisement schedules (Coll.) |
| 2 | Store: Store Manager | Prepare Advertisements and Discounts | Prepare advertisement request | WRITE Advertisement request (Comp.) | Advertisement request (Comp.) |
| 3 | Store: Store Manager | Prepare Advertisements and Discounts | Issue advertisement request to ... | READ Advertisement request (Comp.) | Advertisement request (Comp.) |
| 4 | Store: Marketing Manager | Prepare Advertisements and Discounts | Receive advertisement request | READ Advertisement request (Comp.) | Advertisement request (Comp.) |
| 5 | Store: Marketing Manager | Prepare Advertisements and Discounts | Prepare advertisement strategy … | WRITE Advertisement schedule (Comp.) | Advertisement schedule (Comp.) |
| | | | ... | | |

shows the required read permission for the advertisement schedules during the activity *Prepare advertisement request.* Row two shows the required write permission for the advertisement request during the same activity. After step four, all business level ACRs from the IntBIIS_LP processes are extracted and stored in the ACR mapping model in form of interconnected tuples of role, process, activity, permission and data type.

*Step 5:* During this step, a simple hierarchy is elicited on the basis of the ACRs in the ACR mapping model. This is done according to the formalization in Equation (3.37) of Section 3.1.1. The permissions of each role, e.g., cashier, are inspected whether they are a subset of another role, e.g., manager. If this is the case, the role manager inherits from role cashier. An example was given in Figure 3.15 of Section 3.2.3.2. The procedure is the same as in BAcsTract (further details were provided in Section 3.2.3.2).

*Step 6:* The last step extracts the initial RBAC role model from the ACR mapping model. The formal definition was introduced in Equations (3.35) and (3.36) of Section 3.1.1. Unique tuples of roles and permissions are extracted from the rows of the ACR mapping model. The row data type provides extra precision in the extracted role model. As the exact representation of a data object is known in IntBIIS_LP, the extracted permission will encompass this information. Thus, final permissions of the role model will have the data type instead of the data object inside. Take into consideration, that in the provided example the data types have the same names as their data objects. Taking design guidelines into account this is the normal case. Further on, extracted permissions are combined with the information from step five resulting in a role model with a hierarchy. The resulting role model for the ACR mapping model in Table 3.16 is shown in Table 3.17.

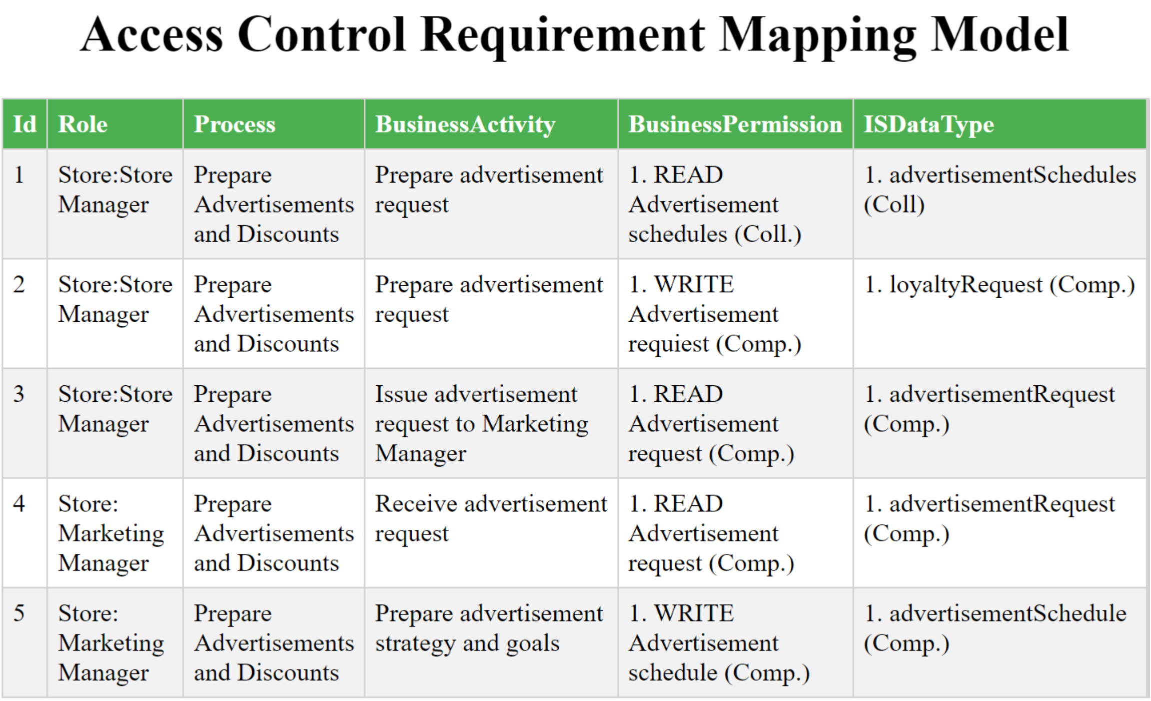
**Table 3.17:** Role model for the excerpt of the ACR mapping model from Table 3.16.

| Nr. | Role | Permission |
|---|---|---|
| 1 | Store: Store Manager | READ Advertisement schedules (Coll.) |
| 2 | Store: Store Manager | READ/WRITE Advertisement request (Comp.) |
| 3 | Store: Marketing Manager | READ Advertisement request (Comp.) |
| 4 | Store: Marketing Manager | WRITE Advertisement schedule (Comp.) |
| | | ... |

The first three rows of the role model in Table 3.17 result from the first three rows of the ACR mapping model in Table 3.16. From the first row of the ACR mapping model in Table 3.17 the role and permission are extracted for the role model and can be found in the first row of Table 3.17. Consider that the objects of the permission column are the data types of the ACR mapping model. From the second row of the ACR mapping model again the role and permission are extracted and compared whether this permission is part of the current role model. As there is no pair with *Store: Store Manager* and *READ Advertisement request (Comp.)* in the role model, it is added to the role model. Otherwise, it would not be added, as the role model has only unique role-permission pairs. The role-permission pair from the third row is extracted and compared whether there is an equivalent in the current role model. As this is not the case, it is added to the role model. Read and write permissions to the same object are combined into one row, thus the role-permission pair is combined with the role-permission pair extracted previously. The result is shown in the second row of Table 3.17: *Store: Store Manager* and *READ/WRITE Advertisement request (Comp.)*. The exemplified procedure is repeated with the fourth and fifth row of the ACR mapping model in Table 3.16, resulting in the third and fourth row of the role model in Table 3.17.

The resulting role model serves security experts as an initial role model comprising business level ACRs extracted from business processes modeled in IntBIIS_LP. The objects of the permissions in the role model are the actual data types of the EAA. Still, technical ACRs are missing, as they are not part of the business level knowledge. Nonetheless, the initial role model eases the role engineering process for security experts who are more technical-oriented and hence, can focus on technical parts. As a result, the overall role engineering process is less error-prone, as parts are automated and the resulting role model is better aligned with the business level ACRs, allowing to trace back design decisions by using the ACR mapping model.

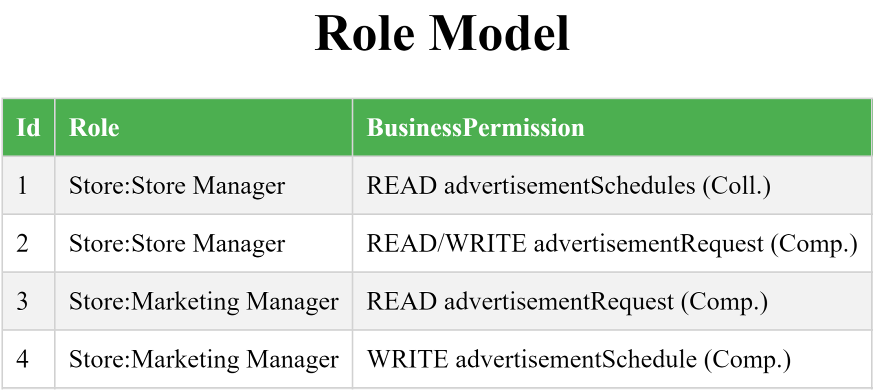
PAcsTract outputs several models in form of HTML tables. They visualize results and help security experts during their work. An example of an ACR mapping model and a role model is provided in Figure 3.27 and in Figure 3.28. Both are identical with the ACR mapping model and the role model from Table 3.16 and Table 3.17.



## Access Control Requirement Mapping Model

| Id | Role | Process | BusinessActivity | BusinessPermission | ISDataType |
|---|---|---|---|---|---|
| 1 | Store:Store Manager | Prepare Advertisements and Discounts | Prepare advertisement request | 1. READ Advertisement schedules (Coll.) | 1. advertisementSchedules (Coll) |
| 2 | Store:Store Manager | Prepare Advertisements and Discounts | Prepare advertisement request | 1. WRITE Advertisement requiest (Comp.) | 1. loyaltyRequest (Comp.) |
| 3 | Store:Store Manager | Prepare Advertisements and Discounts | Issue advertisement request to Marketing Manager | 1. READ Advertisement request (Comp.) | 1. advertisementRequest (Comp.) |
| 4 | Store: Marketing Manager | Prepare Advertisements and Discounts | Receive advertisement request | 1. READ Advertisement request (Comp.) | 1. advertisementRequest (Comp.) |
| 5 | Store: Marketing Manager | Prepare Advertisements and Discounts | Prepare advertisement strategy and goals | 1. WRITE Advertisement schedule (Comp.) | 1. advertisementSchedule (Comp.) |

**Figure 3.27:** Shows an excerpt of the HTML output of the ACR mapping model.



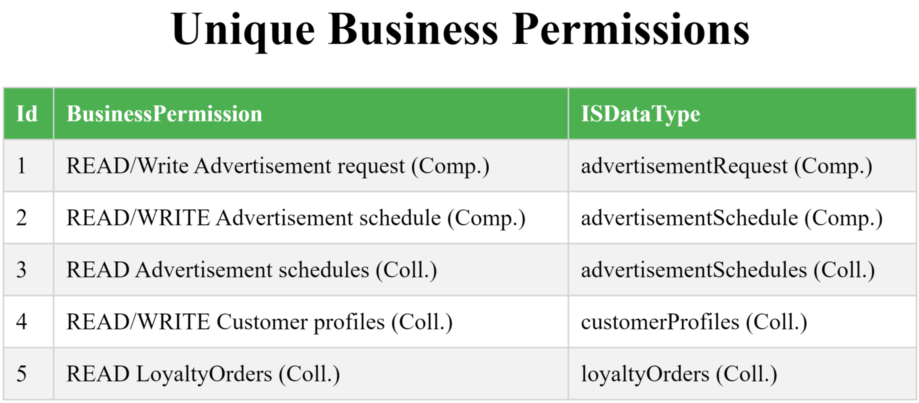
## Role Model

| Id | Role | BusinessPermission |
|---|---|---|
| 1 | Store:Store Manager | READ advertisementSchedules (Coll.) |
| 2 | Store:Store Manager | READ/WRITE advertisementRequest (Comp.) |
| 3 | Store:Marketing Manager | READ advertisementRequest (Comp.) |
| 4 | Store:Marketing Manager | WRITE advertisementSchedule (Comp.) |

**Figure 3.28:** Shows an excerpt of the HTML output of the role model.

As with BAcsTract, PAcsTract outputs several more tables. A table with unique permissions and a table summarizing processes and roles. Typically, the amount of business processes is very large. Thus, the number of roles and permissions is even larger, making it difficult to get an overview over roles and permission or to find a particular one. Hence, the aforementioned tables help in providing a comprehensive overview. An example for the unique permissions output is shown in Figure 3.29. It provides an overview for security experts and the business level over all unique permissions, making it easier to track whether a certain permission is existing or not. Figure 3.30 provides an example for the output about processes and roles. It provides an overview over all processes and their participating roles, helping to get an overview over the vast amount of processes in an organization.

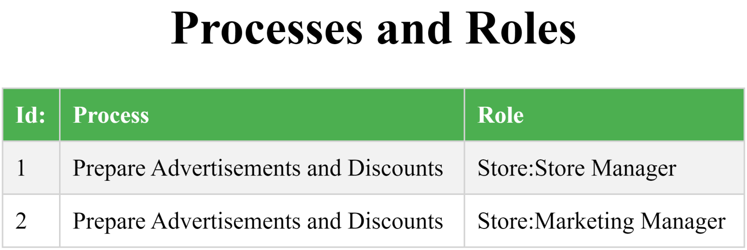
Both outputs help security experts and the business level to get a comprehensive overview and to find particular permissions.



Unique Business Permissions

| Id | BusinessPermission | ISDataType |
|---|---|---|
| 1 | READ/Write Advertisement request (Comp.) | advertisementRequest (Comp.) |
| 2 | READ/WRITE Advertisement schedule (Comp.) | advertisementSchedule (Comp.) |
| 3 | READ Advertisement schedules (Coll.) | advertisementSchedules (Coll.) |
| 4 | READ/WRITE Customer profiles (Coll.) | customerProfiles (Coll.) |
| 5 | READ LoyaltyOrders (Coll.) | loyaltyOrders (Coll.) |

**Figure 3.29:** Shows a HTML output of the unique permissions.



Processes and Roles

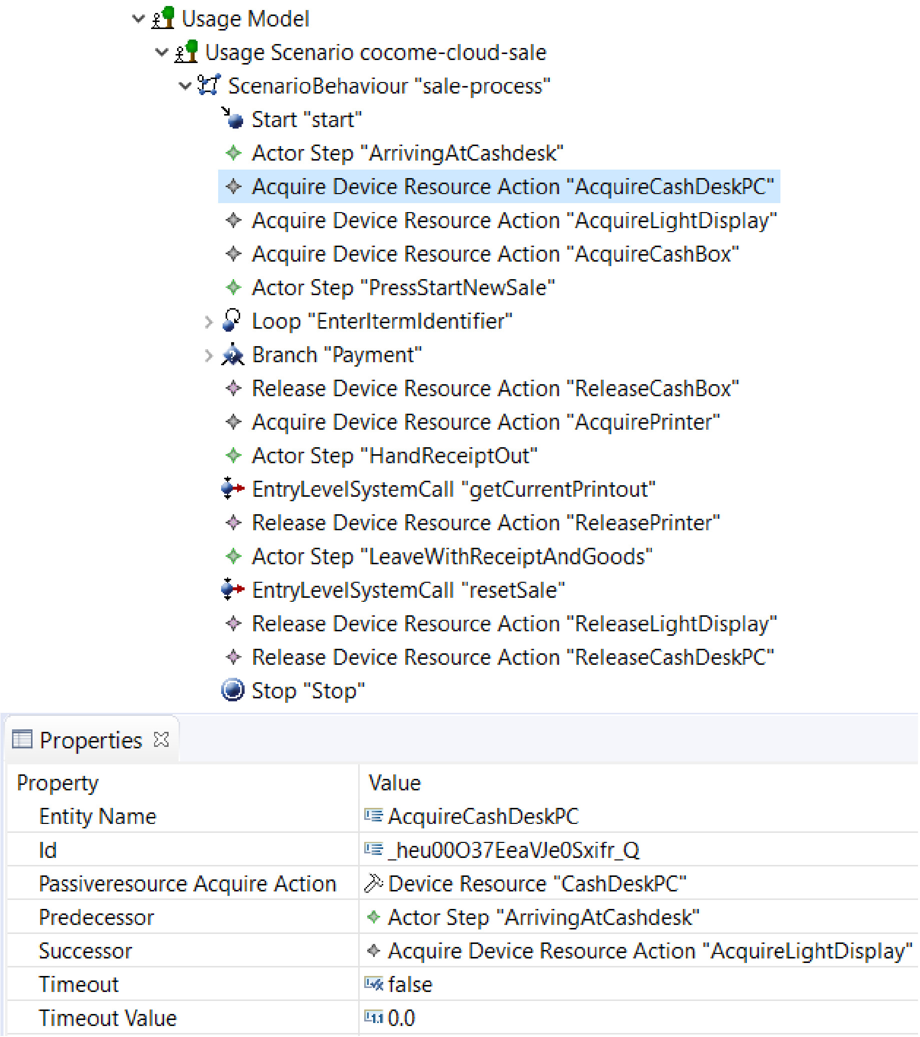
| Id: | Process | Role |
|---|---|---|
| 1 | Prepare Advertisements and Discounts | Store:Store Manager |
| 2 | Prepare Advertisements and Discounts | Store:Marketing Manager |

**Figure 3.30:** Shows a HTML output of the processes and roles.

To sum up, PAcsTract operates on business processes that are modeled in IntBIIS_LP. These processes are analyzed in six steps. This procedure implements the formalized concept from Section 3.1.1. During the six steps relevant parts of the business processes are extracted to build an ACR mapping model that interconnects elements of IntBIIS_LP processes, RBAC and EAA modeled in PCM. Therefore, tuples of role, process, actor step, business permission and data type are built. In the final step the role model is extracted out of the ACR mapping model. Lastly, PAcsTract creates several outputs besides the ACR mapping model and the role model that help security experts and the business level to get a comprehensive overview of the results. Further details on further outputs and extracted ACRs resulting from acquire and release device resource actions are presented in the following section.

#### 3.2.3.4 Palladio Access Permission Extractor and Device Resources

Business processes modeled in IntBIIS_LP are coupled with the corresponding IT architecture. Besides the system step, there is another class of elements that possess interesting information in terms of access control. Namely, the acquire device resource action and release device resource action. They allow to define when a certain device or machine is



```
Usage Model
  Usage Scenario cocome-cloud-sale
    ScenarioBehaviour "sale-process"
      Start "start"
      Actor Step "ArrivingAtCashdesk"
      Acquire Device Resource Action "AcquireCashDeskPC"
      Acquire Device Resource Action "AcquireLightDisplay"
      Acquire Device Resource Action "AcquireCashBox"
      Actor Step "PressStartNewSale"
    > Loop "EnterItermIdentifier"
    > Branch "Payment"
      Release Device Resource Action "ReleaseCashBox"
      Acquire Device Resource Action "AcquirePrinter"
      Actor Step "HandReceiptOut"
      EntryLevelSystemCall "getCurrentPrintout"
      Release Device Resource Action "ReleasePrinter"
      Actor Step "LeaveWithReceiptAndGoods"
      EntryLevelSystemCall "resetSale"
      Release Device Resource Action "ReleaseLightDisplay"
      Release Device Resource Action "ReleaseCashDeskPC"
      Stop "Stop"
```

| Property | Value |
|---|---|
| Entity Name | AcquireCashDeskPC |
| Id | _heu00O37EeaVJe0Sxifr_Q |
| Passiveresource Acquire Action | Device Resource "CashDeskPC" |
| Predecessor | Actor Step "ArrivingAtCashdesk" |
| Successor | Acquire Device Resource Action "AcquireLightDisplay" |
| Timeout | false |
| Timeout Value | 0.0 |

**Figure 3.31:** Shows an IntBIIS_LP process of a supermarket store and the properties for the first acquire device resource action.

used by a person during a business process. Figure 3.31 shows the acquire device resource action and release device resource action usage in an IntBIIS_LP process.

The lower part of Figure 3.31 shows the properties of the highlighted acquire device resource action. *Entity Name* is the property where the name of the particular action is defined. In the property *Passiveresource Acquire Action* the device is selected that is acquired by this action. In this case, the cash desk PC is acquired. Device resources are modeled in the organization environment model alongside with roles. In the *ELSC Match AS Ext* the acquire and release device resource actions are connected to their actor steps in which the device is acquired or released. Properties of the release device resource action are the same as of the acquire device resource action.

In the context of an IntBIIS_LP process an acquire device resource action models that a certain device or machine, e.g., a cash desk PC or a cash box, is taken during an actor step. The acquired device is hold until an actor step is associated with a release device resource action. In such an actor step, the previously acquired device is freed and afterwards, can be acquired by others again. In the context of access control, the acquire device resource action may require a permission to acquire a certain device. Especially in areas of critical infrastructures devices as control units and database terminals have high security regulations and need to be protected. In some cases, they are protected with additional passwords and access rights, in others, they are protected by key cards or restricted access areas. In all of these cases, an additional access right is required to acquire the desired device. On this basis, it is possible to extract an ACR from an acquire device resource action for the executing role.

In the example of Figure 3.31, a cash desk PC and a cash box are acquired by the cashier of the supermarket store. A cashier acquired for the time of the sales process a cash box with a specific amount of money for which he is accountable for. To process sales of customers in a supermarket the cashier requires access to the cash desk PC of the cash desk. As stated by the directive [82] from the German ministry for finance, all sale processes have to be secured and be accountable to the processing employee. Hence, the cashier requires an access right to log in to the cash desk PC and may also require access rights to the deposit box where the cash box is stored. As a result, two ACRs can be extracted for the cashier, one ACR for the cash desk PC and one for the cash box.

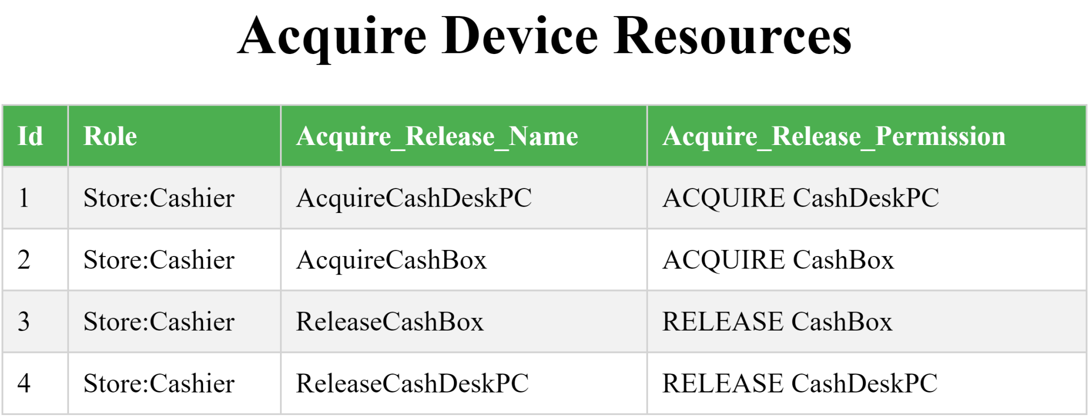
In step three, of the extraction process of PAcsTract explained in Section 3.2.3.3, the IntBIIS_LP process is additionally analyzed for acquire and release device resource actions. For each action the connected actor step is extracted from the *ELSC Match AS Ext*. The acquire device resource action is stored alongside with its actor step in a new row inside the ACR mapping model. Table 3.18 shows an example.

**Table 3.18:** Excerpt of the ACR mapping model with acquire device resource actions.

| Nr. | Role | Process | Activity | IS Permission |
|---|---|---|---|---|
| 1 | Store: Cashier | Process Sale | Arriving at cash desk | Acquire Cash Box |
| 2 | Store: Cashier | Process Sale | Arriving at cash desk | Acquire Cash Desk PC |
| | | ... | | |

There are several models in form of HTML tables that PAcsTract produces. They visualize results and help security experts and the business level in understanding the resulting ACRs. PAcsTract provides an output for the overview of all acquire and release device resources used by roles. An example is shown in fig:PAcsTract:OutputAcquireDeviceResources.

Figure 3.32 illustrates an excerpt of the output that shows the overview about roles and their acquire and release device resources. The first row of Figure 3.32 shows the role cashier, followed by the name of the acquire device resource action and the actual device resource that is acquired. Rows two to four show further acquire and release device resource actions of the role cashier.

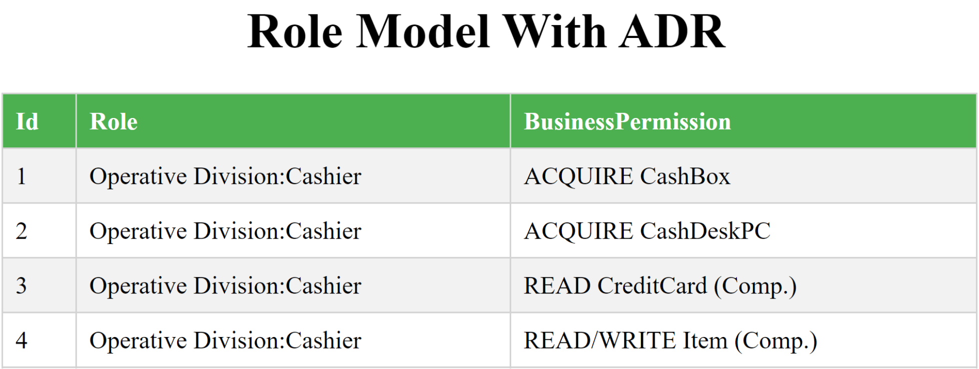


# Acquire Device Resources

| Id | Role | Acquire_Release_Name | Acquire_Release_Permission |
|---|---|---|---|
| 1 | Store:Cashier | AcquireCashDeskPC | ACQUIRE CashDeskPC |
| 2 | Store:Cashier | AcquireCashBox | ACQUIRE CashBox |
| 3 | Store:Cashier | ReleaseCashBox | RELEASE CashBox |
| 4 | Store:Cashier | ReleaseCashDeskPC | RELEASE CashDeskPC |

**Figure 3.32:** Shows an excerpt of the HTML output about the overview of roles and their acquire and release device resource actions.

Another output illustrates the role model with extended ACRs for the acquire device resources. The output of PAcsTract shown in Figure 3.33 depicts this.



# Role Model With ADR

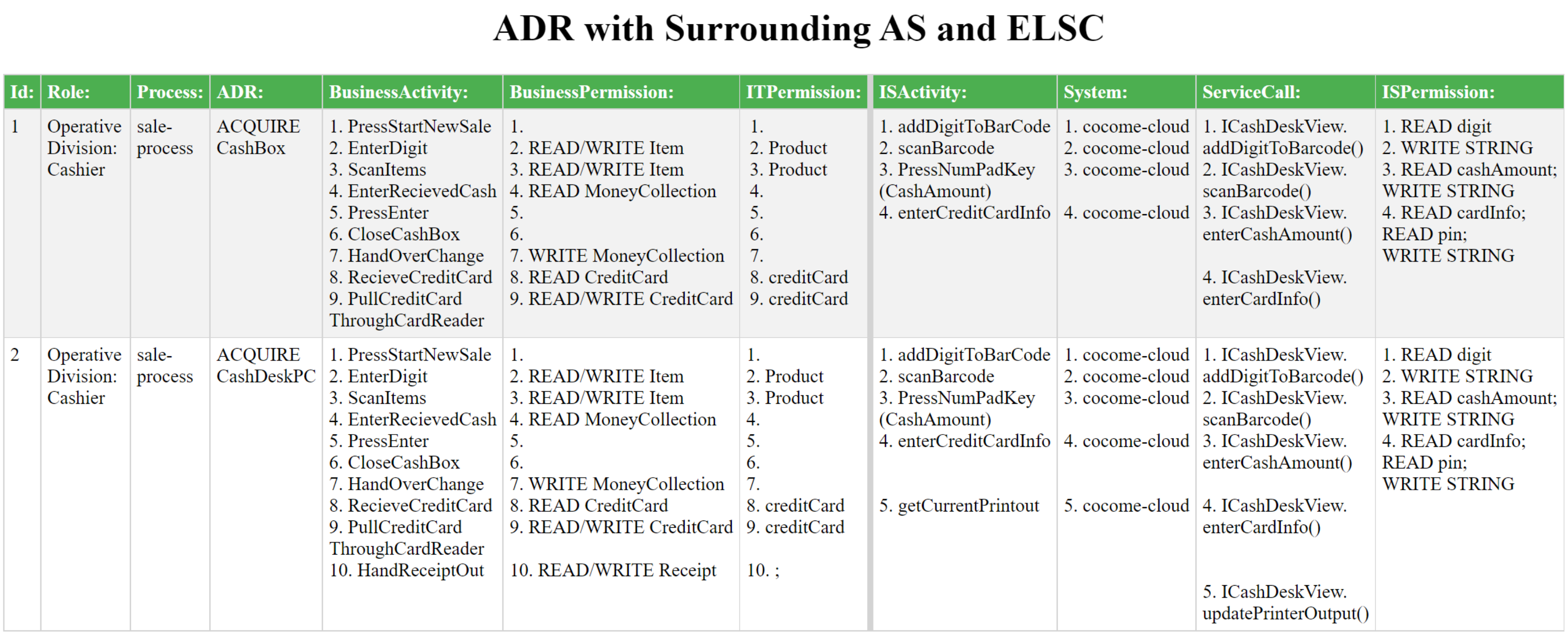
| Id | Role | BusinessPermission |
|---|---|---|
| 1 | Operative Division:Cashier | ACQUIRE CashBox |
| 2 | Operative Division:Cashier | ACQUIRE CashDeskPC |
| 3 | Operative Division:Cashier | READ CreditCard (Comp.) |
| 4 | Operative Division:Cashier | READ/WRITE Item (Comp.) |

**Figure 3.33:** Shows a HTML output of the role model with ACRs for acquire device resource actions.

The first two rows in Figure 3.33 show the ACRs resulting from the acquire device resources actions, implicating that permissions are required to acquire the cash box and the cash desk PC. They are followed by normal permissions of the role model explained in the previous section.

Another output provided by PAcsTract is an extension to the ACR mapping model. Figure 3.34 shows the output. This output relates acquire and release device resources actions with all actor and system steps that are done by the role during the possession of the device resources. In particular, this allows to track across which actor and system steps of a role the device resource is actively required.

The first row of Figure 3.34 illustrates the acquire device resource action *ACQUIRE CashBox* of the role *Store:Cashier* in the process *sale-process*. In the column *BusinessActivity* all actor steps are consecutively numbered during which the cashier has the cash box in use. The columns *BusinessPermission* and *ISDataType*, known from the ACR mapping model, enumerate for each actor step the required access permissions and corresponding data types. In the column *ISActivity* all system steps are consecutively numbered during which the cashier has the cash box in use. The following columns *System*, *ServiceCall* and *ISPermission* depict the permissions for data types referenced by the signatures of the service calls called in the system steps. They form triples of system, service call of an interface and permission.



# ADR with Surrounding AS and ELSC

| Id: | Role: | Process: | ADR: | BusinessActivity: | BusinessPermission: | ITPermission: | ISActivity: | System: | ServiceCall: | ISPermission: |
|---|---|---|---|---|---|---|---|---|---|---|
| 1 | Operative Division: Cashier | sale-process | ACQUIRE CashBox | 1. PressStartNewSale<br>2. EnterDigit<br>3. ScanItems<br>4. EnterRecievedCash<br>5. PressEnter<br>6. CloseCashBox<br>7. HandOverChange<br>8. RecieveCreditCard<br>9. PullCreditCard ThroughCardReader | 1.<br>2. READ/WRITE Item<br>3. READ/WRITE Item<br>4. READ MoneyCollection<br>5.<br>6.<br>7. WRITE MoneyCollection<br>8. READ CreditCard<br>9. READ/WRITE CreditCard | 1.<br>2. Product<br>3. Product<br>4.<br>5.<br>6.<br>7.<br>8. creditCard<br>9. creditCard | 1. addDigitToBarCode<br>2. scanBarcode<br>3. PressNumPadKey (CashAmount)<br>4. enterCreditCardInfo | 1. cocome-cloud<br>2. cocome-cloud<br>3. cocome-cloud<br>4. cocome-cloud | 1. ICashDeskView. addDigitToBarcode()<br>2. ICashDeskView. scanBarcode()<br>3. ICashDeskView. enterCashAmount()<br>4. ICashDeskView. enterCardInfo() | 1. READ digit<br>2. WRITE STRING<br>3. READ cashAmount; WRITE STRING<br>4. READ cardInfo; READ pin; WRITE STRING |
| 2 | Operative Division: Cashier | sale-process | ACQUIRE CashDeskPC | 1. PressStartNewSale<br>2. EnterDigit<br>3. ScanItems<br>4. EnterRecievedCash<br>5. PressEnter<br>6. CloseCashBox<br>7. HandOverChange<br>8. RecieveCreditCard<br>9. PullCreditCard ThroughCardReader<br>10. HandReceiptOut | 1.<br>2. READ/WRITE Item<br>3. READ/WRITE Item<br>4. READ MoneyCollection<br>5.<br>6.<br>7. WRITE MoneyCollection<br>8. READ CreditCard<br>9. READ/WRITE CreditCard<br>10. READ/WRITE Receipt | 1.<br>2. Product<br>3. Product<br>4.<br>5.<br>6.<br>7.<br>8. creditCard<br>9. creditCard<br>10. ; | 1. addDigitToBarCode<br>2. scanBarcode<br>3. PressNumPadKey (CashAmount)<br>4. enterCreditCardInfo<br>5. getCurrentPrintout | 1. cocome-cloud<br>2. cocome-cloud<br>3. cocome-cloud<br>4. cocome-cloud<br>5. cocome-cloud | 1. ICashDeskView. addDigitToBarcode()<br>2. ICashDeskView. scanBarcode()<br>3. ICashDeskView. enterCashAmount()<br>4. ICashDeskView. enterCardInfo()<br>5. ICashDeskView. updatePrinterOutput() | 1. READ digit<br>2. WRITE STRING<br>3. READ cashAmount; WRITE STRING<br>4. READ cardInfo; READ pin; WRITE STRING |

**Figure 3.34:** Shows a HTML output of the acquire device resource actions and their surrounding actor steps and *EntryLevelSystemCalls*.

### 3.2.4 Identification of Access Control Requirement Breaches in Enterprise Application Architectures

The approach AcsALign described in this section realizes the concept from Section 3.1.2 to identify ACR breaches in EAAs automatically. AcsALign operates on an EAA defined in PCM (Section 2.4) and business processes defined in IntBIIS_LP (Section 3.2.1.2). It consumes ACRs extracted with PAcsTract (see Section 3.2.3.3) to generate data flow constraints for service calls of the EAA. During the generation of data flow constraints the previously established ACR mapping model from PAcsTract is extended with elements of the EAA. Hence, the ACR mapping model interconnects elements of access control, business processes with elements of the EAA. This extension is explained in Section 3.2.4.1. Afterwards, Section 3.2.4.2 explains how AcsALign works along with the rule set used to analyze whether the data flows fulfill their corresponding data flow constraints or not.

#### 3.2.4.1 Access Control Requirements Mapping Model Extension

Section 3.1.1 has introduced the concept of an ACR mapping model that is built during the extraction of ACRs from business processes. This concept was refined in Section 3.2.3.1 for BPMN and PCM. The introduced ACR mapping model aligns business processes with RBAC in terms of ACRs. Section 3.1.2 introduced a concept to extend this ACR mapping model with information from the EAA. This section explains how this ACR mapping model extension is built by AcsALign during the generation of data flow constraints. The extended ACR mapping model provides a traceability between elements of business processes and RBAC with elements of the EAA. Apart from the usage during the generation of the data flow constraints, the ACR mapping model provides a documentation of design decisions and allows to trace data flow breaches by service calls back to their ACRs, RBAC permissions and affected business process elements. On the on hand, this allows to better understand the complex, mutual dependencies of business processes and EAAs and provides arguments why certain elements have to be realized. On the other hand, it helps the enterprise architect to elaborate on the breach with responsible employees, for example, the business process owner and helps him to better understand why the data flow breach occurred to resolve it correctly. This support in understanding the cause of the data flow breach is especially helpful during evolution scenarios of business processes and EAAs, as then extensive changes are made that might produce ACR breaches.

The ACR mapping model built by BAcsTract and PAcsTract (see Section 3.2.3) consists of four layers and is shown in the left part of Figure 3.35: role, process, activity and permission. Each role in a business processes requires a set of permissions, for example, an access permission to a certain document, to fulfill his activities. These permissions are stored in an access control system such as RBAC. The ACR mapping model is extended by a system, interface and service call layer representing EAA elements that are invoked during the fulfillment of an activity.

The right part of Figure 3.35 shows the layers that are extended by AcsALign. An example of an extended entry in the ACR mapping model is illustrated in Figure 3.36.

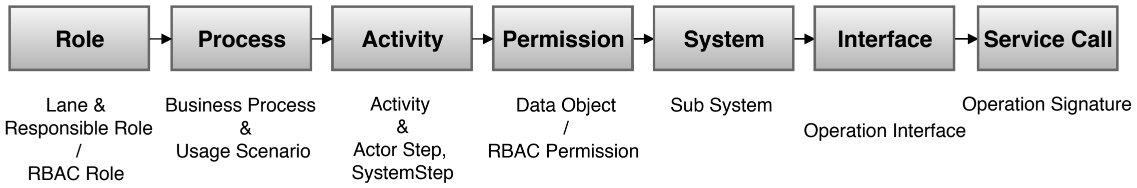

Role
Process
Activity
Permission
System
Interface
Service Call
Lane & Responsible Role / RBAC Role
Business Process & Usage Scenario
Activity & Actor Step, SystemStep
Data Object / RBAC Permission
Sub System
Operation Interface
Operation Signature

**Figure 3.35:** Shows the ACR mapping model extended with EAA elements by AcsALign.

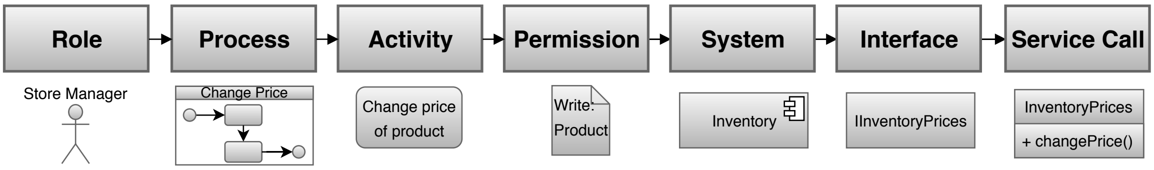


**Figure 3.36:** Illustrates an example of an extended entry in the ACR mapping model.

- **Service call**: This layer is part of the EAA and represents a service call. Throughout the fulfillment of an activity an employee interacts with IT systems. During this interaction he invokes services of IT systems. For example, to change the price of a product the service call *changePrice()* is invoked (shown in Figure 3.36). In order to fulfill an activity several service calls might need to be invoked.
- **Interface**: This layer represents the interface element of the EAA. The interface clusters a set of logically familiar service calls together so that they can be exposed to other systems as a coherent bundle. For example, the Interface *IInventoryPrice* clusters service calls required to modify the prices of products in the inventory (shown in Figure 3.36).
- **System**: This layer represents the system element of the EAA. A system groups interrelated components that interact with each other to encapsulate high-level behavior. It exposes interfaces so that other systems can interact with it. For example, the system *Inventory* encapsulates all functionalities to realize an inventory. The interface *IInventoryPrice* with its service call *changePrice()* is a part of it (shown in Figure 3.36).

#### 3.2.4.2 Access Permission Architecture Aligner

AcsALign basis on the formal concept of identifying ACR breaches in EAAs that was explained in Section 3.1.2. The approach is built on top of PAcsTract meaning that it consumes ACRs extracted by PAcsTract from business processes. These ACRs are transformed into data flow constraints and verified if the given EAA fulfills them. AcsALign does all steps automatically and does not require any human interaction. A violated data flow constraint indicates a flaw in the EAA, meaning that it is not aligned with business level needs regarding ACRs. During this analysis the ACR mapping model established by PAcsTract is extended with elements from the EAA as introduced in Section 3.2.4.1. AcsALign operates on an EAA defined in PCM (Section 2.4) and business processes defined in IntBIIS_LP (Section 3.2.1.2). Section 3.2.1.3 explained the data consumed by AcsALign as input. The following models representing the EAA are consumed: system, repository and service effect specification. A set of ACRs for the given EAA is consumed from the output of PAcsTract. A mapping for data objects from business processes to data types of the EAA and a mapping for activities from business processes to service calls of the EAA is already provided by IntBIIS_LP processes and can be found in the data model and the ELSC match AS ext container. Finally, a set of read and written data types of invoked servicses is provided by the Palladio extension Data Centric Palladio (DC-PCM) [202]. Table 3.19 shows an overview of the input data. This section will explain how AcsALign identifies ACR breaches in EAAs. In the following, the PCM models are explained, which are part of the running example required to understand how AcsALign is working.

**Table 3.19:** Data input for AcsALign.

| Input Data | Description |
|---|---|
| System | Model for the interconnection of systems, components and interfaces. |
| Repository | Model for systems, components, interfaces and data types. |
| Service Effect Specification | Model for the behavioral descriptions of components. |
| ACRs | A set of ACRs for the given EAA. |
| Data Model | Model for data objects and the mapping of data objects from business processes to data types of the EAA. |
| ELSC Match AS Ext Container | Model for the interconnection of system steps and acquire/release device resource actions with actor steps, providing a mapping of activities from business processes to service calls of the EAA. |
| Read and Written Data Types | A set of read and written data types for the service calls of the EAA. |

In the following, the PCM models are explained, which are part of the running example required to understand how AcsALign is working. The running example for AcsALign basis on the running example of the CoCoME supermarket explained in Section 3.2.3.3. It contains the basic CoCoME version with an additional loyalty program and a marketing division. Furthermore, the supermarket undergoes an evolution scenario during which necessary business processes and software systems of CoCoME are extended to include an online shop. In the course of this evolution scenario the enterprise architect makes logical and design mistakes while interpreting the requirements of the business level. Logical mistakes arise from faults and false solution approaches. Design mistakes arise from unclear, false interpretation and misunderstanding of requirements.

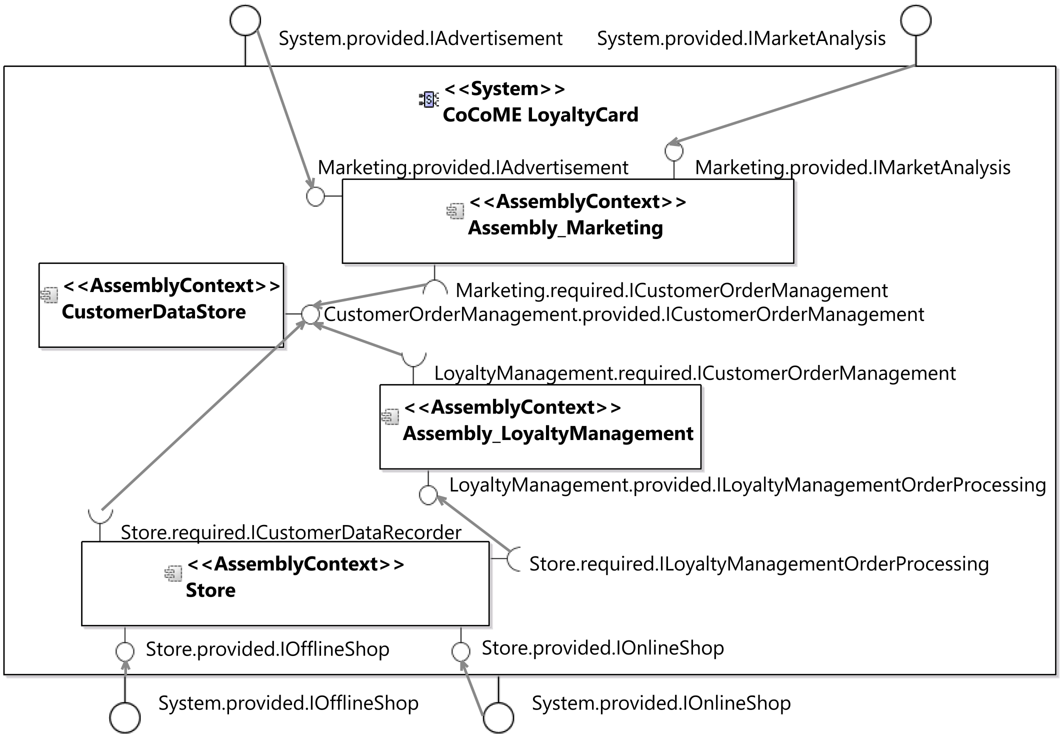

System.provided.IAdvertisement
System.provided.IMarketAnalysis
<<System>>
CoCoME LoyaltyCard
Marketing.provided.IAdvertisement
Marketing.provided.IMarketAnalysis
<<AssemblyContext>>
Assembly_Marketing
<<AssemblyContext>>
CustomerDataStore
Marketing.required.ICustomerOrderManagement
CustomerOrderManagement.provided.ICustomerOrderManagement
LoyaltyManagement.required.ICustomerOrderManagement
<<AssemblyContext>>
Assembly_LoyaltyManagement
LoyaltyManagement.provided.ILoyaltyManagementOrderProcessing
Store.required.ICustomerDataRecorder
<<AssemblyContext>>
Store
Store.required.ILoyaltyManagementOrderProcessing
Store.provided.IOfflineShop
Store.provided.IOnlineShop
System.provided.IOfflineShop
System.provided.IOnlineShop

**Figure 3.37:** Shows the system diagram of the supermarket enterprise CoCoME.

Figure 3.37 shows the PCM system diagram of the running example. It has four systems. The system *Store* is responsible for the processes around the cash desks, the online shop and the inventory. It is connected to the *CustomerDataStore* via the interface *ICustomerDataRecorder*, that is used to store orders done by customers in the supermarket. The *Store* is also connected to the *LoyaltyManagement* via the interface *ILoyaltyManagementOrderProcessing*, that is responsible for the loyalty program of the store. It is in turn connected to the *CustomerDataStore* to store orders from loyalty customers. The system *Marketing* is connected to the *CustomerDataStore* to get information about processed orders, that are required to build customer profiles from which advertisements and discounts are derived.

Figure 3.38 shows the components of the system *Store*. The component *CashDesk* contains the functionalities around the cash desks in the store. The component *Inventory* is responsible for handling all the goods that the store owns. Finally, the new component *OnlineShop* handles the services around the online shop.

Figure 3.39 depicts the business process *Prepare Advertisements and Discounts* defined in IntBIIS_LP. It is equivalent to the BPMN process in Figure 3.11 and has been explained in Section 3.2.3.2. In short, the process describes how new advertisements are made. A store manager requests a marketing manager to create an advertisement schedule and approves it. The marketing manager creates the schedule by creating and analyzing customer profiles build from loyalty orders, defining an advertisement strategy and selecting appropriate discounts to advertise.

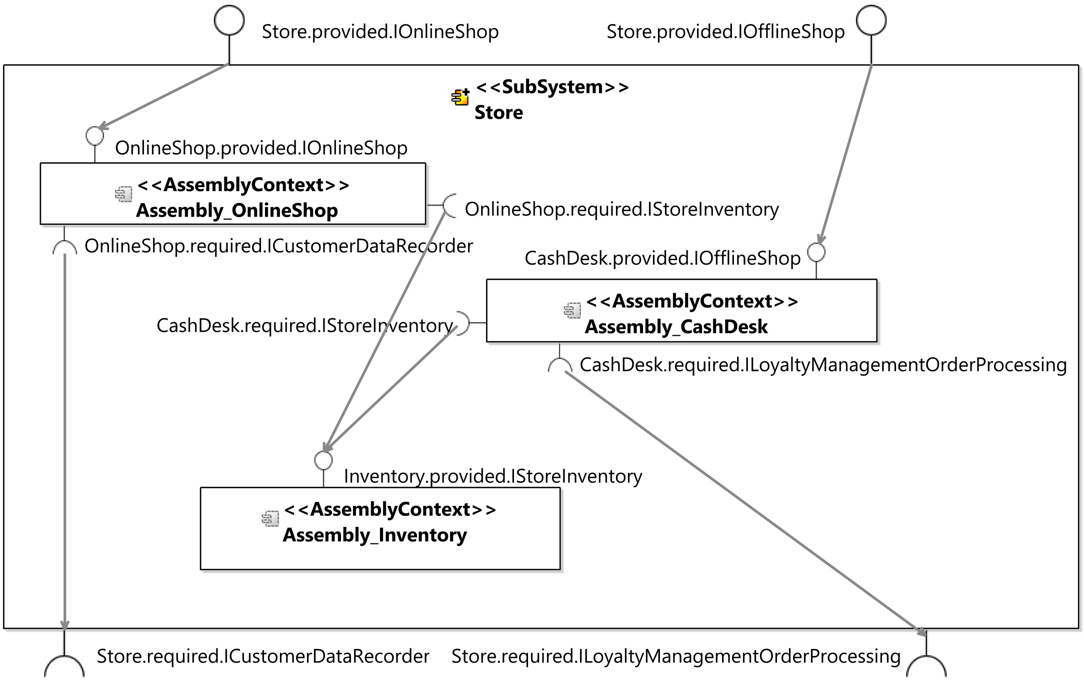

Store.provided.IOnlineShop
Store.provided.IOfflineShop
<<SubSystem>>
Store
OnlineShop.provided.IOnlineShop
<<AssemblyContext>>
Assembly_OnlineShop
OnlineShop.required.IStoreInventory
OnlineShop.required.ICustomerDataRecorder
CashDesk.provided.IOfflineShop
<<AssemblyContext>>
Assembly_CashDesk
CashDesk.required.IStoreInventory
CashDesk.required.ILoyaltyManagementOrderProcessing
Inventory.provided.IStoreInventory
<<AssemblyContext>>
Assembly_Inventory
Store.required.ICustomerDataRecorder
Store.required.ILoyaltyManagementOrderProcessing

**Figure 3.38:** Shows the components of the system *Store*.

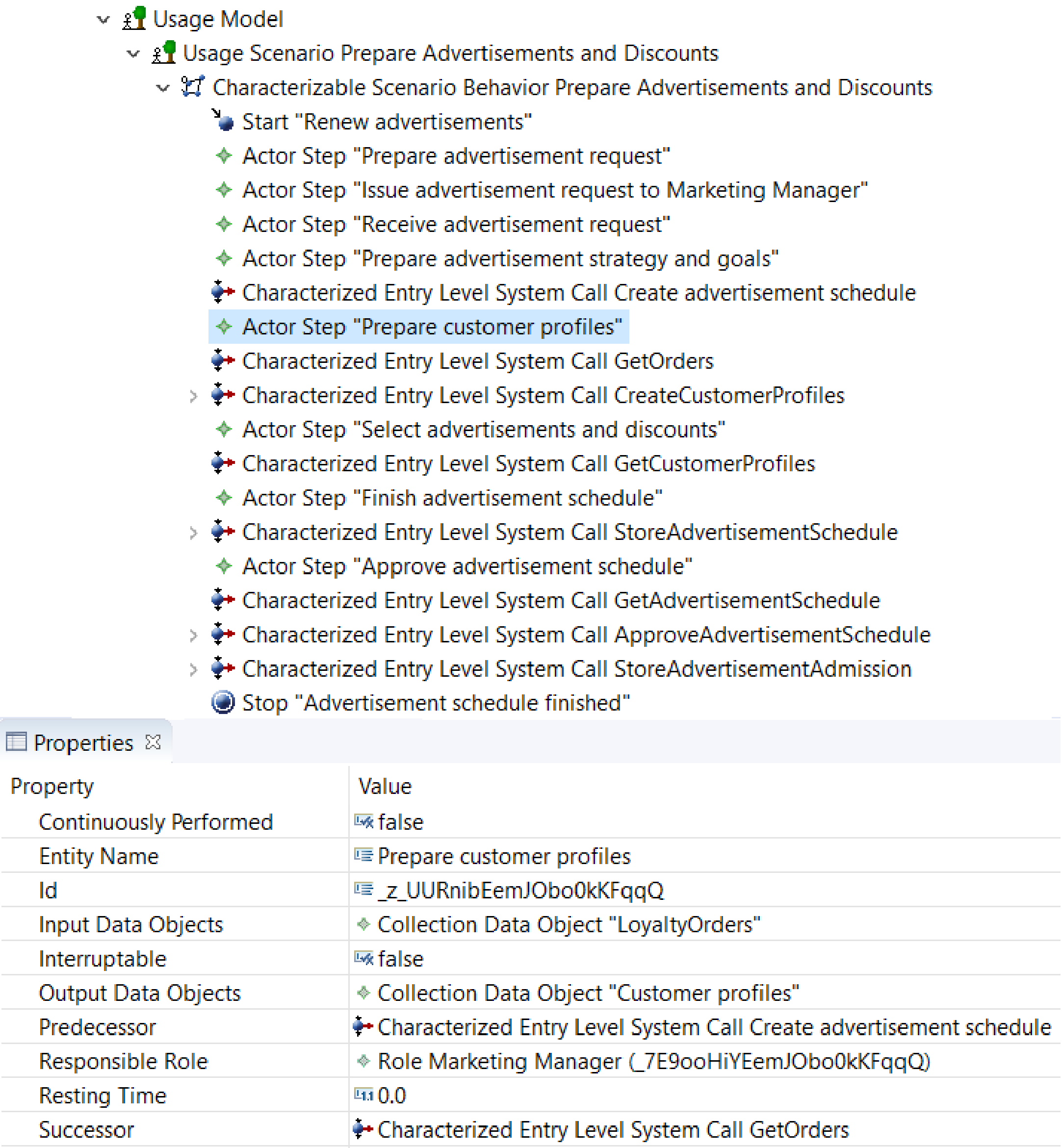
Figure 3.40 shows the *ELSC Match AS Ext Container* that interconnects actor steps with EntryLevelSystemCalls. For example, the actor step *Prepare customer profiles* is connected with the EntryLevelSystemCall *getOrders()* and *createCustomerProfiles()*.

Figure 3.41 shows the data model of the supermarket enterprise CoCoME. It defines the data objects for the business processes and maps them to data types of the EAA. For example, the property *Data Types* shows that the collection data object *OnlineOrders* is mapped to the collection data type *OnlineOrders* and the collection data object *LoyaltyOrders* is mapped to the collection data type *LoyaltyOrders*.

AcsALign analysis the EAA for ACR breaches in two steps. The first step processes ACRs to form data flow constraints for the EAA. The second step analysis whether the data flow constraints are fulfilled by the EAA. Both steps are done automatically and do not require any human interaction. In a subsequent third step the enterprise architect manually resolves the identified mistakes, with the support of additional information provided by AcsALign.

*Step 1) Generating data flow constraints:* During the first step, data flow constraints are generated. Therefore, the ACR mapping model elicited by PAcsTract is extended with EAA elements. Table 3.20 shows an excerpt of the ACR mapping model for the business process *Prepare Advertisements and Discounts*.

AcsALign extracts the mapping of activities from business processes to service calls of the EAA from the ELSC match AS ext container. It interconnects the actor steps, which are the activities of IntBIIS_LP processes with EntryLevelSystemCalls, which are the service



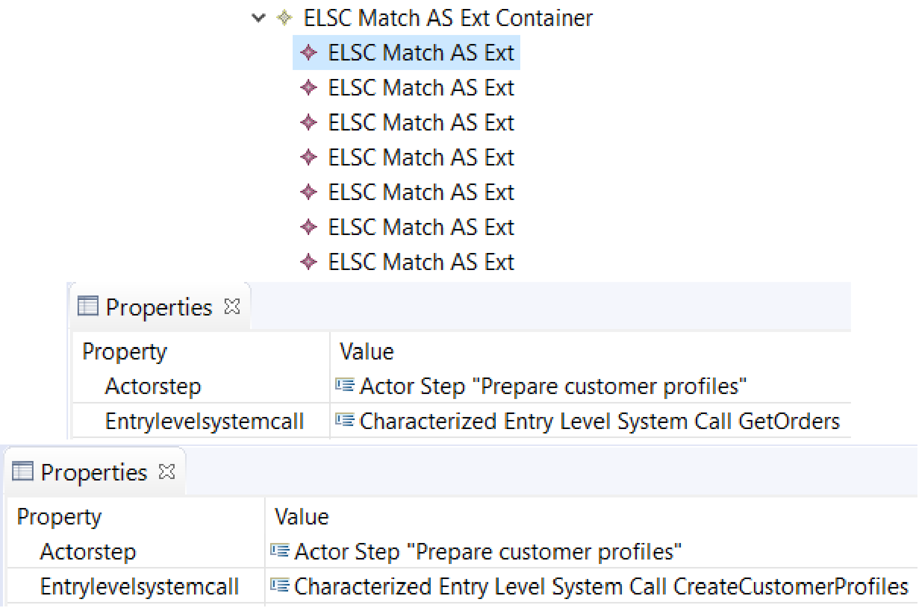
- Usage Model
  - Usage Scenario Prepare Advertisements and Discounts
    - Characterizable Scenario Behavior Prepare Advertisements and Discounts
      - Start "Renew advertisements"
      - Actor Step "Prepare advertisement request"
      - Actor Step "Issue advertisement request to Marketing Manager"
      - Actor Step "Receive advertisement request"
      - Actor Step "Prepare advertisement strategy and goals"
      - Characterized Entry Level System Call Create advertisement schedule
      - Actor Step "Prepare customer profiles"
      - Characterized Entry Level System Call GetOrders
      - Characterized Entry Level System Call CreateCustomerProfiles
      - Actor Step "Select advertisements and discounts"
      - Characterized Entry Level System Call GetCustomerProfiles
      - Actor Step "Finish advertisement schedule"
      - Characterized Entry Level System Call StoreAdvertisementSchedule
      - Actor Step "Approve advertisement schedule"
      - Characterized Entry Level System Call GetAdvertisementSchedule
      - Characterized Entry Level System Call ApproveAdvertisementSchedule
      - Characterized Entry Level System Call StoreAdvertisementAdmission
      - Stop "Advertisement schedule finished"

| Property | Value |
|---|---|
| Continuously Performed | false |
| Entity Name | Prepare customer profiles |
| Id | _z_UURnibEemJObo0kKFqqQ |
| Input Data Objects | Collection Data Object "LoyaltyOrders" |
| Interruptable | false |
| Output Data Objects | Collection Data Object "Customer profiles" |
| Predecessor | Characterized Entry Level System Call Create advertisement schedule |
| Responsible Role | Role Marketing Manager (_7E9ooHiYEemJObo0kKFqqQ) |
| Resting Time | 0.0 |
| Successor | Characterized Entry Level System Call GetOrders |

**Figure 3.39:** Shows the IntBIIS_LP process *Prepare Advertisements and Discounts* of the supermarket enterprise CoCoME.

calls of the EAA. Therefore, the ELSC match AS ext container of each business process is analyzed. Figure 3.40 shows the ELSC match AS ext container for the business processes *Prepare Advertisements and Discounts* in Figure 3.39. Figure 3.40 shows that the EntryLevelSystemCalls *getOrders()* and *createCustomerProfiles()* are invoked during the actor step *Prepare customer profiles*. By analyzing the EntryLevelSystemCalls their corresponding systems and interfaces are found. This information from the EAA (EntryLevelSystemCall, interface and system) is extended in the ACR mapping model for the corresponding actor step. Table 3.21 shows an excerpt of the extended ACR mapping model.

The ACRs extracted from business processes by PAcsTract are part of the ACR mapping model. Hence, AcsALign reads the ACRs along with the information of corresponding



- ELSC Match AS Ext Container
  - ELSC Match AS Ext
  - ELSC Match AS Ext
  - ELSC Match AS Ext
  - ELSC Match AS Ext
  - ELSC Match AS Ext
  - ELSC Match AS Ext
  - ELSC Match AS Ext

Properties ☒

| Property | Value |
|---|---|
| Actorstep | Actor Step "Prepare customer profiles" |
| Entrylevelsystemcall | Characterized Entry Level System Call GetOrders |

Properties ☒

| Property | Value |
|---|---|
| Actorstep | Actor Step "Prepare customer profiles" |
| Entrylevelsystemcall | Characterized Entry Level System Call CreateCustomerProfiles |

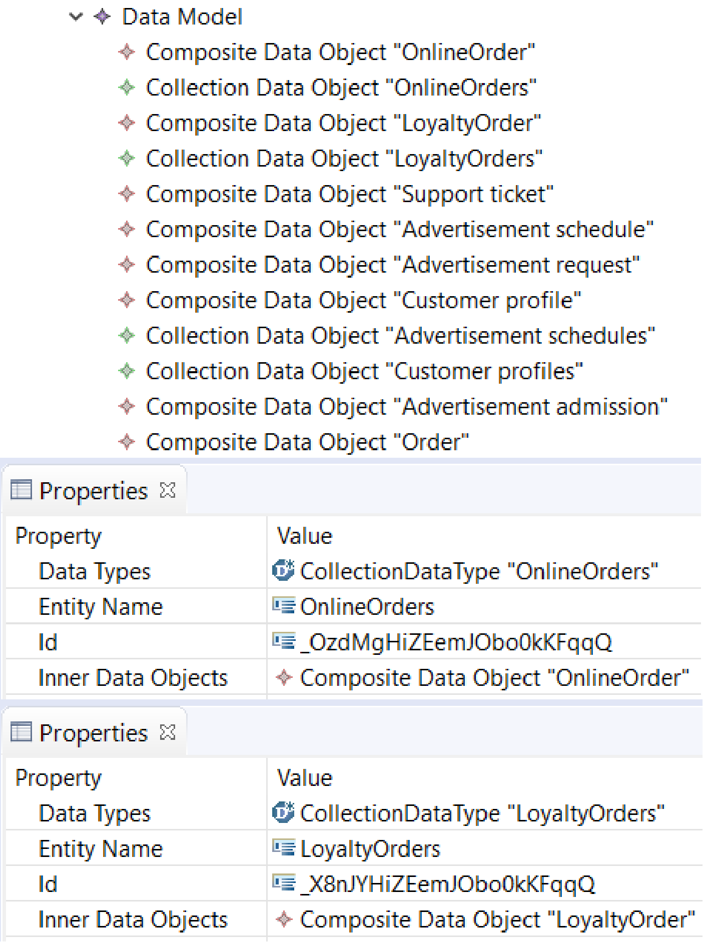
**Figure 3.40:** Shows the *ELSC Match AS Ext Container* that interconnects actor steps with their EntryLevelSystemCalls for the business process *Prepare Advertisements and Discounts*.

**Table 3.20:** Excerpt of the ACR mapping model for the business process *Prepare Advertisements and Discounts*.

| Role | Process | Activity | Permission | Data Type |
|---|---|---|---|---|
| | | ... | | |
| Store: Marketing Manager | Prepare Advertis-ements and Discounts | Prepare customer profiles | READ Loyalty order (Coll.)<br>WRITE Customer profiles (Coll.) | Loyalty order (Coll.)<br>Customer profiles (Coll.) |
| | | ... | | |

EntryLevelSystemCalls from the extended ACR mapping model to generate data flow constraints. A data flow constraint states for an EntryLevelSystemCall which data types a role is allowed to read and write. Table 3.22 shows the generated data flow constraints for the excerpt of the extended ACR mapping model from Table 3.21.

The extended ACR mapping model in Table 3.21 states for each actor step the data objects that are allowed to be read and written by a role. It also states the EntryLevelSystemCalls that are invoked during the actor step. Thus, the EntryLevelSystemCall, the role and the data objects that are allowed to be read and written are extracted from it. For example, during the EntryLevelSystemCall *getOrders()* role *Store: Marketing Manager* is allowed to read the data object *Loyalty orders (Coll.)* and during the EntryLevelSystemCall *createCustomerProfiles()* role *Store: Marketing Manager* is allowed to write the data object *Customer profiles (Coll.)*. Finally, the data model shown in Figure 3.41 is used to transform data objects from business processes to data types of the EAA. As the ACR mapping model also contains this information each data object, the ACR mapping model can be



- Data Model
  - Composite Data Object "OnlineOrder"
  - Collection Data Object "OnlineOrders"
  - Composite Data Object "LoyaltyOrder"
  - Collection Data Object "LoyaltyOrders"
  - Composite Data Object "Support ticket"
  - Composite Data Object "Advertisement schedule"
  - Composite Data Object "Advertisement request"
  - Composite Data Object "Customer profile"
  - Collection Data Object "Advertisement schedules"
  - Collection Data Object "Customer profiles"
  - Composite Data Object "Advertisement admission"
  - Composite Data Object "Order"

Properties

| Property | Value |
|---|---|
| Data Types | CollectionDataType "OnlineOrders" |
| Entity Name | OnlineOrders |
| Id | _OzdMgHiZEemJObo0kKFqqQ |
| Inner Data Objects | Composite Data Object "OnlineOrder" |

Properties

| Property | Value |
|---|---|
| Data Types | CollectionDataType "LoyaltyOrders" |
| Entity Name | LoyaltyOrders |
| Id | _X8nJYHiZEemJObo0kKFqqQ |
| Inner Data Objects | Composite Data Object "LoyaltyOrder" |

**Figure 3.41:** Shows the data model of the supermarket enterprise CoCoME.

**Table 3.21:** Excerpt of the extended ACR mapping model for the business process *Prepare Advertisements and Discounts.*

| Role | Process | Activity | Permission | Data Type | System | Service Call |
|---|---|---|---|---|---|---|
| | | | ... | | | |
| Store: Marketing Manager | Prepare Advertis-ements and Discounts | Prepare customer profiles | READ Loyalty order (Coll.)<br>WRITE Customer profiles (Coll.) | Loyalty order (Coll.)<br>Customer profiles (Coll.) | Mark-eting | IMarketAnalysis. getOrders() |
| Store: Marketing Manager | Prepare Advertis-ements and Discounts | Prepare customer profiles | READ Loyalty order (Coll.)<br>WRITE Customer profiles (Coll.) | Loyalty order (Coll.)<br>Customer profiles (Coll.) | Mark-eting | IMarketAnalysis. createCustomer-Profiles() |
| | | | ... | | | |

**Table 3.22:** Excerpt of the generated data flow constraints for the business process *Prepare Advertisements and Discounts.*

| Service Call | Role | Permission |
|---|---|---|
| | ... | |
| IMarketAnalysis. getOrders() | Store: Marketing Manager | READ Loyalty orders (Coll.)<br>WRITE Customer profiles (Coll.) |
| IMarketAnalysis. createCustomerProfiles() | Store: Marketing Manager | READ Loyalty orders (Coll.)<br>WRITE Customer profiles (Coll.) |
| | ... | |

also used for the transformation (see column data type in Table 3.21). Consequently, data object *Loyalty orders (Coll.)* and *Customer profiles (Coll.)* are transformed to data types *Loyalty orders (Coll.)* and *Customer profiles (Coll.).* In case of the running example the names of the data objects and data types are the same, but this must not be the case. Table 3.22 shows the generated data flow constraints for the running example. The role *Store: Marketing Manager* is allowed to read *Loyalty orders (Coll.)* in the EntryLevelSystemCall *getOrders()* and is allowed write *Customer profiles (Coll.)* in the EntryLevelSystemCall *createCustomerProfiles().*

*Step 2) Architectural alignment analysis:* In this step, ACR breaches are identified in the EAA using the previously generated data flow constraints. The data types that are read and written by an EntryLevelSystemCall are provided by the Palladio extension DC-PCM in form of a JSON file. Table 3.23 summarizes the relevant parts of the JSON file for the running example.

**Table 3.23:** Excerpt of the read and written data types for the invoked EntryLevelSystemCalls of the business process *Prepare Advertisements and Discounts.*

| Service Call | Read/Written Data Types |
|---|---|
| | ... |
| IMarketAnalysis. getOrders() | READ Loyalty orders (Coll.)<br>READ Online orders (Coll.) |
| IMarketAnalysis. createCustomerProfiles() | WRITE Customer profiles (Coll.) |
| | ... |

Table 3.23 shows that the data types *Loyalty orders (Coll.)* and *Online orders (Coll.)* are read during the invocation of the EntryLevelSystemCall *getOrders()* and that the data type *Customer profiles (Coll.)* is written during the invocation of the EntryLevelSystemCall *createCustomerProfiles()*. By comparing the actual read and written data types of an EntryLevelSystemCall with the data flow constraint for that EntryLevelSystemCall, AcsALign identifies ACR breaches. Algorithm 1 shows the pseudocode for the comparison.

**Algorithm 1** Pseudo-code to derive if a given data type $d$ is allowed [179].

```
 1: function ALLOWED($d, D_{known}, D_{allowed}$)
 2:     $allowed \leftarrow (d \in D_{known} \implies d \in D_{allowed}) \land (d \notin D_{known} \implies fallback)$
 3:     if $allowed \land isComposite(d)$ then
 4:         for $d_i \leftarrow d.innerDataTypes$ do
 5:             $allowed \leftarrow allowed \land allowed(d_i, D_{known}, D_{allowed})$
 6:         end for
 7:     end if
 8:     if $allowed \land isCollection(d) \land d \notin D_{known}$ then
 9:         $allowed \leftarrow allowed \land d.innerType \notin D_{known}$
10:     end if
11:     return $allowed$
12: end function
```

A $fallback$ is used if a data type is not known in the business processes. Using $fallback = false$ is appropriate for high risk environments because it denies access to all unknown data types. Using $fallback = true$ is more permissive and grants access to all data types that are not known in the business processes. This can be useful in case of many data type refinements. The latter is used in the running example. $D_{known}$ is the set of data types known from business processes. It contains data types for which data objects in the data model exists. $D_{allowed}$ is the set of data types allowed by the data flow constraint for the EntryLevelSystemCall of the data type that is analyzed. The function *ALLOWED* receives: the data type $d$ of an EntryLevelSystemCall that will be analyzed, $D_{allowed}$ containing the allowed data types from the data flow constraint of the particular EntryLevelSystemCall and $D_{known}$. If the data type $d$ is known to business processes, it has to be allowed explicitly (line 2 in Algorithm 1). Otherwise, the fallback applies. For a composed data type all of its inner data types have to be allowed as well (lines 3–7 in Algorithm 1). In case a collection data type is unknown its inner data type must also be unknown (lines 8–10 in Algorithm 1), otherwise it is forbidden.

In case of the running example, the data flow constraints in Table 3.22 state that the *marketing manager* has the permissions *READ Loyalty orders (Coll.)* and *WRITE Customer profiles (Coll.)* for the service calls *getOrders()* and *createCustomerProfiles()* of the activity *prepare customer profiles*. The data flow analysis for the invoked EntryLevelSystemCall shown in Table 3.23 detects a read to *Loyalty orders (Coll.)* and *Online orders (Coll.)* for the EntryLevelSystemCall *getOrders()* and a write of *Customer profiles (Coll.)* for the EntryLevelSystemCall *createCustomerProfiles()*. The results of the comparison algorithm shown in Table 3.24 yield that: a) the data type *Customer profiles (Coll.)* is allowed to flow, as it is explicitly allowed by the data flow constraint, b) the data type *Loyalty orders (Coll.)* is allowed to flow, as it is explicitly allowed by the data flow constraint and c) the data type *Online orders (Coll.)* is forbidden to flow, as it is a known data object in the business processes but has no data flow constraint that permits it for the given EntryLevelSystemCall. The last part of the algorithm correctly identifies an ACR breach which indicates that the enterprise architect has made a logical or design mistake during the design of the EAA.

**Table 3.24:** Results of the AcsALign comparison algorithm for detecting ACR breaches.

| Service Call | Role | Permission | Read/Written Data Types | Allowed |
|---|---|---|---|---|
| IMarketAnalysis. getOrders() | Store: Marketing Manager | ...<br>READ Loyalty orders (Coll.) | READ Loyalty orders (Coll.) | ALLOWED |
| | | WRITE Customer profiles (Coll.) | READ Online orders (Coll.) | FORBIDDEN |
| IMarketAnalysis. createCustomerProfiles() | Store: Marketing Manager | READ Loyalty orders (Coll.)<br>WRITE Customer profiles (Coll.)<br>... | WRITE Customer profiles (Coll.) | ALLOWED |

The flow of the data type *Online orders (Coll.)* to the marketing manager is correctly identified by AcsALign as an ACR breach. During the evolution of the EAA the newly introduced online orders were falsely passed to the service call *getOrders()*. The mistake was done by the enterprise architect due to false interpretation of requirements. The business level has not intended to use online orders for marketing reasons and has not defined this usage in the business processes. The reason for this is that the GDPR prohibits the processing and use of personal data without an explicit consent of the person. This consent was not obtained by the supermarket enterprise CoCoME, as the use of the online orders was not intendent for marketing purposes. A consent for the processing of loyalty orders does exists. As a result, the marketing manager has access to personal data that is prohibited by the GDPR. Hereinafter, the enterprise architect has to resolve the mistakes identified by AcsALign. This step is not automated and has to be manually done by the enterprise architect. Nevertheless, AcsALign supports this step with additional information such as the extended ACR mapping model including the results for identified ACR breaches.

*Step 3) Mistake resolution:* AcsALign outputs the results of identified ACR breaches together with the extended ACR mapping model in form of a HTML file. A CSV file is also provided. Besides the information about the ACR breach, this output provides additional information to the enterprise architect. This information supports him to resolve the mistakes in the EAA. AcsALign produces the following additional information that helps to understand the ACR breach and identify the logical and design mistake:

a) the violated interface and service call of the affected component/system.

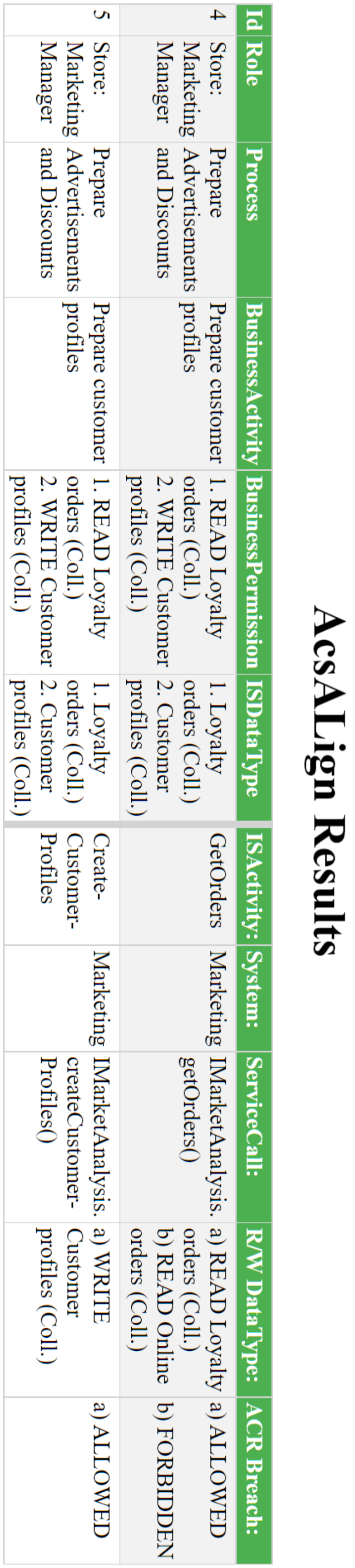
b) the violated data flow. Beginning from the source component, over the service call that violates the ACR, to the sink component. This information is part of the JSON file provided by the Palladio extension DC-PCM.

c) the violated roles and permissions of the access control.

d) the affected business process including the affected activities of lanes.

An example of the extended ACR mapping model with results in form of an HTML file is shown in Figure 3.42. The right side of Figure 3.42 shows the system, interface and service call in the columns *System* and *ServiceCall*. The read and written data types for the service calls are shown in the column *R/W DataType*. They are enumerated. The column *ACR Breach* states whether a data type is allowed to flow or forbidden, indicating an ACR breach. This information supports the enterprise architect in identifying where in the EAA the ACR breach happened. The left side of Figure 3.42 shows the ACR mapping model extracted by PAcsTract with the entries for the process, role, activity and permission. They contain the trace information to the affected business processes and the access permissions of the access control system. The column *BusinessPermission* shows additionally the ACRs for the corresponding service call. The trace information to the affected access permissions of the access control system helps the enterprise architect to understand which ACR was intended by the business level in the particular service call and activity. It enables to understand which access rights to data were violated and how the correct access rights should look like. The trace information regarding the affected business process, lane and activity supports the enterprise architect in understand the bigger picture and reflect design decisions of the business level. It also provides the possibility to contact responsible employees, e.g., the process owner for further counseling. This trace information across models of business and IT is unique and facilitates important information to resolve mistakes in the EAA sustainable.

To sum up, AcsALign is built on top of PAcsTract meaning that it consumes ACRs extracted by PAcsTract from business processes defined in IntBIIS_LP. It operates in two automatic steps to analyze the EAA for ACR breaches. In the first step, data flow constraints are generated for service calls from the ACRs. In a second step, the actual data flows of invoked services are compared against the generated data flow constraints to identify ACR breaches. An ACR breach indicates that the EAA is not aligned with the business processes with regard to access control. This means that the enterprise architect has made logical or design mistakes during the design of the EAA. AcsALign identifies such breaches and outputs additional trace information to support the identification and sustainable resolution of these mistakes.



# AcsALign Results

| Id | Role | Process | BusinessActivity | BusinessPermission | ISDataType | ISActivity: | System: | ServiceCall: | R/W DataType: | ACR Breach: |
|---|---|---|---|---|---|---|---|---|---|---|
| 4 | Store: Marketing Manager | Prepare Advertisements and Discounts | Prepare customer profiles | 1. READ Loyalty orders (Coll.)<br>2. WRITE Customer profiles (Coll.) | 1. Loyalty orders (Coll.)<br>2. Customer profiles (Coll.) | GetOrders | Marketing | IMarketAnalysis.getOrders() | a) READ Loyalty orders (Coll.)<br>b) READ Online orders (Coll.) | a) ALLOWED<br>b) FORBIDDEN |
| 5 | Store: Marketing Manager | Prepare Advertisements and Discounts | Prepare customer profiles | 1. READ Loyalty orders (Coll.)<br>2. WRITE Customer profiles (Coll.) | 1. Loyalty orders (Coll.)<br>2. Customer profiles (Coll.) | CreateCustomerProfiles | Marketing | IMarketAnalysis.createCustomerProfiles() | a) WRITE Customer profiles (Coll.) | a) ALLOWED |

**Figure 3.42:** Shows an excerpt of the HTML output of the results from AcsALign for the running example.

## 3.3 Discussion of BPMN Access Permission Extractor and Palladio Access Permission Extractor

This section discusses in Section 3.3.1 the problem statements and contributions related to BAcsTract and PAcsTract. Section 3.3.2 elaborates on the assumptions and limitations.

### 3.3.1 Discussion of Problem Statements and Contributions

BPMRNME and PAcsTract realize the concept of extracting ACRs from business processes to form a role model for RBAC that was formalized in Section 3.1.1. In addition, the approaches establish an ACR mapping model that interconnects elements of business processes and RBAC. It can be seen as an automated documentation of design decisions, allowing to trace resulting access permission to their originating process, role and activity. Both approaches are top-down role engineering approaches eliciting business roles and permissions. Thus, they are part of the role engineering approaches explained in Section 2.2.2. In contrast to typical role engineering approaches that are carried out manually by experts, the approaches presented in this thesis are mostly automated. With respect to the business level, they help to accomplish the three goals (identify critical business assets, establish organization-wide IT security and privacy strategies and comply with IT security and privacy laws) that were introduced in Section 1.1. Regarding the IT level, the approaches help to establish a secure and aligned role model for RBAC. Further on, problems from Section 1.2 are discussed with regard to the contributions of the presented approaches.

**P1 Missing knowledge on IT level:** Knowledge about which business assets are critical and their required protection degree lies on the business level and thus, is missing on the IT level [25]. The approaches of this thesis close this gap by extracting implicitly modeled business level ACRs from business processes (contribution **C1**). The information about critical assets and their access permissions is extracted and transformed to the IT level by forming a role model for RBAC. BAcsTract extracts roles and permissions from processes in BPMN by analyzing interacting lanes and their associated data objects. PAcsTract does the same on processes modeled in IntBIIS_LP. Additionally, PAcsTract is able to extract data types of data objects, which are the representation of data objects in the EAA. Further on, PAcsTract can extract further ACRs pertaining the physical access to devices and machines. This is especially crucial for organizations dealing with critical infrastructure and high-risk environments.

**P2 Different terminology between business and IT level:** Several discrepancies, e.g., different terminology, domain knowledge, domain-specific models and modeling tools of the business level and IT level widen a communication gap that may lead to errors and security breaches [24, 25]. By transforming knowledge about critical assets and required protection degrees in form of ACRs from the business level to the IT level (contribution **C1**), the approaches close the communication gap with respect to access control. Security experts are getting a role model containing the business level ACRs as well as an ACR mapping model documenting design decisions. These relieves the security experts in understanding some models and terminology of the business level in detail as business level ACRs are inside the generated role model. Comprehensibility for the generated access permissions is provided by the ACR mapping model. It interconnects elements of business processes with elements of RBAC and EAA (contribution **C2**). This allows to track design decisions regarding ACRs across the three mentioned models and hence, couples the domain-specific models together. Experts, but also the business level and IT level, are supported in understanding design decisions in models outside of their subject area. If any question about an access permission arises, they can trace the access permission to the originating business process and conclude its existence. They can also talk with the employees responsible for the activity and the business process owner to clarify any doubts.

**P3 Experts needed to understand business level:** Security experts are needed who know the terminology and models of both business level and IT level. They have to analyze a vast amount of business processes to engineer a role model, which leads to several problems. This engineering process requires skills across several domains of business and IT. As the engineering process is complex, it is time consuming and leads to human errors. During evolution scenarios the process has to be constantly repeated, as ACRs may change due to changes in business processes. There is no way to check automatically whether the engineered role model from the security experts is correct. BAcsTract and PAcsTract reduce the dependencies on skills of security experts due to the automatic extraction of business level ACRs from business processes to form a role model for RBAC (contribution **C1** and **C2**). The approaches reduce complexity while engineering the role model and allow security experts to focus on technical parts. In addition, the provided ACR mapping model allows security experts as well as the business level to understand the reasons for extracted access permissions (contribution **C2**). Besides, the generated access permissions are aligned with the ACRs from the business processes automatically and thus, do not require a check for correctness.

**P4 Costly and error-prone engineering of the RBAC role model:** Role engineering is a manual, slow and complex task making it costly and error-prone. By generating an initial role model for RBAC out of the extracted ACRs automatically (contribution **C3**), BAcsTract and PAcsTract ease the engineering of the role model. The vast amount of business processes that have to be analyzed manually by security experts, are processed automatically. The usage of BAcsTract and PAcsTract requires only little additional effort, as they operate on models that have to be defined anyway. In addition, BAcsTract is tailored to work with BPMN, which is the de facto standard modeling language for business processes and is used across most companies [23, 213]. Security experts are provided with an initial role model comprising the ACRs from business processes. This reduces complexity and avoids human errors during the engineering of the role model. Avoided errors reduce the possibility for security breaches. The improved efficiency of the role model engineering reduces needed time and costs for the organization while increasing the overall security of the access control system.

**P5 Missing alignment between RBAC and business level access control requirements:** Due to the manual and complex engineering of the role model, it is not well aligned with business level ACRs. The approaches of this thesis help with two points. First, a role model for RBAC is built by extracting business level ACRs from business process (contribution **C3**). This aligns the RBAC with the ACRs from the business processes. Second, elements of business processes are interconnected with elements of RBAC establishing an ACR mapping model (contribution **C2**). It allows to track design decisions regarding ACRs across the models. All extracted access permissions are traceable to a particular activity of a business process. The traceability enables security experts and the business level to understand the reasons for access permissions in a comprehensible way. This leads to a better alignment of access permissions with business level needs and thus, to more compliant access permissions. PAcsTract goes even further by providing an output visualizing the usage of devices and machines across actor steps and system steps.

**P8 Missing support of evolution scenarios for RBAC and enterprise application architectures:** Especially during evolution scenarios the role model is not well aligned with business level ACRs, as requirements may constantly change due to changes in business processes and corporate structure [9]. Different employees in the organization are responsible for business processes and RBAC. The evolutionary change of these artifacts is not well studied and understood so far, especially with regard to ACRs [25]. By extracting a role model from business processes automatically (contribution **C3**), the process of engineering the role model becomes faster and more aligned to business level needs. This eases the cumbersome and error-prone work of security experts to go through the vast amount of business processes. As these processes constantly change over time, the role model has to be adapted to these changes. Each time an adaptation becomes necessary, the approaches can extract the changes from the business processes and provide an adapted role model. Alongside the role model, the ACR mapping model allows to understand why changes are occurring and to trace them back to responsible entities and business process elements (contribution **C2**). This supports security experts and the business level in understanding design decisions outside of their expertise. In evolution scenarios, this is beneficial as complex decisions have to be made without understanding fully all resulting changes. The approaches help to resolve this problem in the context of ACRs, enabling the business level to make proper decisions between different evolution scenarios. Considering evolution scenarios, a faster adaptation and better support is provided, due to automation and traceability of elements, giving the opportunity to react faster to changes. The topic of utilizing the approaches during evolution scenarios will be discussed in more detail in Chapter 4.

### 3.3.2 Discussion of Limitations

Several assumptions are proposed for the work in this thesis. Some of them imply limitations on the use of the approaches and their results:

- **Existence of business processes:** In this thesis, I assume that business processes are already modeled. If not, they have to be modeled in order for BAcsTract and PAcsTract to work. However, medium to large organizations and especially organizations with high-security requirements, for example, critical infrastructures are obligated by laws to manage and organize their business according to certain requirements. To fulfill these obligations organizations design business processes according to business processes guidelines as ITIL [34] and COBIT [32]. Hence, many organizations will already have modeled their business processes.
- **Correctly modeled business processes:** In the course of this thesis, I assume that business processes are modeled correctly with regard to syntactics and semantics. If syntactics are not correct, the processes cannot be parsed correctly. Nonetheless, processes are modeled with modeling frameworks that do syntactical checks and may forbid nonsensical modeling (it is still possible to model nonsensical processes that are syntactically correct). In terms of BAcsTract, business processes are consumed in XML format. This format underlies strict definitions of the Object Management Group who is responsible for the development of BPMN. In terms of PAcsTract, the modeling language PCM and IntBIIS_LP are used. Their elements are defined in metamodels and editors support the responsible roles along with constraints that check for wrong usage of elements. Regarding semantics, the business level is responsible to reflect the processes of the organization correctly. There are other works that help them to accomplish this goal, but it is not the objective of the work done in this thesis. These limitations apply also when considering the traditional role engineering process. If business processes are not modeled syntactically correct, security experts are not able to understand them and if business processes are not semantically correct, security experts would propagate these errors into the role model.
- **Scope of processes:** In this thesis, I assume that business processes encompass all departments of the organization and provide a comprehensive picture of the ongoing work done by employees on a daily basis. If parts of the work done in organizations are not modeled in business processes, they will not be reflected in the generated role model. Security experts have to take this into consideration. However, the approaches should help security experts in providing a more aligned and correct role model. This is also the case when only parts of the ongoing work in organizations is modeled. In such scenarios, security experts have to go through other business level artifacts either way.
- **Correctly modeled ACRs:** I assume that ACRs incorporated in business processes by the business level are legally correct and in line with the business goals introduced in Chapter 1, because the focus of this thesis is not to identify erroneous ACRs, but

to define an automated transformation of ACRs from business processes to IT level artifacts.

- **Initial role model:** BAcsTract and PAcsTract extract an initial role model encompassing business level ACRs that reside in business processes. As business processes reflect only the business view of ACR, technical ACRs have to be completed by security experts. The reason for this is that technical ACRs are not part of business processes and thus, cannot be extracted from them. The goal of the approaches is to support the security expert in extracting business level ACRs and this is accomplished by the initial role model. Technical ACRs are not in the scope of this thesis.
- **Effort utilizing approaches:** Both approaches were designed to impose only little additional effort when utilizing them. This is achieved by focusing on de facto standard modeling languages like BPMN and on models that organizations have to model anyway. For sure, this is not the case for every organization. Small organizations and startups may not model their business processes. However, it is another question at what price this comes when considering profitability. Nonetheless, during the growth of organizations and in high security environments, organizations come to a point where they have to model business processes either due to legislative obligations or due to management complexity. At this point, organizations can benefit the most from the approaches of this thesis.
- **Evolution scenarios:** The presented approaches become especially useful during evolution scenarios. Certainly, to utilize the approaches during evolution scenarios the scenarios have to be reflected in the business processes. How the approaches can be utilized during evolution scenarios will be discussed in Chapter 4.

## 3.4 Discussion of Access Permission Architecture Aligner

This section discusses in Section 3.4.1 the problem statements and contributions related to AcsALign. Section 3.4.2 elaborates on the assumptions and limitations.

### 3.4.1 Discussion of Problem Statements and Contributions

AcsALign realizes the concept to identify ACR breaches in EAAs that was formalized in Section 3.1.2. In addition, the approach extends the ACR mapping model established by PAcsTract to interconnect elements of business processes and RBAC with elements of the EAA. This can be seen as an automated documentation of design decisions, allowing to understand ACR breaches by tracing them back to the affected access permissions and the affected activities of business processes. With respect to the business level, AcsALign helps to accomplish the three goals (identify critical business assets, establish organization-wide IT security and privacy strategies and comply with IT security and privacy laws) that were introduced in Section 1.1. Regarding the IT level, the approach helps to establish a secure and aligned EAA. Further on, problems from Section 1.2 are discussed with regard to the contributions of the presented approach.

**P2 Different terminology between business and IT level:** Several discrepancies, e.g., different terminology, domain knowledge, domain-specific models and modeling tools of the business level and IT level widen a communication gap that may lead to errors and security breaches [24, 25]. By transforming ACRs from the business level to data flow constraints on the IT level, AcsALign goes a step further to close this communication gap with respect to access control (contribution **C4**). Enterprise architects are enabled to analyze the EAA for ACR breaches automatically. Furthermore, AcsALign extends the ACR mapping model from PAcsTract so that it interconnects elements of business processes with elements of RBAC and the EAA (contribution **C2**). This allows to track design decisions regarding ACRs across the three mentioned models and couples the domain-specific models together. On the one hand, the enterprise architect is supported in resolving logical and design mistakes leading to ACR breaches. For example, by contacting the business process owner of an affected business process. On the other hand, the business level and IT level are supported in understanding design decisions in models outside of their subject area.

**P3 Experts needed to understand business level:** To analyze the EAA for correctness experts are required who understand terminology and models of both, business level and IT level. Then they have to manually compare the EAA with business level requirements stemming, e.g., from business processes. AcsALign automates this process by using the extracted ACRs of PAcsTract to analyze whether the EAA fulfills them (contribution **C4**). It helps the enterprise architect to design a secure and aligned EAA without the need of further experts. Furthermore, AcsALign produces an extended ACR mapping model that helps the enterprise architect to understand the mistakes and how they affect other models like business processes (contribution **C2**). It establishes a connection between the ACR breach and the affected business process as well as the affected RBAC access permission. This eases the comprehension on how the ACR breach affects other models.

**P6 Complex and error-prone designing of the enterprise application architecture:** Enterprise architects make logical and design mistakes during the design of the EAA for various reasons such as misunderstanding correct requirements, complexity of interrelating models and the widening communication gap due to different terminology. This leads to ACR breaches endangering the overall security of IT systems. AcsALign supports this complex and error-prone process by identifying ACR breaches with the use of ACRs stemming from business processes or the access control system (contribution **C4**). It also supports the enterprise architect in resolving the mistakes by providing additional information to achieve a more secure EAA. This helps to increase the overall security of EAAs.

**P7 Missing alignment between enterprise application architecture and business level access control requirements:** Due to the manual and complex engineering of the EAA, it is not well aligned with business level ACRs. AcsALign helps the enterprise architect at two points to design a secure EAA that is aligned with business level ACRs. First, the EAA is analyzed for ACR beaches (contribution **C4**). This aligns the EAA with ACRs from business processes and RBAC. Second, elements of the EAA are interconnected with elements of business processes and RBAC establishing an extended ACR mapping model (contribution **C2**). It allows to track design decisions regarding ACRs across the three models. All ACR breaches are traceable to the affected service call of a system, the affected activity of a business process and the violated access permission of RBAC or another access control system. The ACR mapping model enables the enterprise architect to understand the mistakes comprehensibly and resolve them correctly. This leads to a better alignment of the EAA with business level needs and thus, to a more compliant EAA.

**P8 Missing support of evolution scenarios for RBAC and enterprise application architectures:** Especially during evolution scenarios the EAA is not well aligned with business level ACRs, as requirements may constantly change due to changes in business processes and corporate structure [9]. Different employees in the organization are responsible for the EAA and the business processes. The evolutionary change of these artifacts is not well studied and understood so far, especially with regard to ACRs [25]. AcsALign enables a more secure adaptation during evolution scenarios of business processes and access permissions by checking automatically if the EAA violates ACRs (contribution **C4**). By analyzing the EAA for ACR breaches automatically, AcsALign helps to increases the security of the EAA and aligns the EAA to business level needs, leading to better compliance. Due to negligible amount of time required to conduct the analysis AcsALign can be utilized to identify ACR breaches each time an adaptation is done. After identifying ACR beaches they can be resolved to align the EAA with business level needs. Besides, AcsALign allows organizations to better understand the mutual interdependence of business processes, access permissions and EAAs. Alongside this, the ACR mapping model is extended with elements from the EAA (contribution **C2**). This helps to understand the mistakes that lead to ACR breaches comprehensible and to resolve them correctly. Considering evolution scenarios, a more secure adaptation and better alignment is provided, due to automation and traceability of elements, giving the opportunity to react faster to evolutionary changes. The topic of utilizing AcsALign and the other approaches during evolution scenarios will be discussed in more detail in Chapter 4.

### 3.4.2 Discussion of Limitations

Several assumptions are proposed for the work in this thesis. Some of them imply limitations on the use of the approach and its results:

- **Existence of an EAA model:** In the course of this thesis, I assume that the EAA is already modeled. If not, the EAA has to be modeled in order for AcsALign to work. However, medium to large organizations have to organize their business to cope with complexity. Therefore, these organizations will already have modeled an EAA. Some other reasons why organizations often model an EAA are, for example, to maximize the organizational value by being able to make better decisions, to trim costs by having a more efficient resource allocation and to establish organization-wide IT security. Organizations with high-security requirements, e.g., critical infrastructures will also have modeled their EAA as security guidelines will force them to do so. Hence, many organizations have already modeled their EAA so that this limitation does apply only to certain organizations.

- **Scope of the EAA:** In this thesis, I assume that the EAA as well as the business processes encompass all departments of the organization and provide a comprehensive picture of the ongoing work done by employees. If any parts of the organization are not modeled in the EAA, then AcsALign will not be able to analyze it for EAA breaches. In such cases AcsALign can at least identify ACR breaches in those parts that are modeled, helping to secure and align them with ACRs.
- **Quality of data flow constraints:** The quality of data flow constraints generated by AcsALign depends on the quality and scope of the ACRs provided as input. At this point AcsALign depends on PAcsTract and thus, on the quality and scope of defined business processes. It is also possible to serve ACRs from other sources, for example, from the access control system. Nevertheless, ACRs need to be correct and cover as many parts of the EAA as possible.
- **No predefined IT-modules:** AcsALign presumes that the enterprise architect designs the EAA according to the requirements of the business level. There might be scenarios where the design is made bottom-up meaning that the business expert has to use predefined IT-modules during the design of the business processes.
- **Limited to data types:** The analysis of AcsALign is limited to data types rather than to actual classes of data. This means that AcsALign cannot differentiate between different classes of data of the same data type. For example, it is possible that two classes of data with the same data type have different ACRs depending on the overall scenario. A newly planed exhibition is confidential during the planning phase but becomes public after the launch. Such expressions of data types are part of the runtime of organizations. AcsALign focuses on the support during the design phase. However, different classes of data types can be designed as individual data types during the design phase. By doing so, AcsALign is able to distinguish them and the limitation can be bypassed.
- **Effort utilizing the approach:** AcsALign was designed to impose only little additional effort to utilize it. This is achieved by focusing on de facto standard modeling languages and models that organizations design anyway. Probably startups and small organizations form an exception. However, during the growth of an organization and in high security environments, organizations come to a point where they have to design an EAA either due to legislative obligations or due to management complexity. Thus, they can benefit the most from AcsALign.
- **Mapping of data types and service calls:** Regarding the modeling language that is used to design the EAA: a) a mapping for data objects from business processes

to data types of the EAA and b) a mapping for activities from business processes to service calls of the EAA is required. In case of PCM and IntBIIS_LP both mappings are part of the business process design. For other modeling languages such as BPMN and UML this mapping has to be provided. However, both mappings require only low effort that needs to be done once. During evolution scenarios these mappings require often only small changes. In case of the mapping from data objects to data types, EAA evolution scenarios do not require a change in the mapping unless a name of an interconnected data type changes. Then only a marginal change of the name is required. In case of business process evolution scenarios changes are only required if new data objects are introduced. Then the mapping needs to be extended for the newly introduced data objects. Depending on the evolution scenario this requires also only minor extensions. In case of the mapping from activities to service calls, EAA evolution scenarios and business process evolution scenarios might imply changes to the mapping. Again, evolution scenarios of the EAA do not imply any changes to the mapping unless names of service calls that are part of the mapping change or are replaced by other service calls. Business process evolution scenarios require changes for any new activity that is introduced.

- **Read and written data types:** AcsALign requires a tool that performs a simple data flow analysis on the EAA to extract the read and written data types of service calls. Such a data flow analysis is very simple and there are tools for many modeling languages that provide this capability.
- **Evolution scenarios:** As with the other approaches of this thesis, AcsALign is especially useful during evolution scenarios. Certainly, to utilize AcsALign during evolution scenarios they have to be reflected in the ACRs (for example, in the business processes or in the access permissions) and the EAA. How AcsALign and the other approaches can be utilized during evolution scenarios will be discussed in Chapter 4.

# 4 Process for Utilizing Approaches in Organizations

This chapter outlines the high-level process of utilizing the approaches introduced in Chapter 3. It describes how organizations can align their business processes with RABC and EAA, especially during evolution scenarios. Section 4.1 briefly discusses the phases that organizations typically undergo when establishing business processes, as well as an EAA and a role model. Afterwards, Section 4.2 outlines the phases when to utilize the approaches described in Chapter 3. Section 4.3 concludes this chapter with a discussion about benefits and limitations regarding evolution scenarios. For the sake of comprehensibility, I will assume a top-down design of models, where models designed first imply requirements for the models designed afterwards. For example, business processes are designed before the EAA and thus, imply requirements to the EAA. However, throughout the sections I will also discuss how organizations can utilize these approaches, if some of their models are already designed. Such scenarios are equivalent to evolution scenarios and will be discussed in more detail in Section 4.3.

Table 4.1 shows the high-level process of organizations when establishing business processes, an EAA and a role model. It is illustrated in a top down manner, where models above imply requirements for the models after. On the left side of Table 4.1, the typical process without the approaches is illustrated, while on the right side the process when utilizing the approaches is illustrated.

## 4.1 Process Typical in Organizations

The typical process for establishing business processes, an EAA and a role model in organizations is outlined in the left part of Table 4.1. First, organizations have to model their business processes and an EAA (phase one in Table 4.1). It is also possible that one or both models already exist. Business processes are developed to achieve business goals and manage workflows in organizations. Software systems are used to support and execute these business processes. The EAA is modeled to establish a link between business processes and software systems. This alignment leads to a better management and optimization with regard to business goals, providing a considerable benefit for organizations [90]. The EAA is often build based on the business processes. To align the artifacts of both Enterprise Architecture Management (EAM) is used. It involves initiating and establishing of processes, governance and the definition of application scenarios. Also

**Table 4.1:** Process with and without the approaches of this thesis.

| Phase | Typical | With Approaches |
|---|---|---|
| 1 | Modeling business processes and enterprise application architecture | Modeling business processes and enterprise application architecture |
| 2 | Engineering the role model | Role model extraction |
| 3 | | Refinement of initial role model |
| 4 | | Enterprise application architecture analysis for access control requirement violations |
| 5 | | Mistake resolution in enterprise application architecture |

models and lifecycles are defined [149]. Frameworks as TOGAF [96] propose approaches for designing, planning, implementing and governing the enterprise architecture. They define different model types with mutual relations. A detailed explanation about EAAs was provided in Section 2.3. In phase two of Table 4.1 security experts engineer the role model for RBAC based on the previously mentioned artifacts. The role model is engineered manually and comprises access permissions according to the needs of the organization. Therefore, the vast amount of business process must be analyzed. In Section 2.2.2 this complex, time consuming and cost-intensive phase is explained in-depth.

## 4.2 Process when Utilizing Approaches

This section introduces the high-level process of how to utilize the approaches of this thesis throughout different evolution scenarios. By doing so, organizations can align business processes, RABC and the EAA. Section 4.2.1 introduces the problems that the high-level process tackles. Afterwards, Section 4.2.2 presents the high-level process as well as the contributions.

### 4.2.1 Problem Statement

The high-level process tackles the following problems from Section 1.2:

- **P5 Missing alignment between RBAC and business level access control requirements:** The process of engineering the role model is complex and error-prone (as explained in Section 2.2). Security experts have to analyze a vast amount of interrelating business processes manually. In addition, there is no traceability between the elicited role model and the business processes. Due to the missing traceability and the complex role engineering process, RBAC is not well aligned with business level ACRs.

- **P7 Missing alignment between enterprise application architecture and business level access control requirements:** During the design of the EAA the enterprise architect has to consider many functional and non-functional requirements. Two of them are IT security and privacy requirements. A fundamental building block of both are ACRs stemming from the business level and laws. As only the business level knows which assets are critical and which protection degree they require, a communication gap widens due to different terminology and domain specific models. Business processes and EAA affect each other in non-trivial ways [9] (see also Section 1.1). As a result, it is difficult to align them with each other [24, 25].
- **P8 Missing support of evolution scenarios for RBAC and enterprise application architectures:** Business processes, RBAC and EAA are interdependent. As organizations constantly evolve, changes of one model require adaptation and alignment of the others. This evolutionary change is not well studied and understood so far, especially for ACRs [25]. Furthermore, models are big, complex and not tightly-coupled (see also Section 1.1). Various employees of different expertise in the organization are responsible for them. This widens a communication gap that additionally aggravates the alignment of these models.

### 4.2.2 High-Level Process

The high-level process outlined on the right side of Table 4.1 reflects the alignment of business level and IT level artifacts in an organization, while utilizing the approaches presented in Chapter 3 of this thesis. Some phases are equal to the typical process (phase one and partially phase three), while others are introduced newly by the approaches. As the concepts of the approaches are applicable to widely used and de facto standard modeling languages, the same is true for the applicability of the high-level process. Figure 4.1 illustrates the sequence of phases of the high-level process in which organizations utilize the approaches from Chapter 3. It is aligned along the typical process of organizations where they design business processes, EAA and engineer a role model for RBAC. In the following, the high-level process is outlined in a top-down manner.

The three green little circles in Figure 4.1 named *Design new business processes or EAA*, *Evolution of RBAC* and *Evolution of RBAC or EAA* indicate starting points in the process. Starting from *Design new business processes or EAA* and following the transitions along the numbers of the phases, results in the top-down order shown in Table 4.1:

- **Phase 1:** Organizations develop business processes reflecting their organizational services and products as well as the EAA to organize IT services in accordance to business needs.
- **Phase 2:** Security experts utilize BAcsTract or PAcsTract to automatically extract the initial role model from the business processes.

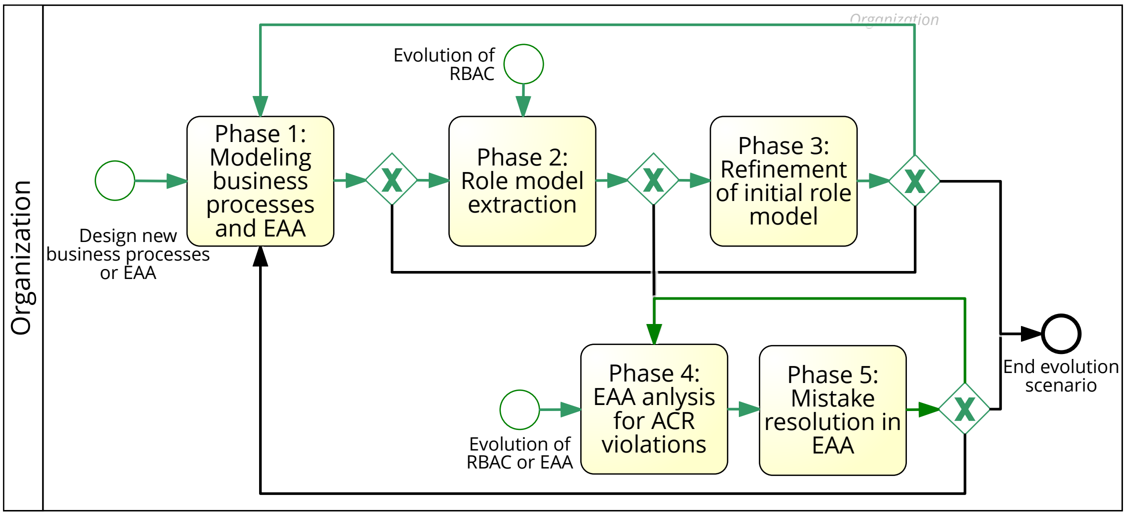

Organization
Organization
Evolution of RBAC
Design new business processes or EAA
Phase 1: Modeling business processes and EAA
X
Phase 2: Role model extraction
X
Phase 3: Refinement of initial role model
X
Evolution of RBAC or EAA
Phase 4: EAA anlysis for ACR violations
Phase 5: Mistake resolution in EAA
X
End evolution scenario

**Figure 4.1:** Shows the high-level process for utilizing the approaches of this thesis.

- **Phase 3:** Security experts refine the initial role model with technical access permissions.
- **Phase 4:** Enterprise architects utilize AcsALign to analyze the EAA for ACR breaches during design time.
- **Phase 5:** Enterprise architects resolve logical and design mistakes responsible for the ACR breaches.

Section 4.3 will go into more details of Figure 4.1 and the various starting points for the different evolution scenarios and purposes. Further on, the individual phases are explained in more detail.

To illustrate the perspective of an organization when utilizing the approaches along the high-level process the previously introduced running example of the CoCoME case study from Section 3.2.3.2 is used. The basic CoCoME supermarket store, including a loyalty program and a marketing division, undergoes an evolution scenario. In this scenario, the supermarket store is extended by an online shop according to the community evolution scenario of adding a pick-up shop [104], including adjustments for handling the loyalty management. Obviously, privacy must be considered by the CoCoME enterprise. Research on the various regulations and resulting security and privacy requirements pertaining CoCoME was conducted in [180].

**Phase 1 Modeling business processes and enterprises application architecture:** Organizations develop business processes reflecting their organizational services and products to manage and optimize their services. They are a central point and key factor for the business level to manage business workflows successfully. The EAA is designed by the enterprise architect, to organize IT services of the organization in accordance to business needs. This phase is independent of the presented approach and most middle to big organizations will already developed both models, as they are a key management factor for organizations.

More details on business processes and the EAA were provided in Section 2.1, Section 1.1 and Section 2.3.

In case of the running example, the following business processes are designed amongst others. Figure 4.2 shows the business process of CoCoME for preparing advertisements and discounts by the marketing manager and store manager. Figure 4.3 shows the customer support process of CoCoME. Afterwards, Figure 4.5 shows an excerpt of the corresponding EAA supporting the business processes.

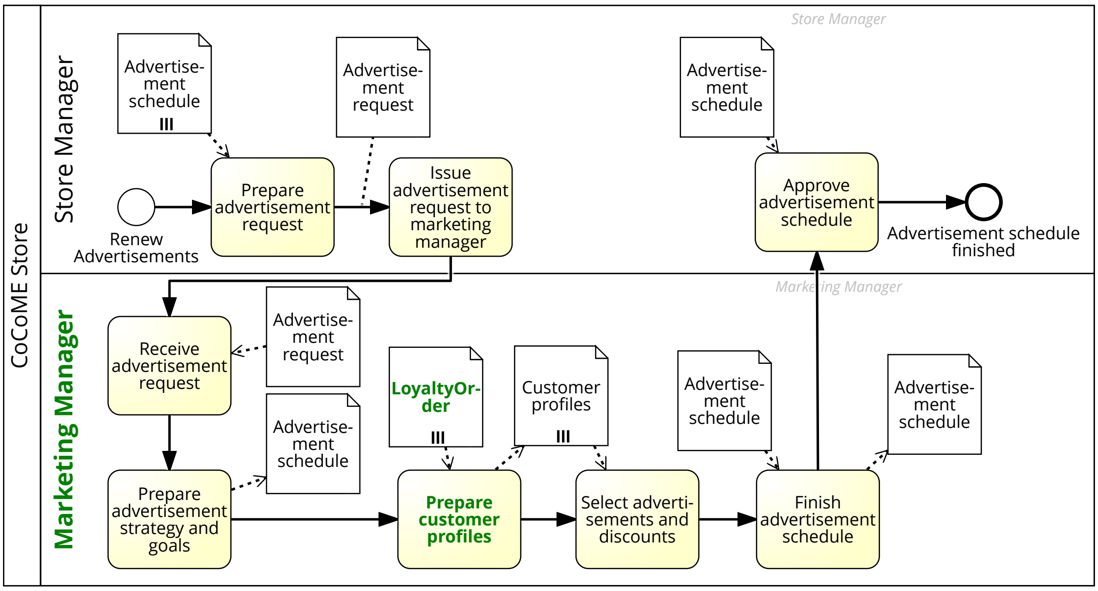

CoCoME Store
Store Manager
Store Manager
Advertise-ment schedule
Advertise-ment request
Advertise-ment schedule
Renew Advertisements
Prepare advertisement request
Issue advertisement request to marketing manager
Approve advertisement schedule
Advertisement schedule finished
Marketing Manager
Marketing Manager
Receive advertisement request
Advertise-ment request
Advertise-ment schedule
LoyaltyOr-der
Customer profiles
Advertise-ment schedule
Advertise-ment schedule
Prepare advertisement strategy and goals
Prepare customer profiles
Select adverti-sements and discounts
Finish advertisement schedule

**Figure 4.2:** Shows the business process *Prepare Advertisements and Discounts* of the CoCoME supermarket store.

In the business process of Figure 4.2 new advertisements and discounts are defined, based on the gathered data of customers participating in the loyalty program. During the selection of appropriate advertisements and discounts the marketing manager prepares customer profiles (green highlighted part of Figure 4.2). During this activity, a set of *LoyaltyOrder* is consumed, resulting in a required read access to the data object *LoyaltyOrder*.

The process in Figure 4.3 shows the customer support process of the CoCoME supermarket store. The process begins with a customer issuing a support ticket. This support ticket is processed by a customer service employee. To solve the problem, he requires read access to a set of *LoyaltyOrder*, represented by the corresponding data objects and dotted arrows pointing to the activity *Solve problem* in Figure 4.3. After the problem is identified and solved, the customer service employee answers to the customer support ticket and the process is finished. Exemplarily, we look at an evolution scenario where the CoCoME supermarket store is extended by an online shop, as defined in [104]. Consequently, some business processes and the EAA changes. Figure 4.4 shows the customer support process after the online shop was added. The input data object *OnlineOrder* in the activity *Solve problem* is new. As not only loyalty customers may issue problems but also online customers, the customer service employee requires also access to loyalty orders. The

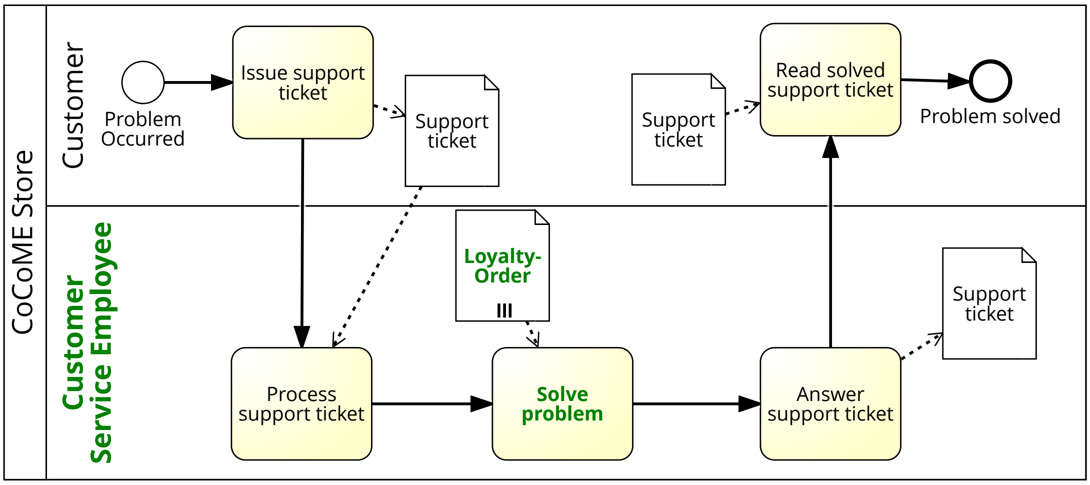


**Figure 4.3:** Shows the business process *Solve Customer Problem* of the CoCoME supermarket store.

business level distinguishes between *LoyaltyOrder* and *OnlineOrder*, as they comprise different information and are used throughout different processes and purposes.

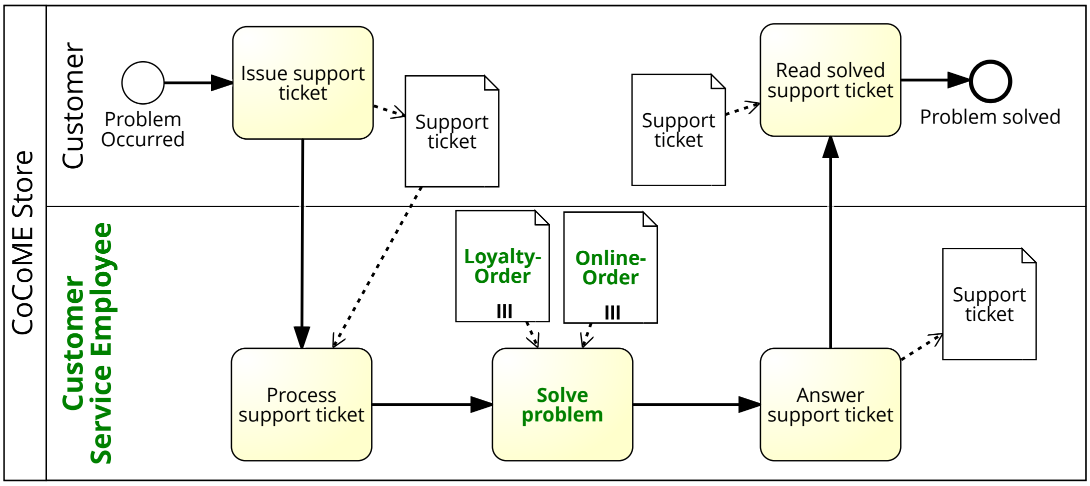


**Figure 4.4:** Shows the business process *Solve Customer Problem* of the CoCoME supermarket store.

An excerpt of the EAA supporting the business processes is defined in Figure 4.5. Grey elements denote parts of the basic CoCoME without any modifications. Shaded elements are extended to fit the proposed evolution scenario.

The *Store* system handles sales and inventory. If a customer uses a loyalty card during payment, the store system informs the *LoyaltyManagement* system with a *LoyaltyOrder* comprising information about the loyalty card and the ordered goods. The *LoyaltyManagement* system performs calculations about gained loyalty points and sends the order to the *CustomerDataStore* system. This information is available to the *Marketing* system

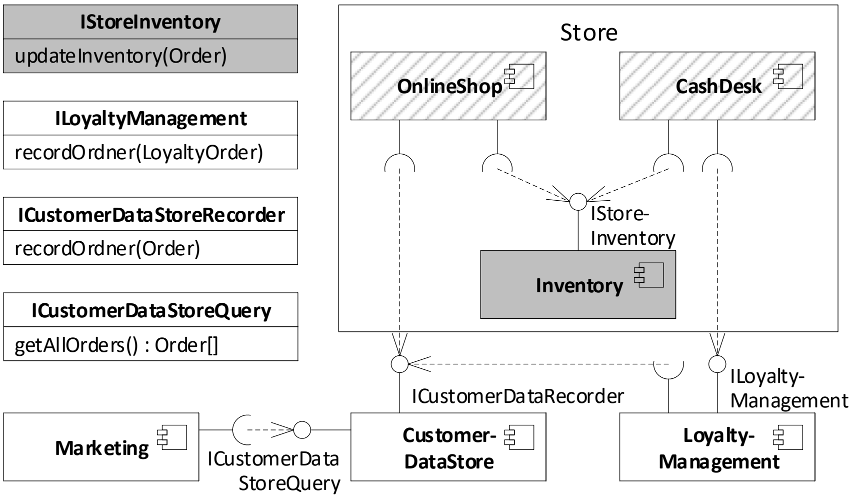


**Figure 4.5:** Simplified enterprise application architecture of the CoCoME supermarket [178].

via the *ICustomerDataStoreQuery* interface. The marketing manager uses the *Marketing* system to create advertisements based on previous orders.

To sum up, in this phase organizations model their business processes and the supporting EAA. Some business processes and EAA parts may be new and others may be undergoing evolution scenarios. The running example showed two processes and the corresponding EAA. One business process reflects the preparation of advertisements and discounts. Another reflects the customer support process. The latter undergoes an evolution scenario while the online shop is added to the CoCoME enterprise. The highlighted green parts of the business processes in Figure 4.4 and Figure 4.3 show the most important parts for the rest of the running example. It is important to note that the two presented business processes are taken as an example from the large amount of business processes of CoCoME. Also, the EAA reflects only a part of the whole EAA. The running example is a simplified version showing only the relevant parts and elements for the purpose of illustrating the different phases of the high-level process in which the approaches of this thesis support organizations in aligning their RBAC and EAA.

**Phase 2 Role model extraction:** In the typical process security experts would engineer the role model in a manual, complex and time-consuming role engineering process as described in Section 2.2.2. In contrast, BAcsTract and PAcsTract ease the engineering of the role model by extracting an initial role model from business processes automatically. This takes only negligible amount of time and effort, as only models are used that are defined anyway. Depending on the used business process language, either BAcsTract or PAcsTract is used. A detailed explanation on the role model extraction was provided in Section 3.2.3.

For the running example, Table 4.2 shows the initial role model extracted by BAcsTract from the two business processes of phase one.

**Table 4.2:** Extracted role model from the business process in Figure 4.2 and Figure 4.4.

| Nr. | Role | Process |
|---|---|---|
| 1 | CoCoME Store: Customer | READ/WRITE Support ticket |
| 2 | CoCoME Store: Customer Service Employee | READ/WRITE Support ticket |
| 3 | **CoCoME Store: Customer Service Employee** | **READ LoyaltyOrder** |
| 4 | **CoCoME Store: Customer Service Employee** | **READ OnlineOrder** |
| 5 | CoCoME Store: Marketing Manager | READ Advertisement request |
| 6 | CoCoME Store: Marketing Manager | READ/WRITE Advertisement schedule |
| 7 | **CoCoME Store: Marketing Manager** | **READ LoyaltyOrder** |
| 8 | CoCoME Store: Marketing Manager | READ/WRITE Customer profiles |
| 9 | CoCoME Store: Store Manager | READ/WRITE Advertisement request |
| 10 | CoCoME Store: Store Manager | READ Advertisement schedule |

On the left side of the role model in Table 4.2 are the extracted roles. On the right side their extracted access permissions. As explained in Section 3.2.3, the initial role model comprises all business level ACRs from the analyzed business processes. Bold roles and permissions of the role model in Table 4.2 correspond to the required permissions resulting from the highlighted green parts of the business processes in Figure 4.2 and Figure 4.4. The green highlighted parts of the first process imply that the *Marketing Manager* requires read access to the data object *LoyaltyOrder*. The green highlighted parts of the second process imply that the *Customer Service Employee* requires read access to the data objects *LoyaltyOrder* and *OnlineOrder*. These access permission can be found in row three, four and seven of the role model in Table 4.2.

To sum up, in this phase security experts are supported in eliciting ACRs from business processes to form an initial role model. This takes only negligible amount of time as BAcsTract and PAcsTract operate on models that are designed anyway in the previously phase. Consequently, security experts do not have to analyze the vast amount of business processes manually anymore. This reduces complexity, time and costs of the overall role engineering process. As a benefit, the role model is aligned with business level needs and security experts whose skills are more technical can focus on technical parts. Another unique benefit is accomplished by the ACR mapping model. It is built alongside the role model and documents design decisions regarding ACRs. The established traceability of ACRs enables the business level to understand resulting access permissions in the role model, as each access permission can be traced to its originating business process, activity and lane. This enables the business level to forecast requirements for access permissions resulting from decisions made in business processes. Besides, the ACR mapping model improves the communication between the business level and security experts.

**Phase 3 Refinement of initial role model:** The initial role model, extracted by BAcsTract or PAcsTract automatically in phase two, has to be completed by security experts. After phase two, the role model encompasses all business level ACRs from business processes. In terms of completeness the initial role model lacks technical ACRs. Subsequently, in this phase security experts refine the initial role model by adding technical access permissions.

When looking on the typical process without the utilization of the approaches, security experts would also have to elicit technical ACRs. Hence, the work done by security experts in this phase has to be done anyway. However, manually analyzing the vast amount of business processes is not necessary anymore, as business level ACRs were already extracted in the previous phase. This saves time, costs and reduces errors.

**Phase 4 Enterprise application architecture analysis for access control requirement violations:** In this phase, the extracted ACRs from the previous phases are transformed into architectural data flow constraints to help the enterprise architect analyze the correctness of the designed EAA. For the analysis it is possible to use the extracted ACRs from phase two or to use the refined ACRs from phase three. This allows not only to align the EAA with the ACRs form the business processes, but also to align the EAA with technical ACRs of the RBAC role model. The generated constraints are used by the enterprise architect in an architectural data flow analysis to identify forbidden data flows. The existence of a forbidden data flow means that the EAA was not correctly designed regarding the business level needs. In more detail, the enterprise architect has done either a logical mistake or a design mistake during the design of the EAA. A detailed explanation on the identification of ACR breaches in EAAs was provided in Section 3.2.4. This analysis takes only negligible amount of time, as all required models are designed by organizations anyway (see phase one).

Specifically in the running example it means that AcsALign is applied to the EAA and business processes shown in phase one. The green highlighted part of the business process in Figure 4.4 has two different types of order defined by the business level. *LoyaltyOrder* and *OnlineOrder* are required as input during the activity *Solve problem* of the role *Customer Service Employee*. The green highlighted part of the business process in Figure 4.2 as well as the extracted ACRs of the role model in Table 4.2 imply that the *Marketing Manager* requires read access to a set of *LoyaltyOrder* in the activity *Prepare customer profiles*. This set is used to prepare *Customer profiles*. On the technical level (see the EAA in Figure 4.5), this is realized by the operation *getAllOrders(): order[]* of the *CustomerDataStore* system. It transfers order data from the *CustomerDataStore* system to the *Marketing* system. The operation itself does not perform any anonymization. At this point the EAA has a design mistake.

Previous to the evolution scenario there was only one order type, *LoyaltyOrder*. Thus, the design decisions concerning the service function *getAllOrders(): order[]* and its technical description “getting all orders” were correct. With the introduction of the online shop during the evolution scenario, the new order type *OnlineOrder* was introduced by the business level. The reason was the obligation to comply with the GDPR [222], as described in detail in my publication [180]. GDPR states that organizations are not allowed to use personal data without an explicit and informed consent of that person. In the evolution scenario for the online shop no consent was obtained by the CoCoME enterprise. Only for the loyalty customers a consent exists to process personal data for marketing reasons. Hence, the GDPR forbids the marketing manager to have access to online order.

Regarding the technical part, *order* became a shorthand writing for data that can be *LoyaltyOrder* or *OnlineOrder*. According to the design decisions and the technical description of the operation *getAllOrders(): order[]*, the operation was modified correctly to return both types of orders. But, due to the communication gap and not aligned models of business and IT (explained in Section 1.1) as well as the complexity of designing EAAs (explained in Section 1.1 and Section 2.3) the marketing manager still uses the service function *getAllOrders(): order[]* to get the required orders for preparing advertisements and discounts, as he also used before the evolution scenario. This is a design mistake in the EAA. The problem is that according to the technical specification it is correct that the service function *getAllOrders(): order[]* returns *LoyaltyOrder* and *OnlineOrder*. However, according to specifications of business processes the marketing manager has only permission to read *LoyaltyOrder* (see highlighted parts of process in Figure 4.2 and role model in Table 4.2). This is due to the restrictions of the GDPR. Consequently, with the designed EAA the marketing manager has access to forbidden data. This access is not allowed according to the GDPR and was not intended by the business level. The resulting EAA is not aligned with ACRs from business processes. However, AcsALign identifies this forbidden data flow and provides this information to the enterprise architect. This is done in negligible amount of time, as AcsALign operates on models that are designed by organizations anyway.

To sum up, the enterprise architect uses AcsALign to identify forbidden data flows in the EAA. This is done either based on the extracted ACRs from phase two or based on the refined ACRs from phase three. This allows to align the EAA with ACRs form business processes and with access permissions from RBAC. Forbidden data flows resulting from logical and design mistakes can be identified, indicating that the EAA is not in line with business processes or RBAC. Detailed information about security breaches is provided to the enterprise architect, so that he can resolve the error in the upcoming phase.

**Phase 5 Mistake resolution in enterprise application architecture:** In this phase, the enterprise architect resolves mistakes found by AcsALign previously. The enterprise architect uses results of phase four to understand each forbidden data flow comprehensively. AcsALign supports this phase by providing an ACR mapping model that interconnects ACR elements between the three models of business and IT. To better understand why a data flow is forbidden, the enterprise architect can trace the forbidden data flow back to its originating access permission in RBAC and their originating lane and activity of the business process. To visually support the information about forbidden data flows across models of business and IT, the enterprise architect gets the following additional information generated from the ACR mapping model:

a) a data flow diagram with a highlighted visualization of violated data flows across relevant systems, including the sources, sinks and service functions.

b) the corresponding access permissions in the role model of RBAC that are violated.

c) the corresponding business processes, lanes, activities and data objects that are violated.

This additional information helps the enterprise architect in understanding design decisions made by the business level regarding ACRs that are relevant for the forbidden data flows. He is able to understand where and why the mistake occurs, so that he can resolve it correctly with regard to previous design decisions. Consequently, the EAA becomes more correct, secure and better aligned with business level ACRs.

Regarding the running example, the enterprise architect gets the following traceability information:

a) the whole data flow that is violated. Beginning from the source component, including all components and service functions that are invoked, to the sink component. The service function that violates ACRs is highlighted. In the example, it is the service function *getAllOrders(): order[]* of the *CustomerDataStore* system in Figure 4.5.

b) the roles and permissions in the role model that are violated. In the role model of Table 4.2 a permission of the role *Marketing Manager* is violated. It is the read permission for *OnlineOrder*.

c) the activities of lanes in business processes that are violated. Regarding the running example, it is the business process in Figure 4.2 and the activity *Prepare customer profiles* of lane *Marketing Manager* along with its data object *OnlineOrder*.

The additional information in other models of business and IT concerning the forbidden data flows help the enterprise architect to understand the mistakes faster and more comprehensively. Information in the EAA helps to identify the forbidden data flows along with the service functions that violate ACRs. Traceability information in the role model helps to understand which access permissions of which roles are violated. Traceability information in business processes help to understand which ACRs are violated by which activities of certain lanes. The latter is most valuable, as the enterprise architect can understand how the business processes and the corresponding data flows should look like and thus, understand business level decisions. It provides comprehensible information about why and what has to be changed in the EAA to be conform with business level ACRs. Another benefit is that the business level is enabled to understand the forbidden data flow as well. They can also trace the mistake back to the lanes, activities and business processes that are violated. This enables the business level to understand the problem and the design decisions of the enterprise architect. Hence, improving the communication between both.

Summed up, the described process shows how business level knowledge about ACRs lying in business processes is extracted and transformed to IT level artifacts. It enables security experts and enterprise architects to align their models with ACRs from the business level. Consequently, the overall security degree is increased, while reducing the expenditure of required time and money in the overall process. Due to the automation and the negligible amount of time to process phase two and four, faster adaptations in evolution scenarios are possible. In addition, the ACR mapping model allows to understand mutual dependencies of business processes, EAA and RBAC. For example, if business processes evolve, conducting phase two and four enable organizations to understand immediately which changes arise in RBAC and whether the EAA is in line with the changes in business processes. When changes happen to RBAC or the EAA phase four can be conducted to analyze whether the EAA is still in line with the changes access permissions in RBAC or the ACRs from business processes. Different evolution scenarios will be discussed in the following section.

Regarding the problems stated in the beginning of this section, the presented high-level process provides the following contributions (see also Section 1.6) to solve them:

- **C5 A high-level process to align RBAC and the enterprise application architecture with business level access control requirements:** The high-level process presented in this section aligns the role model of RBAC with business processes by utilizing BAcsTract or PAcsTract (tackles problem **P5**). Furthermore, extracted information from business processes and RBAC are used to align the EAA with the other models by identifying forbidden data flows with AcsALign (tackles problem **P7**). The generated ACR mapping model additionally supports the alignment by providing a documentation of design decisions across models of business and IT. It also helps to resolve errors in RBAC and mistakes in the EAA by providing relevant information across the three models.
- **C6 A high-level process to identify inconsistencies between models in evolution scenarios of business processes, RBAC and the enterprise application architecture:** During evolution scenarios of either business processes, RBAC or EAA the high-level process can be used to forecast changes regarding the other models. The ACR mapping model helps the business level and IT level in understanding required changes throughout evolution scenarios across the various models. Thus, BAcsTract, PAcsTract and AcsALign enable to understand mutual dependencies more comprehensively and to make tradeoffs among different evolution scenarios (tackles problem **P8**).

## 4.3 Discussion of the Process when Utilizing Approaches

This section will discuss how organizations undergo the high-level process in different evolution scenarios. The high-level process can start at different phases depending on which models evolve, which models already exist and which goals are aimed for the analysis. This is illustrated in Figure 4.1. Later, in this section such evolution scenarios will be discussed in detail, but first some background about the applicability of the approaches has to be provided.

Regarding the applicability of the high-level process a benefit is that the concepts of the approaches are applicable to widely used and de facto standard modeling languages. Phase one, in which organizations model their business processes and EAA, is undergone by organizations anyway, as explained in Section 4.1 and Section 4.2. It is important to note that in this phase all required input information is provided. It means that most organizations will already have all the input required by BAcsTract, PAcsTract and AcsALign and thus, no additional effort of modelling is required. With regard to AcsALign, the most common language for the EAA is UML [6]. The concept described in Section 3.1.2 can be used with UML. Other architectural design languages, e.g., PCM are also possible. In this thesis, I implemented the concept for PCM. I argued this point in Section 3.2. Supposing organizations use PCM to model their EAA, AcsALign is directly applicable. When other modeling languages are used for the EAA, for example, UML, the architectural data flow constraints from AcsALign can be reused with little adaptations, as EAA models in PCM are fundamentally not very different from UML component models [189]. The concrete syntax of PCM is based on the syntax of UML. Nevertheless, the main point is that:

a) for phase two, *Role model extraction*, no additional information is required. Input models are developed by organizations anyway and are often existing (see also Section 4.1). The analysis itself takes only negligible amount of time and the business level and IT level are supported in understanding generated results by various output models like the ACR mapping model.

b) for phase four, *EAA analysis for ACR violations*, if modeled in PCM, no additional information is required. If another modeling language is used, for example, UML, only few additional information might be required. Most of the generated information can still be used with a data flow analysis in UML. The analysis itself takes only negligible amount of time. In addition, the business level and IT level are supported in phase five with various traceability information providing a better comprehensibility of generated results.

Models of business and IT are complex and interrelated, as described in Section 1.1 and Section 1.2. As a result, mutual dependencies between these models cannot be foreseen in evolution scenarios. This leads to a misalignment of models. The high-level process proposed in this thesis aligns business processes, RBAC and EAA in terms of ACRs and thus, allows to understand how during evolution scenarios changes in one model affect changes in other models. The reason for this is threefold:

a) Section 4.1 explained that the required input is modeled by organizations anyway.

b) as explained in Section 3.2.1, business processes and the EAA models do not need to be enriched with extended information.

c) Section 3.2.3 and Section 3.2.4 explained that the analysis done by these approaches takes only negligible amount of time.

Consequently, mutual dependencies and required changes during evolution scenarios can be analyzed in negligible amount of time and without additional effort. This enables to forecast the impact of evolution scenarios on interrelated models and decide among different design decisions appropriately.

The above-mentioned reasons make it valuable to utilize the approaches during evolution scenarios. Hence, the phases of the high-level process may be undergone partially or in different sequences. In Figure 4.1 the high-level process is illustrated as a process with different starting points and transitions.

The three green little circles named *Design new business processes or EAA*, *Evolution of RBAC* and *Evolution of RBAC or EAA* indicate starting points in the process. Starting from

*Design new business processes or EAA* and following the transitions along the numbers of the phases, results in the sequential order depicted in Table 4.1. The green arrows of the process shown in Figure 4.1 indicate the possibility to use BAcsTract, PallBPMRNME and AcsALign separately from each other. Organizations may focus either on aligning their role model with BAcsTract or PAcsTract (green arrows in the upper part of Figure 4.1) or on aligning the EAA with AcsALign (green arrows in the lower part of Figure 4.1).

It is important to mention that not all phases have to be processed and that the processing sequence may vary depending on the needs of the organization. Starting at any of the three starting points (little green circles in Figure 4.1) indicate different evolution scenarios. Organizations that already have business processes, an EAA and/or a role model can start at any of the three starting points and transit according to the flow of the arrows to the desired destination phase. Respectively, organizations that undergo evolutions of any of the three models can use the approaches to crosscheck the models with the generated results. For example, if an organization reworks their role model to increase security, it is possible to start at *Evolution of RBAC* to conduct phase two and crosscheck if the reworked role model meets all ACRs of the business level. This is done by comparing whether the reworked role model encompasses all access permissions of the generated role model from BAcsTract or PAcsTract. Another possibility is to start at *Evolution of RBAC* to conduct phase four and crosscheck if the current EAA meets the ACRs of the reworked role model. In case an organization redesigns their EAA to adapt new software systems, it is possible to start at *Evolution of RBAC or EAA* to conduct phase four to crosscheck if the reworked EAA complies with the ACRs from business processes or with the refined ACRs from the role model.

Another possible use case for the high-level process lies in the possibility to conduct tradeoff analyses between variants of certain business processes, RBAC and EAA. An organization can conduct a tradeoff analysis with variants of any of the three models.

a) A tradeoff analysis between variants of business processes.

   a1) It is possible to analyze how the variants of business processes comply with the current role model. Therefore, phase two in Figure 4.1 is conducted and the generated role model is crosschecked for divergent access permissions with the current role model used by the organization. A divergent access permission means that the business process variant requires a change to the current role model. Whether this change is meaningful and secure has to be decided with authorities in charge.

   a2) It is possible to analyze how the business process variants comply with the current EAA. Therefore, phase two and afterwards phase four in Figure 4.1, is conducted. The amount of identified forbidden data flows is a measure for required changes to the EAA. Identified forbidden data flows may also indicate whether some functionalities break with regard to information security.

b) A tradeoff analysis between variants of role models.

b1) On the one hand, the variants of role models can be analyzed for compliance with business processes. This is done by conducting phase two in Figure 4.1 with a subsequent crosschecking for divergent access permissions between the generated role model and the role model variants. Each divergent access permission indicates a misalignment with business processes.

b2) On the other hand, the role model variants can be analyzed for compliance with the EAA. Phase four in Figure 4.1 is conducted for this. The amount of forbidden data flows that are identified indicate the misalignment with the EAA.

c) A tradeoff analysis between variants of EAAs.

c1) It is possible to analyze the variants of EAAs for compliance with business processes. Therefore, phase four in Figure 4.1 is conducted based on the initial role model from phase two. The resulting amount of forbidden data flows indicate which parts of the EAA are not in line with ACRs from business process.

c2) It is possible to analyze the variants of EAAs for compliance with the role model. Therefore, phase four in Figure 4.1 is conducted based on the current role model of the organization. Again, the amount of identified forbidden data flows indicate which parts of the EAA are not in line with the role model of RBAC.

Such tradeoff analyses reflect decisions between security characteristics and other quality characteristics, for example, performance.

The high-level process of utilizing the approaches of this thesis promotes comprehensibility for the business level and IT level by generating traceability between elements of the EAA, the role model and business processes. Dependencies between these models are often complex and hard to identify. The business level as well as the IT level are enabled to forecast mutual dependencies with regard to ACRs. In addition, the approaches support the engineering of the role model for RBAC and the identification of logical and design mistakes in EAAs. Nevertheless, refining the initial role model and resolving EAA mistakes is still a challenging task. On the one hand, to refine the initial role model a security expert, who mines technical ACRs, is still required. Furthermore, the quality of the extracted role model is dependent on the quality and workflow coverage of the business processes. On the other hand, certain IT compliance implications, for example, separation of duty, still have to be engineered in cooperation with the business level. Regarding the EAA, the approach closes the communication gap between the business level and the enterprise architect by providing traceability information across models and documenting design decisions. After all, communication between both is still recommended. To be clear, enterprise architects still should question business decisions that imply high costs and efforts for realization. However, organizations that utilize the approaches have an unique opportunity: to understand mutual dependencies between business and IT models, to understand their implications for alignment problems and to ease the engineering of correct and business level aligned access permissions and EAAs during evolution scenarios.

# 5 Validation

This section describes the experimental validation of the approaches explained in Chapter 3. Two case studies are conducted to validate these approaches. Both case studies are structured according to the GQM method [181]. That means goals are formulated which represent the desired evaluation objectives. Then research questions are derived from the goals and finally, metrics are defined to validate the research questions. Section 5.1 presents the case study for the role model extraction from business processes with BAcsTract. BAcsTract and PAcsTract extract ACRs from business processes to build an initial role model for RBAC. Section 5.2 presents the case study for the identification of ACR breaches in EAAs with AcsALign. AcsALign uses ACRs extracted from business processes by PAcsTract to analyze service calls of the EAA for violations of these ACRs. In the course of the second case study PAcsTract is used for the extraction of ACRs from business processes and thus, the case study partly validates the approach for role model extraction. Figure 5.1 shows the overall GQM model.

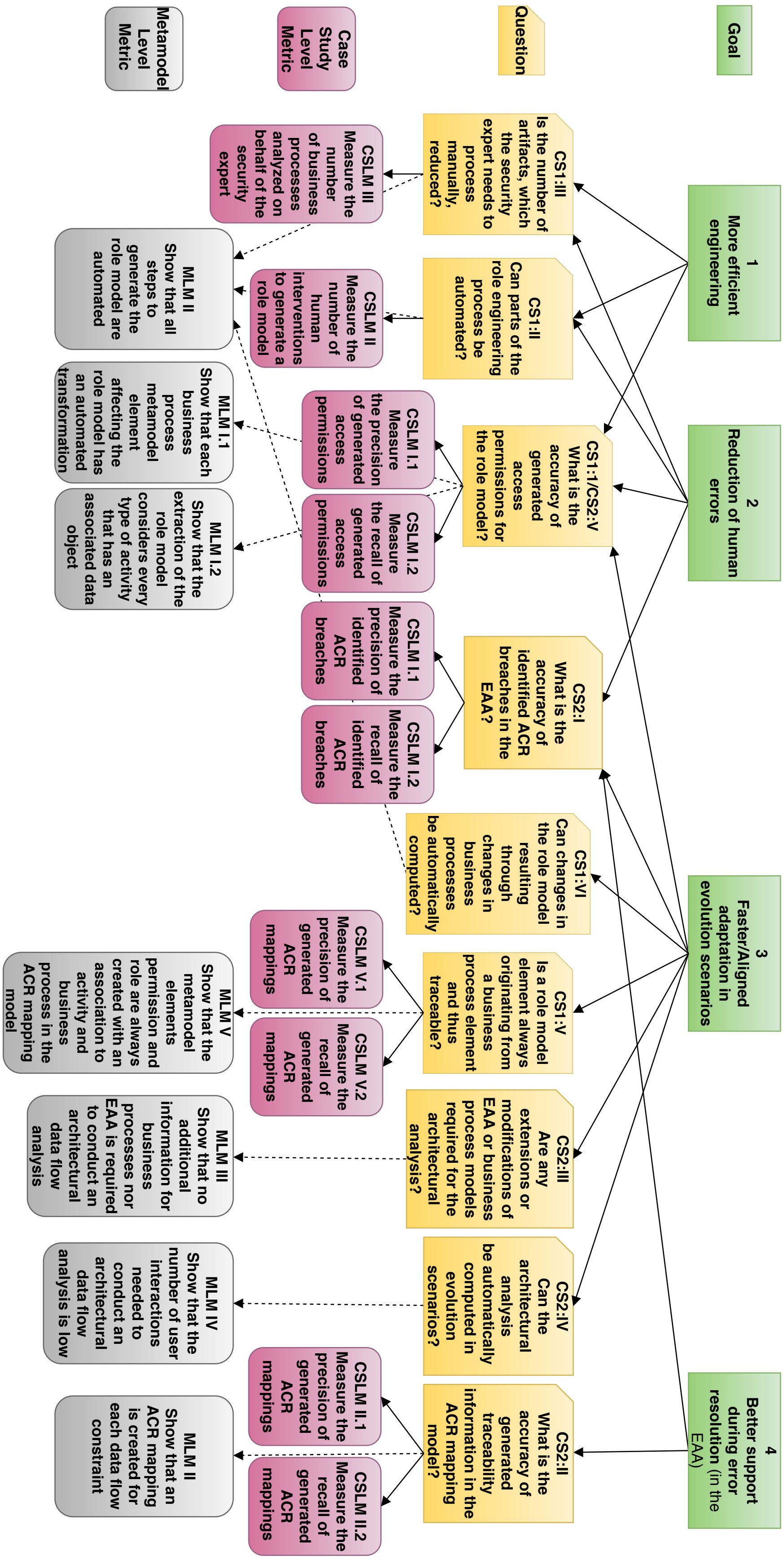

Goal
Question
Case Study Level Metric
Metamodel Level Metric
1 More efficient engineering
2 Reduction of human errors
3 Faster/Aligned adaptation in evolution scenarios
4 Better support during error resolution (in the EAA)
CS1:III Is the number of artifacts, which the security expert needs to process manually, reduced?
CS1:II Can parts of the role engineering process be automated?
CS1:1/CS2:V What is the accuracy of generated access permissions for the role model?
CS2:I What is the accuracy of identified ACR breaches in the EAA?
CS1:VI Can changes in the role model resulting through changes in business processes be automatically computed?
CS1:V Is a role model element always originating from a business process element and thus traceable?
CS2:III Are any extensions or modifications of EAA or business process models required for the architectural analysis?
CS2:IV Can the architectural analysis be automatically computed in evolution scenarios?
CS2:II What is the accuracy of generated traceability information in the ACR mapping model?
CSLM III Measure the number of business processes analyzed on behalf of the security expert
CSLM II Measure the number of human interventions to generate a role model
CSLM I.1 Measure the precision of generated access permissions
CSLM I.2 Measure the recall of generated access permissions
CSLM I.1 Measure the precision of identified ACR breaches
CSLM I.2 Measure the recall of identified ACR breaches
CSLM V.1 Measure the precision of generated ACR mappings
CSLM V.2 Measure the recall of generated ACR mappings
CSLM II.1 Measure the precision of generated ACR mappings
CSLM II.2 Measure the recall of generated ACR mappings
MLM II Show that all steps to generate the role model are automated
MLM I.1 Show that each business process metamodel element affecting the role model has an automated transformation
MLM I.2 Show that the extraction of the role model considers every type of activity that has an associated data object
MLM V Show that the metamodel elements permission and role are always created with an association to activity and business process in the ACR mapping model
MLM III Show that no additional information for business processes nor EAA is required to conduct an architectural data flow analysis
MLM IV Show that the number of user interactions needed to conduct an architectural data flow analysis is low
MLM II Show that an ACR mapping is created for each data flow constraint

**Figure 5.1:** Shows the overall GQM model for both case studies.

Figure 5.1 shows four goals regarding the approaches introduced in Chapter 3. Nine questions are derived from these goals. The goals are formulated based on the contributions from Section 1.6 of this thesis. Questions are formulated based on the research questions from Section 1.3 of this thesis. In the top of each question the prefixes *CS1* and *CS2* indicate whether the question is part of the first case study, the second case study or both. Section 5.1 and Section 5.2 will explain the respective parts of the GQM model in more detail. The first case study validates the extraction of ACRs from business processes to form a role model for RBAC. The corresponding concept was introduced in Section 3.1.1. This case study uses BAcsTract, which was described in Sections 3.2.1.1 and 3.2.3, as the realization of the concept. The second case study validates the identification of ACR breaches in EAAs. The corresponding concept was introduced in Section 3.1.2. Therefore, AcsALign is used as the realization of the concept. AcsALign was described in Sections 3.2.1.3 and 3.2.4. In the second case study AcsALign is used in conjunction with PAcsTract. PAcsTract is used to extract ACRs from business processes that are an input of AcsALign. Consequently, the accuracy of extracted ACRs from business processes by PAcsTract is also validated in the second case study.

## 5.1 Case Study I

This section presents the validation of the role model extraction from business processes using BAcsTract. In this case study, BAcsTract is validated on the case study CoCoME. CoCoME is a community driven case study for collaborative empirical research on software evolution approaches [108]. It represents a comprehensive trading system of a supermarket chain and consists of 17 business processes that reflect the business around the cash desk, the inventory of the stores and the management of stores from an enterprise perspective. They span a comprehensible set of business processes that contain interactions between actors, data type definitions and data usage descriptions. Section 5.1.1 explains the validation goals and derives research questions of the GQM model [181]. The case study examines to which extend the utilization of BAcsTract reduces the amount of human errors during the role engineering process, increases the efficiency of engineering the role model and quickens the adaptations during evolution scenarios. The case study system is described in Section 5.1.2. Afterwards, Section 5.1.3 discusses the results, followed by a discussion of threats to validity in Section 5.1.4. Finally, Section 5.1.5 summarizes the scientific findings of the case study.

### 5.1.1 Validation Goals and Research Questions

The case study of this section has the goal to validate the research questions introduced from Section 1.3 that pertain to the extraction of a role model from business processes. The validation is structured according to the GQM method [181]. Goals represent the validation objectives that are desired to be achieved. Thus, they are subdivided into research questions. These research questions are validated using metrics. This allows to answer questions that contribute to goals.

Figure 5.2 illustrates the GQM model for this case study. The green boxes in the top part of Figure 5.2 represent the goals of the GQM model. The yellow cards represent the research questions that are derived from the goals. Arrows connect goals with questions that need to be answered in order to achieve the goals. To confirm a hypothesis proposed by a question, the corresponding metrics need to be reached. Questions and metrics are also connected with arrows. The red rounded boxes in the lower part of Figure 5.2 describe the case study level metrics (CSLM). These metrics are formulated for the case study under research and need to be evaluated by applying BAcsTract to the case study system and by analyzing the results. If all CSLMs of a question are met, it testifies that the hypothesis of the question is confirmed for the given case study. The gray rounded boxes describe metamodel level metrics (MLM). They are formulated on the metamodel level, so that they can be answered by the fundamental constructs of the approach. If a hypothesis of a question is confirmed by an MLM, it indicates that the hypothesis is confirmed for every specific occurrence of the problem (e.g., a case study experiment) and thus, the corresponding goal is satisfied for every instance of the problem.

**Goal 1 - More efficient engineering of the role model:** In Chapter 1 and Section 2.2 I explained that the typical role engineering process without BAcsTract, in which the role model for RBAC is engineered, is manually carried out by security experts. They have to go through a vast amount of business processes, which are complex and interrelated, to elicit a role model for RBAC. The complexity of the role engineering process (and further problems described in Section 1.2), are the main causes why engineering a role model is slow, error-prone and cost-intensive. One of the major concepts of this thesis, namely to automatically extract a role model from business processes tackles these problems. Therefore, the first goal is to validate whether the engineering of the role model with BAcsTract is more efficient than the typical 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 engineering of the role model 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 role engineering process is tedious and requires analyzing a vast amount of BPs by hand. Chapter 1 and Section 2.2 explained that the number of business processes can easily grow into hundreds of complex and interrelated processes. As the role engineering process is complex, it creates room for human errors during the engineering of the role model. Each error poses a major threat to the security of the overall system as a small error can lead to a severe security breach leaking sensitive information and allowing access to forbidden services (see Chapter 1). Therefore, the second goal is to validate that the proposed approach to extract a role model from business processes reduces the amount of human errors which security experts make during the engineering of the role model. Hence, the goal is to validate that BAcsTract automates the parts of the role engineering process in which the security experts have to analyze the vast amount of business processes manually and by doing so, reduces human errors.

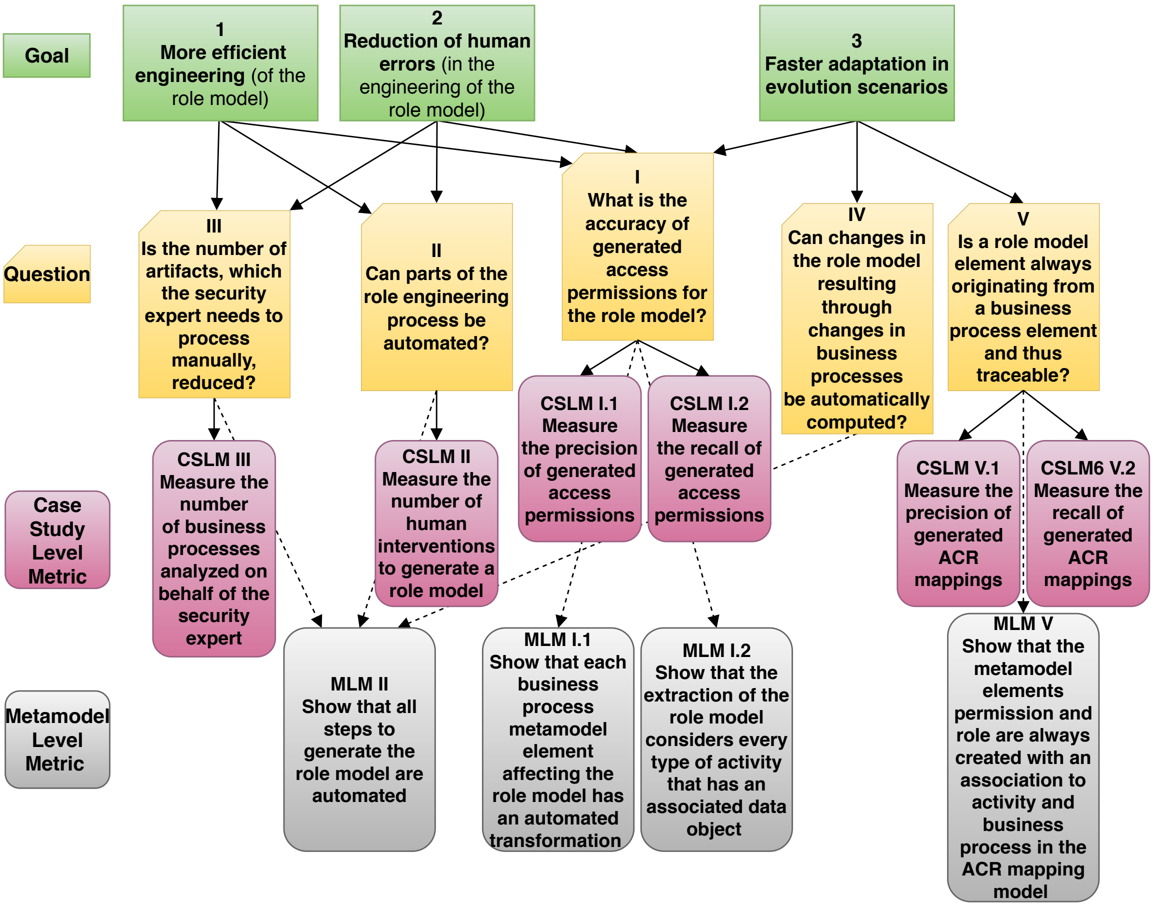

Goal
1
More efficient engineering (of the role model)
2
Reduction of human errors (in the engineering of the role model)
3
Faster adaptation in evolution scenarios
Question
I
What is the accuracy of generated access permissions for the role model?
II
Can parts of the role engineering process be automated?
III
Is the number of artifacts, which the security expert needs to process manually, reduced?
IV
Can changes in the role model resulting through changes in business processes be automatically computed?
V
Is a role model element always originating from a business process element and thus traceable?
Case Study Level Metric
CSLM I.1
Measure the precision of generated access permissions
CSLM I.2
Measure the recall of generated access permissions
CSLM II
Measure the number of human interventions to generate a role model
CSLM III
Measure the number of business processes analyzed on behalf of the security expert
CSLM V.1
Measure the precision of generated ACR mappings
CSLM6 V.2
Measure the recall of generated ACR mappings
Metamodel Level Metric
MLM I.1
Show that each business process metamodel element affecting the role model has an automated transformation
MLM I.2
Show that the extraction of the role model considers every type of activity that has an associated data object
MLM II
Show that all steps to generate the role model are automated
MLM V
Show that the metamodel elements permission and role are always created with an association to activity and business process in the ACR mapping model

**Figure 5.2:** The GQM model for the case study regarding the extraction of a role model from business processes [177].

**Goal 3 – Faster adaptation in evolution scenarios:** I explained in Chapter 1 that business processes have a lifecycle and thus, change constantly over time. In addition, their interrelations with access permissions are complex. Thus, evolution scenarios of business processes may lead to changes in ACRs and as a result, require changes of access permissions in the role model. To identify and adjust these changes security experts are required to conduct the role engineering process, at least for the changed processes as well as for interrelated processes. This evolutional adjustment is slow, error-prone and cost-intensive, as it is the typical role engineering process (see Chapter 1 and Section 2.2). The third goal is to validate that utilizing the approach to extract a role model from business processes automatically allows a faster adaptation of the role model during evolutional changes of business processes. In addition, it is to validate that the approach increases the understanding of mutual dependencies between business processes and access permissions of the role model. Hence, the third goal is to validate that BAcsTract can automatically compute the adapted role model in evolution scenarios of business processes and establish an ACR mapping model that helps the business level and IT level to understand mutual dependencies between business processes and RBAC.

Each goal consists of research questions which formulate hypotheses that need to be confirmed in order to achieve the goal. In the following sections, the questions and corresponding metrics will be explained.

#### 5.1.1.1 Accuracy of the Role Model

In order to achieve any of these goals a crucial point is the accuracy of the extracted role model. To be precise, it is fundamental to examine whether BAcsTract extracts correct ACRs from business processes and whether they are complete in terms of their number. Therefore, question I:

- What is the accuracy of generated access permissions for the role model?

claims hypotheses

- **H I.1:** The transformation of ACRs from business processes into the role model is semantically correct.
- **H I.2:** All ACRs from business processes are transferred into access permissions of the role model.

To answer question I, BAcsTract is executed with the case study system. Each access permission of the generated role model is classified based on a reference list of ACRs for the given business processes. The reference list is made independently by two postgraduates. They analyzed the business processes manually. Their results were compared to avoid mistakes. An access permission is a true positive $t_p$ if the access permission has an exact counterpart in the reference list. An exact counterpart means that the lane and the pool of an ACR correspond to the role in the access permission, that the data object of an ACR corresponds to the data object of the access permission and that the input/output association of the data object corresponds to the read/write operation in the access permission. It is a false positive $f_p$ if there is no exact counterpart in the reference list. 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 following two established CSLMs:

- **CSLM I.1:** Precision $P = \frac{t_p}{t_p+f_p}$, to address hypothesis H I.1.
- **CSLM I.2:** Recall $R = \frac{t_p}{t_p+f_n}$, to address hypothesis H I.2.

Hypotheses H I.1 and H I.2 can be also answered on the metamodel level. Therefore, the following MLMs need to be proven:

- **MLM I.1:** To address hypothesis H I.1 BAcsTract requires a transformation for every metamodel element of business processes that do affect the access permission in the role model. In Section 3.2.3 I reviewed which metamodel elements of business processes affect the role model. The role model consists of permissions. Permissions are tuples of roles, objects and operations. In the business process the metamodel element *data object* and its input/output association are relevant, as data objects associated with activities state access privileges needed to fulfill the activity. In addition, the participant who carries out the activity has to be considered as he requires access to the data objects. Participants are represented by the metamodel elements *lane* and *pool.* The lane is the executing instance of activities inside the lane. Hence, the algorithm of BAcsTract needs to be examined whether it has a transformation for a) the metamodel elements *lane* and *pool* of the business processes to the metamodel element *role* of the role model and b) the metamodel element *data object* and its input/output association of the business processes to the metamodel element *object* and *operation* of the role model.

- **MLM I.2:** To address hypothesis H I.2 BAcsTract needs to generate an access permission for every activity that has an associated data object. Therefore, the steps of BAcsTract need to be analyzed regarding whether they consider every type of activity that has an associated data object during the extraction of the role model.

#### 5.1.1.2 Automation of Role Engineering

To achieve a more efficient engineering of the role model (goal 1) and a reduction of human errors during the engineering of the role model (goal 2) BAcsTract automates the extraction of the role model from business processes. Hence, the automation has to be examined. Question II:

- Can parts of the role engineering process be automated?

claims hypothesis

- **H II**: A part of the role engineering process, in which security experts elicit the role model from business processes, can be automated.

To answer question II, the process of executing BAcsTract with the case study system is examined. The required amount of human interventions is analyzed for all parts starting from the preparation of input models, to the utilization of BAcsTract and to the generation of the role model. A human intervention *i* means that the security experts have to conduct some manual task in order for BAcsTract to begin or continue its work. Simple tasks as selecting the input model and pressing correct buttons to launch BAcsTract are excluded in this case study, as they impose no significant effort. Instead, this case study focuses on tasks that impose serious effort, for example, extending or designing new models. This classification is used to calculate:

- **CSLM II:** Number of Human Interventions $I = \sum i$, to address hypothesis H II.

Hypothesis H II can be also answered on the metamodel level. Therefore, the following MLM needs to be proven:

- **MLM II:** To address hypothesis H II it has to be shown that a) the input models for BAcsTract do not require any extended modeling and b) that all transformation steps to generate the role model are automatic. Therefore, the input models for BAcsTract introduced in Section 3.2.1.1 and the steps of BAcsTract to generate the role model introduced in Section 3.2.3 have to be examined.

#### 5.1.1.3 Manual Processing of Artifacts

In order to achieve a more efficient engineering of the role model (goal 1) BAcsTract automates the analysis of the vast amount of business processes to generate a role model. Hence, it has to be examined how many business level artifacts BAcsTract analyzes on behalf of security experts automatically. Therefore, question III:

- Is the number of artifacts, which the security expert needs to process manually reduced?

claims hypothesis

- **H III:** The security experts do not have to analyze business processes manually to engineer the role model.

To answer question III, BAcsTract is executed with the case study system. Each business process that is analyzed automatically by BAcsTract and does not require a human intervention, is counted as $bp_a$, an artifact that the security experts do not need to analyze manually to engineer the role model. This number is compared to the number of processes that need to be analyzed by hand during the traditional role engineering $bp_m$. To classify the number of business processes that are analyzed on behalf of the security experts the percentage of automatically analyzed business processes is calculated with the following CSLM:

- **CSLM III:** Percentage of Automatically Analyzed Business Processes $A = 100 * (\frac{bp_a}{bp_m})$, to address hypothesis H III.

Hypothesis H III can be also answered on the metamodel level. Therefore, the following MLM needs to be proven:

- **MLM III:** To address hypothesis H III all transformation steps of BAcsTract in which the business processes are analyzed to generate the role model have to be automatic and without human interventions. Therefore, the metric MLM II is reused to examine this hypothesis.

#### 5.1.1.4 Automatic Computation of Changes in Evolution Scenarios

To achieve a faster adaptation of RBAC during evolution scenarios (goal 3) BAcsTract automates the generation of the role model. Thus, it Can changes in the role model resulting from changes in business processes has to be examined whether changes in the role model resulting from evolution of business process can be automatically generated. Question IV:

- Can changes in the role model resulting from changes in business processes be automatically computed?

claims hypothesis

- **H IV:** BAcsTract can compute role model changes resulting from changes of business processes automatically.

Question IV is answered on the metamodel level. Therefore, two points have to be examined. It has to be analyzed that a) BAcsTract does not require any human interventions to further adapt the input models after the evolution of business processes and b) all transformation steps of BAcsTract to generate the new role model are automatic. A human intervention means that the security experts have to conduct some manual task in order for BAcsTract to begin or continue its work. We exclude tasks as selecting the input models and pressing correct buttons to launch BAcsTract, as we focus on tasks that impose serious effort, for example, extending or designing new models. Therefore, the following MLM needs to be proven:

- **MLM II:** To address hypothesis H IV it has to be shown that a) the input models for BAcsTract do not require any extension and b) that all transformation steps to generate the role model are automatic. Here, the metric MLM II will be reused for part b).

#### 5.1.1.5 Traceability of Role Model Elements

To achieve a faster adaptation of RBAC during evolution scenarios (goal 3) an ACR mapping model is established by BAcsTract. It interconnects elements from the role model with elements of the business processes. This ACR mapping model automatically documents design decisions and helps the business level and IT level to understand mutual dependencies between business processes and RBAC. Hence, the precision and recall of the generated ACR mappings in the ACR mapping model have to be examined. Therefore, question V:

- Is a role model element always originating from a business process element and thus, traceable?

claims hypotheses

- **H V.1:** The ACR mapping information for generated access permissions of the role model are semantically correct.
- **H V.2:** All role model elements (role and permission) are traceable to their originating business process elements (process, lane, activity and data object) in the ACR mapping model.

To answer question V, BAcsTract is executed with the case study system. Each generated ACR mapping in the ACR mapping model is classified based on a reference list of ACR mappings for the given business processes. The reference list is made independently by two postgraduates. They analyzed the business processes manually. Their results were compared to avoid mistakes. An ACR mapping is a true positive $t_p$ if the ACR mapping has an exact counterpart in the reference list. An exact counterpart means that the tuple role, process, activity and permission correspond to the entry in the reference list. It is a false positive $f_p$ if there is no exact counterpart in the reference list. A false negative $f_n$ occurs, if there is an ACR mapping in the reference list for which no ACR mapping is generated by BAcsTract. This classification is used to calculate the following two established CSLMs:

- **CSLM V.1:** Precision $P = \frac{t_p}{t_p+f_p}$, to address hypothesis H V.1.
- **CSLM V.2:** Recall $R = \frac{t_p}{t_p+f_n}$, to address hypothesis H V.2.

Hypotheses H V.1 and H V.2 can be also answered on the metamodel level. Therefore, the following MLM needs to be proven:

- **MLM V:** To address hypotheses H V.1 and H V.2 it needs to be proven that each generated role model element (role and permission) is created based on a tuple of business process elements process, lane, activity and data object. If this is true, each generated role model element has originating business process elements to which it can be traced. Therefore, the steps of BAcsTract to generate an element for the role model introduced in Section 3.2.3 have to be analyzed.

### 5.1.2 Case Study System

The Common Component Modeling Example (CoCoME) is used as a case study system to validate BAcsTract. CoCoME is a community driven case study for collaborative empirical research on software evolution approaches [108]. It represents a comprehensive trading system of a supermarket chain illustrated in Figure 5.3. The case study covers the EAA of a supermarket enterprise with several supermarket stores as well as the corresponding business processes. The business processes cover store management, sales, inventory management and reporting. All respective IT systems that support the business processes are covered by the EAA.

The upper part of Figure 5.3 shows the CoCoME enterprise, from which all supermarket stores are managed. The enterprise manager is able to review various documents of the stores to manage them. Therefore, the CoCoME enterprise server connects to each CoCoME store via the store server. A store manager manages a particular store. The stock manager of a store is responsible for its inventory. Each store server connects a set of cash desks forming a cash desk line located in the lower part of Figure 5.3. The cash desk PC is the central unit of each cash desk and connects a cash box, card reader and bar code scanner. At the cash desk, the customer places products and the cashier scans them to process the sale.

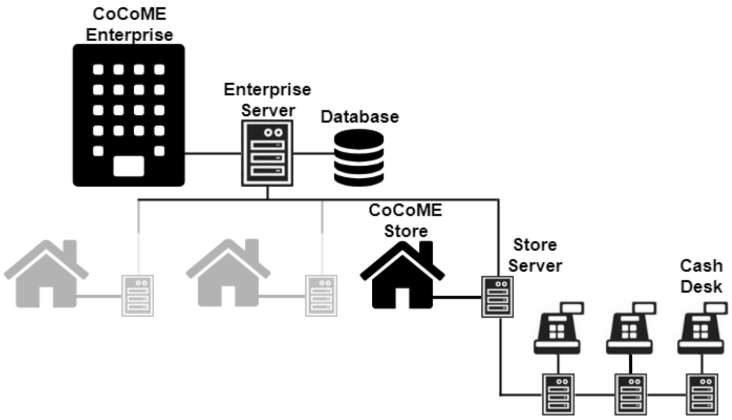


**Figure 5.3:** Simplified architecture of the CoCoME supermarket chain.

The case study consists of 17 business processes, eight processes are based on [108] and further nine processes extend the services of the CoCoME architecture developed during a bachelor thesis [233]. Figure 5.4 gives an overview over the business processes.

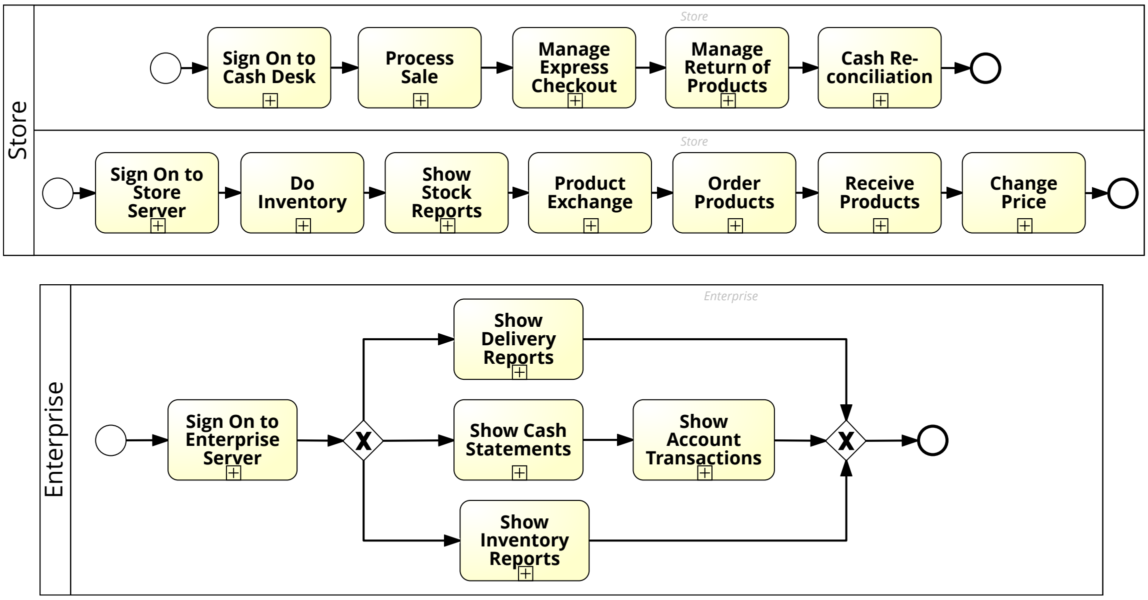


**Figure 5.4:** Overview of sub-processes for the CoCoME supermarket chain.

The upper part of Figure 5.4 shows the sub-processes that are part of the store. The upper lane gathers the sub-processes related to the cash desk. The sign on of the cashier at the cash desk, the processing of purchased goods by customers, the management of express checkouts of customers, the management of products that are returned by customers and finally, the cash reconciliation of the cash desk. The lower lane gathers the sub-processes related to the inventory of a store. There are the sign on at the store server by the marketing manager or store manager, the process for making inventory in the store, the access to stock reports, the exchange of products with other stores, the ordering and receiving of new products and the price change of products. The lower part of Figure 5.4 shows the sub-processes that are part of the enterprise. There are the sign on of the enterprise manager at the enterprise server, the access to delivery reports of a store, the access to cash statements as well as the account transactions of a store and finally, the access to inventory reports of a store.

The size of the case study is reasonable as can be seen in the business process characteristics of Table 5.1. Overall, there are 17 business processes containing seven unique roles in 48 lanes which comprise 166 activities and 294 flow transitions. In addition to the business processes, there are two reference lists. One reference list contains the ACR mappings and one reference list contains the ACRs of the business processes.

**Table 5.1:** Characteristics of the business processes of the CoCoME supermarket chain [177].

| Business Process Characteristic | # | Business Process Characteristic | # |
|---|---|---|---|
| Business processes | 17 | Activities | 166 |
| Lanes | 48 | Flow transitions | 294 |
| Roles | 7 | Data objects | 38 |
| Data object usage | 112 | Access control req. | 112 |
| Access control req. unique per role | 81 | | |

CoCoME is appropriate for examining ACRs and RBAC permissions, as many different IT security requirements stemming from various sources, e.g., laws, pertain the supermarket chain. Many of them impose ACRs. In [180] I examined the broad amount of security regulations that pertain the CoCoME supermarket chain. In case of privacy, the GDPR [222] regulates the collection, processing and use of personal data. There are manifold ACRs pertaining, for example, the establishment of an access control system, access rights, and the amount and storage of collected information. Other IT security related requirements stem from various fields of regulations, for example, company laws, commercial legislation, tele-media legislation, criminal law, contract law and labor legislation. Every country has their own jurisdiction in these fields. The company laws regulate the duties of organizations to organize, establish, and monitor their information security management. Practical advice is found in technical standards, for example, the ISO/IEC 27000-series [210] and the IT Baseline Protection [117] but also in business process guidelines, for example, ITIL [34] and COBIT [32]. They all propose various ACRs. Commercial legislation governs the bookkeeping as well as the storage and access of all financially relevant information. This legislation is particularly important for CoCoME, as CoCoME is a trading company with stores. In Germany, the law for IT-supported accounting systems [82] regulates all IT specific requirements for electronic data processing systems, as used by CoCoME. Authorization concepts, logging and traceability of transaction changes and tamper-resistant storage of financial information are some of the manifold ACRs imposed here. Criminal law, contract law and labor legislation propose further IT security and privacy requirements and thus, ACRs that are also important for CoCoME. Finally yet importantly, CoCoME itself has to define own IT security guidelines to protect information and processes that are critical to their business. In [180] I discussed these various sources of IT security requirements with regard to CoCoME and identified ACRs as one essential and major group of requirements.

### 5.1.3 Results and Discussion

The case study was conducted and the metrics were calculated as described in the previous sections. BAcsTract processed 17 business processes of the community case study CoCoME and produced an ACR mapping model and a role model. Further outputs were produced according to the description in Section 3.2.3.2. In the following, the results for the metrics and their implications for their associated research question will be presented. Afterwards, the implications for each goal will be discussed in Section 5.1.3.6. To fulfill a hypothesis proposed by a question, the corresponding metrics that are interconnected with arrows need to be accomplished. If any hypothesis of a question is confirmed by all of its metamodel level metrics, evidence is provided that the hypothesis is confirmed for every specific occurrence of the problem (e.g., a case study experiment) and thus, that the corresponding goal is satisfied for every instance of the problem. Table 5.2 summarizes the results for the CSLMs.

**Table 5.2:** Summarizes results for the CSLMs.

| Nr. | Metric | Result |
|---|---|---|
| 1 | CSLM I: access permissions in the role model | 81 |
| 2 | CSLM I.1: precision of the role model | 1.0 |
| 3 | CSLM I.2: recall of the role model | 1.0 |
| 4 | CSLM II: number of human interventions before execution | 0 |
| 5 | CSLM II: number of human interventions during execution | 0 |
| 6 | CSLM III: amount of artifacts analyzed on behalf of security experts | 17 |
| 6 | CSLM III: amount of artifacts in the case study | 17 |
| 6 | CSLM III: percentage of automatically analyzed business processes | 100 |
| 7 | CSLM V: ACR mappings in the ACR mapping model | 112 |
| 8 | CSLM V.1: precision of the ACR mapping model | 1.0 |
| 9 | CSLM V.2: recall of the ACR mapping model | 1.0 |

#### 5.1.3.1 Accuracy of the Role Model

To measure the accuracy of generated access permission from the role model the CSLM I.1 precision and CSLM I.2 recall is calculated for the case study. The classification of generated access permissions against the reference list of ACRs yields the following results for the accuracy measurement: 81 true positives, zero false positives and zero false negatives. Hence, the result for CSLM I.1 precision is $P = \frac{81}{81+0} = 1.0$ and for CSLM I.2 recall is $R = \frac{81}{81+0} = 1.0$. This confirms hypothesis H I.1 and H I.2 and means that BAcsTract extracted all access permissions correctly from the business processes according to the previously defined rules. Table 5.3 shows an excerpt of the generated access permissions of the role model.

**Table 5.3:** Excerpt of the generated role model [177].

| Role | Permission |
|---|---|
| Store:Store Manager | Read Stock report |
| Store:Store Manager | Write Staff schedule |
| Store:Store Manager | Write Inventory listing |
| Store:Cashier | Read Cashier ID |
| Ent.:Ent. Manger | Read Business transaction |
| Ent.:Ent. Manger | Read Inventory report |

To answer question I, *What is the accuracy of generated access permissions for the role model?* on the metamodel level MLM I.1 and MLM I.2 are answered:

- MLM I.1: To address hypothesis H I, that *the transformation of ACRs from business processes into the role model is semantically correct*, it has to be examined whether BAcsTract has a transformation for every metamodel element of the business processes that affects the access permissions in the role model. Hence, the algorithm of BAcsTract needs to be examined for a transformation from a) the metamodel elements *lane* and *pool* of a business process to the metamodel element *role* in the role model and b) the metamodel element *data object* and its input/output association of a business process to the metamodel elements *object* and *operation* of role model. BAcsTract applies six extraction steps, explained in Section 3.2.3.2, the same way to all business processes served as input. Regarding a) the first step is relevant which is described in Section 3.2.3.2 comprehensively. In the first step, BAcsTract analyzes the metamodel elements *lane* and *pool* to extract the role for the role model. This means that the algorithm of BAcsTract has a transformation for the metamodel elements *lane* and *pool* of a business process. Regarding b) the fourth step is relevant. It is described comprehensively in Section 3.2.3.2. In the fourth step, BAcsTract analyzes the metamodel element *data object* and its input/output association and transforms them to the metamodel elements *object* and *operation* of role model. Hence, the transformation regarding b) exists. As a conclusion, MLM I.1 confirms hypothesis H I.1 on the metamodel level meaning that the algorithm of BAcsTract is designed in a way that it extracts business level ACRs semantically correct.
- MLM I.2: To address hypothesis H I.2, that *all ACRs from business processes are transferred into access permissions of the role model*, it has to be examined whether BAcsTract generates an access permission for every activity that has an associated data object. BAcsTract applies six steps to every business process served as input (see Section 3.2.3.2). As these steps are applied to every business process uniformly, it remains to examine whether during these steps an access permission is generated for every activity that has an associated data object. The steps that are relevant for this are the first and fourth step. They are explained comprehensively in Section 3.2.3.2.

In the first step, BAcsTract analyzes the metamodel elements *lane* and *pool* to extract the role for the role model. Consequently, each role is extracted from a business process as this step analyzes all lanes and pools of the business process. In the fourth step, BAcsTract analyzes the metamodel element *data object* and its input/output association to extract the object-operation pair if the data object is associated with an activity. Hence, every data object associated with an activity is analyzed and a permission in form of an object-operation pair is extracted. As data objects without any association have no influence on the business process (and also no meaning), BAcsTract successfully generates an access permission for every relevant data object that is associated to an activity. Hence, MLM I.2 confirms hypothesis H I.2 on the metamodel level meaning that the algorithm of BAcsTract transforms all business level ACRs of the type role-permission pair from business processes to access permissions in the role model.

#### 5.1.3.2 Automation of Role Engineering

To identify whether parts of the role engineering process can be automated with BAcsTract the number of human interventions to execute BAcsTract with the case study is measured. A human intervention is defined as a significant additional effort, for example, required modifications of models in order for BAcsTract to work. The baseline in this experiment is that the organization has designed state of the art business processes. The 17 business processes are modeled according to the de facto standard BPMN 2.0 [5]. In this experiment, two steps are examined. The first step is to prepare the input for BAcsTract. While preparing the input models for BAcsTract neither adaptation nor extensions of the BPMN input models were required. Thus, while preparing the input no additional human intervention is measured. The second step is the utilization of BAcsTract to generate the role model. BAcsTract extracts the role model with 81 entries automatically according to the defined steps in Section 3.2.3.2. During the execution, no human intervention was measured. This leads us to CSLM II the number of human interventions $I = 0 + 0 = 0$. This experiment confirms hypothesis H II that BAcsTract can extract the role model automatically and without any human intervention before and during the extraction process.

Question II *Can parts of the role engineering process be automated?* can be also answered on the metamodel level. Therefore, two points are examined: a) whether the input models for BAcsTract require any extension or modification and b) whether all transformation steps of BAcsTract to generate the role model are automatic. For a) Section 3.2.1.1 explains in detail which input models BAcsTract requires. BAcsTract operates on plain BPMN models which are the de facto standard for modeling business processes. These, models do not require any extension or modification in order for BAcsTract to extract the role model. Regarding b) Section 3.2.3.2 explains the six steps during which BAcsTract extracts the role model from the business processes. The first and second step extract the roles and the process names from the business processes. The third and fourth step extract the activities and permission from the business processes. Finally, the fifth and sixth generate the role model from the extracted information. As explained in Section 3.2.3.2, all six steps are transformations made by the algorithm itself. No human intervention is required. As a result, BAcsTract automates the part of the role engineering in which the business processes are analyzed to extract the business permissions for the role model. MLM II confirms hypothesis HII.

#### 5.1.3.3 Manual Processing of Artifacts

To measure the amount of artifacts which the security experts do not need to analyze manually, in CSLM III the number of business process that are analyzed automatically by BAcsTract are counted. The total number of business processes analyzed by BAcsTract on behalf of the security experts is $bp_a = 17$. The amount of business processes that otherwise would be needed to be analyzed manually is $bp_m = 17$. Thus, the percentage of automatically analyzed business processes is $A = 100 * (\frac{17}{17}) = 100\%$, meaning that all business processes of the case study are analyzed by BAcsTract on behalf of the security experts. Hence, the extracted role model comprises all ACRs from these business processes. This case study has only a small amount of business processes when compared to global operating organizations. In medium to large organizations, business processes grow easily into hundreds (see Chapter 1). This fact additionally underpins hypothesis H III, that the amount of artifacts the security experts do not need to process manually is reduced.

Question III *Is the number of artifacts, which the security expert needs to process manually reduced?* can be also answered on the metamodel level. The results of MLM II are discussed in the results of question 2 (see Section 5.1.3.2). They show that a) input models for BAcsTract do not require any extension nor modification and b) the steps of BAcsTract to extract the role model are fully automated. Hence, BAcsTract analyzes the business processes on behalf of the security experts and thus, reduces the amount of artifacts which security experts have to analyze manually. This confirms hypothesis HIII.

#### 5.1.3.4 Automatic Computation of Changes in Evolution Scenarios

In order to research whether changes in the role model resulting through changes in business process can be automatically computed, two points are examined in this metric:

a) does BAcsTract require any adaptation of input models after the evolution of business processes.

b) are all transformation steps of BAcsTract to generate the new role model automatic.

During the evolution of business processes they 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. Regarding a) BAcsTract operates on plain BPMN models and, thus no further modifications or extensions of the business processes are required after the evolution scenario. This means no additional manual effort is required in order for BAcsTract the analyze the business processes after the evolution scenario. Regarding b) the results of CLSM II and MLM II for question II (see Section 5.1.3.2) show that BAcsTract extracts the role model from the input models automatically and without any significant human intervention. This means that all transformation steps of BAcsTract are automatic which confirms hypothesis H IV that changes in the role model resulting through changes of business processes can be automatically computed and without any additional effort.

#### 5.1.3.5 Traceability of Role Model Elements

To measure the accuracy of the generated ACR mapping model CSLM V.1 precision and CSLM V.2 recall are calculated for the case study. The classification of generated ACR mappings against the reference list of ACR mappings yields the following results for the accuracy measurement: 112 true positives, zero false positives and zero false negatives. This yields a precision in CSLM V.1 of $P = \frac{112}{112+0} = 1.0$ and a recall in CSLM V.2 of $R = \frac{112}{112+0} = 1.0$. These results confirm hypothesis H V.1 and H V.2 and mean that BAcsTract generates correct entries in the ACR mapping model for the business process ACRs of the case study. Hence, each generated access permission is traceable to its originating business process, lane and activity.

To answer question V, *Is a role model element always originating from a business process element and thus, traceable?* on the metamodel level MLM V is answered. To address hypothesis H V.1 and HV.2 it has to be examined whether the elements role and permission of the generated role model from BAcsTract are created based on the business process elements lane, activity and data object. Therefore, the algorithm of BAcsTract from Section 3.2.3.2 is analyzed. BAcsTract does only extract role-permission pairs within the six steps described in Section 3.2.3.2. In the first four steps, the business process elements process, lane, activity and data object are analyzed to build the tuple of the ACR mapping model. This tuple is complete if the analyzed lane of the process has an activity with an associated data object. It means that a) BAcsTract generates tuples containing a role for each lane of the analyzed process and b) these tuples have a permission for each activity which is associated with a data object. The fifth step creates a simple hierarchy and thus, it does not change anything relevant in the ACR mapping model. In the sixth step, the role-permission pairs are extracted from the ACR mapping model. As stated in Section 3.2.3.2 a role-permission pair is only extracted if the tuple of the ACR mapping model has an entry for role and permission. As explained above, this is only the case if the tuple is based on the business process elements lane, activity and data object. Hence, MLM V confirms hypothesis H V.1 and H V.2 that each role-permission pair in the role model originates from a lane, activity and data object of a business process and thus, is traceable.

#### 5.1.3.6 Goals

The CSLM results as well as the MLM results confirm hypotheses H I to H V raised by the questions of the GQM model. This means, that the hypotheses H I to H V are confirmed for the case study and on the metamodel level. This indicates that the hypotheses will be confirmed by any case study with same characteristics regarding the input models.

Hypotheses H I.1 and H I.2 confirm that the extraction of ACRs from business processes and the generation of the initial role model is complete and semantically correct. This finding increases the significance of the three goals. On this basis, the following sections explain how hypotheses H I to H V answer these goals.

**More Efficient Engineering of the Role Model**

The results for question I show that the extracted access permissions have a high accuracy. BAcsTract correctly identifies all 81 unique access permission of the role model according to the scheme explained in Section 3.2.3. Results for question II and question 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 adaptation or modification of input models is required in order for BAcsTract to work as it operates on de facto standard BPMN models without any extensions. The case study results indicate that the vast amount of business processes an organization has can be automatically analyzed by BAcsTract to produce an initial role model that comprises business level ACRs of role-permission type. This relieves security experts of manually analyzing business processes making the role engineering process quicker. The amount of time needed for the role engineering process is reduced which leads to a reduction of costs. In addition, the complexity of the overall role engineering process is reduced. These arguments show that BAcsTract makes the role engineering process more efficient.

Another finding which makes the role engineering process more efficient arises from the automatic extraction of the initial role model. According to the results of question II the extraction requires no human intervention and takes only negligible amount of time so that the impact from variations of business processes on the role model can be predicted. BAcsTract can generate an initial role model and an ACR mapping model for variations of same business processes. The business level can use these results to compare how the variations of business processes affect access permission of RBAC. Along with the automated documentation of design decisions by the ACR mapping model BAcsTract enables IT and business level to make proper decisions regarding access control during the design of business processes.

**Reduction of Human Errors**

The case study results for question I show that BAcsTract extracts all 81 unique access permission of the role model correctly. Results for question II show that the extraction process is fully automated and does not require any human interventions like the extension or modification of input models. In conjunction with the results for question III this shows that the usage of BAcsTract reduces the amount of manual steps during the role engineering processes. The vast amount of business processes is analyzed on behalf of the security experts for the purpose of generating a role model comprising business level ACRs. This automation reduces the amount of human errors [101] during the role engineering process, as the manual analysis of the vast amount of complex and interrelated business processes is the major cause for human errors (explained in Chapter 1). BAcsTract relieves security experts of manually analyzing business processes to extract business level ACRs for the role model, making the role engineering process less error-prone.

#### Faster Adaptaion in Evolution Scenarios

The results for question I show that BAcsTract correctly extracts 81 unique access permissions. Results for question V show that the ACR mapping model is built correctly for all 112 ACRs of the business processes. Consequently, the accuracy of the ACR mapping model is high. The amount of ACRs in the ACR mapping model is higher than the amount of access permissions in the role model, as the role model comprises only unique access permissions. It means that two or more ACRs may lead to the same access permission. This permission is stored only once in the role model. Contrary, an access permission can originate from several business processes. Thus, to provide full traceability and gather all design decisions it is inevitable that ACR mappings for all originating business processes are identified. Results for question V show that they are identified and stored completely.

The results for question IV show that BAcsTract can be utilized during evolutional changes of business processes without imposing additional effort. As BAcsTract operates on de facto standard models, it does not require any modifications or extensions of input models. This means that BAcsTract can be utilized during the evolution of business processes to adapt the role model. Due to the fact that the extraction is automatic and imposes no additional effort, the adaptation of the role model becomes faster compared to the manual engineering by security experts. The relevance of this is also supported by the results of goal 1, which shows that BAcsTract makes the role engineering process more efficient in terms of alignment, performance and complexity.

Besides, BAcsTract allows to predict the impact of changes in the role model resulting from changes in business processes. This is achieved by the fact that computing adaptations for the role model requires only negligible amount of time and requires no human intervention. Thus, BAcsTract enables the business level and IT level to decide better 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, which otherwise would not be easy. This is especially true for the business level, as RBAC and the role engineering process is not part of their expertise. Consequently, mutual dependencies between business processes and RBAC can be understood better due to the traceability provided by the ACR mapping model.

### 5.1.3.7 Reduction of needed business knowledge

In this section, a goal is discussed that is not part of the case study but for which the case study results allow to draw a first conclusion. The goal is the reduction of needed business knowledge for the engineering of the role model. Results of questions I to question V indicate that required business knowledge of security experts is reduced. Typically, throughout the role engineering process, security experts have to analyze a vast amount of business processes (see Chapter 1 and Section 2.2 for further information). Depending on the size of the organization, the number of business processes can easily grow into hundreds of complex and interrelated processes. Results for question I to III show that BAcsTract can analyze these business processes on behalf of the security experts. In addition, results for question V show that the ACR mapping model comprises a trace for every generated access permission to its origination business process, lane and activity. These results indicate that security experts do not need to analyze business processes extensively and can focus more on technical ACRs which is the core part of their expertise. Hence, a first evidence can be drawn that the amount of needed business knowledge during the role engineering process is reduced.

### 5.1.4 Threats to Validity

Runeson et al. stated in [192] that four aspects of validity need to be discussed during case study research. Thus, the following sections discuss internal validity, external validity, construct validity and reliability.

#### 5.1.4.1 Internal Validity

Internal validity refers to the degree in which the claim about the cause of a case study is reliable and not influenced by unexpected factors.

I expect the input models, the algorithm of BAcsTract and the result classifications to influence the evaluation results. The factor that is analyzed in this case study is the BAcsTract algorithm. Regarding the input models I relied on the community driven case study CoCoME [104] that provides models for a realistic and comprehensive supermarket chain. CoCoME is used throughout many publications for empirical research and was developed by different research groups. The business process characteristics presented in Table 5.1 underpin the appropriateness of the case study size. In my publication [180] I showed that CoCoME is appropriate for examining security requirements as well as ACRs, which are analyzed in this case study. By choosing CoCoME I avoided creating a case study that is tailored to the approach. Regarding the result classifications a classification scheme is provided for every metric and explained in Section 5.1.1. If possible, established metrics are used for measurement. All reference lists are made separately by two postgraduates. Therefore, they analyzed the business processes manually. Afterwards, the resulting versions were compared to avoid mistakes.

#### 5.1.4.2 External Validity

External validity refers to the degree to which the conclusions of the case study can be generalized to other situations and environments.

According to Runeson et al. [192, p. 71] results of case studies cannot be generalized in a universal way as no statistically relevant sample has been drawn. This is a general problem in case study research. In the future work further case studies can be conducted in which the approach is applied on different cases. Nevertheless, the results of this case study can be generalized to cases with similar characteristics. The most relevant characteristic is certainly the input model language BPMN. BPMN is the de facto standard modeling language for business processes [23, 213]. This makes the results meaningful for a broad amount of other cases. The appropriateness of CoCoME for ACR research was already discussed in the previous section and was a research objective in my publication [180].

#### 5.1.4.3 Construct Validity

Construct validity refers to the extent the taken measures represent the matter of research.

If possible, established metrics as precision and recall for accuracy are used. Furthermore, a reasonable classification scheme is provided and explained. Section 5.1.1 explains for each goal how the research questions and metrics are derived. Finally, the whole case study evaluation is structured according to the GQM method [181].

#### 5.1.4.4 Reliability

Reliability refers to the degree in which the conclusions of the case study depend on the conducted researchers.

For the evaluation, the following steps have been conducted: gathering input models, running the analysis and classifying results. Input models are not created by the author but provided from the community driven case study CoCoME. The steps of the algorithm are explained in Section 3.2.3.2. They are fully automated and do not require any significant human intervention. Hence, I could not influence the results during the first two steps. Regarding the last step, I explained in Section 5.1.1 how metrics and classifications are derived.

### 5.1.5 Summary

In this case study, BAcsTract (explained in Section 3.2.1.1 and Section 3.2.3.2) is validated with the community driven case study CoCoME. The validation is structured according to the GQM method [181] where goals represent the validation objectives that are desired to be achieved. To research these goals, they are subdivided into research questions which are validated using metrics. Three desired goals, illustrated in Figure 5.2, were analyzed in this case study: the efficiency of the role model extraction, the reduction of human errors and the adaptation of the role model during evolution scenarios. Therefore, BAcsTract was applied to 17 business processes of CoCoME. The five questions were answered by measuring ten metrics. The results of the case study show that the accuracy of generated access permissions is high. BAcsTract automates the analysis of the vast amount of business processes to extract a role model with business permissions. This is achieved without imposing additional effort through human interventions before and during the utilization of BAcsTract. The complexity of the role engineering process is reduced as the amount of artifacts which security experts have to manually analyze is reduced. Consequently, the engineering of the role model with BAcsTract is more efficient while reducing the potential for human errors. Furthermore, the results show that the accuracy of the generated ACR mapping model is high, allowing to trace generated access permissions to its originating business process, lane and activity. This model automatically documents design decisions allowing the business and IT level to better understand mutual dependencies between business processes and RBAC. The case study results also show that adaptations of the role model required during the evolution of business processes become faster when utilizing BAcsTract. Finally, the generated role model is better aligned with the business processes.

## 5.2 Case Study II

In this section, the identification of ACR breaches in EAAs using AcsALign is validated on a real-world case study of a national art gallery. It covers an EAA and a set of business processes for the preparation of an exhibition. There are ten business processes that cover the preparation for a new exhibition, internal negotiations and the borrowing and lending of artworks. They span a comprehensible set of business processes that contain interactions between actors, data type definitions and data usage descriptions. The business processes are supported by an EAA that consists of 16 systems with 22 interfaces. A central middleware interconnects all backend and frontend systems. The middleware aggregates services and data of the backend systems. Based on this aggregation, the frontend systems provide certain views tailored for different roles of the art gallery. Section 5.2.1 lists the validation goals and derives research questions of the GQM model [181]. This case study examines whether the utilization of AcsALign reduces the amount of human errors during the design of the EAA, helps during the resolution of errors and quickens the adaptations throughout evolution scenarios. In addition, PAcsTract is used for the extraction of ACRs from business processes and thus, is also partly the focus of the validation. The accuracy of the extracted ACRs with PAcsTract is validated. The case study system is described in Section 5.2.2. Afterwards, Section 5.2.3 discusses the results, followed by a discussion of threats to validity in Section 5.2.4. Finally, Section 5.2.5 summarizes the scientific findings of the case study.

### 5.2.1 Validation Goals and Research Questions

The case study of this section aims to validate the research questions from Section 1.3 that pertain to the identification of ACR breaches in EAAs. The validation is structured according to the GQM method [181]. Goals represent the validation objectives that are desired to be achieved. Thus, they are subdivided into research questions. These research questions are validated using metrics. This allows to answer questions that contribute to goals.

Figure 5.5 illustrates the GQM model for this case study. The green boxes in the top of Figure 5.5 represent the goals of the GQM model. The yellow cards represent the research questions that are derived from the goals. Arrows connect goals with questions that need to be answered in order to achieve the goals. To confirm a hypothesis proposed by a question, the corresponding metrics need to be reached. Questions and metrics are also connected with arrows. The red rounded boxes in the lower part of Figure 5.5 describe the case study level metrics (CSLM). These metrics are formulated for the case study under research and need to be validated by applying AcsALign to the case study system and by analyzing results. If all CSLMs of a question are met, it testifies that the hypothesis of the question is confirmed for the given case study. The gray rounded boxes describe metamodel level metrics (MLM). They are formulated on the metamodel level, so that they can be answered by the fundamental constructs of the approach. If a hypothesis of a question is confirmed by all MLMs, the hypothesis is confirmed for every specific occurrence of the problem (e.g., a case study experiment) and thus, the corresponding goal is satisfied for every instance of the problem.

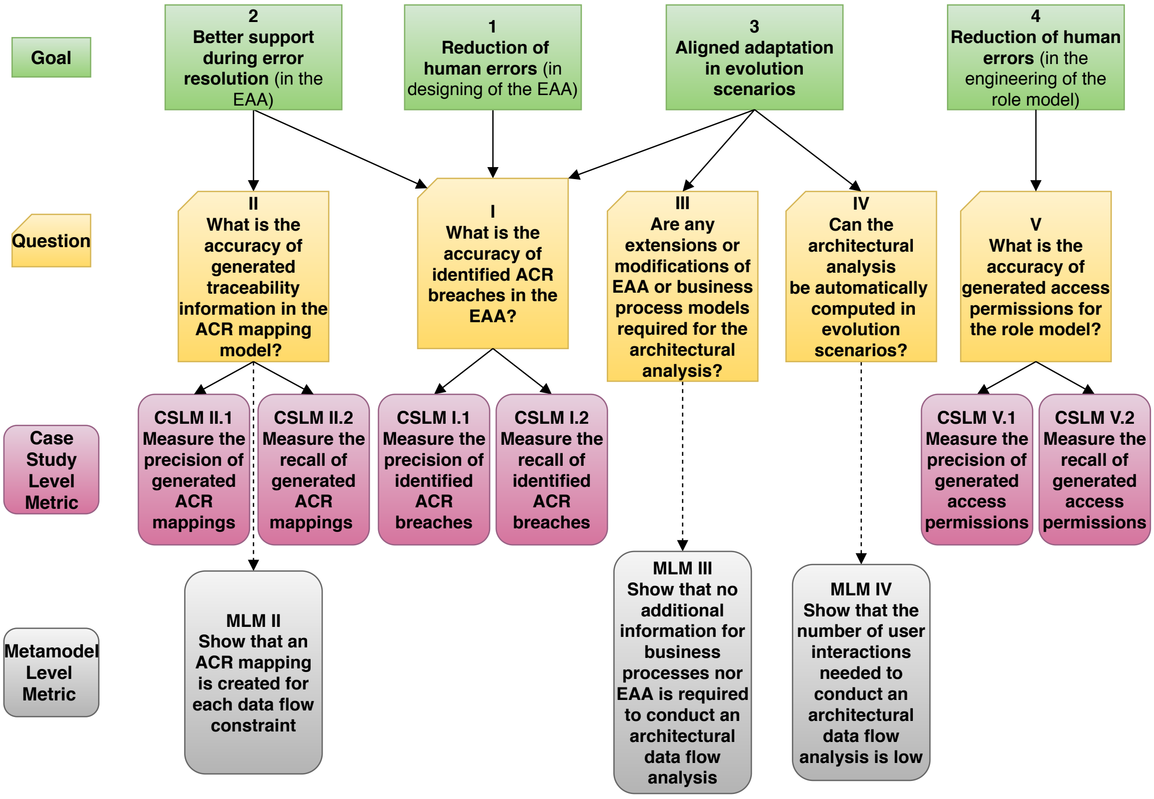

Goal
2 Better support during error resolution (in the EAA)
1 Reduction of human errors (in designing of the EAA)
3 Aligned adaptation in evolution scenarios
4 Reduction of human errors (in the engineering of the role model)
Question
II What is the accuracy of generated traceability information in the ACR mapping model?
I What is the accuracy of identified ACR breaches in the EAA?
III Are any extensions or modifications of EAA or business process models required for the architectural analysis?
IV Can the architectural analysis be automatically computed in evolution scenarios?
V What is the accuracy of generated access permissions for the role model?
Case Study Level Metric
CSLM II.1 Measure the precision of generated ACR mappings
CSLM II.2 Measure the recall of generated ACR mappings
CSLM I.1 Measure the precision of identified ACR breaches
CSLM I.2 Measure the recall of identified ACR breaches
CSLM V.1 Measure the precision of generated access permissions
CSLM V.2 Measure the recall of generated access permissions
Metamodel Level Metric
MLM II Show that an ACR mapping is created for each data flow constraint
MLM III Show that no additional information for business processes nor EAA is required to conduct an architectural data flow analysis
MLM IV Show that the number of user interactions needed to conduct an architectural data flow analysis is low

**Figure 5.5:** The GQM model for the identification of ACR breaches in EAAs.

**Goal 1 - Reduction of human errors in the designing of the EAA:** Designing the EAA is a complex task as an increasing amount of functional and non-functional requirements stemming from different stakeholders and sources need to be considered. In particular, modeling security and privacy requirements is complex and challenging [25]. Additionally, Chapter 1 explained that that the gap between the business level and IT level impedes a correct design of the EAA. Business processes and the EAA are mutually dependent and affect each other in non-trivial ways [9]. Consequently, it is challenging to align the EAA with business level ACRs correctly. As a result, the EAA has logical and design mistakes (see Chapter 1). While logical mistakes simply arise from faults and false solution approaches, design mistakes arise from unclear, false interpretation and misunderstanding of requirements. Therefore, goal one is to validate that the approach to identify ACR breaches in EAAs reduces the amount of logical and design mistakes in the EAA. Hence, the goal is to validate the accuracy of identified ACR breaches by AcsALign.

**Goal 2 – Better support during error resolution in the EAA:** Chapter 1 explained that the complexity of designing the EAA and the gap between the business level and IT level leads to logical and design mistakes in the EAA. If such mistakes are found, they need to be comprehensively understood so that they can be correctly resolved. One of the major concepts of this thesis, namely to automatically identify ACR breaches in EAAs, tackles this problem. Hence, the second goal is to validate whether AcsALign is able to generate an ACR mapping model with traceability information that interconnects elements of the EAA with affected parts of the business processes and access control.

**Goal 3 – Aligned adaptation in evolution scenarios:** I explained in Chapter 1 that business processes and the EAA have a lifecycle and thus, change constantly over time. In addition, their interrelations regarding ACRs are complex. This is why evolution scenarios of business processes may require changes in the EAA. To identify and adjust these changes the enterprise architect has to work through the business processes and adjust the EAA manually. During this complex manual process the potential for human errors is high leading to logical and design mistakes in the EAA (see Chapter 1). The third goal is to validate that utilizing the automatic approach to identify ACR breaches in EAAs allows to align the required adaptations of the EAA with regard to ACR during evolutional changes of business processes, as mistakes can be identified and resolved during the design time. Hence, the third goal is to validate that AcsALign can identify ACR breaches in the EAA during evolution scenarios with a reasonable amount of effort.

**Goal 4 - Reduction of human errors in the engineering of the role model:** The role engineering process is tedious and requires analyzing a vast amount of BPs by hand. Chapter 1 and Section 2.2 explained that the number of business processes can easily grow into hundreds of complex and interrelated processes. As the role engineering process is complex, it offers room for human errors during the engineering of the role model. Each error poses a major threat to the security of the overall system, as a small error can lead to a severe security breach leaking sensitive information and allowing access to forbidden services (see Chapter 1). Therefore, goal four is to validate that the proposed approach to extract a role model from business processes reduces the amount of human errors made by security experts during the engineering of the role model. Hence, the goal is to validate the accuracy of the automatically generated role model of PAcsTract.

Each goal consists of research questions which formulate hypotheses that need to be confirmed in order to achieve the goal. In the following sections, the questions and corresponding metrics will be explained.

#### 5.2.1.1 Accuracy of Identified ACR Breaches

To achieve a reduction of human errors during the design of the EAA (goal 1), a better support during the error resolution of ACR breaches in the EAA (goal 2) and a faster adaptation of the EAA in evolution scenarios (goal 3), it is fundamental to consider the accuracy of identified ACR breaches. To be precise, it is fundamental to examine whether AcsALign identifies ACR breaches correctly and completely in terms of number. Therefore, question I:

- What is the accuracy of identified ACR breaches in the EAA?

claims hypotheses

- **H I.1:** ACR breaches in the EAA are identified correctly.
- **H I.2:** All ACR breaches from the EAA are identified.

To answer question I, several mistakes are injected into the correct EAA. First, a categorization of all possible mistakes into mistake types of the same root cause is done. To violate an ACR a service call has to provide access for data to which no read access is intended or it has to trigger a write action for data for which no write access is intended. By considering all PCM elements that have an influence on data four mistake types can be derived:

1. a data type produced in a SEFF leading to an explicit data type assignment.
2. a service call in a SEFF leading to a wrong data type that is sent or received.
3. a wiring of systems or components that lead to a wrong data type that is sent or received.
4. a data type refinement that lead to an illegal composition of data.

For the validation of AcsALign it is sufficient to inject one mistake of each mistake type into the EAA. More mistakes of the same mistake type are handled the same way and thus, yield the same result. Second, for each mistake type a mistake is developed for the case study system and finally, injected into the models. Then AcsALign is executed with the case study system containing the injected mistakes. Each identified ACR breach in the EAA is classified based on a reference list of ACR breaches for the given EAA. The reference list is developed based on the injected mistakes and contains the injected mistakes, their expected effects and the violated ACRs. An ACR breach is a true positive $t_p$ if the ACR breach has an exact counterpart in the reference list. It is a false positive $f_p$ if there is no exact counterpart in the reference list. A false negative $f_n$ occurs if there is an ACR breach in the reference list for which AcsALign did not found a violation in the EAA. This classification is used to calculate the following two established CSLMs:

- **CSLM I.1:** Precision $P = \frac{t_p}{t_p+f_p}$, to address hypothesis H I.1.
- **CSLM I.2:** Recall $R = \frac{t_p}{t_p+f_n}$, to address hypothesis H I.2.

#### 5.2.1.2 Traceability of ACR Breaches

In order to achieve better support during the error resolution of ACR breaches in EAAs (goal 2) AcsALign generates an ACR mapping model. It interconnects elements from the EAA with elements of the business processes and RBAC. This ACR mapping model automatically documents design decisions and helps the business level and IT level to understand mutual dependencies between business processes and the EAA in terms of ACRs. It is especially helpful during the resolution of ACR breaches, done by the enterprise architect, as it allows to better understand the mistake that resulted in an ACR breach. Hence, the precision and recall of the generated ACR mappings in the ACR mapping model have to be examined. Therefore, question II:

- What is the accuracy of generated traceability information in the ACR mapping model?

claims hypotheses

- **H II.1:** The generated ACR mapping information for the ACRs of the EAA is semantically correct.
- **H II.2:** All data flow constraints are traceable to their originating business process elements (process, lane, activity and data object) in the ACR mapping model.

To answer question II, AcsALign is executed with the case study system. Each generated ACR mapping in the ACR mapping model is classified based on a reference list of ACR mappings for the given business processes. The reference list is made independently by two postgraduates. They analyzed the business processes manually. Their results were compared to avoid mistakes. An ACR mapping is a true positive $t_p$ if the ACR mapping has an exact counterpart in the reference list. An exact counterpart means that the tuple role, process, activity, business permission, data type, system, interface and service call correspond to the entry in the reference list. It is a false positive $f_p$ if there is no exact counterpart in the reference list. A false negative $f_n$ occurs if there is an ACR mapping in the reference list for which no ACR mapping is generated by AcsALign. This classification is used to calculate the following two established CSLMs:

- **CSLM II.1:** Precision $P = \frac{t_p}{t_p+f_p}$, to address hypothesis H II.1.
- **CSLM II.2:** Recall $R = \frac{t_p}{t_p+f_n}$, to address hypothesis H II.2.

Hypotheses H II.1 and H II.2 can be also answered on the metamodel level. Therefore, the following MLM needs to be proven:

- **MLM II:** To address hypotheses H II.1 and H II.2 it needs to be proven that a) for each access permission extracted from the business processes an entry is generated in the ACR mapping model and b) for each data flow constraint, on basis of which ACR breaches are identified, an entry is generated in the ACR mapping model. Therefore, the steps of PAcsTract to extract access permissions from business processes and AcsALign to identify ACR breaches in EAAs, introduced in Sections 3.2.3 and 3.2.4, have to be analyzed. If for each data flow constraint an entry in the ACR mapping model is generated that also links to a business process, the metric is true meaning that each ACR breach is traceable to its access permission in RBAC and the relevant parts of the business processes.

#### 5.2.1.3 Modification and Extension of Input Models

To achieve an aligned adaptation of the EAA during evolution scenarios (goal 3) it needs to be possible to utilize AcsALign during evolutional changes of business processes and EAAs. Therefore, it has to be examined whether the input models of AcsALign require any extensions or modifications during the evolution scenario of business processes or EAAs in order for AcsALign to identify ACR breaches. Therefore, question III:

- Are any extensions or modifications of EAA or business process models required for the architectural analysis?

claims hypothesis

- **H III:** AcsALign requires only a reasonable amount of effort to identify ACR breaches during evolution scenarios of business processes and EAAs.

Question III is answered on the metamodel level. Therefore, it has to be examined which modifications and extensions the input models of AcsALign require a) after the evolution of business processes and b) after the evolution of the EAA. In this thesis, extension means whether the input models require any extension of the modeling language and as such require the modeling of extended elements (e.g., when a BPMN extension is used, the extended elements need to be adapted to the evolution scenario). Modification means whether any further adaptation of the input models is required after the regular adaptations made during the evolution scenario. Using this definition the following MLM needs to be proven:

- **MLM III:** To address hypothesis H III it has to be shown that a) the input models for AcsALign do not require any extensions or modifications after the evolution of the business processes and b) the input models for AcsALign do not require any extensions or modifications after the evolution of the EAA. As AcsALign uses the outputs of PAcsTract in this case study, PAcsTract is also subject of research in this metric. Therefore, the characteristics of input models for AcsALign and PAcsTract described in Sections 3.2.1.2 and 3.2.1.3 are examined to answer hypothesis H III.

#### 5.2.1.4 Automatic Computation of ACR Breaches in Evolution Scenarios

In order to achieve an aligned adaptation of the EAA during evolution scenarios (goal 3) it needs to be possible to utilize AcsALign during evolution scenarios of business processes and EAAs. Therefore, this question examines whether AcsALign can identify ACR breaches after an evolution scenario automatically. Therefore, question IV:

- Can the architectural analysis be automatically computed in evolution scenarios?

claims hypothesis

- **H IV:** AcsALign can identify ACR breaches during evolution scenarios of business processes and EAAs automatically.

Question IV is answered on the metamodel level. Therefore, it has to be examined whether AcsALign requires any additional effort in form of human interventions during the identification of ACR breaches after the input models have undergone an evolution scenario. A human intervention means that the enterprise architect has to conduct some task in order for AcsALign to start or continue its work. Simple tasks, e.g., selecting the models to analyze and pressing correct buttons to launch AcsALign are excluded, as they impose no serious effort. Instead, this case study focuses on tasks that impose additional effort as the input of additional information or the modification of AcsALign specific models. Hence, the following MLM needs to be proven:

- **MLM IV:** To address hypothesis H IV it has to be shown that all transformation steps of AcsALign to identify ACR breaches do not require any human intervention after the evolution of business processes and EAAs. As AcsALign uses the outputs of PAcsTract in this case study, PAcsTract is also subject of research in this metric. Therefore, the algorithm steps of AcsALign and PAcsTract described in Sections 3.2.3.3 and 3.2.4.2 are examined for human interventions to answer hypothesis H IV.

#### 5.2.1.5 Accuracy of Extracted Access Permissions

In order to achieve a reduction of human errors during the engineering of the role model (goal 2) the accuracy of extracted access permissions needs to be measured. To be precise, it is fundamental to examine whether PAcsTract extracts correct access permissions from business processes and whether they are complete in terms of their number. Therefore, question V:

- What is the accuracy of generated access permissions for the role model?

claims hypotheses

- **H V.1:** The transformation of ACRs from business processes into the role model is semantically correct.

- **H V.2:** All ACRs from business processes are transferred into access permission of the role model.

To answer question V, PAcsTract is executed with the case study system. Each access permission of the generated role model is classified based on a reference list of ACRs for the given business processes. An access permission is a true positive $t_p$ if the access permission has an exact counterpart in the reference list. An exact counterpart means that the responsible role of an ACR corresponds to the role in the access permission, that the input/output data objects of an ACR corresponds to the data objects and their read/write operations of the access permission. It is a false positive $f_p$ if there is no exact counterpart in the reference list. A false negative $f_n$ occurs if there is an ACR in the reference list for which no access permission is generated by PAcsTract. This classification is used to calculate the following two established CSLMs:

- **CSLM V.1:** Precision $P = \frac{t_p}{t_p+f_p}$, to address hypothesis H V.1.
- **CSLM V.2:** Recall $R = \frac{t_p}{t_p+f_n}$, to address hypothesis H V.2.

### 5.2.2 Case Study System

The case study system is a real-world case study of a national art gallery. It covers an EAA and a set of business processes for the preparation of an exhibition. All models were already created, i.e. the author did not create them.

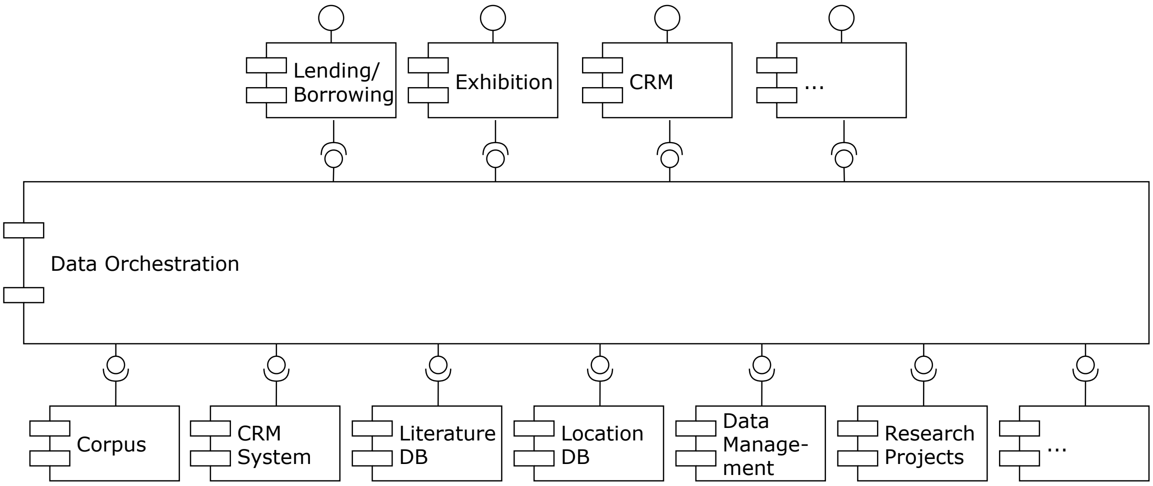


**Figure 5.6:** Excerpt of the EAA from the national art gallery.

Figure 5.6 shows an excerpt of the EAA from the national art gallery. The supporting system is illustrated in the middle of Figure 5.6. It is a middleware that interconnects different systems from the lower part of Figure 5.6, e.g., *Corpus* with systems from the upper part of Figure 5.6, e.g., *Lending/Borrowing*, to provide various views on the data of the lower systems required by the upper systems. The systems of the upper part of

Figure 5.6 provide the views by aggregating services and data. This is required for the different employee roles in the art gallery. For example, the systems *Corpus* and *CRM System* provide data and services which are combined in the *Data Orchestration*. Relevant parts of it are provided to systems like *Lending/Borrowing* or *Exhibition* which are then accessible to employees.

The case study consists of ten business processes. Figure 5.7 gives an overview over the business processes.

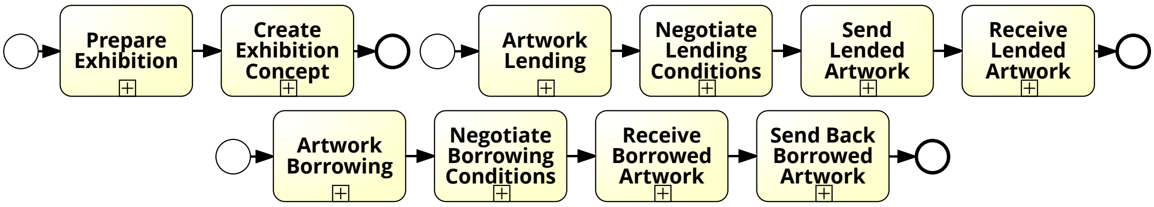


**Figure 5.7:** Overview of sub-processes for the exhibition preparation.

All business processes shown in Figure 5.7 are sub-processes required for the preparation of an exhibition. First, the curator prepares a concept for the exhibition including the planned artworks. Then the exhibition concept is finalized to go into the agreement phase where, for example, the budget agreement is negotiated with the directors. If this is done, the borrowing of external artworks begins. After the negotiation of conditions, the transport of the artwork is managed. It is usual that the gallery lends their artworks to other galleries, too. Therefore, also conditions are negotiated and afterwards, the transport is managed.

The size of the case study is reasonable for the evaluation of the approach as can be seen in the business process characteristics of Table 5.4. Overall, there are ten business processes defined in IntBIIS_LP containing twelve unique roles in 34 lanes which comprise 75 activities. They span a comprehensible set of business processes that contain interactions between actors, data type definitions and data usage descriptions. The EAA is defined in PCM including the system, repository and SEFFs. It contains 16 systems with 22 interfaces. Each system consists of several components. In total they use 48 different data types. In addition to the business processes and the EAA, there are three reference lists. One reference list contains the ACR breaches for the EAA, one reference list contains the ACR mappings of the ACR mapping model for the business processes and one reference list contains the ACRs of the business processes.

The business processes and the EAA of the national art gallery are appropriate for examining ACRs and ACR breaches in the EAA, as it is a real-world scenario of an EAA evolution due to digitalization. The set of business processes encompasses data flows of confidential information such as financial budgets, insurance values and customer/client data. The art gallery has also to comply with diverse laws, for example, the GDPR and financial regulation. Hence, it is a suitable case for analyzing ACRs. Regarding the ACR breaches, which are explained in Section 5.2.1.1, the EAA contains the usual data processing patterns for information systems, including delegation, merging, reading from and writing

**Table 5.4:** Characteristics of the business processes and the EAA of the case study [179].

| Business Process Characteristic | # | EAA Characteristic | # |
|---|---|---|---|
| Business processes | 10 | Systems | 16 |
| Lanes | 34 | Interfaces | 22 |
| Activities | 75 | Connections between systems | 21 |
| Data objects | 18 | Service calls | 41 |
| Data object usage | 56 | Data types | 48 |
| Roles | 12 | | |
| Access control req. | 83 | | |

to databases. There is a variety of confidential data flows of the previously mentioned sensitive information that spans over a complex EAA that interconnects various systems. Hence, designing a secure and correct EAA is challenging, making the case study suitable for the analysis of ACR breaches. Therefore, the four mistakes described in Section 5.2.1.1 are injected into the EAA to cover all relevant mistake types regarding ACR breaches.

### 5.2.3 Results and Discussion

The case study was conducted and the metrics were calculated as described in the previous sections. AcsALign processed ten business processes of the real-world case study of a national art gallery and produced an ACR mapping model and identified ACR breaches. Further outputs were produced according to the description in Section 3.2.4.2. Hereafter, the results for the metrics and their implications for the associated research question will be presented. Afterwards, the implications for each goal will be discussed in Section 5.2.3.6. To fulfill a hypothesis proposed by a question, the corresponding metrics that are connected by an arrow need to be accomplished. If any hypothesis of a question is confirmed by a metamodel level metric, evidence is provided that indicates that the hypothesis is confirmed for every specific occurrence of the problem (e.g., a case study experiment) and thus, that the corresponding goal is satisfied for every instance of the problem. Table 5.5 summarizes the results for the CSLMs.

#### 5.2.3.1 Accuracy of Identified ACR Breaches

The accuracy of identified ACR breaches for the case study is measured with CSLM I.1 precision and CSLM I.2 recall. The classification of identified ACR breaches from AcsALign against the reference list of ACR breaches yields the following results for the accuracy measurement: 6 true positives, zero false positives and zero false negatives. Thus, the precision in CSLM I.1 is $P = \frac{6}{6+0} = 1.0$ and the recall in CSLM I.2 is $R = \frac{6}{6+0} = 1.0$. The identified mistakes are shown in Table 5.6.

**Table 5.5:** Summarized results for the CSLMs.

| Nr. | Metric | Result |
|---|---|---|
| 1 | CSLM I: identified ACR breaches | 6 |
| 2 | CSLM I.1: precision of identified ACR breaches | 1.0 |
| 3 | CSLM I.2: recall of identified ACR breaches | 1.0 |
| 4 | CSLM II: ACR mappings in the ACR mapping model | 83 |
| 5 | CSLM II.2: precision of the ACR mapping model | 1.0 |
| 6 | CSLM II.2: recall of the ACR mapping model | 1.0 |
| 7 | CSLM V: generated access permissions unique per role | 44 |
| 8 | CSLM V.1: precision of generated access permissions | 1.0 |
| 9 | CSLM V.2: recall of generated access permissions | 1.0 |

**Table 5.6:** Results for the identification of ACR breaches with AcsALign (MT means mistake type and mistakes are set in italic) [179].

| MT | EntryLevelSystemCall | Read Data Types | Written Data Types |
|---|---|---|---|
| 1 | SaveLendingConfirmation | *LendingRequest* | LendingConfirmation |
| 2 | GetForeignArtwork | *OwnArtwork* | – |
| 3 | GetArtwork | Artwork, *ForeignArtwork* | – |
| 3 | GetArtworkForVerifying Transportation | Artwork, *ForeignArtwork* | – |
| 4 | CreateLendingContract | – | *LendingContract* |
| 4 | AddLendingContract | – | *LendingContract* |

1. First mistake type: In the service call for saving the approved lending request the system returns a lending request. Here nothing should be returned as only the approval should be persisted in the database.

2. Second mistake type: In this service call the system returns an own exhibit object but it must return a foreign exhibit object. The reason is that a wrong service is called internally.

3. Third mistake type: During the service call for getting the collection of exhibit objects the system returns too many data objects. Beside the exhibit object, it also returns the foreign exhibit objects which should not be returned in this case. The injected mistake is identified again during the service call for getting the collection of exhibit objects to verify the transportation of exhibit objects (see row four in Table 5.6).

4. Fourth mistake type: The service call to create a borrowing agreement returns the correct data objects. However, the data object borrowing agreement has been refined in a wrong way. It contains an exhibit object but should contain a public exhibit object. The injected mistake is identified again during the service call to store the borrowing agreement (see row six in Table 5.6). As with the previous service call,

this service call returns the correct data objects. However, the data object borrowing agreement has been refined in a wrong way.

All reported mistakes match the injected mistakes. For the mistake types three and four two ACR breaches are identified for each mistake type. The reason is that the injected mistakes are used by more than one service call. Thus, each of these service calls is forbidden with respect to the ACRs. The results confirm hypothesis H I.1 and H I.2, meaning that AcsALign successfully identified the injected EAA mistakes. This yields a good accuracy for the identification of ACR breaches in the current case study.

#### 5.2.3.2 Traceability of ACR Breaches

To measure the accuracy of the generated ACR mapping model CSLM II.1 precision and CSLM II.2 recall are calculated for the case study. The classification of generated ACR mappings from AcsALign against the reference list of ACR mappings yields the following results for the accuracy measurement: 83 true positives, zero false positives and zero false negatives. This yields a precision in CSLM II.1 of $P = \frac{83}{83+0} = 1.0$ and a recall in CSLM II.2 of $R = \frac{83}{83+0} = 1.0$. These results confirm hypothesis H II.1 and H II.2. This means AcsALign generates correct entries in the ACR mapping model for the analyzed ACRs of the case study. Hence, each data flow constraint and thus, each identified ACR breach is traceable to its originating access permission in RBAC and the relevant parts of the business processes.

To answer question II, *What is the accuracy of generated traceability information in the ACR mapping model?* on the metamodel level MLM II is answered. MLM II shall answer whether an entry in the ACR mapping model is generated for every data flow constraint that should be analyzed in the EAA.

To address hypothesis H II.1 and HII.2 it has to be examined whether a) for each access permission extracted from the business processes an entry is generated in the ACR mapping model and b) for each data flow constraint, on basis of which ACR breaches are identified, an entry is generated in the ACR mapping model. Regarding a) the algorithm of PAcsTract from Section 3.2.3.3 is analyzed for the generation of access permissions based on the business process elements actor step, responsible role, and input/output data object. PAcsTract extracts the access permissions in form of role-permission pairs within six steps described in Section 3.2.3.3.

In the first four steps, the business process elements bp usage scenario, responsible role, actor step and input/output data objects are analyzed to build the tuple of the ACR mapping model. This tuple is complete if the analyzed actor step of the process has input/output data objects. It means that a) PAcsTract generates tuples containing a role for each responsible role of an actor step of the analyzed process and b) these tuples have a permission for each actor step that has input/output data objects. The fifth step creates a simple hierarchy and thus, does not change anything relevant regarding the hypotheses. In the sixth step, the access permissions in form of role-permission pairs are extracted from the ACR mapping model. As stated in Section 3.2.3.3 a role-permission pair is only extracted if the tuple of the ACR mapping model has an entry for role and permission. As explained above, this is only the case, if the tuple is based on an actor step of a business process that has an entry for responsible role and input/output data objects. Hence, regarding a) for each access permission that is extracted from the business processes a complete tuple exists in the ACR mapping model.

To answer b) the algorithm of AcsALign from Section 3.2.4.2 is analyzed for the generation of an ACR mapping entry for each data flow constraint. AcsALign works in two steps. In the first step, information about the ACRs that need to be analyzed in the EAA are processed. Therefore, the ACRs in form of access permissions generated by PAcsTract are taken as input. For these ACRs an ACR mapping entry, connecting them to their originating business process elements, already exists as explained before. The first step generates for each ACR a data flow constraint based on the *EntryLevelSystemCalls*. At the same time the information regarding the EAA from the *EntryLevelSystemCalls* is stored in the ACR mapping model. In the second step, the previously generated data flow constraints are used to analyze the EAA for ACR breaches. As this step does only use the data flow constraints from the previous step and does not modify the ACR mapping model, only the first step is relevant to answer b). As stated before, in the first step the EAA information is extended in the ACR mapping model for each generated data flow constraint. Hence, regarding b) an entry in the ACR mapping model is generated for each generated data flow constraint. Consequently, MLM II confirms hypothesis H II.1 and H II.2 that each data flow constraint has an entry in the ACR mapping model connecting it with the access permission for RBAC and the relevant parts of the business processes. Thus, every identified ACR breach is traceable to the violated access permission and business process.

#### 5.2.3.3 Modification and Extension of Input Models

In order to research whether AcsALign can identify ACR breaches after an evolution scenario automatically, two points are examined:

a) does AcsALign require any adaptation of input models after the evolution of business processes and the EAA.

b) are all transformation steps of AcsALign to identify ACR breaches automatic.

While MLM III addresses a), MLM IV of Section 5.2.3.4 addresses b). In the following, MLM III of question III *Are any extensions or modifications of EAA or business process models required for the architectural analysis?* will address a) on the metamodel level. During an evolution scenario business processes are modified within the scope of IntBIIS_LP. The same applies to the EAA which is modified within the scope of PCM. We do not count these modifications as they reflect the evolution scenario itself and are done anyway. Regarding a) Section 3.2.1.3 describes the input required for AcsALign:

1. EAA models: the EAA is consumed in form of normal PCM models. Whether the EAA is changed during an evolution scenario does not affect the analysis of AcsALign,

as no further modifications or extensions besides the evolution scenario itself are required.

2. Set of ACRs: the ACRs that are analyzed on the EAA are provided by PAcsTract in this case study. It is also possible to use BAcsTract to get the required ACRs. Either IntBIIS_LP or BPMN models are required as input. In both cases, only normal IntBIIS_LP or BPMN models without any modifications or extensions are used. This means that evolution scenarios of business processes do not affect the analysis of AcsALign, as no further modifications or extensions besides the evolution scenario itself are required.

3. Mapping of data objects to data types: In case of PCM and IntBIIS_LP this is already part of the IntBIIS_LP models as architecture and business processes are tightly coupled. Besides, this mapping is static and requires rarely changes. Changes are only required if new data objects are introduced in the business processes and then the amount of modifications is very low as only one data type has to be connected for each new data object.

4. Mapping of activities to service calls: As in the previous case this is already part of the IntBIIS_LP models. The reason is that architecture and business processes are tightly coupled in PCM and thus, business processes in PCM have service calls as part of their design.

5. Actual read and written data types of invoked services: This input is provided by a simple data flow analysis which can be applied to any normal EAA and has to be executed after each evolution of the EAA. It takes only negligible amount of time and does not require any modification of the EAA models, as it only calculates the data flows of data types that are read and written during service invocations.

Consequently, no further modifications or extensions of input models are required after an evolution scenario when EAA and business processes are modeled in PCM and IntBIIS_LP. If BPMN is used to model business processes, minor modifications might be required in the mapping of data objects to data types and activities to service calls. However, for the case study under research in a) no additional manual effort to extend or modify the input models is required in order for AcsALign to start the identification of ACR breaches after an evolution scenario of business processes or the EAA.

#### 5.2.3.4 Automatic Computation of ACR Breaches in Evolution Scenarios

In order to research whether AcsALign can identify ACR breaches after an evolution scenario automatically, two points are examined:

a) does AcsALign require any adaptation of input models after the evolution of business processes and the EAA.

b) are all transformation steps of AcsALign to identify ACR breaches automatic.

MLM III from Section 5.2.3.3 addresses a). In the following, MLM IV will address b). During an evolution scenario business processes are modified within the scope of the IntBIIS_LP. The same applies to the EAA which is modified within the scope of PCM. As mentioned in Section 5.2.3.3 we do not count these modifications as they reflect the evolution scenario itself and are done anyway. Regarding b) the steps of AcsALign and PAcsTract are analyzed for human interventions. The algorithm of PAcsTract is explained in Section 3.2.3.3. It operates in six steps. The first four steps build the ACR mapping model by analyzing various elements of the business processes. In these steps no manual intervention is required (see Section 3.2.3.3). The fifth step creates a hierarchy and the sixth step generates the access permissions. Both steps are done based on the existing information in the ACR mapping model and thus, do not require any human interventions (see Section 3.2.3.3).

The algorithm of AcsALign is explained in Section 3.2.4.2. It operates in two steps. The first step generates the data flow constraints from the input models. This step does not require any human interventions (see Section 3.2.4.2). Afterwards, in the second step the EAA is analyzed for ACR breaches based on the data flow constraints and service calls. This step does not require any human interventions and the required service calls are provided by a simple data flow analysis which is also automatic (see Section 3.2.4.2). Hence, both steps do not require any human interventions. This MLM shows that all transformation steps of AcsALign are automatic and do not required any additional effort. This confirms hypothesis H IV that ACR breaches can be identified by AcsALign after evolution scenarios of business processes and EAAs automatically.

#### 5.2.3.5 Accuracy of Extracted Access Permissions

To measure the accuracy of generated access permission by PAcsTract CSLM V.1 precision and CSLM V.2 recall is measured for the case study. The classification of generated access permissions against the reference list of business process ACRs yields the following results for the accuracy measurement: 44 true positives, zero false positives and zero false negatives. Hence, the result for CSLM V.1 precision is $P = \frac{44}{44+0} = 1.0$ and for CSLM V.2 recall is $R = \frac{44}{44+0} = 1.0$. This confirms hypothesis H V.1 and H V.2 and means that PAcsTract extracted all access permissions correctly from the business processes according to the previously defined rules. Table 5.7 shows an excerpt of the generated access permissions of the role model.

#### 5.2.3.6 Goals

The CSLM results as well as the MLM results confirm hypotheses H I to H V raised by the questions of the GQM model. In addition, hypotheses H II, HII and H IV are confirmed on the metamodel level. This indicates that the hypotheses will be confirmed by any case study with same characteristics regarding the input models. On this basis, the following sections explain how hypotheses H I to H V answer the goals of the GQM model.

**Table 5.7:** Excerpt of the generated role model.

| Role | Permission |
|---|---|
| Executiv:Director | Read Financing plan |
| Executiv:Director | Read/Write Borrowing request |
| Executiv:Director | Read/Write Borrowing contract |
| Executiv:CEO | Read Financing plan |
| Corpus&Science:Curator | Read/Write Exhibition concept |
| Communication:Communication | Read/Write Exhibition concept |

#### Reduction of Human Errors in the EAA Design

The case study results for question I show that the identification of ACR breaches in the EAA has a high accuracy. AcsALign successfully identified all injected mistakes for each mistake type according to the scheme explained in Section 3.2.4. The case study results show that AcsALign enables the enterprise architect to identify and correct the mistakes in an early design phase. This leads to the conclusion that AcsALign identifies logical and design mistakes of the aforementioned mistake types. By doing so, the complex task of designing the EAA becomes less error-prone with regard to ACRs.

#### Better Support during Error Resolution

Results for question I and II demonstrate that AcsALign identifies ACR breaches in the EAA and generates an ACR mapping model with high accuracy. The generated ACR mapping model is built correctly for all 83 ACRs of the business processes. The amount of ACRs in the ACR mapping model is higher than the amount of access permissions in the role model, as the role model comprises only unique permissions. It means that two or more ACRs may lead to the same access permission. This is essential as for each ACR a potential ACR breach may exist in the EAA. Thus, it is inevitable that the ACR mappings are generated completely in order for each ACR breach to be traceable to its violated access permission and business process. Results for question II show that the ACR mappings are identified and stored correctly and completely.

Furthermore, the ACR mapping model establishes a traceability between identified ACR breaches and violated parts of the EAA, violated access permission and violated ACRs of business processes. This is an automatically generated documentation of design decisions which enables the enterprise architect to understand which service calls violate which ACRs, which otherwise would not be easy. This supports the enterprise architect during the error resolution by providing context information about the violated access permissions and the violated parts of the business processes. For the business level, the ACR mapping model helps to better understand how and why their business level ACRs reflect in the EAA, as the EAA is not part of their expertise. Moreover, the automated documentation of design decisions helps to remember the design decisions over time and in upcoming evolution scenarios. Consequently, mutual dependencies between business processes, RBAC and EAA can be understood better due to the traceability provided by the ACR mapping model.

**Aligned Adaptation in Evolution Scenarios**

The case study results for question III indicate that AcsALign can identify ACR breaches in evolution scenarios of business processes and EAA without any further extensions or modifications of the input models. Modifications made to reflect the evolution scenario itself are taken as a baseline, as they have to be done anyway. Furthermore, results for question III indicate that no additional manual effort to extend or modify the input models is required in order for AcsALign to start the analysis for ACR breaches after the evolution scenario. The results for question IV indicate that all transformation steps of AcsALign are automatic and do not require any additional effort. In conclusion, both questions show that AcsALign can be utilized without additional effort during evolution scenarios to identify ACR breaches automatically and thus, align the EAA with business process ACRs. Identified ACR breaches are then resolved during design time. The results for question I undermine the significance of this goal with a high accuracy for the identification of ACR breaches in EAAs.

**Reduction of Human Errors during the Role Model Engineering**

The results for question V show that PAcsTract successfully extracts the 44 unique access permissions. All extracted access permissions are correct. The extraction is done automatically and without human interventions, like the extension or modification of input models. Thus, the vast amount of business processes is analyzed on behalf of the security experts for the purpose of generating access permissions for the role model. The automatic generation of access permissions, which reflect business level ACRs, directly influences the amount of human errors during the role engineering process [101]. PAcsTract relieves security experts of manually analyzing business processes to extract business level ACRs for the role model. As otherwise the manual analysis of the vast amount of complex and interrelated business processes would be error-prone (explained in Chapter 1), the support and automation of this manual step makes the role engineering process less error-prone.

### 5.2.4 Threats to Validity

Runeson et al. stated in [192] that four aspects of validity need to be discussed during case study research. Thus, the following sections discuss internal validity, external validity, construct validity and reliability.

#### 5.2.4.1 Internal Validity

Internal validity refers to the degree in which the claim about the cause of a case study is reliable and not influenced by unexpected factors.

I expect the input models, the algorithm of AcsALign, the result classifications and the injected mistakes to influence the evaluation results. The factor that is analyzed in this case study is the algorithm of AcsALign. Regarding the input models I relied on a real-world case study of a national art gallery. It provides appropriate models for ACR analysis, which cover all parts of the algorithm. The business process and EAA characteristics presented in Table 5.4 underpin the appropriateness of the case study size. Business processes span a comprehensible set that contains interactions between actors, data type definitions and data usage descriptions. In Section 5.2.2 I elaborated on the appropriateness of the case study for examining ACRs and ACR breaches. It is a real-world scenario of an EAA evolution due to digitalization. The set of business processes encompasses data flows of confidential information as financial budgets, insurance values and customer/client data. Hence, the art gallery has to comply with diverse laws, for example, the GDPR and financial regulation, which impose ACRs. The EAA contains usual data processing patterns for information systems, including delegation, merging, reading from and writing to databases. There is a variety of confidential data flows of the previously mentioned confidential information that is spanned over a complex EAA that interconnects various systems. Hence, designing a secure and correct EAA is challenging and makes the case study suitable for ACR analysis. By choosing models of a real-world case study I avoided creating a case study that is tailored to the approach. With regard to the injected mistakes, I categorized all possible mistake types that influence ACRs in Section 5.2.1.1 and injected one mistake for each mistake type. More mistakes of the same mistake type would be handled in the same way and yield the same results. Thus, the injected mistakes are sufficient to evaluate accuracy. Regarding the result classifications a classification scheme is provided for every metric and explained in Section 5.2.1. If possible, established metrics are used for measurement. All reference lists are made separately by two postgraduates. Therefore, they manually analyzed the business processes. Afterwards, the versions have been compared to avoid mistakes.

#### 5.2.4.2 External Validity

External validity refers to the degree to which the conclusions of the case study can be generalized to other situations and environments.

According to Runeson et al. [192, p. 71] results of case studies cannot be generalized in a universal way when no statistically relevant sample has been drawn. This is a general problem in case study research. In the future work further case studies can be conducted in which the approach is applied on different cases. Nevertheless, the results of this case study can be generalized to cases with similar characteristics. The most relevant characteristic is certainly the input model languages PCM and IntBIIS_LP. This makes the results meaningful for a broad amount of other cases modeled in these languages. Furthermore, in Section 1.4 I argued on the similarity between IntBIIS_LP and BPMN and mentioned the similarity of modeling EAAs in PCM and UML. Conceptually, IntBIIS_LP is based on the BPMN standard and introduces a minimal required set of BPMN 2.0 elements. EAAs modeled in PCM are fundamentally not very different from UML component models. The concrete syntax of PCM is based on the syntax of UML [189]. Section 3.1 formalizes the algorithm of AcsALign in an uniformly way and states relevant assumptions so that it can be applied to similar modeling languages such as UML and BPMN. This makes the results also meaningful for cases modeled in other modeling languages. On the appropriateness of the case study regarding ACR research I elaborated already in the previous section and in Section 5.2.2.

#### 5.2.4.3 Construct Validity

Construct validity refers to the extent the taken measures represent the matter of research.

If possible, established metrics like precision and recall for accuracy are used. Furthermore, a reasonable classification scheme is provided and explained for each metric in Section 5.2.1. Besides, it explains for each goal how the research questions and metrics are derived. Finally, the whole case study evaluation is structured according to the GQM method [181].

#### 5.2.4.4 Reliability

Reliability refers to the degree in which the conclusions of the case study depend on the conducted researchers.

For the evaluation, the following steps have been conducted: creating the input models, running the analysis and classifying results. Input models are not created by the author but provided from a real-world case study of a national art gallery. The steps of the algorithm are explained in Section 3.2.4.2. It is fully automated and does not require any human intervention during the identification of ACR breaches. Hence, I could not influence the results during the first two steps. For the last step, I explained in Section 5.1.1 how research questions, metrics and classification schemes are derived. For the measurement established metrics for accuracy are used, which provide reasonable evidence and reduce the scope for interpretations. Consequently, the study design hardly allows another interpretation that may lead to a different conclusion.

### 5.2.5 Summary

In this case study, AcsALign (explained in Section 3.2.1.3 and Section 3.2.4.2) is validated on a real-world case study of a national art gallery. The validation is structured according to the GQM method [181] where goals represent validation objectives that are desired to be achieved. To research these goals, they are subdivided into research questions which are validated using metrics. Four desired goals, illustrated in Figure 5.5, were analyzed in this case study: the reduction of human errors in the designing of the EAA, the improved support during error resolution in the EAA, the adaptation of the EAA during evolution scenarios and the reduction of human errors in the engineering of the role model. Therefore, AcsALign and as such PAcsTract were applied to ten business processes from the national art gallery. Five questions were answered by measuring nine metrics. The results of the case study show that the accuracy of generated access permissions by PAcsTract is high. PAcsTract automates the analysis of the vast amount of business processes to extract a role model with business permissions. In this case study access permissions of PAcsTract are used as input for AcsALign. The identification of ACR breaches by AcsALign has a high accuracy. In addition, results show that AcsALign does not require any extensions or modifications of input models after an evolution scenario in order to conduct the architectural analysis to identify ACR breaches. The analysis itself imposes no additional effort during evolution scenarios. This allows to utilizes AcsALign during evolution scenarios of business processes and EAAs to identify ACR breaches. The results show that AcsALign provides support during the resolution of ACR breaches by providing an ACR mapping model which allows to trace ACR breaches to violated access permissions and violated activities of a business process. Consequently, utilizing AcsALign aligns the EAA with ACRs from business processes and human errors are reduced at design time.

Furthermore, results show that the accuracy of the generated ACR mapping model is high, allowing to trace ACR breaches to the violated access permissions and violated activities of a business process. This model automatically documents design decisions allowing the business and IT level to better understand mutual dependencies between business processes, access control and EAA. This becomes especially useful during upcoming evolution scenarios, when design decisions become forgotten and know-how may be lost due to an employee or responsibility shift.

# 6 Related Work

The following sections discuss the related work concerning the presented contributions of this thesis. There are three major research areas that are closely related to the work of this thesis. Section 6.1 discusses IT security and privacy extensions for business process languages and architecture languages as well as transformation approaches for IT security and privacy attributes based on these language extensions. Section 6.2 surveys related approaches regarding RBAC. Therefore, the contributions of this thesis are contrasted to existing role engineering, role mining and hybrid approaches. Role mining approaches that optimize the role model hierarchy and the role of the approaches of this thesis in the context of other access control concepts that are complementary to RBAC are also discussed. Furthermore, Section 6.3 elaborates on the relation of enterprise architecture management to the approaches of this thesis. Afterwards, differences between the approach of this thesis and security analysis approaches on de facto standard IT architectures are described. In the following sections, I will focus on security and privacy approaches that are related to access control as this thesis proposes an approach to align ACRs, stemming from IT security or privacy requirements, across models of business and IT.

## 6.1 Security Extensions for Business Process and Architecture Languages

This section discusses related approaches from the research area of security and privacy extensions for business process and architecture languages and corresponding transformation approaches. For this matter, this thesis focuses on security and privacy extensions that are related to access control as the approaches proposed in this thesis aim to align ACRs. First, Section 6.1.1 surveys security and privacy extensions for business processes. Second, Section 6.1.2 surveys security and privacy extensions for IT architectures. Finally, Section 6.1.3 discusses approaches that transform IT security and privacy attributes on language extensions from the previously introduced sections. Nonetheless, the approaches of this thesis intentionally focus on de facto standard modeling languages (as explained in Chapter 1) and thus, exclude language extensions for security and privacy attributes as input models. Many approaches extend plain modeling languages, for example, BPMN and UML to model additional security and privacy attributes. However, such approaches are rarely used and not that common as the de facto standard modeling languages. The required additional effort for organizations to explicitly model security or privacy attributes in non-security models as business processes and architectures is rather high.

Furthermore, this effort has to be done periodically, for example, during each evolution scenario and thus, costs expensive resources. In order to impose least possible effort for organizations to utilize the approaches of this thesis, this thesis focuses on information that most organizations model anyway. On the one hand, this keeps the barrier to utilize the approaches low and on the other hand, the approaches can be better utilized during evolution scenarios in order to align the models of business and IT. Hence, compared to the approaches presented in this section a major difference is that the that the approaches of this thesis focus on aligning models of business and IT by analyzing implicitly modeled security information in de facto standard models that most organization model anyway.

In a literature review [24] I have systematically analyzed various approaches that extend business process languages and architecture languages to express security and privacy attributes. One scientific finding was that security and privacy are treated differently between business processes and architectures. Current security and privacy approaches do not allow to model all aspects of security and privacy in both, business process and architecture models. Consequently, there is a need to model security and privacy attributes coherent across the models of business and IT.

### 6.1.1 Business Process Security Extensions

IT security and privacy extensions for business processes allow to enrich plain business process models with various aspects of IT security and privacy. The following paragraphs discuss the related work in this area with the focus on access control and confidentiality and contrasts the discussed extensions from the approaches presented in this thesis.

The following approaches describe different BPMN extensions to enrich business processes with security and privacy requirements. The work of Rodriguez et al. [190] proposes a BPMN extension to incorporate security and privacy requirements into business process models. Therefore, they enable business analysts to model, for example, non-repudiation, integrity, privacy, access control and security permissions and attach them to different BPMN elements as pools, lanes, activities and data objects. The work of Brucker et al. [49] introduces the BPMN extension SecureBPMN that allows to model various security and privacy requirements, for example, access control, separation of duty, binding of duty and trust in business processes. Afterwards, the explicitly modeled security requirements can be transformed into XACML rules for policy enforcement points. Salnitri et al. [195] presents the BPMN extension SecBPMN that allows to express security requirements for confidentiality, integrity and availability. A query language SecBPMN-Q allows to specify security policies. It allows to verify whether a given set of SecBPMN processes complies with a SecBPMN-Q security policy. This analysis is realized as a path analysis in the SecBPMN processes. Mülle et al. [169] propose a way to model authorization, authentication, audit, confidentiality and integrity requirements without extending the BPMN metamodel. However, they use a well-defined syntax similar to XML to annotate requirements as BPMN annotations. Hence, special knowledge for the concrete syntax of the approach is required. The work of Rekik et al. [188] introduces the BPMN extension

BPMN-Sec that aims on modeling secure business processes for cloud applications. Therefore, they extend BPMN with elements to express security and privacy requirements, for example, access control, privacy, availability and non-repudiation. In [135, 136] the authors Klarl et al. present IdM-BPMN a BPMN extension to model access control requirements in the context of identity management. Therefore, lanes and pools are extended with new elements to model roles and role hierarchies, and activities and groups are extended to model access control policies. They propose a model-driven development process to define policies for service-oriented architectures during the business process design. The work of Sang et al. [196] focuses on the healthcare sector. They introduce several new events, for example, authentication, authorization, access control and secure communication to express security requirements in the business processes. This enables to model secure business processes. Wolter et al. [234] describe a BPMN extension that focuses on modeling authorization in business processes. Therefore, they distinguish between manual and IT supported activities. They introduce new elements to express confidentiality levels similar to the Bell-La Padula model in lanes and data objects and clearance levels and trusted subjects in lanes. Their approach allows model checking by transforming the extended BPMN models into colored Petri nets. The aforementioned approaches as well as similar approaches [242, 208, 215, 58, 143, 68, 193] all have in common that they introduce BPMN extensions to express different sets of security and privacy requirements as part of the BPMN elements. However, each of these approaches requires additional effort to model the requirements in the extended BPMN models. Furthermore, business experts require security and privacy knowledge, which is typically not their expertise, to be able to model the security and privacy requirements. If considering that business processes can easily grow into several hundreds and that all of them evolve constantly, the amount of resources an organization has to invest in order to model the extended security and privacy requirements in each business process and to adapt them during the evolution is high and cost-intensive. In contrast, the approaches of this thesis aim to solve this problem by building upon de facto standard models, so that organizations can exploit their existing models. Moreover, the approaches of this thesis are automatic, i.e. that no additional effort is imposed on organizations to utilize them. Another difference is that the related approaches do not focus on business-IT alignment, whereas the approaches of this thesis aim to align IT models with business processes.

In [156] from Menzel et al. a BPMN extension is proposed to model security goals, trust and threat relationships between participants of business processes and so-called security groups. This is realized, for example, by defining security intentions and security ratings to a set of pools and activities. Based on the extended BPMN models, security configurations can be derived for service-based systems. In contrast to the approaches of this thesis, the work of [156] explicitly requires additional effort to model trust, confidentiality and integrity requirements as part of the business processes. They focus on providing means to express general security goals in business processes and to transform them into security policies for Apache Rampart configurations.

Varela-Vaca et al. [226] propose a BPMN extension that allows excessive modeling of risk characteristics of business processes, for example, asset value, cost, frequency, vulnerability, threat, etc. These characteristics can be converted into a constraint satisfaction problem for a constraint solver tool that verifies whether the business processes satisfy the risk characteristics. A similar approach is presented by Altuhhov et al. [26]. The authors establish a risk analysis in business processes by incorporating the Information System Security Risk Management Concept (ISSRM) into BPMN. Therefore, they relate elements of BPMN and ISSRM with each other, for example, a BPMN data object is an ISSRM IS-asset. Then assets, risks and measures can be expressed in BPMN, which allows to perform a risk analysis in the business processes. The aforementioned approaches as well as similar approaches from Monokova et al. and Meland et al. [161, 155] have a different goal than the approaches presented in this thesis. They want to enable business experts to make risk analyses on business process. In contrast, the approaches of this thesis focus on aligning business processes and IT models with regard to ACRs.

Another work of Altuhhov et al. [27] introduces a BPMN extension to model high-level business security objectives including variants of security choices as part of BPMN models. This extension is designed for business analysts in order to improve their business decisions with regard to risk and IT security.

Wolter et al. [236, 235, 237] propose different BPMN extensions to model separation of duty and binding of duty in business processes. They use visual representations to distinguish between manual and automatic tasks as well as to annotate separation of duty and binding of duty constraints on activities. In fact, separation of duty and binding of duty are constraints that cannot be expressed in plain BPMN. The approaches of this thesis could be easily extended to consume the additionally modeled information about separation of duty and binding of duty. However, this thesis focuses to align business and IT models without the burden of modeling additional information in extensions, as explained in detail in the beginning of the Section 6.1.

Another business process language is Petri nets. Many approaches exist that enrich Petri nets with security and privacy attributes. The approach of Huang et al. [113] uses colored Petri nets including Petri net properties as completeness and consistency to verify whether GDPR policies are fulfilled by the business process models. This approach has a different aim than the approaches in this thesis. They focus on aligning business processes with requirements stemming from laws. In this thesis, I assume that the business processes are modeled correctly and compliantly to laws, as the goal of this thesis is to align business processes and IT models with regard to ACRs.

Approaches similar to [138, 14, 13, 147, 31, 216, 144, 30, 245] focus on analyzing various aspects of confidentiality and access control in extended business processes modeled in Petri nets. The works of Accorsi et al. [14, 13] propose an approach to analyze the propagated information in business processes. This can be done statically or during process execution with the use of previously defined confidentiality levels. The work of Li et al. [147] proposes a similar approach based on a colored Petri net extension. It allows to detect confidentiality issues within the information flow of business processes. Confidentiality levels are modeled as additional attributes of Petri net tokens. The work of Knorr [138] also uses confidentiality levels but allows to verify multilevel security policies. Therefore, control and information flow has to be modeled as different arcs. Atluri et al. [31] propose a different approach to verify multilevel security policies based on confidentiality levels.

In [30] they extend their previous work to allow the additional expression of concepts as separation of duty and RBAC in colored, timed Petri nets. Similar approaches are proposed in [216, 144]. They also focus on confidentiality of information flow in business processes by extending Petri nets with RBAC policies or other confidentiality policies. The approach from Zhang et al. [245] focuses on modeling the Chinese wall policy in colored Petri nets in order to facilitate a policy compliance analysis on the business level. However, all of the aforementioned approaches require to model additional security information as part of Petri net extensions in order to facilitate the proposed analysis. Furthermore, they only focus on an analysis on the business level. In contrast, the approaches of this thesis transfer implicitly modeled business knowledge about ACRs to the IT level in order to generate an initial role model and align the EAA. Furthermore, this is done without the burden of further modeling effort in any of the business and IT models. Another major difference is the business process language. BPMN has achieved widespread adoption as stated before. It is the de facto standard modeling language for business processes [23, 213] and thus, is used as the business process modeling language for the input models consumed by the approaches of this thesis (explained in more detail in Section 3.2.1).

Approaches from Akbarzadeh et al., Bouroulet et al. and Crazzolara et al. [16, 45, 63] aim for assessing and analyzing security protocols from the perspective of an attacker. This requires either to model an attacker as well as attacks, threats and vulnerabilities or to make an analysis for flaws in the information flow of the given model. Such approaches have a different purpose than the approaches of this thesis, as they focus on security protocols and the attacker view. They do not align models across business and IT but rather focus on identifying existing flaws in security protocols.

To sum up, there are different approaches that extend business processes to enable business experts or security experts to express security attributes as part of the business processes. However, such approaches impose additional effort on organizations to utilize them. Hence, they are not widely used compared to the de facto standard modeling language BPMN. Furthermore, the utilization of such approaches becomes even more complex and resource-intensive if considering that business processes and information systems evolve over time. In both cases business processes must be adapted and thus, additional modelling effort is required every time to adapt the information in the business process extensions. This makes such approaches less suitable for the business-IT alignment during evolution scenarios. In contrast, the approaches of this thesis are build upon de facto standard modeling languages to overcome this problem.

### 6.1.2 Architecture Security Extensions

In the area of IT architecture many security and privacy extensions are proposed for the different UML diagrams. They allow to enrich the UML elements with various IT security and privacy aspects. The following paragraphs discuss the related work in this area with focus on access control and confidentiality and highlight the differences to the approaches of this thesis.

Jürjens [125] proposes an extension called UMLsec to express security requirements within a variety of UML diagrams. They aim for organizations that want to model security-critical systems. Their extension is intended for non-security employees so that they can express their security needs easily. For example, the extension enables software engineers to express basic security requirements including security concepts, security primitives, security management and threat scenarios. Among other things, this allows to model the confidentiality of information and information flows. There is a list of stereotypes that can be attached to various UML elements as classes, components and associations. Constraints can be used to refine specific stereotype elements. An analysis proposed in [124] allows to verify if a given specification fulfills the constraints associated with the stereotypes and by this, to identify vulnerabilities. The difference to the approaches of this thesis is that UMLsec requires additional specification of fine-grained security related information on the architecture level. In contrast, the approaches of this thesis exploit de facto standard models that organizations model anyway. Furthermore, the approaches presented in this thesis go beyond an analysis on the IT level by alignment of the architecture to business level ACRs.

Heldal et al. [106] propose an UML extension called UMLsProfile to incorporate decentralized labels into UML class diagrams. This enables to model confidentiality of data flows at design time in a fine-grained manner. The concept of declassification is included. The work of Goudalo et al. [93] also proposes an UML extension to model confidentiality and confidential information flow in architectures. Their extension uses confidentiality and security levels for systems, objects and resources to specify access control policies. Klarl et al. [136] extend UML activity diagrams to express access control policies based on identity management concepts. Therefore, they introduce means to model roles, role relationships, permissions and assertion attributes. The work of Fernandez-Medina et al. [79] introduces an UML extension called SECDW to model confidentiality in UML class diagrams that are specifically tailored for the domain of data warehouses. The extension allows to specify security classes for information and users. Through the use of tuples composed of security classifications, set of user compartments and user roles it is later possible to specify access constraints. An extension called SECDW+ [220] extends the previous work and introduces the ability to model leakage of confidential information (e.g., health information or company turnover) based on conflicts of interest. Conflict of interest specifies a problem in which isolated access to several data sets is secure but combined access to these data sets results in an information leakage. On the one hand, SECDW+ allows to specify conflicts of interest among multidimensional concepts of data warehouses, for example, dimensions, hierarchies and attributes and on the other hand, it allows to specify data dependent conflicts of interest, for example, the address of a patient and his illness type, if it is equal to cancer. In [148] of Lodderstedt et al. an UML extension called SecureUML is proposed in order to model RBAC requirements in IT architectures. SecureUML is a policy definition language that extends UML but in theory is independent of it. The policy definition language is bind to UML via UML stereotypes. While the approach provides means to extract the specified access control policies for RBAC systems, it lacks the ability to check if the architecture complies with the policies. This limitation is overcome in the work of Basin et al. [38]. The authors extend SecureUML and use object constraint language (OCL) expressions to analyze if the modeled security properties are fulfilled by the UML class models or model instances. Another approach to model access control requirements in architecture models is proposed by Koch et al. [139]. The authors present an own extension that builds upon UML class and UML object diagrams. OCL expressions are used to verify the consistency of modeled access control policies. All of the aforementioned approaches focus on modeling confidentiality aspects through specification of additional information in IT architectures in order to verify if the architecture complies with the confidentiality aspects. In contrast, the approaches of this thesis align the architecture and RBAC with ACRs stemming from business processes without the overhead of modeling additional information on the business or on the IT side.

In [73, 74] Emig et al. propose an approach to model access control requirements for the context of identity management with the focus on service-oriented architectures. They develop a metamodel for access control that combines aspects of RBAC and attribute-based access control (ABAC) and is specific for the context of identity management. Based on this metamodel access control policies can be expressed in a platform independent domain language called WSACML. This additional language should decouple the specification of access control policies from the architectural design. Later, WSACML specifications can be transformed to a domain specific policy language of choice. The authors focus on providing a platform independent language for modeling access control policies in the context of identity management. An approach with a similar idea is presented by Alam et al. in [17] and refined in [18, 19]. The authors propose a framework called SECTET to secure service-oriented architectures by modeling RBAC policies with OCL expressions. XACML policies can be derived from the extended architecture models. In contrast to the approaches of this thesis, the aforementioned approaches focus only on the architecture level and do not aim for a business-IT alignment of ACRs. Furthermore, these approaches rely on the explicit modeling of additional information during the architectural design phase. The approaches of this thesis specifically avoid this to keep the modeling overhead for organizations at a minimum. This also allows an automatic business-IT alignment during evolution scenarios.

The work of Alghathbar et al. [22] proposes an UML extension that allows to analyze access control policies in UML use case diagrams. Their primary goal is not necessarily to model access control policies in the architecture but to analyze them for consistency and completeness in an early design phase. The authors extend their UML extension in [21, 20] to express RBAC policies and authorization requirements in UML use case diagrams. A similar approach is described by Fernandez et al. [80]. The authors use extended UML use case diagrams, which are enriched with access control policies, in order to extract access permissions and roles for access control systems. Bertolino et al. [41] present a similar methodology with tool support. Their approach enables to model access control policies in extended UML class diagrams to verify their consistency and subsequently, to generate code for the enforcement of the access control policies of the implementation phase. All three approaches aim for a different purpose than the approaches presented in this thesis as they do not focus on closing the gap between the business level and the IT level. They focus on specifying access control related information explicitly at design time rather than extracting them from business models. This imposes additional effort in utilizing the aforementioned approaches and makes them less suitable for a business-IT alignment during evolution scenarios.

The work of Sindre et al. [206] focuses on defining misuse cases within extended UML class diagrams by describing different attackers and how they misuse the systems. In the work of Gomma et al. [92] a similar approach is proposed but the authors separate functional use cases from security use cases. Therefore, an UML extension for UML use case diagrams is presented that allows to express security requirements as part of the UML use cases. The work of both is different to the approaches of this thesis, as they focus on enabling architects or other IT roles to model security attributes within architectural models. They do not aim on aligning EAAs with business processes or other business level models. Similar approaches are proposed in [86, 200, 201, 81]. They all have in common that they apply the idea of design patterns to security attributes. After introducing the so-called security patterns they propose methodologies that describe how to build secure systems with the help of the security patterns. Thereby, Kim et al. [131, 132] focus on access control, Bouaziz et al. [44] focus on security patterns that are specific for component-based modeling and Schnjakin et al. [199] focus on security patterns for the configurations of security modules in service-oriented architectures. These approaches have the same limitations as the previously mentioned two approaches. They focus on the phases from architectural design to code implementation but do not consider business models or the business process design phase. Another methodology that also only focuses on the architectural and implementation domain is introduced by Hafner et al. [100]. Instead of security pattern the authors describe a model driven security methodology that is based on a combination of a framework for requirements engineering called ProSecO and the model driven development framework called SECTET. This allows to transform security attributes from the architectural design phase level to configuration code for the implementation phase.

The work of Jutla et al. [126] proposes an extension to the UML use case diagram for representing privacy specifications as pseudonymization, anonymization and consent methods in an easy understandable way. The extension does not use the UML profile extension mechanism but a Microsoft Visio extension ribbon to offer the required elements. Privacy specifications and ACRs can be expressed in free text fields of extended UML use case diagrams. The extension introduces a so-called super container in the UML use case diagram. The super container lies between the actors and use cases connecting them through labeled lines named by the corresponding actor. Privacy controls stating a privacy control class or a privacy control obligation are inside the super container. Actors and their use cases can be connected to the whole container, to particular privacy control classes or to particular privacy control obligations. This means that all privacy controls, those of the privacy control classes or only those of the privacy control obligations have to be realized for the connected use cases. The presented extension allows to model not only any kind of privacy principles but also other security principles as confidentiality. The work of Basso et al. [39] introduces an UML extension via UML profiles to express various privacy concepts through the incorporation of privacy policies in several UML diagrams. Privacy policies are composed by one or more statements that describe the rules specified by the privacy policy.

Besides, they specify the purpose of data collection, management and prerequisites which need to be met. The introduced stereotypes allow to design privacy-aware applications through the specification of the application's privacy policy and by keeping track of the elements responsible for enforcing them. This allows to express access control for private data and also privacy principles as consent, data security and purpose limitation. Another privacy related extension is proposed by Simons [205]. The UML extension enables to define privacy restrictions in UML class diagrams. The profile is developed for the area of mobile distributed systems but it is applicable to other areas. The main idea is to bind access rights to context information by using confidentiality levels and confidentiality sources and the validity of the context information. In this UML extension, constraints are used to validate the model. This is done by imposing restrictions on the stereotypes to enforce the correct usage of the extension. The two major differences to the approaches presented in this thesis are that the previously mentioned approaches introduce additional modeling elements to cope with security and privacy related information and that they only focus on IT models rather than on the business-IT alignment.

Houmb et al. [112] propose an UML extension called SecurityAssessmentUML that extends UML sequence and UML activity diagrams to incorporate risk analysis concepts into architecture modeling. Therefore, they enable to model vulnerabilities, undesired events, attackers, threats and assets in the previously mentioned UML models. This approach aims for a different purpose than the approaches presented in this thesis. They want to enable IT roles to make risk analyses at the architecture level. In contrast, the approaches of this thesis focus on aligning business processes and IT models with regard to ACRs.

A more general modeling approach for security is proposed by Hatebur et al. [102]. The authors use UML profiling to express problem frames in UML class diagrams. Problem frames are patterns used to define problem classes by their context and characteristics. In the context of security the traditional goals confidentiality, availability and integrity can be modeled. They are expressed by stereotypes including specifications for the data to be secured, the attacker and the stakeholder of data. This allows to express any possible confidentiality requirement by using problem frames. The authors expect the main advantage of their approach in the ability to express dependability requirements without the anticipation of a solution. This separates the problem space from the solution space. The works of Mouheb et al. [165, 164] are similar with regard to the abstraction level. The authors propose an UML extension that captures security requirements and allows specifying security solutions. This is achieved by weaving security aspects into UML class, sequence, state machine and activity diagrams in an aspect-oriented manner. Thus, security concerns are separated from software functionalities. In contrast to the aforementioned approaches, the approaches of this thesis focus on the business-IT alignment of ACRs by leveraging implicitly modeled access control information in non-extended business processes.

To sum up, there are different approaches that extend architecture modeling languages in order to express security attributes as part of the architecture. However, such approaches impose additional effort on organizations during the architectural design phase and they are not widespread compared to the plain de facto standard models. Furthermore, the utilization of such approaches becomes even more complex and resource-intensive when considering that business processes and information systems evolve over time. In both cases, the architecture must be adapted and thus, every time additional modeling effort is required to adapt the architecture extension. This makes these approaches less suitable for the business-IT alignment during evolution scenarios. In contrast, the approaches of this thesis build upon de facto standard modeling languages to overcome this problem.

### 6.1.3 Transformation on Language Extensions

This section examines security analyses and transformation approaches that are based on the architecture and business process extensions from Section 6.1.1 and Section 6.1.2. While transformation means that some security attributes are transformed between two or more development phases (business processes design, architectural design, implementation), security analysis means that a security related question is analyzed in a model of the development phase. This section mostly focuses on approaches for the business process design phase and architectural design phase but other approaches are also mentioned if they are considered relevant with regard to the contributions of this thesis.

Klarl et al. [136, 135] introduce extensions for BPMN processes and UML activity diagrams to express access control policies in the context of identity management. The extended models can be transformed to access control policies. However, the transformed information is exactly the same information that has to be extended in the UML and BPMN models prior to transformation. Hence, there is no information difference between modeling the information directly in an access control policy and in the extensions for UML and BPMN. No new information is generated and the modelling effort is similar. A major difference to the approaches of this thesis is that the proposed transformation has not the goal to align business and IT models, but rather to extend business models with access control specific information.

Jürjens [125] proposes an extension called UMLsec to express security requirements within a variety of UML diagrams. The extensions enable software engineers to express basic security requirements including security concepts, security primitives, security management and threat scenarios. In [124] the author introduces an analysis that allows to verify if a given architecture specification fulfills the constraints associated to the UMLsec stereotypes. Any unfulfilled constraint indicates a potential vulnerability. A similar analysis is used by Simons [205] to verify if the architecture complies with the privacy related information that is additionally modeled in their specific UML extension. The work of Heldal et al. [106] proposes a similar analysis. The authors propose an UML extension that focuses on confidentiality and confidential information flow in UML class diagrams. Through the generation of Jif code skeletons the architect is able to verify if the architecture complies with the constraints of the extended security stereotypes. The difference to the approaches of this thesis is that the aforementioned approaches require additional specification of fine-grained security related information on the architecture level. This information has to be specified, for example, by enterprise architects and then allows to verify if a given architecture fulfills the security requirements. In contrast, the approaches of this thesis automatically extract ACRs from business processes to identify misalignments of the EAA with regard to the extracted ACRs. This achieves a business-IT alignment at design time and during evolution scenarios which is especially crucial as organizations and their models evolve constantly.

In the work of Lodderstedt et al. [148] an UML extension called SecureUML is proposed in order to model RBAC requirements in IT architectures. Based on the specified access control policies in the IT architecture RBAC or other access control policies can be derived automatically. In contrast to the approaches presented in this thesis, the derived RBAC polices have to be explicitly modeled in extended architecture models. Furthermore, their approach does not align RBAC with business ACRs stemming from business processes. A similar approach is presented by Pavlich-Mariscal et al. [175]. The authors extend several UML diagrams to express access control policies during the architectural design phase. Afterwards, specified access control policies can be transformed into code that enforces the policies at runtime. The limitations are the same as with the above-mentioned approach. Access control polices have to be modeled explicitly in architecture models and the approach does not help to align RBAC with business ACRs stemming from business processes. Another approach is presented by Fernandez-Medina et al. [79]. Again, UML diagrams are extended to model confidentiality requirements which then can be transformed into SQL queries to instantiate databases that are aligned with the confidentiality requirements. Similar approaches that focus on instantiating secure databases are proposed by Vela et al. [228, 227]. These approach have the same limitations as the related approaches described in this section. Furthermore, these approaches focus on the transformation between the architectural design phase and the implementation phase, while the approaches of this thesis focus on the transformation between the business process design phase and the architectural design phase. Further approaches that focus on a transformation from architectural design to code implementation are [73, 74, 17, 18, 19, 44, 199, 171] and vice versa [42].

Mouratidis et al. [166] propose a methodology to design security attributes by defining functional and non-functional requirements together in an extended goal-driven requirements engineering methodology. Functional and security requirements are elicited and their constraints are analyzed. Then the operational environment is taken into consideration together with the functional and non-functional requirements. Afterwards, the architecture is designed in a traditional manner by taking the previously engineered information into account. In contrast to the approaches presented in this thesis, this methodology mostly focuses on the requirements engineering phase, whereas the architectural design phase is not supported adequately with regard to security requirements. This limitation is overcome in [167]. The authors extended the previous methodology to propose a secure software development methodology where the architectural phase is supported with UMLsec models that provide means to express security requirements as part of the architecture models. When compared to the approaches of this thesis, the authors propose a mostly manual methodology on how to systematically develop secure software systems. They do not focus on business-IT alignment in an automatic manner. Furthermore, the authors rely on architecture extensions that impose additional modeling effort which makes their approach unsuitable for alignments during evolution scenarios.

Mülle et al. [169] propose a well-defined syntax similar to XML to annotate confidentiality and integrity requirements in BPMN models as part of BPMN annotations. Based on the annotated BPMN models they describe a transformation of the security requirements to security constraints for IT systems, for example, policy enforcements points and logging components. In contrast to the approaches presented in this thesis, the transformation is used for real time enforcement of constraints rather than the alignment of IT models at design time. Additionally, their research focus lies not at EAAs as it is the case with AcsALign.

The work of Altuhhova et al. [27] introduces a BPMN extension to model high-level business security objectives including variants of security choices in BPMN models. They propose a transformation in order to align high-level organizational goals with business processes. This is done by deriving security-annotated process skeletons from organizational goal models. This is an alignment only in the scope of the business level. In contrast, the approaches of this thesis align models of the business and IT level with each other.

Menzel et al. [156] propose a BPMN extension to express trust, confidentiality and integrity requirements in a general manner. Based on the extended BPMN models the authors propose a transformation to security configurations for service-based systems in Apache Rampart. This allows to derive security policies that are realized as part of web service implementations. They focus on aligning business processes with code of the implementation phase. In contrast, the approaches of this thesis focus on aligning models from the business process design phase with the intermediate layer, the architectural design phase, that lies between the business process design and implementation phase. Furthermore, the approaches presented in this thesis focus on transforming implicitly modeled security information from business processes rather than imposing additional effort to extend models with security information.

The works of Abramov et al. [10, 11] propose a systematic way to enforce ACRs from the business level in architectures and databases. Therefore, the authors propose a methodology with four phases in which different stakeholders work together to engineer artifacts as security patterns and UML diagrams. In the first phase, security officers and domain experts specify organizational security patterns. During the second phase, the conceptual model and functional models are designed based on user requirements and artifacts of the previous phase. In this phase, class diagrams including data classes, their attributes and relationships are defined. The functional model is developed in an extended UML use case diagram that allows to connect functions with data classes and function calls and to describe authorization policies. In the third phase, the artifacts of the previous phase are combined into an unified class diagram that is an extended UML class diagram that allows to express authorization rules and security patterns. Finally, in the fourth phase, SQL statements are derived from the unified model in order to establish database schemas and authorization constraints for databases. The differences to the approaches of this thesis are that the authors focus on architecture diagrams rather than on business process models and ACRs and, also security patterns need to be modeled explicitly by security experts in extended UML class diagrams and extended UML use case diagrams. The approaches of this thesis focus on supporting commonly used methodologies and de facto standard modeling languages without imposing additional overhead through extensions. Implicitly modeled ACRs are extracted to analyze the correctness of the EAA without an additional effort for security experts. The aforementioned work aims for a different goal. The authors want to build databases that are compliant with organizational policies. Their alignment process is a manual methodology with the focus of transforming the engineered models into SQL statements to design secure and compliant databases. In contrast, this thesis has a different goal. The goal is to align EAAs of the IT level and business processes of the business level with regard to ACRs.

Ramadan et al. [186] present an approach to transfer security requirements, for example, confidentiality and integrity requirements from extended business processes to extended architecture models. Therefore, they rely on secBPMN models that extend business processes with explicitly modeled security requirements as input and on UMLsec as architecture models that extend UML with explicitly modeled security requirements. The authors propose transformation rules for the explicitly modeled security requirements from secBPMN to UMLsec. Although the approach does transfer security requirements from business processes to IT architectures, it heavily depends on extended models on the business and IT side. Furthermore, the approach does not consider implicitly modeled ACRs and does not generate an initial role model for RBAC.

The work of Brucker et al. [49] introduces the BPMN extension SecureBPMN that allows to model various security and privacy requirements as access control, separation of duty, binding of duty and trust in business processes. In [48] the authors propose a method to verify the code implementation for compliance with the modeled security requirements in SecureBPMN. Their general goal is similar to the goal in this thesis as they transfer business knowledge about security attributes to the IT level. The major differences are that they focus on the code implementation as the IT artifact while the approaches of this thesis focus on EAAs and that they rely on extended business process models in which security requirements are modeled explicitly.

The work of Rodriguez et al. [191] is based on the BPMN extension of [190]. The authors introduce transformation rules to generate implementation specifications based on the extended security requirement annotations of the extended BPMN models. The transformations are used to semi-automatically generate initial UML class and use case diagrams. However, the generated initial UML diagrams have to be refined and completed and the transformation rules heavily depend on the additionally modeled security requirement annotations of the BPMN extension.

Ramadan et al. [185] present another approach that aligns architecture models with business processes with regard to security requirements. Therefore, they transform security policies from secBPMN2 models to architecture models in UMLsec. While the idea is similar, the realization is based on different assumptions and goals. They focus on non-standard models that introduce security specific elements which have to be incorporated at design time by experts. However, such extended models are not widespread and impose additional modeling effort on organizations. While the approaches of this thesis also focus on business-IT alignment, they exploit implicitly modeled security information in de facto standard modeling languages and are tailored to impose least possible effort during their utilization.

To sum up, there are several approaches that facilitate security analyses in architecture and business models (and code implementation) and also transformations between them. However, all of the presented approaches rely on extended models on the business side, IT side or on both sides. Hence, security attributes have to be modeled explicitly by experts in order to complete the standard models. This imposes additional effort on organizations during the business process and architectural design in order to utilize these approaches. None of the extensions are conventionally accepted as the de facto standard models. Furthermore, the complexity of utilizing such approaches rises due to the constant evolution of business processes and information systems. Such evolution scenarios require repeated adaptation of the modeling extension in order for the approaches to work. This is a manual and resource-intensive process resulting in a heavy burden for organizations to utilize the approaches. Hence, these approaches are less suitable for the business-IT alignment in evolution scenarios. In contrast, the approaches of this thesis build upon de facto standard modeling languages and require no additional modeling effort. One of their primary aims is to overcome the problem of additional modeling in order to facilitate business-IT alignment for ACRs.

## 6.2 Access Control

This section assesses similarities and differences between the RBAC role model extraction approaches of this thesis and related approaches in this research area. Furthermore, the role of the approaches of this thesis is discussed in the context of other access control concepts that are complementary to RBAC. Section 6.2.1 makes a comparison to other role engineering approaches and afterwards, in Section 6.2.2 to role mining approaches. Section 6.2.3 introduces approaches that optimize the role model hierarchy. These approaches can be seen as complementary to the approaches of this thesis. Section 6.2.4 makes a comparison with hybrid approaches that combine properties of role engineering and role mining. Finally, Section 6.2.5 elaborates on the utilization of BAcsTract and PAcsTract in the context of other access control concepts that are complementary to RBAC (hybrid RBAC concepts), i.e., concepts that either combine certain parts of RBAC with other access control concepts or concepts that modify RBAC in order to introduce new functionalities for access control.

### 6.2.1 Role Engineering Approaches

This section examines related work in the area of role engineering. Role engineering approaches are carried out top-down. Experts decompose business artifacts, for example, business processes or IT artifacts mostly manually into permissions that are required to carry out specific tasks [66]. Afterwards, these permissions are grouped into roles and a role hierarchy. Roles elicited with role engineering are business roles reflecting the hierarchy of an organization. Section 2.2.2 has introduced role engineering in more detail. In the following paragraphs, role engineering approaches are subdivided into approaches that operate on business artifacts and approaches that operate on IT artifacts. I will begin with the approaches that operate on business artifacts.

Early role engineering approaches like the work of Coyne et al. [72] describe the role engineering process as a manual methodology from a high-level point of view and thus, lack many practical details. It describes a decomposition from business processes and functional structures to system-specific access control requirements. A similar engineering approach is described by Chandramouli [57] and Jaeger et al. [119]. The first approach focuses on the healthcare sector. The authors propose specific roles and a hierarchy tailored for healthcare systems that are based on their practical experience in the healthcare sector. However, permissions still have to be defined by security experts. The second approach focuses on protection domains for remote programs. The authors propose rules and transformations on how to derive and adjust permissions based on specific protection domains and protection domain changes. Thomson et al. [219] present a similar engineering approach. The methodology called Napoleon has seven phases in which application developers and security experts manually engineer access control related information, for example, objects, application constraints, key chains and enterprise constraints. Afterwards, a tool supports the transformation of the engineered security information into access control policies for RBAC. Another role engineering approach is proposed by Roeckle et al. [98]. The authors explain a methodology for experts where they manually engineer roles and permissions to form a RBAC role model based on business artifacts. First, they elaborate that business artifacts, e.g., job descriptions and organigrams lack sufficient information to build an appropriate role model. Then they show their methodology that is based on information from business processes and expert knowledge to fill the previous gaps. However, the amount of business processes, which needs to be analyzed in the aforementioned approaches, grows increasingly with the organizational size. As all steps are carried out manually, the approaches are time-consuming and resource-intensive. The rigorous amount of human interventions required to analyze all business processes makes the proposed approaches error-prone [35]. Hence, decisions about roles and permissions for the role model cannot be made reliably, since important business process information may be missed [9]. The approaches presented in this thesis are also top-down role engineering approaches and focus on business processes, as the business level artifacts. The major difference is that the approaches of this thesis automate the analysis of business processes and thus, tackle the above-mentioned problems of scalability and human errors. Another difference is that they generate an ACR mapping model that interconnects elements of business processes and RBAC, allowing to understand mutual dependencies and providing an automated documentation of design decisions. Furthermore, the approaches presented in this thesis do not only focus on RBAC role models, as IT level artifacts, but also on EAAs. They provide means to align RBAC and EAAs with business process ACRs. This becomes especially useful during evolution scenarios.

The work of Gustaf et al. [172] proposes a role engineering approach in which ACRs are defined along a bottom-up business development process. The authors propose a methodology with seven phases. They begin by modeling scenarios of the system usage that describe ACRs in form of action-event sequences. Then permissions are derived from the scenarios and constraints, as separation of duty, are identified and made explicit for each scenario. During the next steps, scenarios are aggregated into tasks and work-profiles which represent business processes, a role hierarchy including roles, is derived and finally, the role model is defined. Compared to the approaches of this thesis, this role engineering approach is a manual methodology in which experts systematically engineer permissions, roles and a hierarchy along the business development process. Thus, this approach suffers from human errors, is resource-intensive and lacks scalability as described in the previous paragraph in more detail. The approach is unsuitable to quickly align IT architectures, RBAC and business processes during evolution scenarios.

Epstein et al. [77] describe an approach that decomposes roles into permissions with several additional layers from which roles, permissions and a role hierarchy can be derived manually. Therefore, they introduce three additional layers (jobs, workpatterns, tasks) between roles and permissions to divide them into smaller and better manageable parts. Roles are responsible for one or more jobs in which they have to do workpatterns. Each workpattern consists of atomic tasks in order to complete the workpattern. Tasks may require permissions to accomplish the task. The work of [172] suggests a methodology on how to manually engineer these layers in an organization. Compared to the approaches of this thesis, the engineering approach is resource-intensive, lacks scalability and suffers from human errors. Furthermore, their approach is not able to facilitate a business-IT alignment during evolution scenarios while the approaches of this thesis specifically tackle the formerly mentioned problems. In the work of this thesis a decomposition between roles and permissions is introduced by the concept of the ACR mapping model. However, this thesis proposes different layers with a different purpose. Their purpose is to interconnect RBAC with elements of business processes and EAAs.

Fuchs et al. [84] describe a structured process-oriented methodology to engineer ACRs and implement access control policies with regard to identity management in organizations. They propose three major steps in which different stakeholders from business and IT work together. All three steps are done with a high-level identity management strategy in place. The first step breaks down the high-level identity management strategy into smaller projects, specifies a detailed plan and involves all relevant stakeholders. In the second step results from step one are refined and transformed into business processes and security policies reflecting the requirements of the smaller projects. During the third step business processes and security policies are implemented and the technical implementation is tested and re-engineered if necessary. The authors present a high-level methodology for a manual, structured process to establish identity management including ACRs and access control policies in organizations. Security policies have to be specified along with the business processes. All sources can be taken into account for the elicitation of ACRs and the specification of security policies. During the final step the previously engineered artifacts, business processes and security policies are implemented. The approach has several differences from the approaches of this thesis: a) it is manual and resource-intensive, b) requirements may stem from all possible sources, c) ACRs and security policies have to be defined explicitly, d) the authors do not describe how to test implemented security policies and e) the goal of the authors is to implement business processes along with security policies. In contrast, the approaches of this thesis propose an automated way to extract ACRs from business processes and align the architectural design along with RBAC to business process ACRs. One of the main goals is to overcome the gap of manual engineering, that is proposed by the aforementioned approaches. The automation of the approaches of this thesis enables them to be used throughout evolution scenarios to align business processes, the EAA and RBAC with respect to ACRs. This is achieved without imposing additional effort on organizations.

The work of Epstein et al. [78] presents a role engineering approach that utilizes UML diagrams to express different parts of RBAC. In a case study in the healthcare sector the authors show how UML can be used throughout the architectural phase to document some of the RBAC related policies in order to finally build a role model. However, some aspects are not addressed in their engineering approach, for example, constraints on ACRs and the role hierarchy. A similar approach is proposed by Fernandez et al. [80]. The authors suggest to derive a role model from extended UML use case diagrams. Therefore, authorizations are derived from use case preconditions, roles from actors and the residual access control information from the extended stereotypes that have to be additionally specified in the UML use case diagram. However, this engineering approach also does not consider constraints on ACRs and the role hierarchy. Another approach from Crook et al. [182] focuses on requirements engineering artifacts. The authors propose a methodology in which organizational documents that contain ACRs are analyzed to engineer access control policies based on roles. They analyzed various organizational documents to identify those documents which contain access restrictions and authorizations procedures. Then they propose an analytical methodology for analyzing the previously selected organizational documents in order to manually specify roles and corresponding access control policies from the role's perspective. In contrast to the approaches of this thesis, the aforementioned engineering approaches are not able to bridge the gap between the business level and IT level in order to align business processes, RBAC and EAAs in terms of ACRs. The reason is that they do not take the main business artifact, namely business processes, into account and they do not align RBAC with ACRs stemming from business processes.

An automatic role engineering approach is proposed by Narouei et al. [152]. The major difference is that they do not focus on business processes, as business level artifacts. Instead, they use natural language processing to extract roles and permissions from high-level requirement specifications that include access control policies expressed in human-understandable language. Hence, the approach can be seen complementary to the approaches of this paper, as both approaches together could produce a more precise and complete role model. Further natural language processing approaches are proposed by Xiao et al. [239] and Slankas et al. [207]. The first approach uses shallow parsing techniques to match sentences with predefined access control patterns and the second approach uses inductive reasoning. However, compared to the previous approach both approaches have several weaknesses. For example, it is hard to get data sets with labeled data that are similar to the documents being analyzed and the approaches do not take contextual information into account [152].

To sum up, role engineering approaches can be differentiated between approaches which consume business artifacts and approaches that consume IT artifacts. Business processes are consumed often by the first group. They describe how to elicit, establish or implement ACRs into productive systems. However, these approaches only describe manual methodologies. This is one of the problems described in Section 1.2. These manual approaches often require experts, they are resource-intensive and error-prone, they do not scale and they are slow especially during the constantly required adaptations in evolution scenarios [35]. The work of this thesis tackles these problems by proposing approaches that are automatic, that consume de facto standard models in order to make them easily utilizable for organizations and that are applicable during evolution scenarios to align business processes, EAAs and RBAC with respect to ACRs without imposing additional effort. The second group consists of manual and some automatic approaches but they focus on IT artifacts rather than business processes. Thus, these approaches do not extract business ACRs and do not align RBAC with business ACRs.

### 6.2.2 Role Mining Approaches

This section examines related work in the area of role mining. Role mining approaches are carried out bottom-up, meaning that an algorithm analyzes existing permissions in an organization automatically to group them into roles and a role model [66]. Roles elicited with role mining are technical roles reflecting the underling usage of systems and services. Role mining was introduced in detail in Section 2.2.3. In the following paragraphs, role mining approaches to elicit a role model are briefly summarized. Role mining approaches that focus on optimizing the role hierarchy are discussed in the next section and hybrid approaches are discussed in the subsequent section.

A survey on role mining approaches was recently done by Mitra et al. [35] and previously by Vaidya et al. [223]. Role mining analyzes permissions of existing access control systems, which are IT level artifacts, providing roles from a technical point of view. Such roles only reflect the performed actions on data objects but lack business meaning in form of the daily work of employees. The authors of the surveys identified [128, 142, 197, 225] to be the most fundamental role mining approaches. The works of Kern et al. and Kuhlmann et al. [128, 142] provide a link between traditional data mining algorithms and RBAC. The authors propose a clustering algorithm that is based on the k-means algorithm to mine an RBAC role model. In the work of Schlegelmilch et al. [197] a hierarchical clustering algorithm is used to derive roles from merged access permissions. Results are presented in a user-friendly graphical form allowing security experts to incorporate expert knowledge to guide the role mining algorithm. Vaidya et al. [225] present the unsupervised role mining approach called RoleMiner. They adapt traditional data mining algorithms in order to allow overlapping access permissions, as access permissions of roles typically have overlapping permissions. This allows them to mine more appropriate roles and role hierarchies. The approaches of Colantonio et al. and Molloy et al. [62, 59, 160] advance the work of [225] by improving semantics and introducing further cost and performance optimizations. Altogether, role mining operates on the technical level. Their premise is that the existence of access permissions in productive access control systems can be used to mine a role model. Hence, role mining approaches can neither bridge the gap between the business level and IT level nor can they analyze ACRs from the business point of view in contrast to the approaches presented in this thesis. The approaches in this thesis focus on eliciting a role model from de facto standard business process models that most organizations already have. Furthermore, the approaches of this thesis focus on the alignment of business processes, RBAC and EAAs with regard to business ACRs.

### 6.2.3 Role Mining for Hierarchy Optimization

This section examines related work in the area of role mining that focuses on the optimization of role hierarchies. Such role mining approaches do not mine access permissions from productive access control systems but have the assumption that all required access permissions are already existing in an analyzable format. These approaches are carried out bottom-up, meaning that an algorithm analyzes an existing role model to optimize the role hierarchy [66]. Role mining was introduced in detail in Section 2.2.3. In the following paragraphs, these approaches are briefly summarized.

A detailed survey on role mining approaches including role mining approaches that focus on optimizing the role hierarchy was recently done by Mitra et al. [35]. In the following paragraphs, several approaches are representatively discussed. The work of Guo et al. [97] elaborates on a role hierarchy building problem in which the set of roles already exists and the goal is to build an optimal role hierarchy. They use a directed acyclic graph where edges represent the relationships of roles and optimize the number of edges to a minimum. Similar approaches that use graph-based strategies are presented by Zhang et al. in [243] and [244]. [243] uses another graph-based optimization to build role hierarchies. They optimize a matrix that includes the user assignments, permission assignments and roles and afterwards, identify pairs of roles that can be merged based on permission overlaps. In [244] the authors use a heuristic as part of the graph-based approach. They iteratively add and remove roles from the graph based on a heuristic to optimize administrative costs of the overall role model. Dong et al. [71] introduce a role mining approach that leverages the network-clique-finding model to optimize role hierarchies. Roles are mapped to a network in which they build cliques that are optimized with regard to specific parameters. The work of Molloy et al. [159] uses formal concepts to optimize the role hierarchy. They formulate triples of objects, attributes and a relation that is based on access permissions. The role hierarchy has to satisfy the specified triples which optimize certain parameters with regard to the relation. Vaidya et al. [224] extend the work of [225] in order to provide a role hierarchy optimization for mined roles. Their aim is to minimize the number of roles. Therefore, they define a minimal perturbation role mining problem which derives a minimal set of roles from a user-permission assignment where the roles have to be similar to the roles mined with the role mining algorithm of [225]. It means that the approach computes similarities between roles produced by two different role mining algorithms where the first algorithm mines roles and the second one optimizes the role hierarchy. Takabi et al. [217] improve the work of [224] as the authors noticed that the measure for minimal perturbation was not appropriate for calculating the role similarity. They propose a new metric that focuses either on permission similarities, user similarities or hierarchy-relation similarities. Another role mining approach to optimize the role hierarchy is proposed by Lu et al. [150]. The authors view the problem from an end-user perspective. It means that users should be assigned to least possible amount of roles, as the problem with a high user-role assignment is that it becomes hard for users to handle their amount of roles. HyungHyo et al. [145] propose a role hierarchy optimization in which virtual roles are introduced in order to optimize the hierarchy. The difference to normal roles is that virtual roles are technically never assigned to employees. Such roles serve only the purpose to reduce the amount of duplicate permissions and ease the permission management. Therefore, virtual roles are introduced between roles that have a partial subset of access permissions (not a full subset, meaning that only some access permissions are the same). The aforementioned approaches do not mine a role model by analyzing productive access control systems but aim at optimizing an existing role model in terms of its role hierarchy. Hence, such approaches have a different purpose compared to the approaches of this thesis. The purpose of the aforementioned approaches is not to bridge the gap between the business level and IT level and they do not analyze ACRs from the business point of view. However, such approaches can be combined with the approaches presented in this thesis in order to optimize the role hierarchy with regard to certain optimization parameters. Nonetheless, these optimization parameters depend heavily on the requirements of organizations. Thus, there is no one-size-fits-all optimization but rather each organization will require a specific optimization based on their requirements.

### 6.2.4 Hybrid Approaches

This section discusses related work in the area of hybrid approaches. Hybrid approaches generate roles including permissions for RBAC based on business information (the role engineering part) and existing access control information from access control systems (the role mining part). Hence, these approaches combine role engineering and role mining in order to elicit roles and permissions that take business meaning into account.

A first attempt showing the need to consider business information in the context of role mining is described by Kuhlmann et al. [142]. The authors provide a link between traditional data mining algorithms and RBAC and explain the difference between business roles and technical roles. They discuss that considering business information in role mining is meaningful and propose a general methodology for role mining that allows to take business artifacts into account. However, in their realization of the role mining approach they did not rely on business artifacts. The work of Ma et al. [153] introduces a role mining approach that considers weights that are associated with permissions. These weights determine the importance of these permissions. It is possible to determine those weights based on different business artifacts or with the help of business experts. However, the authors do not describe how to weight the permissions based on business artifacts as they focus on weights stemming from other sources. A similar approach is presented by Xu et al. [240]. The authors propose a role mining approach that optimizes several quality metrics that may, for example, stem from policies. However, in their realization they do not focus explicitly on business policies but rather on IT policies and other metrics, e.g., role interpretability.

The first explicit combination of role engineering and role mining is proposed by Fuchs et al. [85]. The authors present a hybrid methodology that mines permission from access control systems and takes business information into account during the generation of roles. They classify basic roles, organizational roles and technical roles as an outcome from their methodology. Basic roles are roles with a basic set of access permission from which many other roles extend. Organizational roles are generated based on the organizational structure and technical roles are elicited from mined access permissions. However, many of the proposed steps have to be done manually by experts and the authors do not provide information on concrete steps how to generate the final role model. They describe a high-level methodology in which role engineering approaches and role mining approaches can be combined. Their methodology proposes to use role mining to elicit technical permission and bundle them in business roles stemming from organizational structures. Compared to the approach of this thesis, this hybrid approach is highly manual and does not consider the access permissions of business ACRs (only the business roles). Thus, this hybrid approach is not suitable to facilitate a business-IT alignment during the evolution of business processes.

Colantonio et al. [60] propose a formal framework that allows to combine existing role mining approaches with role engineering approaches. In particular they enable role mining algorithms to work on business artifacts as business processes and organizational structures. Therefore, they introduce a metric that allows role mining algorithms to derive additional roles from business artifacts and take these roles into consideration during the mining of the access permissions from access control systems. The metric measures roles in business processes based on the role's involvement in activities and cooperation with other roles of the same division. They call this metric "the spreading of the role among business processes or organization units" [60]. A similar approach is proposed by Colantonio et al. in [61]. The authors propose a methodology that combines a role engineering approach with certain types of role mining approaches. Therefore, they describe a manual role engineering approach in which experts decompose business information into dataset partitions. Each dataset partition groups users and permissions that belong to a certain division. Afterwards, a role mining approach uses these dataset partitions in addition to the access control systems to elicit access permissions and roles with business meaning. Both aforementioned approaches, utilize business information stemming from different business artifacts in order to enable role mining approaches to take business information with regard to the organizational structure into account. While the first approach proposes a formal metric which allows to transform information about roles from business processes and organization charts, the second approach proposes a manual methodology in which experts engineer dataset partitions with information about business roles that the role mining algorithm can analyze. Although these approaches consider business processes to elicit a RBAC role model, they do not consider the access permissions of business ACRs (only the business roles). The approaches focus only on business information that allows the role mining part to elicit roles that are closer to the organizational structure. Furthermore, the second approach requires rigorous human intervention in order to provide the required dataset partitions with business information for the role mining approach. Consequently, both approaches are not suitable to align RBAC and EAAs with business ACRs. The approaches presented in this thesis aim in particular to align EAAs and RBAC with business ACRs (including access permissions) stemming from business processes as well as to enable organizations to align those models during evolution scenarios without imposing significant overhead.

In [154] Mandala et al. propose a hybrid approach. They describe a role mining approach that is based on a bipartite graph and takes user attributes as a business information input. In essence, access permissions are mined from existing access control systems and roles as well as a role hierarchy are mined by taking user attributes into account. However, they assume that user attributes are already existing but in fact they have to be engineered in a manual role engineering process. Furthermore, business information is only used in order to mine roles with some kind of business meaning. Thus, this approach is neither able to align RBAC with business ACRs nor is it able to align the EAA with ACRs from RBAC and business processes.

Molloy et al. [159] propose another hybrid approach. They assume that roles already exist that are manually engineered with a role engineering approach. Then a role mining approach is used to mine further roles based on existing access control systems. In a next step, the role mining approach combines the roles from the role engineering approach and the role mining approach in order to optimize the role model. This optimization involves, for example, merging, splitting and deleting of roles. A similar approach is described by Hernandez et al. [107]. The authors propose an approach that mines roles including access permissions stemming from permissions engineered with a role engineering approach and permissions mined with a role mining approach. Their proposed role engineering approach relies on the manual analysis of questionnaires, user attributes gathered from various business information and user skill sets in order to engineer the access permissions for the employees of an organization. Finally, roles are elicited from the combination of permissions stemming from the role engineering and role mining approach. Because of the way both hybrid approaches are structured, it is possible to combine them with any role engineering approach that creates roles and permissions. Thus, both approaches can be seen complementary to the approaches, BAcsTract and PAcsTract, of this thesis, as the related approaches can be used to complement the role model with technical permissions. However, in contrast to the approaches presented in this thesis, those approaches are not able to align RBAC with business ACRs during evolution scenarios efficiently, because both approaches rely on rigorous, manual human interventions. Furthermore, the approaches are not able to align the EAA with RBAC or with business processes with regard to ACRs.

### 6.2.5 Hybrid RBAC Concepts

This section discusses related work in the area of hybrid RBAC concepts. Hybrid RBAC concepts are approaches that extended RBAC with further functionalities in order to modify or extend the RBAC concepts. They either combine certain parts of RBAC with other access control concepts or modify RBAC concepts in order to introduce new functionalities for access control. The following paragraphs will discuss how the approaches of this thesis, specifically BAacsTract and PAcsTract, can be used in presence of a hybrid RBAC concept.

Kern et al. [127, 129] describe a hybrid RBAC concept called enterprise role-based access control (ERBAC). It extends the concepts of RBAC. ERBAC introduces the enterprise role which is a business role that groups access permissions for one or more systems. Thus, it describes a role that is a layer above the typical RBAC role which lies at the application layer. The idea of the authors is to introduce a role concept that is more suitable for organizations and is better manageable by the human resources department. Therefore, access permissions of enterprise roles may consist of, for example, application layer roles, permissions of application layer roles and LDAP groups. ERBAC introduces another concept called joker permissions which allow to specify wildcards as part of an access permission. The enterprise role is responsible to define the attribute that is required to resolve the wildcard. For example, the access permission *read financeReport$ProjectName$* and enterprise attribute *ProjectPRIM* resolves into the permission *read financeReportProjectPRIM*. This allows to include more abstraction as part of the enterprise roles in order to easier reflect and manage the ACRs of business roles. The approaches of this thesis are also applicable to ERBAC as ERBAC focuses explicitly on business roles. In the context of ERBAC, roles that are extracted by BAcsTract and PAcsTract can be seen as enterprise roles containing the business permissions. As explained in Chapter 3, technical permission have to be complemented by security experts. In case of ERBAC, this would be the part of the application layer permissions that are not reflected in the business processes. Furthermore, it is possible to extend BAcsTract and PAcsTract in order to generate joker permissions based on the extracted permissions from the business processes. However, this thesis focuses on RBAC as it is widely adopted and other concepts often include the core RBAC concepts.

A similar concept is presented by Wortmann [238]. The author describes a system-overlapping authorization schema that is based on the concepts of ERBAC [127, 129]. There are two major differences. First, the enterprise role does not specify access permissions explicitly but rather groups application roles that define the actual access permission. This resolves a problem from ERBAC that enterprise roles may define access permission of completely different granularities and further enhances the clarity and manageability of roles. Second, on top of the enterprise roles the author defines process roles. Process roles are simply the roles of the business processes and their activities. These roles should help to define appropriate enterprise and application roles by interconnecting them with the business processes. However, the author did not make it clear what the difference between enterprise roles and process roles is and what value process roles contribute. Due to the similarity of enterprise roles and process roles it is a valid question whether process roles are practically useful as part of the role model. It seems that process roles are rather a construct of the author in order to provide a connection to the business processes [135]. Thus, one could consider using the approaches of this thesis to extract enterprise roles and application role permissions that have to be complemented by security experts with application permissions that are not part of the business processes. Nonetheless, the approaches of this thesis can contribute by extracting relevant RBAC parts for the proposed hybrid RBAC concept.

Hybrid RBAC concepts as in [122, 75, 73] propose a concept to combine attribute-based access control (ABAC) [4] with RBAC. ABAC uses fine-grained attributes with which data objects or services are labeled. Afterwards, users receive a set of attributes. If the attribute of a user matches the attribute of a data object or service, then the user is allowed to access the data object or service. A combination of ABCA and RBAC is possible, for example, by specifying the roles of RBAC as additional attributes of ABAC. Another possibility is to instantiate both concepts and define rules on how the RBAC system and ABAC system can work together. Independently of the way how the hybrid combination of RBAC and ABAC is realized RBAC roles and access permission need to be engineered. Thus, the approaches of this thesis can contribute to these hybrid RBAC concepts by generating the required RBAC parts. Furthermore, it is possible to extend the approaches of this thesis with additional sources from which the attributes for ABAC can be extracted. However, this is not in the scope of this thesis but it is described as part of the future work in Section 7.4.

Klarl et al. [134] propose the B&S-RBAC concept. It is a hybrid RBAC concept that extends the role concept of RBAC by dividing the traditional role into a business and a system role. Business roles are described as roles that have the same granularity of roles that can be used in business processes. System roles are described as technical roles that have to be defined by security experts or application managers. The aim of this distinction is to ease the user-role assignment for the human resources department while allowing a fine-grained definition of technical roles. The human resources department has only to assign business roles to users. Each business role is connected to a set of system roles that are managed by the application managers. As RBAC is a fundamental building block of this approach, the approaches of this thesis can be utilized in order to generate business roles as well as the corresponding access permissions. As explained in Chapter 3, technical permission have to be complemented by security experts. In the case of B&S-RBAC, this is anyway required and is done by the application manager or security expert. Thus, the approaches of this thesis can be utilized to generate the RBAC parts of the B&S-RBAC concept. Nevertheless, this thesis focuses on RBAC rather than a hybrid RBAC concept as RBAC is widely adopted and thus, is included by other concepts allowing the approaches of this thesis to be used regardless of the actually utilized hybrid concept.

In [3] the authors describe a NIST standard for the next generation access control (NGAC) concept. While NGAC is on its own an access control concept, it combines features of RBAC and ABAC and allows to express RBAC policies. NGAC links users and data objects through user and object attributes with different relations that represent the access rights. Altogether, these elements are stored as a graph to express the set of access control policies. This graph is equivalent to the role model in RBAC. The actual privileges can be derived from the graph on the fly as triples of user, access permission and data object. By using the relations NGAC enables to express access control policies of any granularity including fine-grained policies. To express RBAC policies the user attributes can be used to express roles while their relations express the access permissions. The approaches of this thesis can be utilized in order to get an initial set of interconnected user attributes, relations and data objects. However, as NGAC is way more specific and fine-grained than RBAC, the generated information has to be complemented with obligatory technical permissions and optional fine-grained access control policies. While NGAC is a rather new but interesting access control concept, it lacks the widespread adoption of RBAC. Hence, this thesis aims for an alignment of RBAC with business processes and EAAs because at the moment organizations can benefit the most from it. Furthermore, the focus on RBAC allows the approaches of this thesis to be extendable. It means that they can be tailored to any of the other aforementioned hybrid RBAC concepts.

## 6.3 IT Architecture Approaches

The subsequent sections outline related approaches that focus on security analysis in standard models of the architectural design phase. This section does not consider any extensions as extended models and security analysis on extended models were already discussed in Section 6.1. The focus of this section lies on approaches that encompass an analysis of security or privacy attributes that are related to access control, as the approaches presented in this thesis focus on access control. Section 6.3.1 makes a comparison with approaches that align the architecture with business processes. Among others this encompasses the enterprise architecture frameworks that provide guidelines on how to establish EAAs. Afterwards, Section 6.3.2 contrasts the approaches of this thesis from security analysis approaches on architecture models.

### 6.3.1 Architecture Alignment Approaches

This section discusses related work in the area of architecture alignment. The following paragraphs, begin with a discussion about enterprise architecture management (EAM) and how these methodologies are related to the approaches of this thesis. Afterwards, a comparison between the approaches presented in this thesis and approaches that align the architecture with business processes and RBAC is done.

The alignment between IT architectures and business models provides considerable benefits [90]. It is called enterprise architecture management (EAM) and involves initiating and establishing business processes along with governance as well as the definition of application scenarios and the IT architecture landscapes. EAM can be subdivided into four categories [149].

- EAM initiatives that focus on the taxonomy of business and architecture models and their lifecycles.
- EAM processes that describe recurring processes that can be used to achieve business-IT alignment.

- EAM application scenarios that focus on the viewpoints of stakeholders during the alignment of IT architecture models and business processes.
- EAM governance describes organizational structures that are required to achieve business-IT alignment.

EAM initiatives focus on the taxonomy of required business and architecture models along the different requirements of achieving business-IT alignment. Thus, these methodologies emphasize model-driven aspects and describe the lifecycles of those models. Enterprise architecture frameworks are methodologies to achieve this. They define the respective metamodels for the crosscutting concerns and how they should be used in order to align the IT architecture and business models. Enterprise architecture frameworks, for example, the Zachman Framework [241, 209], the Federal Enterprise Architecture Framework (FEAF) [94], The Open Group Architecture Framework (TOGAF) [96], the Gartner's Framework [114] and the Department of Defense Architecture Framework (DoDAF) [69] belong to the category of EAM initiatives. They introduce the required architecture and business models with regard to the crosscutting concerns of the business-IT alignment. Besides the taxonomy of business and architecture models these methodologies describe the lifecycles of those models and how they should be used during the EAM. EAAs as well as business processes are always part of these methodologies. A subset of enterprise architecture frameworks are the enterprise information security architecture (EISA) frameworks, for example, the Gartner's EISA Program [141] and the Sherwood Applied Business Security Architecture (SABSA) [118]. Broadly speaking, EISA frameworks describe the same topics as enterprise architecture frameworks but with a special focus on IT security. They describe a comprehensive methodology for EAM with the focus on specific, security related models for the organizational security including security processes, business security architecture, information system security and performance management. The primary goal is to align IT security from the organizational perspective with the core business and IT strategies. Other methodologies, which belong to the EAM initiatives, focus on the success factors of EAM methodologies, for example, the works from Bricknall et al. [47], Bussells [55], Janssen et al. [121] or Seppanen et al. [204]. The authors elaborate on critical success factors of the aforementioned enterprise architecture frameworks from the practical point of view and describe among others, the importance of top management involvement, the implementation of governance processes and the relations to other organizational lifecycles.

EAM processes describe typical recurring business processes or activities that help organizations to achieve business-IT alignment during the EAM. Buckl et al. [52, 54], Schmidt et al. [198], van der Raadr et al. [184] focus on specific activities with regard to EAM, for example, defining target states, analyzing current states and evaluating established EAM measures. Publications like the works of Hafner et al. [99] and Niemann [173] focus on the high-level view by providing business processes that have to be established for EAM. Examples for high-level processes are IT-strategy processes, modeling processes, application portfolio management processes and policy deployment processes. These publications also elaborate on the lifecycle of those processes and on the specific challenges in each lifecycle of a process.

EAM application scenarios focus on the viewpoints and concerns of stakeholders that are involved in the EAM. Publications in this area, for example, from Bucher et al. [50], Hjort-Madsen et al. [110], Buckl et al. [51, 53] or Moser et al. [162] demonstrate the usage of EAM and their models for the development of applications in specific domains. Based on practical experience the authors describe best practices tailored for the specific domains.

EAM governance focuses on establishing of organizational structures, for example, roles, committees, principles and standards with regard to EAM. While the works of Strano et al. and Niemi [214, 174] describe important roles and their responsibilities and involvements throughout different EAM steps, the works of Venkatesh et al. and Hoogervorst [229, 111] define high-level organizational structures in order to integrate governance activities into EAM methodologies. The works of Winter et al. and Greefhorst et al. [232, 95] focus on principles and standards.

Compared to the goals and approaches of this thesis, EAM aims for a high-level view of business-IT alignment. EAM proposes manual methodologies on how to establish and manage business-IT alignment in organizations. EAAs and business processes are two of the more important and widespread models of EAM. However, EAM encompasses much more than these two models. EAM provides a taxonomy of the required models including their lifecycles, describes crucial activities and business processes in order to facilitate business-IT alignment, elaborates on experiences and best practices with regard to the application of EAM in specific domains and defines various governance activities including roles, organizational structures and principles that have to be integrated during EAM. Compared to the contributions of this thesis, EAM is much more high-level and focuses on defining organizational fundamentals in order to organize EAM from the organizational point of view. It describes rigorous manual processes and activities and sets them in a broader organizational context. This makes EAM challenging [149]. While EAM is widely known, it is hard to apply for various reasons [140]. One reason is the complexity of EAM and the high number of processes and models that have to be designed carefully. Another reason is the necessity for bringing many different stakeholders together who require deep knowledge in the various models of EAM. This thesis tries to overcome this gap with regard to business ACRs. In contrast to EAM, the approaches of this thesis exploit well-known and de facto standard models of business and IT and thereby, lower the threshold and effort for achieving business-IT alignment with respect to ACRs.

In [56] Castellanos et al. present a semi-automatic approach called KALCAS to detect misalignments of business processes and IT architectures with regard to data. They utilize ontology matching to automatically infer mappings between business processes and architecture models including EAAs and data architectures. Afterwards, users have to verify the inferred mappings. The authors propose some heuristics based on the ontologies that are able to identify whether data specified in business processes are represented as part of the architecture models. By using a query language enterprise architects are enabled to express further alignment heuristics. The authors differentiate between three states a) *aligned* means that data from business processes is instantiated in the architecture, b) *misaligned* means that data from business processes is not represented in the architecture and c) *omitted aligned* means that data from business processes is represented by components of a different domain. KALCAS allows to identify whether data from business processes is represented as part of the architecture or not. This includes redundant representations. However, the approach does not consider ACRs nor does it identify violations of ACRs in EAAs. In contrast, the approaches of this thesis align business processes and EAAs with regard to ACRs and allow an automatic extraction of business ACRs to form an initial RBAC role model. These delineations also apply to similar approaches as [33] that aim to align business processes and EAAs by calculating different metrics that represent the coverage of business elements in the architecture models.

Heinrich et al. [105] propose an approach to align business processes and architecture models in order to analyze their interrelations. They introduce a simulation that interconnects information relevant for the business processes with the architecture model in order to make a performance prediction. On the one hand, this performance prediction is more accurate as it takes relevant information from business processes with regard to performance into account. On the other hand, the performance prediction allows to reason about design alternatives and verify them against performance requirements. However, the authors focus on analyzing performance requirements rather than ACRs.

There are several approaches which propose algorithms to derive IT artefacts from business processes for the pre-architectural design phase, the requirements engineering phase. Cruz et al. [64] present a model-driven approach to derive UML use case diagrams including descriptions and UML class diagrams, representing the domain, from business processes. Afterwards, the authors generate a user interface model from the extracted models of the business processes. In contrast to the approaches of this thesis, the goal of the approach is different as it focuses on deriving architecture models in order to generate user interfaces. Together with UML use case diagrams and UML class diagrams they focus on the requirements engineering phase rather than the architectural design phase where EAAs are designed. The work of Brdjanin et al. [46] describes an approach to derive UML class diagrams from business processes in order to ease and quicken the generation of this model. They achieve their goal with high correctness and high completeness. Another approach is described by Khilf et al. [130]. The authors present transformation rules in order to generate UML use case diagrams, UML sequence diagrams and UML class diagrams from business processes. These rules should align models of the requirements engineering phase with business requirements. In comparison to the approaches of this thesis, approaches as the aforementioned approaches have two major differences. First, they do not directly focus on the architectural design phase by aligning the architecture but rather focus on the previous phase of requirements engineering and its models. Second, they do not focus on the transformation or alignment of security attributes as ACRs. In comparison, the approaches of this thesis align EAAs and RBAC policies with ACRs from business processes.

In Section 6.2 I discussed several publications, e.g., [182, 78, 80, 152] that propose either manual approaches to align access control policies with models of the architectural design phase or automatic approaches to mine access control policies from IT models of the architectural design phase. These approaches can be seen as a way to align parts of the architecture with the access permissions of the access control system. Approaches of the first category propose methodologies in which experts manually engineer access control policies during the architectural design phase in order to extract access permissions for the access control systems. They propose different ways on how to document the access control policies as part of the IT models. However, these approaches are slow and resource-intensive as they impose rigorous manual interventions and additional modeling effort in order to achieve their goals. Thus, these approaches are not suitable for a business-IT alignment in the course of evolution scenarios, where the duration and resource-intensity of the alignment becomes crucial. Furthermore, these approaches do not consider EAAs in particular but focus on other IT models and thus, do not bridge the gap of aligning EAAs with ACRs and in particular not business processes with ACRs. Although approaches of the second category are partially automatic, they do not take EAAs into account. Thus, these approaches do not bridge the gap of aligning ACRs between EAAs, business processes and access control systems. Moreover, approaches of both categories do not take business ACRs into account.

### 6.3.2 Security Analysis on Architecture Models

This section discusses related work in the area of security analysis approaches of models related to the architectural design phase. This encompasses above all the analysis and development of ACRs during the architectural design phase and the verification of access control related security attributes in architecture models.

Ahn et al. [15] present an empirical methodology called Assurance Management Framework (AMF) for modeling ACRs in parallel with the architecture models and afterwards, transform the modeled ACRs into code to enforce access control policies. AMF has four phases. In phase one and two the security model is designed with UML and access control policies are specified in a formal language. In phase three the consistency and validity of the security model and access control policies are validated. Phase four is a manual phase in which identified conflicts are resolved by the security expert. Based on the AMF methodology code can be generated to enforce the specified ACRs. A similar methodology is proposed by Kim et al. [133]. It allows a systematic configuration of access control systems by capturing variabilities of RBAC. The authors also use UML models that have to be designed in parallel with the architecture models in order to define all static and behavior properties of the RBAC features. The methodology allows to verify the correctness of specified access control features. Another approach is presented by Mouelhi et al. [163]. The authors propose an UML metamodel with which security aspects and generic access control policies are designed during the architectural design phase. Their model enables security experts to do early consistency checks and then automatically transform the access control policies into XACML policies and aspect-oriented code for applications to enforce these policies. While the aforementioned approaches rely on the de facto standard modeling language UML, it is still required to explicitly design a security model and the access control policies with UML and a formal language or a RBAC feature model with UML. Thus, a second model is built in parallel with the architecture. In contrast to the approaches of this thesis, the aforementioned approaches require additional effort in order to design the required models. This makes the approaches resource-intensive and less effective during evolution scenarios. In contrast to the approaches of this thesis, the authors of the previously mentioned approaches aim for a different purpose. First, they aim to validate the designed access control policies for consistency and second, they aim to generate code for the implementation phase. In contrast, the approaches of this thesis aim for a business-IT alignment by aligning business processes with RBAC and EAAs.

Gerking et al. [87] propose an approach to verify information flow requirements in component-based architectures. Therefore, they extend MechatronicUML, a top-down methodology to develop component-based architectures of a system, with a lightweight specification of information flow policies for services. During the architectural design architects specify the security policy that describes the sensitivity of the service (secret, public, neutral). The authors propose rules for the input/output delegation and assembly of data which enable the approach to verify if the architecture complies with the specified information flow policies. Therefore, the architecture is complemented with behavioral specifications for services of components, which is typically done in the MechatronicUML methodology and a verification tool is utilized to verify the information flow policies. Compared to the approaches of this thesis, this approach focuses only on the architecture and requires the specification of information flow policies in order to verify them on a given architecture. In contrast, the approaches of this thesis do not require any further specification of policies and they focus on the alignment of RBAC and EAAs with business process ACRs.

Seifermann et al. [202, 203] present an approach to analyze confidentiality in architecture models through the use of data flow diagrams. Therefore, they describe how to integrate data flow-oriented behavior descriptions into the architecture model. This enables architects to verify the architecture for compliance with confidentiality requirements. Confidentiality requirements have to be explicitly modeled as part of the data flow-oriented behavior description. They use roles and access permissions attached to components to express confidentiality requirements and a verification tool to identify confidentiality issues. Although this approaches also focuses on confidentiality, it focuses only on the IT level and does not explicitly take business artifacts into account. Thus, the approach cannot achieve business-IT alignment. Furthermore, their approach requires manual effort in order to design the data flow-oriented behavior descriptions and to express confidentiality requirements.

Approaches that are similar to the work of Tuma et al. [221] enable to verify confidentiality polices during the architectural design phase. However, such approaches rely on data flow diagrams that have to be modeled during the architectural design phase and have to be additionally enriched with security policies, for example, in form of confidentiality levels as in the case of [221]. A data flow analysis is used to statically verify if any of the specified security policies are violated. Although such approaches also focus on violations of access control policies, they do this by using data flow diagrams rather than EAAs. Another difference is that access control polices have to be modeled manually. The approaches of this thesis go beyond this by focusing on a business-IT alignment with regard to ACRs.

Heyman et al. [109] describe an approach to verify security patterns during the architectural design phase. Therefore, the authors use the Alloy language to describe the architecture model and verify if the architecture complies with the predefined security patterns that are also expressed in Alloy. Compared to the approaches of this thesis, the approach does not focus on ACRs and does not facilitate a business-IT alignment, but rather focuses on assessing applied security patterns as part of the architecture.

# 7 Conclusion

This chapter concludes this thesis by beginning with a summary of problems, research questions and contributions of this thesis in Section 7.1. Afterwards, Section 7.2 points out the benefits of the presented approaches for the business level of organizations, security experts and enterprise architects. Section 7.3 recapitulates the assumptions and limitations that were discussed throughout this thesis. Finally, Section 7.4 elaborates on future work.

## 7.1 Summary

In this thesis I addressed three important goals of the business level a) identifying and protecting critical assets and sensitive data b) establishing appropriate organization-wide IT security and privacy strategies and c) complying with the rising amount of security and privacy laws. Access control requirements (ACRs) play a significant role in the realization of all three goals and the IT level is required to realize them. While the business level defines business processes to express how the organization is running, the IT level engineers the RBAC role model and designs an enterprise application architecture (EAA) to organize the information systems that support the business processes. Business processes, role-based access control (RBAC) and EAAs evolve constantly over time and affect each other in non-trivial ways. Thus, aligning ACRs between those models is a challenging task (explained in Section 1.1). There is a need to realize:

- an appropriate and compliant establishment of access control policies in RBAC.
- an alignment between the EAA and the ACRs from the business level.

In order to align business processes of the business level with RBAC and the EAA of the IT level with regard to ACRs several problems have to be solved, as described in Section 1.2. First, the enterprise architects and security experts of the IT level do not have the knowledge about which business assets are critical and their required protection degree (problem **P1** in Section 1.2). Hence, they are not able to define appropriate access control policies. This means that essential knowledge about which systems are allowed to access which assets and how to design the access control policies is missing on the IT level. Only the holistic view of the business level has the required knowledge. Hence, the IT level requires the information about ACRs from the business level. Second, the business level and the IT level have different terminology, domain knowledge, domain-specific models and modeling tools which lead to a communication gap (problem **P2** in Section 1.2). Misunderstanding may lead to errors and security breaches. This leads to the third problem.

Expert knowledge is required on the IT level for the specific business level terminology (problem **P3** in Section 1.2).

Forth, engineering a role model for RBAC is costly and error-prone as security experts have to manually analyze a vast amount of business processes to understand the ACRs (problem **P4** in Section 1.2). Depending on the size of the organization, business processes grow easily into hundreds, resulting in a vast amount of complex and interrelated artifacts demanding a specific business knowledge to understand them. As this complex role engineering process is manually done human errors occur. However, each error is a potential security threat to the organization, as it may result in vulnerabilities and data leakage and thus, undermine the three aspired goals of the business level. While organizations evolve, ACRs change over time and demand adaptations. This increases the problem of errors throughout the role engineering process as well as the overall costs for RBAC due to the requirement of repetitive manual adaptations. Fifth, the resulting role model for RBAC is misaligned with regard to business level ACRs (problem **P5** in Section 1.2). There is no traceability between the manually engineered role model for RBAC and the business processes and there is no automatism to check the role model against the ACRs from the business processes.

Sixth, designing the EAAs is complex and error-prone because the enterprise architect has to cope with a large amount of functional and non-functional requirements stemming from various stakeholders (problem **P6** in Section 1.2). Many stakeholders of different domains have to be involved and understanding the correct requirements is a severe challenge due to the different domain knowledge and domain models. This leads to logical and design mistakes during the design of the EAAs. While logical mistakes simply arise from faults and false solution approaches, design mistakes arise from unclear, false interpretation and misunderstanding of requirements. Seventh, the continuous evolution of organizations leads to a misalignment between the EAA and business level ACRs (problem **P7** in Section 1.2). The enterprise architect has to consider IT security and privacy requirements. A fundamental building block of both are the ACRs. Due to the communication gap, it is challenging to align the EAA with business level ACRs correctly. EAAs and business processes are developed separately and without an appropriate and automatic transfer of ACRs between them. Finally, there is missing support to keep RBAC and EAAs aligned with ACRs from business process during the required manual adaptations of evolution scenarios (problem **P8** in Section 1.2). Evolutionary change is not well studied and understood so far, especially for ACRs. However, a correct adaptation of the aforementioned models is crucial to achieve the aspired goals of the business level.

To address the aforementioned problems this thesis examined the following research questions, which were described in Section 1.3:

RQ1 What kind of business knowledge can be extracted from business processes about access control requirements?

RQ2 How can an alignment of business processes, RBAC and the enterprise application architecture help the business level and IT level to better understand mutual dependencies stemming from access control requirements?

RQ3 What kind of business knowledge is no longer needed on the IT level when RBAC and the enterprise application architecture are automatically aligned with business level access control requirements?

RQ4 To what extent can an automatic extraction of business level access control requirements make role engineering more efficient?

RQ5 How can RBAC be aligned with business level access control requirements?

RQ6 To what extent can an identification of access control requirement breaches in the enterprise application architecture make error resolution more efficient?

RQ7 How can the enterprise application architecture be aligned with business level access control requirements?

RQ8 How can an alignment of business processes, RBAC and the enterprise application architecture support evolution scenarios of business processes, RBAC and the enterprise application architecture?

Throughout this thesis, two major concepts were introduced. The first concept, presented in Section 3.1.1, describes how a role model can be extracted from business processes automatically. The second concept, presented in Section 3.1.2, describes how ACR breaches can be identified in EAAs automatically. Afterwards, Section 3.2 elaborated on the concrete approaches which realize the foregoing concepts. While the BPMN Access Permission Extractor (BAcsTract) and the Palladio Access Permission Extractor (PAcsTract) realize the extraction of access permissions for a RBAC role model from the business process modeling languages BPMN and IntBIIS_LP, the Access Permission Architecture Aligner (AcsALign) realizes the identification of ACR breaches in data flows of services in EAAs. The following paragraphs summarize the main contributions of this thesis, that were described in Section 1.6.

BAcsTract and PAcsTract extract implicitly modeled business level ACRs from business processes in six steps (contribution **C1** in Section 1.6). This transfers the knowledge about critical assets and protection degrees from the business level to the IT level and helps the IT level to better understand the demands of the business level. During the extraction of the ACRs an ACR mapping model is built that interconnects access control relevant elements between business processes and RBAC (contribution **C2** in Section 1.6). Later, AcsALign extends the ACR mapping model with access control relevant elements from EAAs. The ACR mapping model bridges the communication gap between the business level and the IT level with regard to ACRs. It allows to track design decisions regarding ACRs across the three mentioned models and thus, couples the domain-specific models together. This enables the business level and IT level in the understanding of design decisions in models outside of their expertise and helps the enterprise architect in particular to understand how to resolve an identified ACR breach. Based on the extracted ACRs from business processes BAcsTract and PAcsTract generate an initial role model (contribution **C3** in Section 1.6). This helps security experts during the tedious role engineering process and automates the alignment of the RBAC role model with business level ACRs. AcsALign uses the extracted ACRs to generate data flow constraints (contribution **C4** in Section 1.6). Using these data flow constraints AcsALign is able to analyze the data flows of EAAs for violations of ACRs. This helps the enterprise architect to identify errors, resolve them and by doing so, aligning the EAA with business level ACRs. The approaches of this thesis can be used throughout various scenarios together but also independent from each other. A high-level process, described in Chapter 4, explains how organizations can utilize BAcsTract, PAcsTract and AcsALign throughout different evolution scenarios to understand mutual dependencies and to align RBAC and the EAA with ACRs from the business processes (contribution **C5** and **C6** in Section 1.6).

Within the scope of this thesis two case studies were conducted to validate the approaches and proposed contributions. The case studies were described in Chapter 5. The first case study in Section 5.1 is based on the Common Component Modeling Example (CoCoME). CoCoME is a community driven case study of a realistic supermarket chain developed by the scientific community. It was developed to research software evolution and has several evolution scenarios developed by various research groups. CoCoME contains business processes and an EAA which both have to comply with various IT security and privacy regulations as I demonstrated in [180]. This case study on CoCoME focuses on the validation of the role model extraction from business processes with the use of BAcsTract. The second case study in Section 5.2 is a real-world case study, resulting from a cooperation with a national art gallery that revises its information systems. The national art gallery provided its business processes as well as the EAA. The business processes encompass critical data flows of confidential information as financial budgets, insurance values and customer/client data. The EAA spans over multiple systems to process common data processing patterns of information systems, including delegation, merging, reading from and writing to databases. The second case study focuses on the validation of the identification of data flows in EAAs that violate ACRs. It also validates the role model extraction from business processes. The case study is conducted with the use of AcsALign and PAcsTract.

Both case studies were conducted following the goal question metric (GQM) method [181] to systematically validate the contributions of this thesis. The following aspects were examined throughout the case studies:

- Quality of generated access permissions.
- Quality of identified data flows in services of the EAA that violate ACRs.
- Completeness and correctness of the generated ACR mapping model with regard to the traceability of ACRs across business and IT models.
- Applicability of the approaches in evolution scenarios of business processes and EAAs.

The overall GQM model that describes the validations goals, research questions and metrics can be found in the beginning of Chapter 5. The results of the case studies indicate that the extracted access permissions for the initial role model have a high accuracy. BAcsTract and PAcsTract can successfully automate parts of the role engineering processes and by this ease the work of security experts and align the role model with business process ACRs. Results show that this leads to a reduction of human errors made throughout the role engineering process. Furthermore, BAcsTract and PAcsTract are applicable during evolution scenarios to generate a role model containing the required adaptions that otherwise have to be engineered manually. A comparison between several generated role models as well as the ACR mapping model allows to forecast how different evolution scenarios affect the RBAC role model. The ACR mapping model allows to understand how design decisions in business processes, RBAC and EAA affect each other with regard to ACRs. The case study results show that AcsALign successfully identifies logical and design mistakes in EAAs with respect to ACRs. Furthermore, the results indicate that AcsALign can be utilized during evolution scenarios without imposing additional effort. By doing so, it can align the EAA with ACRs during various evolution scenarios. Moreover, results of the case studies show that the generated ACR mapping model is built with high accuracy, meaning that it contains correct ACR mappings and is complete in terms of the number of ACR mappings. With the help of the ACR mapping model AcsALign provides the enterprise architect context information about the violated access permissions and the violated parts of the business processes during the resolution of identified ACR breaches.

## 7.2 Benefits

The contributions of this thesis improve the alignment between the business level and the IT level in terms of ACRs. While the approaches BAcsTract and PAcsTract align the RBAC role model with ACRs from business processes, the approach AcsALign aligns the EAA with ACRs from business processes or other sources. The approaches target to help the business level, security experts and enterprise architects of organizations. The following paragraphs summarize how these roles benefit by utilizing the approaches.

**Security experts** utilize BAcsTract and PAcsTract either during the role engineering process to extract business ACRs from business processes and transfer them into an initial role model or during required role model adaptions in evolution scenarios. The conducted research shows that the approaches provide the following benefits for security experts during its utilization:

- The approaches facilitate the extraction of business ACRs in form of role-permission pairs that are implicitly modeled as part of the business processes. This allows to systematically and automatically transfer the knowledge about critical assets and protection degrees, which reside on the business level.
- The automatic extraction of ACRs from business processes to establish the ACR mapping model and to build the initial RBAC role model bridges the communication gap between the business level and the IT level with respect to ACRs. Security experts can understand why certain access permission are required by tracing them back to their originating activities in business processes. Furthermore, they are relived from analyzing the vast amount of business processes manually as the ACRs from business processes are automatically and systematically transferred into the

initial role model. This automatic transformation also reduces the dependency on the skills of security experts with regard to understanding business level terminology and domain models.

- The approaches reduce the complexity of the role engineering by building an initial role model that incorporates all ACRs from the business processes automatically. As the ACRs from the vast amount of business processes are extracted automatically, the overall role engineering process becomes quicker and more cost effective. Due to the automation of the otherwise error-prone manual extraction of access permissions from the business processes the role engineering with BAcsTract and PAcsTract becomes less error-prone. Thus, security experts are able to increase the correctness of the role model while keeping additional effort low.
- Security experts are able to align the RBAC role model with the ACRs from the business processes by utilizing the approaches. This is the case because the approaches extract ACRs from business process according to a formalized algorithm that interconnects access control relevant elements of business processes with RBAC.
- Through the use of the ACR mapping model security experts are able to track and understand current and also previously made design decisions. They can trace access permissions of the role model to their originating activities in business processes and by this understand their necessity as well as boundary conditions. If necessary, they can speak with the employees responsible for the activities as well as with the business process owners to clarify questions. The ACR mapping model also helps in understanding mutual dependencies between business processes and the RBAC role model, as influences between those models can be computed.
- During evolution scenarios, security experts achieve a faster adaptation of the role model due to the automation of the approaches. Thus, security experts can react faster to changes resulting through evolution scenarios while increasing the correctness and alignment of the RBAC role model. Furthermore, the ACR mapping model enables security experts to keep old design decision with regard to access control in mind and to understand how evolution scenarios affect the role model.

**Enterprise architects** primarily utilize AcsALign but also BAcsTract and PAcsTract. While AcsALign is utilized to identify data flows in the EAA that violate ACRs, BAcsTract and PAcsTract are utilized to provide a set of ACRs stemming from business processes. The conducted research shows that the approaches provide the following benefits for enterprise architects:

- The approaches provide means to transfer the required knowledge about ACRs from the business level to the enterprise architects, as the knowledge about critical assets and protection degrees resides at the business level.
- By utilizing the approaches the communication gap between the enterprise architects and the business level is closed because information about business ACRs is systematically transferred to the IT level and used to identify ACR breaches in the EAA. Furthermore, the ACR mapping model enables enterprise architects to

better understand mutual dependencies between the EAA, RBAC and the business processes. It increases the comprehensibility about ACRs and their impact on the EAA.

- The enterprise architects are enabled to identify logical and design mistakes during the design phase of the EAA by utilizing the approaches to identify data flows of services that violate ACRs stemming from business processes. This allows to increase the correctness of the EAA and keep it secure.
- By identifying ACR breaches in the EAA the approaches help enterprise architects to keep the EAA aligned with the ACRs from the business processes. By resolving identified violations, they align the EAA.
- The ACR mapping model interconnects access control relevant elements across EAA, RBAC and business processes. This allows to track design decisions regarding ACRs across the three mentioned models and couples these domain-specific models together. On the one hand, enterprise architects are supported in resolving logical and design mistakes that led to ACR breaches, for example, by providing traceability to affected business processes and by this providing means to contact the business process owners of affected business processes. On the other hand, the enterprise architects are supported in understanding design decisions in models outside of their expertise.
- Enterprise architects can utilize the approaches during evolution scenarios to identify if the EAA is aligned with the evolution scenario and to understand the impact of the evolution of business processes or RBAC on the EAA. The latter is done by checking if changes in the role model or business processes imply ACR breaches in the EAA and if so, indicate required changes of the EAA. During evolution scenarios of EAA enterprise architects can identify if changes of the EAA are still aligned with the ACRs from business processes and RBAC. Furthermore, the ACR mapping model enables to understand previously made design decisions and how changes in one model affect the other models. This allows to keep the EAA correct and secure throughout evolution scenarios.

The **business level** also profits from the approaches of this thesis. The approaches support the business level in achieving the three goals, identifying and protecting critical assets and sensitive data, establishing appropriate organization-wide IT security and privacy strategies and complying with the rising amount of security and privacy laws, better, faster and more cost effective. The conducted research shows that the business level can benefit through the approaches in the following ways:

- The business level has the goal to establish appropriate organization-wide IT security and privacy strategies, but the knowledge about critical assets and appropriate protection degrees resides at the business level. In order to achieve this goal ACRs are fundamental and have to be transferred correctly from the business level to the IT level. The business level profits from the approaches as they transfer business knowledge about ACRs to the IT level and provide an automatic way to align RBAC and the EAA with these ACRs. This increases the overall security of the organization.

- Another goal of the business level is to comply with the rising amount of security and privacy laws to avoid penalties. ACRs are a crucial part of these security and privacy laws. As such they are reflected inside the business processes. The approaches enable to align RBAC and the EAA with the ACRs from the business processes. Any misalignment can be identified and resolved during an early design phase. By doing so, the approaches facilitate a compliance with ACRs of security and privacy laws that are reflected in business processes.
- There is a communication gap between the business level and the IT level. The ACR mapping model established by the approaches enables the business level to understand how ACRs affect other domain specific models. It allows to better understand and trace design decisions regarding ACRs across business processes, RBAC and EAAs. During evolution scenarios of business processes, the business level can understand how the other models are affected and which changes have to be conducted in those models.
- The approaches enable the business level to conduct tradeoff analyses between variants of certain business processes, RBAC role models and EAAs. The high-level process explains which approaches can be utilized during which evolution scenarios. This helps the business level to understand when they can utilize which approach for a tradeoff analysis. With regard to business processes it is possible to analyze how variants of business processes comply with the current role model or how they comply with the current EAA. Variants of RBAC role models can be analyzed for compliance with business processes or with the EAA and finally, variants of EAAs can be analyzed for compliance with RBAC or with business processes.

Lastly, **the organization** as a whole does profit from the utilization of the approaches in the following ways:

- The conducted research shows that the approaches impose only little additional effort, by relying on de facto standard modeling languages and by reusing already existing models of business processes and IT architectures that have to be designed anyway. Hence, the approaches can be used by most organizations from scratch and without additional effort.
- The approaches reduce costs for organizations a) during the engineering of the RBAC role model, b) by identifying ACR breaches in EAA in an early design phase and c) by aligning RBAC and the EAA with ACRs from business processes. Regarding a) parts of the role engineering are automated and complexity of the overall role engineering process is reduced. For b) in an early design phase of the EAA data flows of services are identified that violate ACRs and the enterprise architect is supported during the resolution of identified mistakes. This reduces costs that arise during a security breach and that have to be invested in order to resolve the mistakes in a later phase. Regarding c) costs are reduced as RBAC and the EAA are aligned with the ACRs from business processes and thus, are aligned with the ACRs from IT security and privacy laws that are expressed in business processes.

- Organizations that did not use RBAC so far can benefit from the approaches due to the fact that BAcsTract and PAcsTract reduce complexity and costs for establishing a role model for RBAC. Hence, the migration process to RBAC becomes faster and less error-prone making RBAC more attractive for these organizations.
- There are manifold scenarios where BAcsTract, PAcsTract and AcsALign can be utilized in organizational processes. The engineering of RBAC and the design of the EAA, are only the obvious scenarios. As organizations and their domain-specific models evolve constantly over time, a periodic adaptation of the models is required. This creates a wide scope of evolution scenarios where the approaches can be utilized to either enhance the adaptation processes or predict changes in one of the three aforementioned models. In such scenarios the organization benefits from the high-level process that explains how to utilize BAcsTract, PAcsTract and AcsALign in different evolution scenarios to understand mutual dependencies and compare model alternatives with each other.

## 7.3 Assumptions and Limitations

This section summarizes and references the assumptions and limitations of this thesis. First of all, Section 1.5 makes first assumptions and provides a scope for the approaches of this thesis.

Generally, it is clear that organizations which explicitly model and use business processes and EAAs benefit more from the approaches as they already possess all required input models. This is the case for many large organizations. Nevertheless, other organizations can begin to design the required models and through this, they will not only profit from an alignment of ACRs across business processes, RBAC and EAA but also from the benefits of having the models themselves. As this thesis focuses on ACRs from the area of IT security, organizations with many or complex access control rules and organizations with high-security requirements can benefit most from the utilization of the approaches.

In the area of enterprise architecture this thesis focuses on aligning the EAA with business processes. Therefore, I assume that business processes and IT architecture are built in a top-down manner, meaning that the IT level has to meet requirements of the business level. Business experts do not choose from ready-to-use IT modules, but rather the enterprise architect designs the EAA according to the requirements of the business processes designed by business experts. However, this assumption is only made to explain the approaches in a systematic way. There are many different evolution scenarios in which an organization can benefit from utilizing the approaches. These were described in detail in Chapter 4.

Section 3.2.2 provides an overview over the input models and puts the assumptions into a bigger frame. In the following sections, Section 7.3.1 will summarize the assumptions and limitations with regard to BAcsTract and PAcsTract and Section 7.3.2 with regard to AcsALign.

### 7.3.1 Assumptions and Limitations of BAcsTract and PAcsTract

Some of the assumptions may bear limitations on the utilization of BAcsTract and PAcsTract and their results. This section summarizes the assumptions and limitations that were made throughout this thesis. Section 3.3.2 provided a broader discussion regarding the assumptions and limitations for BAcsTract and PAcsTract.

- **Existence of business processes:** I assume that organizations have already modeled their business processes. If not, additional effort has to be done in order to design them as they are required as input for BAcsTract and PAcsTract. However, medium to large organizations and especially organizations with high-security requirements, for example, critical infrastructures are obligated by laws to manage and organize their business according to certain requirements. To fulfill these obligations organizations design business processes according to business processes guidelines such as ITIL [34] and COBIT [32]. Hence, many organizations already have modeled their business processes.
- **Correctly designed business processes:** In the course of this thesis, I assume that business processes are designed correctly with regard to syntactics and semantics. If syntactics are not correct, the processes cannot be parsed correctly. Nonetheless, processes are often designed with modeling frameworks that do syntactical checks and may forbid nonsensical modeling (it is still possible to model nonsensical processes that are syntactically correct). Regarding semantics, the business level is responsible to reflect the processes of the organization correctly. There are other works that help them to accomplish this goal, but it is not the objective of the work done in this thesis. These limitations also affect the traditional role engineering process. If business processes are not designed syntactically correct, security experts are not able to understand them and if business processes are not designed semantically correct, security experts will propagate the errors into the role model. Hence, this limitation affects any approach in this area.
- **Scope of processes:** In this thesis, I assume that business processes encompass all departments of the organization and provide a comprehensive picture of the ongoing work done by employees on a daily basis. If parts of the business are not reflected in the business processes, they will not be reflected in the generated role model. This has to be considered by security experts. However, the approaches should help security experts in providing a more aligned and correct role model. This is still the case if only parts of the ongoing work of organizations are modeled. In such scenarios, security experts have to go through other business level artifacts either way.
- **Correctly modeled ACRs:** I assume that ACRs incorporated in business processes by the business level are legally correct and in line with the business goals introduced in Chapter 1, because the work of this thesis does not focus on identifying erroneous ACRs, but on defining an automated transformation of ACRs from business processes to IT level artifacts.

- **Initial role model:** BAcsTract and PAcsTract extract an initial role model encompassing business level ACRs that are implicitly modeled in business processes. As business processes reflect only the business view of ACR, technical ACRs have to be completed by security experts. The reason for this is that technical ACRs are not part of the business processes and thus, cannot be extracted from them. The goal of the approaches is to support security experts in extracting business level ACRs and this goal is accomplished by generating the initial role model.
- **Simple hierarchy:** BAcsTract and PAcsTract generate a simple hierarchy for the role model based on proper subsets of the role's permissions. If permissions of role A are a proper subset of permissions of role B then role B extends from role A. There is a wide research field in the area of role mining that aims at optimizing the hierarchy of role models with regard to various optimization goals. It is possible to combine such approaches with BAcsTract and PAcsTract to further optimize the role model hierarchy. However, the optimization goal heavily depends on the organization and the primary aim of BAcsTract and PAcsTract is to transfer ACRs from the business level to the IT level. Thus, building another optimization algorithm was not in the focus of this thesis but rather it leaves this matter open to allow organizations to decide for themselves which hierarchy algorithm fits them best.
- **Effort utilizing approaches:** Both approaches are designed to impose least possible effort for organizations to utilize them. This is achieved by focusing on de facto standard modeling languages, for example, BPMN and on models that organizations design anyway, for example, due to legislative obligation or for business performance reasons. It is clear that this assumption is not the case for every organization. Small organizations and startups may not design any business processes at all. However, this assumption is true for most medium to large organizations.
- **Evolution scenarios:** The presented approaches become especially useful during evolution scenarios. Certainly, to utilize the approaches during evolution scenarios these scenarios have to be reflected in the business processes. If this is fulfilled, the approaches can be utilized with low effort. How to utilize the approaches throughout various evolution scenarios was discussed in Chapter 4.

### 7.3.2 Assumptions and Limitations of AcsALign

Some of the assumptions may bear limitations on the utilization of AcsALign and its results. This section summarizes the assumptions and limitations that were made throughout this thesis. A broader discussion regarding the assumptions and limitations was provided in Section 3.4.2.

- **Existence of an EAA model:** I assume that the EAA is already designed in an organization. If not, additional effort has to be done in order to design it as it is required as an input for AcsALign. However, medium to large organizations have to organize their business to cope with its complexity. Therefore, these organizations will already have designed an EAA. Other reasons why organizations design an EAA are, for example, to maximize the organizational value by being able to make better decisions, to trim costs by having a more efficient resource allocation and to establish organization-wide IT security. Organizations with high-security requirements, for example, critical infrastructures will also have designed an EAA as security guidelines will force them to do so. Hence, for most medium to large organizations and organizations with high-security requirements these assumptions are true.

- **Scope of the EAA:** In this thesis, I assume that the EAA encompass all systems and services of the organization that are used throughout the business processes or for which ACRs are provided. If any parts of the organization are not modeled in the EAA, AcsALign will not be able to analyze it for EAA breaches. This has to be taken into account by enterprise architects. In such scenarios AcsALign can at least identify ACR breaches in those parts that are modeled, helping to secure and align them with correct ACRs.
- **Quality of data flow constraints:** The quality of data flow constraints generated by AcsALign depends on the quality and scope of the ACRs provided as input. At this point AcsALign depends, for example, on PAcsTract and thus, on the quality and scope of defined business processes. It is also possible to serve ACRs from other sources, for example, from an access control system. Nevertheless, ACRs need to be correct and cover as many parts of the EAA as possible.
- **No predefined IT-modules:** An assumption in this thesis is that the enterprise architect designs the EAA according to the requirements of the business level. There might be scenarios where the design is made bottom-up meaning that the business expert has to use predefined IT modules during the design of the business processes.
- **Limited to data types:** The analysis of AcsALign is limited to data types rather than to actual classes of data. This means that AcsALign cannot differentiate between different classes of data of the same data type. For example, it is possible that two classes of data with the same data type have different ACRs depending on the overall scenario. A newly planned exhibition is confidential during the planning phase but becomes public after the launch. Such expressions of data types are part of the runtime of organizations and AcsALign focuses on the support during the design phase. However, if different classes of data types are designed as individual data types during the design phase, AcsALign is able to distinguish them and the limitation can be bypassed.
- **Effort utilizing the approach:** AcsALign is designed to impose least possible effort for organizations to utilize it. This is achieved by focusing on models that organizations design anyway. Some startups and small organizations might be an exception. However, during the growth of an organization and in high security environments, organizations come to a point where they have to design an EAA either due to legislative obligations or due to management complexity. Thus, they can benefit from AcsALign.
- **Mapping of data types and service calls:** Depending on the modeling language that is used to design the EAA a) a mapping for data objects from business processes

to data types of the EAA and b) a mapping for activities from business processes to service calls of the EAA might be required. In case of PCM and IntBIIS_LP both mappings are part of the business process design. For other modeling languages as BPMN and UML this mapping has to be provided. However, both mappings require only low effort to create them and they have to be created only once. During evolution scenarios these mappings require only minor changes.

- **Read and written data types:** AcsALign requires a tool that performs a simple data flow analysis on the EAA to extract the read and written data types of service calls. Such a data flow analysis is very simple and there are tools that provide this capability for many established modeling languages. In this thesis, a data flow extension for PCM is used that was described in Section 2.4.3.
- **Evolution scenarios:** AcsALign is especially useful during evolution scenarios. Certainly, to utilize AcsALign during evolution scenarios the scenarios have to be reflected in the ACRs and in the EAA. If this is fulfilled, AcsALign can be utilized with low effort. How AcsALign and the other approaches can be utilized during evolution scenarios was discussed in Chapter 4.

## 7.4 Future Work

The following paragraphs elaborate on the future work with regard to the contributions and approaches presented in this thesis.

**Extending input information for the ACR mapping model:** The ACR mapping model provides traceability of access control information across models of business and IT (see Section 3.1 for the formal concept and Section 3.2.3.1 as well as Section 3.2.4.1 for the realization). Therefore, the model interconnects relevant elements of business processes, RBAC and EAAs. So far, the ACR mapping model is built upon the information extracted from the business processes, EAAs and the generated access permissions for RBAC. Regarding future work it is possible to use the extended technical access permissions of RBAC and link them with the information from business processes and EAAs. This has potential to increase the overall security of access control throughout evolution scenarios of business processes and information systems.

It is possible to take further domain models of business and IT into account and interconnect their access control relevant information with the current ACR mapping model. Enterprise architecture frameworks have four major domains, business architecture, data architecture, applications architecture and technology architecture. Each of them has a set of domain specific models. In addition to other models from the business architecture and application architecture domain, for example, system use case diagrams and organizational charts, models from the data architecture would be the most interesting ones to integrate into the ACR mapping model. This integration would provide a more detailed view of the data in an organization. Including these models into the ACR mapping model would increase the comprehensibility of how ACRs affect further parts of business and IT. The traceability across more domain specific models would ease the understanding for domain experts for models outside of their expertise. I elaborated in [25] that there is a need for holistic modeling of IT security and especially ACR across models of different domains, to increase the overall security and reduce errors done throughout the evolution of organizations.

Further interesting models, to extend the ACR mapping model, come from the IT security domain. Enterprise information security frameworks as the Sherwood Applied Business Security Architecture (SABSA) [118] and security threat modeling methodologies as STRIDE [157] propose models that are specific to IT security. Some of them encompass access control information as well as their realization in the organization. More specific and interconnected security information as part of the ACR mapping model would increase the comprehensibility of how ACRs and other security related measures affect models of the business domain and would increase the provided traceability information of the ACR mapping model. The latter would increase the understanding of domain specific models for experts of other domains.

**Combining PAcsTract and BAcsTract with hierarchy mining approaches:** PAcsTract and BAcsTract generate an initial role model by extracting implicitly modeled information about ACRs from the business process modeling languages IntBIIS_LP and BPMN (see Section 3.1 for the formal concepts and Section 3.2 for the realization). So far, both approaches generate only a simple hierarchy by identifying roles with a full subset of another role's permissions. In the area of role mining there is a research field that focuses on how to optimize the hierarchy of a role model under certain parameters. A possibility to increase the efficiency of the role model would be to combine PAcsTract and BAcsTract with such hierarchy optimization approaches. An example for this is the approach of [145]. They introduce virtual roles in a role model to optimize the hierarchy. The difference to normal roles is that virtual roles are technically never assigned to employees. They serve only the purpose to reduce the amount of duplicate permissions and ease the permission management. Therefore, virtual roles are introduced between roles that have a subset (not a full subset, meaning that only some access permissions are the same) of identical access permissions. To decide which of the numerous hierarchy mining approaches is best suited is a matter of research. The hierarchy optimization would be beneficial for organizations during the management and assignment of roles to employees.

**Extending input information of AcsALign:** AcsALign consumes a set of ACRs to identify data flows in EAAs that violate these ACRs (see Section 3.1 for the formal concept and Section 3.2.4 for the realization). The information about ACRs is provided, for example, by PAcsTract that extracts the information from business processes. Clearly, the quality of identified data flows that violate ACRs depends on the completeness and quality of provided ACRs. Thus, further research might be done in order to analyze other domain models and artifacts for additional ACR information as input. To accomplish this goal artifacts of the legislation domain, business domain and IT domain need to be analyzed. It is inevitable to understand which artifacts contain which information about ACRs and how much effort it takes to extract this information. Obviously, the extended input must provide information that complements the ACR information provided by PAcsTract as otherwise no new information would be provided. The extension of AcsALign with further input models might increase the amount of identified data flow violations and provide the enterprise architect with further relevant information on how to resolve identified errors.

**How do the models of business and IT affect each other in the backwards direction:** In this thesis, I made research on how business processes affect domain models of IT with regard to access control. However, as described in Section 1.1 models of business and IT affect each other in non-trivial ways especially during evolution scenarios of business processes and information systems. It means that domain models of IT affect business processes as well. Further research might be interesting for understanding how decisions made for RBAC and for the EAAs affect the business processes with regard to access control. Research questions might be *How does the lifecycle of access permissions affect the integrity of business processes?*, *How do different sets of role models influence the effectiveness of business processes?*, *Which effect have different realizations of EAAs on the implementation of business processes?* or *How do various access control measures change quality attributes of business processes?*. Altogether, this field of research might unfold further potential to increase IT security as well as data protection and enforce the correct realization of ACRs during the design time of business processes and especially during evolution scenarios.

**Transferring generated information of BAcsTract, PAcsTract and AcsALign to other domain models:** While BAcsTract and PAcsTract extract ACRs from business processes, AcsALign uses information about ACRs to analyze data flows of services in EAAs for violations of theses ACRs (see Section 3.1 for the formal concepts and Section 3.2 for the realizations). The three approaches operate on business processes, RBAC and EAAs. It is possible to take further domain models of business and IT into account and transfer the generated information to those models for similar analysis purposes. For example, an analysis purpose can be the identification of ACR breaches. Enterprise architecture frameworks have four major domains, business architecture, data architecture, applications architecture and technology architecture. Each of them has a set of domain specific models. On the one hand, information about ACRs could be transferred to other domain models of business architecture and application architecture, for example, system use case diagrams and organizational charts. On the other hand, models of the data architecture seem very promising as changes of data structures have a big impact on ACRs. Research is required to identify the most promising models that would benefit most from an alignment of ACRs. Research questions like *Which domain models of business and IT would benefit most from an alignment of ACRs?*, *How accepted are those models throughout organizations?*, *Is any further information required to conduct an analysis for ACR breaches?* and *How large is the amount of effort imposed on organizations to conduct the analysis?* have to be answered. There are several benefits that would arise from the alignment of ACRs in further domain specific models. First, it might be possible to generate initial models out of the available information and by doing so, easing the engineering process of those models. Second, an identification of ACR breaches would identify errors in an early design phase of those domain specific models. This increases the overall security of an organization. Additionally, this saves costs for organizations, because errors become more expansive the later they are identified throughout a development process [40]. Third, the early identification of ACR breaches becomes especially crucial in evolution scenarios of the domain models, as these evolution scenarios are often complex and error-prone. My research in [25] confirmed that there is a need for holistic modeling of IT security and especially ACRs across models of different domains, to increase the overall security and reduce errors done throughout the evolution of organizations.

Data flow diagrams are another example for domain models that could benefit from a transfer of ACR information. First, it is possible to align the data flow diagram in the same way as AcsALign does with the EAA. A data flow analysis could be made to analyze the data flow diagram for potential violations of ACRs. Afterwards, these violations can be resolved in an early design phase. Second, it is possible to generate an initial data flow diagram from the information extracted out of the business processes. Therefore, not only the ACR information of the business processes is used. This would ease the modeling process of data flow diagrams and keep them aligned with other domain models. In a next step, the initial data flow diagram is completed by a data expert and then can be used to analyze confidentiality aspects of the underlying systems.

I can see that the more information the approaches consume as input, for example, either through more fine-grained or extended models or through additional sources, the more aspects of the models can be aligned with each other automatically. However, depending on the organizational case this increases the effort and thus, has to be considered individually for each organization.

# Bibliography

[1] Allianz Global Corporate & Specialty (AGCS). *Allianz Risk Barometer.* Accessed: May 31, 2021. 2021. URL: https://www.allianz.com/content/dam/onemarketing/azcom/Allianz_com/economic-research/publications/specials/en/2021/january/Allianz-Risk-Barometer-2021.pdf.

[2] American National Standards Institute (ANSI) and InterNational Committee for Information Technology (INCITS). *INCITS 359-2012 - Information technology - Role Based Access Control Standard.* Accessed: May 31, 2021. 2012. URL: https://csrc.nist.gov/projects/role-based-access-control#rbac-standard.

[3] American National Standards Institute (ANSI) and InterNational Committee for Information Technology (INCITS). *INCITS 499-2018 - Information technology - Next Generation Access Control – Functional Architecture (NGAC-FA).* Accessed: May 31, 2021. 2018. URL: https://global.ihs.com/doc_detail.cfm?document_name=ANSI%2FINCITS%20499&item_s_key=00583942.

[4] American National Standards Institute (ANSI) and InterNational Committee for Information Technology (INCITS). *NIST Special Publication 800-162 - Guide to Attribute Based Access Control (ABAC) Definition and Considerations.* Accessed: May 31, 2021. 2014. URL: https://nvlpubs.nist.gov/nistpubs/specialpublications/NIST.SP.800-162.pdf.

[5] Object Management Group (OMG). *Business Process Model and Notation (BPMN) v2.0.2.* Accessed: May 31, 2021. 2011. URL: http://www.omg.org/spec/BPMN.

[6] Object Management Group (OMG). *Unified Modeling Language v2.5.* Accessed: May 31, 2021. 2015. URL: http://www.omg.org/spec/UML/2.5/PDF.

[7] O'Connor A. and Loomis R. *NIST - Economic Analysis of Role-Based Access Control.* Tech. rep. Accessed: May 31, 2021. National Institute of Standards and Technology (NIST), 2010.

[8] Tsolkas A. and Schmidt K. *Rollen und Berechtigungskonzepte: Ansaetze fuer das Identity- und Access Management im Unternehmen.* Vieweg und Teubner Verlag, 2010.

[9] Aerts A.T.M., Goossenaerts J.B.M., Hammer D.K., and Wortmann J.C. "Architectures in context: on the evolution of business, application software, and ICT platform architectures". In: *Information and Management* 41.6 (2004), pp. 781–794.

[10] Jenny Abramov, Omer Anson, Michal Dahan, Peretz Shoval, and Arnon Sturm. "A Methodology for Integrating Access Control Policies within Database Development". In: *Computer Security* 31.3 (2012), pp. 299–314. DOI: 10.1016/j.cose.2012.01.004.

[11] Jenny Abramov, Arnon Sturm, and Peretz Shoval. “Evaluation of the Pattern-Based Method for Secure Development (PbSD): A Controlled Experiment”. In: *Information and Software Technology* 54.9 (2012), pp. 1029–1043. DOI: 10.1016/j.infsof.2012.04.001.

[12] Accenture. *Cost of Cybercrime Study.* Accessed: May 31, 2021. 2019. URL: https://www.accenture.com/_acnmedia/pdf-96/accenture-2019-cost-of-cybercrime-study-final.pdf.

[13] Rafael Accorsi, Andreas Lehmann, and Niels Lohmann. “Information Leak Detection in Business Process Models”. In: *Information Systems* 47.C (2015), pp. 244–257. DOI: 10.1016/j.is.2013.12.006.

[14] Rafael Accorsi and Claus Wonnemann. “InDico: Information Flow Analysis of Business Processes for Confidentiality Requirements”. In: *Security and Trust Management.* Springer Berlin Heidelberg, 2011, pp. 194–209. DOI: 10.1007/978-3-642-22444-7_13.

[15] Gail-Joon Ahn and Hongxin Hu. “Towards Realizing a Formal RBAC Model in Real Systems”. In: *Proceedings of the 12th ACM Symposium on Access Control Models and Technologies.* SACMAT ’07. ACM, 2007, pp. 215–224. DOI: 10.1145/1266840.1266875.

[16] M. Akbarzadeh and M. Abdollahi Azgomi. “A framework for probabilistic model checking of security protocols using coloured stochastic activity networks and PDETool”. In: *5th International Symposium on Telecommunications.* IEEE, 2010, pp. 210–215. DOI: 10.1109/ISTEL.2010.5734026.

[17] M. M. Alam, R. Breu, and M. Breu. “Model driven security for Web services (MDS4WS)”. In: *Proceedings of 8th International Multitopic Conference (INMIC).* IEEE, 2004, pp. 498–505. DOI: 10.1109/INMIC.2004.1492930.

[18] Muhammad Alam, Michael Hafner, and Ruth Breu. “A Constraint Based Role Based Access Control in the SECTET a Model-Driven Approach”. In: *Proceedings of the 2006 International Conference on Privacy, Security and Trust: Bridge the Gap Between PST Technologies and Business Services.* PST ’06. ACM, 2006, pp. 1–13. DOI: 10.1145/1501434.1501451.

[19] Muhammad Alam, Michael Hafner, Ruth Breu, and Stefan Unterthiner. “A Framework for Modeling Restricted Delegation in Service Oriented Architecture”. In: *Trust and Privacy in Digital Business.* Springer Berlin Heidelberg, 2006, pp. 142–151. DOI: 10.1007/11824633_15.

[20] Khaled Alghathbar. “Enhancement of Use Case Diagram to Capture Authorization Requirements”. In: *2009 Fourth International Conference on Software Engineering Advances.* ACM, 2009, pp. 394–400. DOI: 10.1109/ICSEA.2009.63.

[21] Khaled Alghathbar. “Validating the Enforcement of Access Control Policies and Separation of Duty Principle in Requirement Engineering”. In: *Information and Software Technology* 49.2 (2007), pp. 142–157. DOI: 10.1016/j.infsof.2006.03.009.

[22] Khaled Alghathbar and Duminda Wijesekera. “AuthUML: A Three-Phased Framework to Analyze Access Control Specifications in Use Cases”. In: *Proceedings of the 2003 ACM Workshop on Formal Methods in Security Engineering.* FMSE ’03. ACM, 2003, pp. 77–86. DOI: 10.1145/1035429.1035438.

[23] Thomas Allweyer. *BPMN 2.0 - Business Process Model and Notation: Einführung in den Standard für die Geschäftsprozessmodellierung.* BoD – Books on Demand, 2009.

[24] Sascha Alpers, Roman Pilipchuk, Andreas Oberweis, and Ralf Reussner. “Identifying Needs for a Holistic Modelling Approach to Privacy Aspects in Enterprise Software Systems”. In: *Proceedings of the 4th International Conference on Information Systems Security and Privacy (ICISSP).* Vol. 1. 2018, pp. 74–82. DOI: 10.5220/0006606200740082.

[25] Sascha Alpers, Roman Pilipchuk, Andreas Oberweis, and Ralf Reussner. “The Current State of the Holistic Privacy and Security Modelling Approach in Business Process and Software Architecture Modelling”. In: *Information Systems Security and Privacy* (2019). Ed. by Paolo Mori, Steven Furnell, and Olivier Camp, pp. 109–124. DOI: 10.1007/978-3-030-25109-3.

[26] Olga Altuhhov, Raimundas Matulevičius, and Naved Ahmed. “An Extension of Business Process Model and Notation for Security Risk Management”. In: *International Journal of Information System Modeling and Design* 4.4 (2013), pp. 93–113. DOI: 10.4018/ijismd.2013100105.

[27] Olga Altuhhova, Raimundas Matulevičius, and Naved Ahmed. “Towards Definition of Secure Business Processes”. In: *Advanced Information Systems Engineering Workshops.* Springer Berlin Heidelberg, 2012, pp. 1–15. DOI: 10.1007/978-3-642-31069-0_1.

[28] N. Argyropoulos, C. Kalloniatis, H. Mouratidis, and A. Fish. “Incorporating privacy patterns into semi-automatic business process derivation”. In: *IEEE Tenth International Conference on Research Challenges in Information Science (RCIS).* IEEE, June 2016, pp. 1–12. DOI: 10.1109/RCIS.2016.7549305.

[29] Nikolaos Argyropoulos, Haris Mouratidis, and Andrew Fish. *Towards the Derivation of Secure Business Process Designs.* Springer International Publishing, 2015, pp. 248–258. DOI: 10.1007/978-3-319-25747-1_25.

[30] Vijayalakshmi Atluri and Wei-Kuang Huang. “A Petri Net Based Safety Analysis of Workflow Authorization Models”. In: *Journal of Computer Security* 8.2,3 (2000), pp. 209–240. DOI: 10.5555/1297828.1297829.

[31] Vijayalakshmi Atluri and Wei-Kuang Huang. “An Extended Petri Net Model for Supporting Workflows in a Multilevel Secure Environment”. In: *Database Security: Status and prospects.* Springer US, 1997, pp. 240–258. DOI: 10.1007/978-0-387-35167-4_15.

[32] Information Systems Audit and Control Association (ISACA). *COBIT 5.* Accessed: May 31, 2021. 2012. URL: http://www.isaca.org/COBIT/Pages/COBIT-5-Framework-product-page.aspx.

[33] Lerina Aversano, Carmine Grasso, and Maria Tortorella. “Managing the alignment between business processes and software systems”. In: *Information and Software Technology* 72 (2016), pp. 171–188. DOI: 10.1016/j.infsof.2015.12.009.

[34] AXELOS. *ITIL Edition 2011.* Accessed: May 31, 2021. 2011. URL: https://www.axelos.com/best-practice-solutions/itil/what-is-itil.

[35] Mitra B., Sural S., Vaidya J., and Atluri V. “A Survey of Role Mining”. In: *ACM Computing Surveys* 48.4 (2016), pp. 1–37.

[36] Basel Committee on Banking Supervision (BCBS). *Third Basel Accord.* Accessed: May 31, 2021. 2011. URL: https://www.bis.org/publ/bcbs189.pdf.

[37] Joseph Barjis. “The importance of business process modeling in software systems design”. In: *Science of Computer Programming* 71.1 (2008), pp. 73–87. DOI: 10.1016/j.scico.2008.01.002.

[38] David Basin, Manuel Clavel, Jürgen Doser, and Marina Egea. “Automated Analysis of Security-Design Models”. In: *Information and Software Technology* 51.5 (2009), pp. 815–831. DOI: 10.1016/j.infsof.2008.05.011.

[39] T. Basso, L. Montecchi, R. Moraes, M. Jino, and A. Bondavalli. “Towards a UML Profile for Privacy-Aware Applications”. In: *IEEE International Conference on Computer and Information Technology; Ubiquitous Computing and Communications; Dependable, Autonomic and Secure Computing; Pervasive Intelligence and Computing.* IEEE, 2015, pp. 371–378. DOI: 10.1109/CIT/IUCC/DASC/PICOM.2015.53.

[40] Ted Bennett and Paul Wennberg. “Eliminating Embedded Software Defects Prior to Integration Test”. In: *Journal of Defence Software Engineering* (2005), pp. 13–18.

[41] Antonia Bertolino, Marianne Busch, Said Daoudagh, Francesca Lonetti, and Eda Marchetti. “A Toolchain for Designing and Testing Access Control Policies”. In: *Engineering Secure Future Internet Services and Systems: Current Research.* Springer International Publishing, 2014, pp. 266–286. DOI: 10.1007/978-3-319-07452-8_11.

[42] C. Blanco, E. Fernández-Medina, and J. Trujillo. “Modernizing Secure OLAP Applications with a Model-Driven Approach”. In: *The Computer Journal* 58.10 (2015), pp. 2351–2367. DOI: 10.1093/comjnl/bxu070.

[43] Barry Boehm and Victor R. Basili. “Software Defect Reduction Top 10 List”. In: *IEEE Computer* 34.1 (2001), pp. 135–137. DOI: 10.1109/2.962984.

[44] R. Bouaziz, S. Kallel, and B. Coulette. “An Engineering Process for Security Patterns Application in Component Based Models”. In: *IEEE Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises.* IEEE, 2013, pp. 231–236. DOI: 10.1109/WETICE.2013.27.

[45] Roland Bouroulet, Raymond Devillers, Hanna Klaudel, Elisabeth Pelz, and Franck Pommereau. “Modeling and Analysis of Security Protocols Using Role Based Specifications and Petri Nets”. In: *Applications and Theory of Petri Nets.* Springer Berlin Heidelberg, 2008, pp. 72–91. DOI: 10.1007/978-3-540-68746-7_9.

[46] Drazen Brdjanin, Goran Banjac, Danijela Banjac, and Slavko Maric. “Controlled Experiment in Business Model-Driven Conceptual Database Design”. In: *Enterprise, Business-Process and Information Systems Modeling*. Springer International Publishing, 2017, pp. 289–304. DOI: 10.5220/0006852006050616.

[47] Richard Bricknall, Gunilla Darrell, Hans Nilsson, and Kalevi Pessi. “Enterprise architecture: Critical factors affecting modelling and management”. In: *Proceedings of the 14th European Conference on Information Systems (ECIS)* (2006), pp. 2349–2361.

[48] Achim D. Brucker and Isabelle Hang. “Secure and Compliant Implementation of Business Process-Driven Systems”. In: *Business Process Management Workshops*. Springer Berlin Heidelberg, 2013, pp. 662–674. DOI: 10.1007/978-3-642-36285-9_66.

[49] Achim D. Brucker, Isabelle Hang, Gero Lückemeyer, and Raj Ruparel. “SecureBPMN: Modeling and Enforcing Access Control Requirements in Business Processes”. In: *ACM Symposium on Access Control Models and Technologies (SACMAT)*. ACM, 2012, pp. 123–126.

[50] Tobias Bucher, Ronny Fischer, Stephan Kurpjuweit, and Robert Winter. “Enterprise Architecture Analysis and Application - An Exploratory Study”. In: *Journal of Enterprise Architecture* 3.3 (Mar. 2007), pp. 33–43.

[51] S. Buckl, Alexander M Ernst, et al. *Enterprise Architecture Management Pattern Catalog (Version 1.0)*. Tech. rep. Accessed: May 31, 2021. Chair for Informatics 19 (sebis), Technical University of Munich, 2008. URL: https://wwwmatthes.in.tum.de/pages/ugsyi19wmmvl/EAMPC-V2-Enterprise-Architecture-Management-Pattern-Catalog-V2.

[52] S. Buckl, F. Matthes, and C. M. Schweda. “A viable system perspective on enterprise architecture management”. In: *IEEE International Conference on Systems, Man and Cybernetics*. IEEE, 2009, pp. 1483–1488. DOI: 10.1109/ICSMC.2009.5346262.

[53] S. Buckl, C. M. Schweda, and F. Matthes. “A Design Theory Nexus for Situational Enterprise Architecture Management”. In: *14th IEEE International Enterprise Distributed Object Computing Conference Workshops*. IEEE, 2010, pp. 3–8. DOI: 10.1109/EDOCW.2010.27.

[54] Sabine Buckl, F. Matthes, and Christian M. Schweda. “Towards a Method Framework for Enterprise Architecture Management – A Literature Analysis from a Viable System Perspective”. In: *5th International Workshop on Business/IT Alignment and Interoperability (BUSITAL)*. CiteSeerx, 2010.

[55] S. E. Bussells. “Assessment of a Government Agency’s Enterprise Architecture Program”. In: *Journal of Enterprise Architecture* 2.1 (2006), pp. 43–50.

[56] Camilo Castellanos and Dario Correal. “KALCAS: A FrameworK for Semi-automatic ALignment of Data and Business ProCesses ArchitectureS”. In: *Advances in Databases and Information Systems*. Springer Berlin Heidelberg, 2012, pp. 111–124. DOI: 10.1145/1363686.1363820.

[57] Ramaswamy Chandramouli. “Framework for defining an Access Control Service for Healthcare Information System Using Roles”. In: *Proceedings 4th ACM Workshop on Role-Based Access Control.* ACM, 1999, pp. 135–140.

[58] Michele Chinosi and Alberto Trombetta. “Integrating Privacy Policies into Business Processes”. In: *Journal of Research and Practice in Information Technology* 41.2 (Jan. 2008), pp. 155–170.

[59] Alessandro Colantonio, Roberto Di Pietro, and Alberto Ocello. “A Cost-Driven Approach to Role Engineering”. In: *Proceedings of the 2008 ACM Symposium on Applied Computing.* SAC ’08. ACM, 2008, pp. 2129–2136. DOI: 10.1145/1363686.1364198.

[60] Alessandro Colantonio, Roberto Di Pietro, Alberto Ocello, and Nino Vincenzo Verde. “A Formal Framework to Elicit Roles with Business Meaning in RBAC Systems”. In: *Proceedings of the 14th ACM Symposium on Access Control Models and Technologies.* ACM, 2009, pp. 85–94.

[61] Alessandro Colantonio, Roberto Di Pietro, and Nino Vincenzo Verde. “A Business-Driven Decomposition Methodology for Role Mining”. In: *Computers and Security* 31.7 (2012), pp. 844–855. DOI: 10.1016/j.cose.2012.01.005.

[62] Alessandro Colantonio, Roberto Di Pietro, and Alberto Ocello. “Leveraging Lattices to Improve Role Mining”. In: *Proceedings of the 23rd International Information Security Conference (IFIP).* Springer US, 2008, pp. 333–347. DOI: 10.1007/978-0-387-09699-5_22.

[63] Federico Crazzolara and Glynn Winskel. “Events in Security Protocols”. In: *Proceedings of the 8th ACM Conference on Computer and Communications Security.* CCS ’01. ACM, 2001, pp. 96–105. DOI: 10.1145/501983.501998.

[64] Estrela Cruz and Antonio Miguel Rosado da Cruz. “Deriving Integrated Software Design Models from BPMN Business Process Models”. In: *Procedings of the International Conference on Software Technologies (ICSOFT).* ScitePress, July 2018, pp. 605–616. DOI: 10.5220/0006852006050616.

[65] Ferraiolo D. F., Sandhu R., Gavrila S., Kuhn D. R., and Chandramouli R. “Proposed NIST Standard for Role-Based Access Control”. In: *ACM Trans. Inf. Syst. Secur.* 4.3 (2001), pp. 224–274.

[66] F. Ferraiolo D. et al. *Role-Based Access Control.* Artech House Publishers, 2007.

[67] Ferraiolo D.F. and Kuhn D.R. “Role-Based Access Controls”. In: *15th National Computer Security Conference.* NIST, 1992, pp. 554–563.

[68] Julio Damasceno, Bruno Silva, Robson Medeiros, Fernando Lins, Nelson Rosa, Paulo Maciel, Bryan Stephenson, Hamid R. Motahari Nezhad, Jun Li, and Caio Northfleet. “Towards Generating Richer Code by Binding Security Abstractions to BPMN Task Types”. In: *Revista de Informática Teorica e Aplicada (RITA)* Volume 16 (Jan. 2009), pp. 97–98. DOI: 10.22456/2175-2745.12581.

[69] U.S. Department of Defense. *Department of Defense Architecture Framework (DoDAF) v.2.02.* Accessed: May 31, 2021. 2003. URL: https://dodcio.defense.gov/Library/DoD-Architecture-Framework/dodaf20_background/.

[70] G. Dhillon. *Information Security Management: Global Challenges in the New Millennium.* IGI Publishing, 2001.

[71] Lijun Dong, Jinxia Wu, Cheng Gong, and Benjie Pi. “A Network-Cliques Based Role Mining Model”. In: *Journal of Networks* 9.8 (2014), pp. 2079–2088. DOI: 10.4304/jnw.9.8.2079-2088.

[72] Coyne Edward J. “Role engineering”. In: *Proceedings of the first ACM Workshop on Role-based access control (RBAC ’95).* ACM, 1996, pp. 4–6.

[73] C. Emig, F. Brandt, S. Abeck, J. Biermann, and H. Klarl. “An Access Control Metamodel for Web Service-Oriented Architecture”. In: *IEEE International Conference on Software Engineering Advances (ICSEA 2007).* IEEE, 2007, pp. 57–57. DOI: 10.1109/ICSEA.2007.15.

[74] Christian Emig, Sebastian Kreuzer, Sebastian Abeck, Jürgen Biermann, and Heiko Klarl. “Model-Driven Development of Access Control Policies for Web Services”. In: *Proceedings of the 9th IASTED International Conference Software Engineering and Applications.* IEEE, 2008, pp. 165–171.

[75] EmpoweredID. *Best Practices in Enterprise Authorization: The RBAC/ABAC Hybrid Approach v3.* Tech. rep. 2018. 18 pp.

[76] Pete Epstein. “Engineering of Role/Permission Assignments”. dissertation. George Mason University, 2002.

[77] Pete Epstein and Ravi Sandhu. “Engineering of RolePermission Assingment”. In: *Proceedings Of the 17th Annual Computer Security Application Conference (ACSAC).* IEEE, 2001, pp. 127–136.

[78] Pete Epstein and Ravi Sandhu. “Towards a UML Based Approach to Role Engineering”. In: *Proceedings of the Fourth ACM Workshop on Role-Based Access Control.* RBAC ’99. ACM, 1999, pp. 135–143. DOI: 10.1145/319171.319184.

[79] Eduardo Fernandez-Medina, Juan Trujillo, Rodolfo Villarroel, and Mario Piattini. “Extending UML for Designing Secure Data Warehouses”. In: *Conceptual Modeling – ER 2004.* Springer Berlin Heidelberg, 2004, pp. 217–230. DOI: 10.1007/978-3-540-30464-7_18.

[80] E. B. Fernandez and J. C. Hawkins. “Determining Role Rights from Use Cases”. In: *Proceedings of the Second ACM Workshop on Role-Based Access Control.* RBAC ’97. ACM, 1997, pp. 121–125. DOI: 10.1145/266741.266767.

[81] Eduardo Fernandez, María Larrondo Petrie, Tami Sorgente, and Michael Van Hilst. “A Methodology to Develop Secure Systems Using Patterns”. In: *Integrating Security an software engineering: advances and future vision.* IDEA Press, 2006. DOI: 10.4018/9781599041476.ch005.

[82] German Federal Ministry of Finance. *Principles for Properly Maintaing and Storing Books, Records and Documents in Electronic Form and for Data Access (GoBD).* Accessed: May 31, 2021. 2014. URL: `https://www.bundesfinanzministerium.de/Content/DE/Downloads/BMF_Schreiben/Weitere_Steuerthemen/Abgabenordnung/2019-11-28-GoBD.html`.

[83] Mario Frank, Joachim M. Buhmann, and David Basin. "On the Definition of Role Mining". In: *Proceedings of the 15th ACM Symposium on Access Control Models and Technologies.* SACMAT '10. ACM, 2010, pp. 35–44.

[84] L. Fuchs and G. Pernul. "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).* ACM, 2007, pp. 374–384.

[85] Ludwig Fuchs and Günther Pernul. "HyDRo – Hybrid Development of Roles". In: *Information Systems Security.* Springer Berlin Heidelberg, 2008, pp. 287–302.

[86] E Gamma, R Helm, R Johnson, and J. Vlissides. *Design patterns: elements of reusable object-oriented software.* Addison-Wesley Professional, 1995.

[87] C. Gerking and D. Schubert. "Component-Based Refinement and Verification of Information-Flow Security Policies for Cyber-Physical Microservice Architectures". In: *IEEE International Conference on Software Architecture (ICSA).* IEEE, 2019, pp. 61–70. DOI: `10.1109/ICSA.2019.00015`.

[88] Federal Financial Supervisory Authority (BaFin) of Germany. *Minimum Requirements for Risk Management.* Accessed: March 22, 2018. 2017. URL: `https://www.bundesbank.de/resource/blob/623102/bca5bafd72a669115b15c4125e063feb/mL/minimum-requirements-for-risk-management-mindestanforderungen-an-das-risikomanagement-marisk-data.pdf`.

[89] Federal Republic of Germany. *IT Security Act.* Accessed: May 31, 2021. 2015. URL: `https://www.bgbl.de/xaver/bgbl/start.xav?startbk=Bundesanzeiger_BGBl&start=//*%255B@attr_id=%27bgbl115s1324.pdf%27%255D#__bgbl__%2F%2F*%5B%40attr_id%3D%27bgbl115s1324.pdf%27%5D__1521538190954`.

[90] George M. Giaglis. "A Taxonomy of Business Process Modeling and Information Systems Modeling Techniques". In: *International Journal of Flexible Manufacturing Systems* 13.2 (Apr. 2001), pp. 209–228.

[91] Secorvo Security Consulting GmbH. *Zentrale Bausteine der Informationssicherheit.* Web-Site-Verlag, 2014.

[92] Hassan Gomaa and Michael E. Shin. "Separating Application and Security Concerns in Use Case Models". In: *Proceedings of the 15th Workshop on Early Aspects.* EA '09. ACM, 2009, pp. 1–6. DOI: `10.1145/1509825.1509827`.

[93] W. Goudalo and D. Seret. "Toward the Engineering of Security of Information Systems (ESIS): UML and the IS Confidentiality". In: *Second IEEE International Conference on Emerging Security Information, Systems and Technologies.* IEEE, 2008, pp. 248–256. DOI: `10.1109/SECURWARE.2008.66`.

[94] U.S. Federal Government. *FEA Consolidated Reference Model Document (FEAF) v2.3.* Accessed: May 31, 2021. 2007. URL: https://www.reginfo.gov/public/jsp/Utilities/FEA_CRM_v23_Final_Oct_2007_Revised.pdf.

[95] Danny Greefhorst. *Architecture Principles: The Cornerstones of Enterprise Architecture.* 2011.

[96] The Open Group. *The Open Group Architecture Framework (TOGAF) v9.2.* Accessed: May 31, 2021. 2018. URL: http://pubs.opengroup.org/architecture/togaf9-doc/arch/.

[97] Q. Guo, J. Vaidya, and V. Atluri. "The Role Hierarchy Mining Problem: Discovery of Optimal Role Hierarchies". In: *Annual Computer Security Applications Conference (ACSAC).* IEEE, 2008, pp. 237–246. DOI: 10.1109/ACSAC.2008.38.

[98] Roeckle H., Schimpf G., and Weidinger R. "Process-oriented approach for role-finding to implement role-based security administration in a large industrial organization". In: *Proceedings of the fifth ACM workshop on Role-based access control (RBAC '00).* ACM, 2000, pp. 103–110.

[99] M. Hafner and R. Winter. "Processes for Enterprise Application Architecture Management". In: *Proceedings of the 41st Annual Hawaii International Conference on System Sciences (HICSS 2008).* IEEE, 2008, pp. 396–396. DOI: 10.1109/HICSS.2008.362.

[100] Michael Hafner and Ruth Breu. *Security Engineering for Service-Oriented Architectures.* Springer, 2008.

[101] Joel Haight. "Automated Control Systems Do They Reduce Human Error And Incidents?" In: *Professional safety* 52 (2007), pp. 20–27.

[102] Denis Hatebur and Maritta Heisel. "A UML Profile for Requirements Analysis of Dependable Software". In: *Computer Safety, Reliability, and Security.* Springer Berlin Heidelberg, 2010, pp. 317–331. DOI: 10.1007/978-3-642-15651-9_24.

[103] U.S. Department of Health and Human Services. *Health Insurance Portability and Accountability Act (HIPAA).* Accessed: May 31, 2021. 2015. URL: https://www.hhs.gov/hipaa.

[104] R. Heinrich et al. *The CoCoME Platform for Collaborative Empirical Research on Information System Evolution.* Tech. rep. 2. Karlsruhe Institute of Technology, 2016. 57 pp.

[105] Robert Heinrich, Philipp Merkle, Jörg Henß, and Barbara Paech. "Integrating business process simulation and information system simulation for performance prediction". In: *Software Engineering 2016.* Gesellschaft für Informatik e.V., 2016, pp. 51–52.

[106] Rogardt Heldal, S. Schlager, and Jakob Bende. "Supporting Confidentiality in UML: A Profile for the Decentralized Label Model". In: *Proceeding Workshop on Critical Systems Development with UML.* Chalmers, 2004, pp. 56–70.

[107] M. H. Hernandez, J. A. Laredo, S. Mandala, Y. Ruan, V. C. Sreedhar, and M. Vukovic. *System and Method for Hybrid Role Mining*. Tech. rep. Accessed: May 31, 2021. United States Patent, 2014, pp. 1–10. URL: https://patents.google.com/patent/US8635689B2/en.

[108] Sebastian Herold, Holger Klus, Yannick Welsch, Constanze Deiters, Andreas Rausch, Ralf Reussner, Klaus Krogmann, Heiko Koziolek, Raffaela Mirandola, Benjamin Hummel, Michael Meisinger, and Christian Pfaller. "CoCoME - The Common Component Modeling Example". In: *The Common Component Modeling Example: Comparing Software Component Models*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, pp. 16–53. DOI: 10.1007/978-3-540-85289-6_3.

[109] T. Heyman, R. Scandariato, and W. Joosen. "Reusable Formal Models for Secure Software Architectures". In: *Joint Working IEEE/IFIP Conference on Software Architecture and European Conference on Software Architecture*. IEEE, 2012, pp. 41–50. DOI: 10.1109/WICSA-ECSA.212.12.

[110] K. Hjort-Madsen and J. Pries-Heje. "Enterprise Architecture in Government: Fad or Future?" In: *42nd Hawaii International Conference on System Sciences*. IEEE, 2009, pp. 1–10. DOI: 10.1109/HICSS.2009.194.

[111] Jan Hoogervorst. *Enterprise Governance and Enterprise Engineering*. Springer, 2009. DOI: 10.1007/978-3-540-92671-9.

[112] Siv Hilde Houmb and Kine Kvernstad Hansen. "Towards a UML profile for Security Assessment". In: *Workshop on Critical Systems Development with UML*. CiteSeerx, 2003, pp. 1–9.

[113] Hejiao Huang and Helene Kirchner. "Secure interoperation design in multi-domains environments based on colored Petri nets". In: *Information Sciences* 221 (2013), pp. 591–606. DOI: 10.1016/j.ins.2012.09.027.

[114] Gartner Inc. *Gartner Enterprise Architecture Framework*. Accessed: May 31, 2021. 2003. URL: https://www.gartner.com/en/documents/405453.

[115] Gartner Inc. *Gartner Identifies Ten Enterprise Architecture Pitfalls*. Accessed: May 31, 2021. 2009. URL: https://www.gartner.com/newsroom/id/1159617.

[116] German Federal Office for Information Security (BSI). *BSI Lagebericht zur IT-Sicherheit in Deutschland*. Accessed: May 31, 2021. 2020. URL: https://www.bsi.bund.de/SharedDocs/Downloads/DE/BSI/Publikationen/Lageberichte/Lagebericht2020.pdf?__blob=publicationFile&v=1.

[117] German Federal Office for Information Security. *IT Baseline Protection*. Accessed: May 31, 2021. 2006. URL: https://www.bsi.bund.de/EN/Topics/ITGrundschutz/itgrundschutz_node.html.

[118] SABSA Institute. *Sherwood Applied Business Security Architecture (SABSA)*. Accessed: May 31, 2021. 2009. URL: https://sabsa.org/sabsa-executive-summary/.

[119] Trent Jaeger, Frederique Giraud, Nayeem Islam, and Jochen Liedtke. “A Role-Based Access Control Model for Protection Domain Derivation and Management”. In: *Proceedings of the Second ACM Workshop on Role-Based Access Control.* RBAC ’97. ACM, 1997, pp. 95–106. DOI: 10.1145/266741.266764.

[120] Stefan Jakoubi, Simon Tjoa, Gernot Goluch, and Gerald Quirchmayr. “A Survey of Scientific Approaches Considering the Integration of Security and Risk Aspects into Business Process Management”. In: *Proceedings of the 2009 20th International Workshop on Database and Expert Systems Application.* DEXA ’09. IEEE, 2009, pp. 127–132. DOI: 10.1109/DEXA.2009.71. URL: http://dx.doi.org/10.1109/DEXA.2009.71.

[121] M. Janssen and K. Hjort-Madsen. “Analyzing Enterprise Architecture in National Governments: The Cases of Denmark and the Netherlands”. In: *40th Annual IEEE Hawaii International Conference on System Sciences (HICSS’07).* IEEE, 2007, 218a–218a. DOI: 10.1109/HICSS.2007.79.

[122] Xin Jin, Ram Krishnan, and Ravi Sandhu. “A Role-Based Administration Model for Attributes”. In: *Proceedings of the First International Workshop on Secure and Resilient Architectures and Systems.* SRAS ’12. ACM, 2012, pp. 7–12. DOI: 10.1145/2420936.2420938.

[123] Jim Johnson and Hans Mulder. *Rule of Ten.* Tech. rep. Accessed: May 31, 2021. The Standish Group International, Jan. 2014, pp. 1–4. DOI: 10.13140/RG.2.2.36689.28004. URL: https://www.standishgroup.com/sample_research_files/RuleTen.pdf.

[124] Jan Jürjens. “Model-Based Security Engineering with UML”. In: *Foundations of Security Analysis and Design III: FOSAD 2004/2005 Tutorial Lectures.* Ed. by Alessandro Aldini, Roberto Gorrieri, and Fabio Martinelli. Springer Berlin Heidelberg, 2005, pp. 42–77. DOI: 10.1007/11554578_2.

[125] Jan Jürjens. “UMLsec: Extending UML for Secure Systems Development”. In: *Proceedings of the 5th International Conference on The Unified Modeling Language.* Springer Berlin Heidelberg, 2002, pp. 412–425. DOI: 10.1007/3-540-45800-X_32.

[126] D. N. Jutla, P. Bodorik, and S. Ali. “Engineering Privacy for Big Data Apps with the Unified Modeling Language”. In: *IEEE International Congress on Big Data.* IEEE, 2013, pp. 38–45. DOI: 10.1109/BigData.Congress.2013.15.

[127] Axel Kern. “Advanced features for enterprise-wide role-based access control”. In: *Proceedings of the 18th Annual Computer Security Applications Conference.* IEEE, 2002, pp. 333–342. DOI: 10.1109/CSAC.2002.1176305.

[128] Axel Kern, Martin Kuhlmann, Andreas Schaad, and Jonathan Moffett. “Observations on the Role Life-Cycle in the Context of Enterprise Security Management”. In: *Proceedings of the Seventh ACM Symposium on Access Control Models and Technologies.* SACMAT ’02. ACM, 2002, pp. 43–51. DOI: 10.1145/507711.507718.

[129] Axel Kern, Martin Kuhlmann, Andreas Schaad, and Jonathan Moffett. “Observations on the Role Life-Cycle in the Context of Enterprise Security Management”. In: *Proceedings of the Seventh ACM Symposium on Access Control Models and Technologies.* SACMAT ’02. ACM, 2002, pp. 43–51. DOI: 10.1145/507711.507718.

[130] Wiem Khlif, Nourchene Elleuch Ben Ayed, and Hanêne Ben-Abdallah. “From a BPMN Model to an Aligned UML Analysis Model”. In: *Proceedings of the 13th International Conference on Software Technologies (ICSOFT).* ScitePress, 2018, pp. 623–631. DOI: 10.5220/0006866606230631.

[131] D. Kim and P. Gokhale. “A Pattern-Based Technique for Developing UML Models of Access Control Systems”. In: *30th Annual IEEE International Computer Software and Applications Conference (COMPSAC’06).* Vol. 1. IEEE, 2006, pp. 317–324. DOI: 10.1109/COMPSAC.2006.14.

[132] Dae-Kyoo Kim, Indrakshi Ray, Robert France, and Na Li. “Modeling Role-Based Access Control Using Parameterized UML Models”. In: *Fundamental Approaches to Software Engineering.* Springer Berlin Heidelberg, 2004, pp. 180–193. DOI: 10.1007/978-3-540-24721-0_13.

[133] Sangsig Kim, Dae-Kyoo Kim, Lunjin Lu, Suntae Kim, and Sooyong Park. “A feature-based approach for modeling role-based access control systems”. In: *Journal of Systems and Software* 84.12 (2011), pp. 2035–2052. DOI: 10.1016/j.jss.2011.03.084.

[134] H. Klarl, K. Molitorisz, C. Emig, K. Klinger, and S. Abeck. “Extending Role-Based Access Control for Business Usage”. In: *Third IEEE International Conference on Emerging Security Information, Systems and Technologies.* IEEE, 2009, pp. 136–141. DOI: 10.1109/SECURWARE.2009.28.

[135] Heiko Klarl. *Zugriffskontrolle in Geschäftsprozessen - Ein modellgetriebener Ansatz.* Vieweg und Teubner Verlag, 2011. DOI: 10.1007/978-3-8348-9913-2.

[136] Heiko Klarl, Christian Wolff, and Christian Emig. “Identity Management in Business Process Modelling: A Model-Driven Approach”. In: *9. Internationale Tagung Wirtschaftsinformatik (WI2009).* 2009, pp. 1–10.

[137] Tobias Knopf. “Analyse eines Meta-Modells zur Abbildung von BPMN 2.0 im Kontext der Extraktion von Zugriffsberechtigungen”. PhD thesis. University of Applied Siences (HTW Berlin), 2018.

[138] Konstantin Knorr. “Multilevel Security and Information Flow in Petri Net Workflows”. In: *Proceedings of the 9th International Conference on Telecommunication Systems - Modeling and Analysis, Special Session on Security Aspects of Telecommunication Systems.* Springer, 2001, pp. 1–16.

[139] Manuel Koch and Francesco Parisi-Presicce. “UML specification of access control policies and their formal verification”. In: *Software and System Modeling* 5.4 (2006), pp. 429–447. DOI: 10.1007/s10270-006-0030.

[140] S. Kotusev. “Critical Questions in Enterprise Architecture Research”. In: *International Journal of Enterprise Information Systems (IJEIS)* 13.2 (2017), pp. 50–62.

[141] G. Kreizman and B. Robertson. *Incorporating Security into the Enterprise Architecture Process.* Tech. rep. Gartner, Inc., 2006.

[142] Martin Kuhlmann, Dalia Shohat, and Gerhard Schimpf. "Role Mining - Revealing Business Roles for Security Administration Using Data Mining Technology". In: *Proceedings of the 8th ACM Symposium on Access Control Models and Technologies.* SACMAT '03. ACM, 2003, pp. 179–186. DOI: 10.1145/775412.775435.

[143] Wadha Labda, Nikolay Mehandjiev, and Pedro Sampaio. "Modeling of Privacy-Aware Business Processes in BPMN to Protect Personal Data". In: *Proceedings of the 29th Annual ACM Symposium on Applied Computing.* SAC '14. ACM, 2014, pp. 1399–1405. DOI: 10.1145/2554850.2555014.

[144] H. F. Lai, J. L. Hong, and W. H. Jeng. "Model E-contract Update by Coloured Activity Net". In: *IEEE Asia-Pacific Services Computing Conference.* IEEE, 2008, pp. 488–493. DOI: 10.1109/APSCC.2008.191.

[145] HyungHyo Lee, YoungLok Lee, and BongNam Noh. "A Framework for Modeling Organization Structure in Role Engineering". In: *Applied Parallel Computing. State of the Art in Scientific Computing: 7th International Workshop, PARA 2004.* Springer, 2004, pp. 1017–1024.

[146] Maria Leitner, Michelle Miller, and Stefanie Rinderle-Ma. "An Analysis and Evaluation of Security Aspects in the Business Process Model and Notation". In: *Proceedings of the International Conference on Availability, Reliability and Security (ARES).* ACM, Sept. 2013, pp. 262–267. DOI: 10.1109/ARES.2013.34.

[147] W. Li, R. Wu, and H. Huang. "Colored Petri Nets Based Modeling of Information Flow Security". In: *Second International Workshop on Knowledge Discovery and Data Mining.* IEEE, 2009, pp. 681–684. DOI: 10.1109/WKDD.2009.171.

[148] Torsten Lodderstedt, David Basin, and Jürgen Doser. "SecureUML: A UML-Based Modeling Language for Model-Driven Security". In: *The unified Modeling Language Volume 2460 of the series Lecture Notes in Computer Science.* Springer Berlin Heidelberg, 2002, pp. 426–441. DOI: 10.1007/3-540-45800-X_33.

[149] Jan Löhe and Christine Legner. "Overcoming implementation challenges in enterprise architecture management: a design theory for architecture-driven IT Management (ADRIMA)". In: *Information Systems and e-Business Management* 12.1 (Feb. 2014), pp. 101–137.

[150] Haibing Lu, Yuan Hong, Yanjiang Yang, Lian Duan, and Nazia Badar. "Towards User-Oriented RBAC Model". In: *Data and Applications Security and Privacy XXVII.* Springer Berlin Heidelberg, 2013, pp. 81–96. DOI: 10.1007/978-3-642-39256-6_6.

[151] Gallaher M., O'Connor A., Kropp B., and Tassey G. *Planning Report 02-1: The Economic Impact of Role-Based Access Control.* Tech. rep. National Institute of Standards and Technology (NIST), 2002.

[152] Narouei M. and Takabi H. “Towards an Automatic Top-down Role Engineering Approach Using Natural Language Processing Techniques”. In: *Proceedings of the 20th ACM Symposium on Access Control Models and Technologies (SACMAT ’15).* ACM, 2015, pp. 157–160.

[153] Xiaopu Ma, Ruixuan Li, and Zhengding Lu. “Role Mining Based on Weights”. In: *Proceedings of the 15th ACM Symposium on Access Control Models and Technologies.* SACMAT ’10. ACM, 2010, pp. 65–74. DOI: 10.1145/1809842.1809854.

[154] S. Mandala, M. Vukovic, J. Laredo, Y. Ruan, and M. Hernandez. “Hybrid Role Mining for Security Service Solution”. In: *Ninth IEEE International Conference on Services Computing.* IEEE, 2012, pp. 210–217. DOI: 10.1109/SCC.2012.57.

[155] P. H. Meland and E. A. Gjaere. “Representing Threats in BPMN 2.0”. In: *Seventh International Conference on Availability, Reliability and Security (ARES).* ACM, 2012, pp. 542–550. DOI: 10.1109/ARES.2012.13.

[156] M. Menzel, I. Thomas, and C. Meinel. “Security Requirements Specification in Service-Oriented Business Process Management”. In: *IEEE International Conference on Availability, Reliability and Security (ARES).* IEEE, 2009, pp. 41–48. DOI: 10.1109/ARES.2009.90.

[157] Microsoft. *The STRIDE Threat Model.* Accessed: May 31, 2021. 2005. URL: https://docs.microsoft.com/en-us/previous-versions/commerce-server/ee823878(v=cs.20)?redirectedfrom=MSDN.

[158] Hafedh Mili, Guy Tremblay, Guitta Bou Jaoude, Eric Lefebvre, Lamia Elabed, and Ghizlane El Boussaidi. “Business Process Modeling Languages: Sorting Through the Alphabet Soup”. In: *ACM Comput. Surv.* 43.1 (Dec. 2010), 4:1–4:56. DOI: 10.1145/1824795.1824799.

[159] Ian Molloy, Hong Chen, Tiancheng Li, Qihua Wang, Ninghui Li, Elisa Bertino, Seraphin Calo, and Jorge Lobo. “Mining Roles with Multiple Objectives”. In: *ACM Transactions on Information and System Security* 13.4 (2010). DOI: 10.1145/1880022.1880030.

[160] Ian Molloy, Hong Chen, Tiancheng Li, Qihua Wang, Ninghui Li, Elisa Bertino, Seraphin Calo, and Jorge Lobo. “Mining Roles with Semantic Meanings”. In: *Proceedings of the 13th ACM Symposium on Access Control Models and Technologies.* SACMAT ’08. ACM, 2008, pp. 21–30. DOI: 10.1145/1377836.1377840.

[161] Ganna Monakova, Achim D. Brucker, and Andreas Schaad. “Security and Safety of Assets in Business Processes”. In: *Proceedings of the 27th Annual ACM Symposium on Applied Computing.* SAC ’12. Trento, Italy: ACM, 2012, pp. 1667–1673. DOI: 10.1145/2245276.2232045.

[162] Christoph Moser, Stefan Junginger, Matthias Brückmann, and Klaus-Manfred Schöne. “Some Process Patterns for Enterprise Architecture Management”. In: *Fachtagung des GI-Fachbereichs Softwaretechnik.* Gesellschaft für Informatik, Jan. 2009, pp. 19–30.

[163] Tejeddine Mouelhi, Franck Fleurey, Benoit Baudry, and Yves Le Traon. "A Model-Based Framework for Security Policy Specification, Deployment and Testing". In: *Model Driven Engineering Languages and Systems*. Springer Berlin Heidelberg, 2008, pp. 537–552. DOI: 10.1007/978-3-540-87875-9_38.

[164] D. Mouheb, C. Talhi, M. Nouh, V. Lima, M. Debbabi, L. Wang, and M. Pourzandi. "Aspect-Oriented Modeling for Representing and Integrating Security Concerns in UML". In: *Software Engineering Research, Management and Applications 2010*. Springer Berlin Heidelberg, 2010, pp. 197–213. DOI: 10.1007/978-3-642-13273-5_13.

[165] Djedjiga Mouheb, Chamseddine Talhi, Vitor Lima, Mourad Debbabi, Lingyu Wang, and Makan Pourzandi. "Weaving Security Aspects into UML 2.0 Design Models". In: *Proceedings of the 13th Workshop on Aspect-Oriented Modeling*. AOM '09. ACM, 2009, pp. 7–12. DOI: 10.1145/1509297.1509300.

[166] Haralambos Mouratidis and Paolo Giorgini. "Secure Tropos: A Security-Oriented Extension of the Tropos Methodology". In: *International Journal of Software Engineering and Knowledge Engineering* 17.02 (2007), pp. 285–309. DOI: 10.1142/S0218194007003240.

[167] Haralambos Mouratidis and Jan Jurjens. "From Goal-Driven Security Requirements Engineering to Secure Design". In: *International Journal on Intelligent Systems* 25.8 (2010), pp. 813–840. DOI: 10.5555/1841349.1841355.

[168] Jutta A. Mülle, Silvia von Stackelberg, and Klemens Böhm. *A Security Language for BPMN Process Models*. Tech. rep. 9. Karlsruhe Institute of Technology, 2011, pp. 1–23.

[169] Jutta Mulle, Silvia von Stackelberg, and Klemens Bohm. "Modelling and Transforming Security Constraints in Privacy-aware Business Processes". In: *Proceedings of the 2011 IEEE International Conference on Service-Oriented Computing and Applications*. SOCA '11. IEEE, 2011, pp. 1–4. DOI: 10.1109/SOCA.2011.6166257. URL: http://dx.doi.org/10.1109/SOCA.2011.6166257.

[170] T. Murata. "Petri Nets: Properties, Analysis and Applications". In: *IEEE Trans Reliab* 51 (Jan. 1989), pp. 541–580.

[171] Y. Nakamura, M. Tatsubori, T. Imamura, and K. Ono. "Model-driven security based on a Web services security architecture". In: *IEEE International Conference on Services Computing (SCC'05)*. Vol. 1. IEEE, 2005, pp. 7–15. DOI: 10.1109/SCC.2005.66.

[172] Gustaf Neumann and Mark Strembeck. "A Scenario-Driven Role Engineering Process for Functional RBAC Roles". In: *Proceedings of the Seventh ACM Symposium on Access Control Models and Technologies*. ACM, 2002, pp. 33–42.

[173] Klaus D. Niemann. *Von der Unternehmensarchitektur zur IT-Governance*. Springer Vieweg, 2005.

[174] Eetu Niemi. "Enterprise Architecture Stakeholders - a Holistic View". In: *Americas Conference on Information Systems (AMCIS)*. AIS eLibrary, Jan. 2007, p. 41.

[175] Jaime A. Pavlich-Mariscal, Steven A. Demurjian, and Laurent D. Michel. "A Framework of Composable Access Control Features: Preserving Separation of Access Control Concerns from Models to Code". In: *Computer Security* 29.3 (2010), pp. 350–379. DOI: 10.1016/j.cose.2009.11.005.

[176] Roman Pilipchuk. "Coping with Access Control Requirements in the Context of Mutual Dependencies between Business and IT". In: *Proceedings of the ACM Central European Cybersecurity Conference 2018.* CECC'18. ACM Association for Computing Machinery, 2018, 16:1–16:4. DOI: 10.1145/3277570.3277587.

[177] Roman Pilipchuk, Robert Heinrich, and Ralf Reussner. "Automatically Extracting Business Level Access Control Requirements from BPMN Models to Align RBAC Policies". In: *Proceedings of the 7th International Conference on Information Systems Security and Privacy (ICISSP).* Vol. 1. ScitePress, 2021, pp. 300–307. DOI: 10.5220/0010184403000307.

[178] Roman Pilipchuk, Stephan Seifermann, and Robert Heinrich. "Aligning Business Process Access Control Policies with Enterprise Architecture". In: *Proceedings of the ACM Central European Cybersecurity Conference 2018.* CECC'18. ACM Association for Computing Machinery, 2018, 17:1–17:4. DOI: 10.1145/3277570.3277588.

[179] Roman Pilipchuk, Stephan Seifermann, Robert Heinrich, and Ralf Reussner. "Challenges in Aligning Enterprise Application Architectures to Business Process Access Control Requirements in Evolutional Changes". In: *Proceedings of the 18th International Conference on e-Business (ICE-B).* ScitePress, 2021, pp. 13–24. DOI: 10.5220/0010511800130024.

[180] Roman Pilipchuk, Stephan Seifermann, and Emre Taspolatoglu. "Defining a Security-Oriented Evolution Scenario for the CoCoME Case Study". In: *4nd Collaborative Workshop on Evolution and Maintenance of Long-Living Software Systems (EMLS'17).* Vol. 37. Softwaretechnik Trends 2. 2017, pp. 70–73.

[181] Victor R. Basili, Gianluigi Caldiera, and Dieter Rombach. "The goal question metric approach". In: *Encyclopedia of Software Engineering* 1 (1994). Accessed: May 31, 2021, pp. 1–10. URL: https://www.cs.umd.edu/users/mvz/handouts/gqm.pdf.

[182] Crook R., Ince D., and Nuseibeh B. "Modelling access policies using roles in requirements engineering". In: *Information and Software Technology* 45 (2003), pp. 979–991.

[183] Sandhu R.S., Coyne E.J., Feinstein H.L., and Youman C.E. "Role-Based Access Control Models". In: *IEEE Computer* 29.2 (1996).

[184] Bas van der Raadt and Hans van Vliet. "Designing the Enterprise Architecture Function". In: *Quality of Software Architectures. Models and Architectures.* Springer Berlin Heidelberg, 2008, pp. 103–118. DOI: 10.1007/978-3-540-87879-7_7.

[185] Q. Ramadan, M. Salnitriy, D. Strüber, J. Jürjens, and P. Giorgini. "From Secure Business Process Modeling to Design-Level Security Verification". In: *IEEE 20th International Conference on Model Driven Engineering Languages and Systems (MODELS).* IEEE, 2017, pp. 123–133. DOI: 10.1109/MODELS.2017.10.

[186] Qusai Ramadan, Mattia Salnitria, Daniel Strüber and Jan Jürjens, and Paolo Giorgini. "Integrating BPMN- and UML-based Security Engineering via Model Transformation". In: *Proceedings of the SE 2018: Fachtagung des GI-Fachbereichs Softwaretechnik.* Gesellschaft für Informatik, 2018, pp. 63–64.

[187] Wolfgang Reisig. *Understanding Petri Nets: Modeling Techniques, Analysis Methods, Case Studies.* Springer Publishing Company, Incorporated, 2013.

[188] M. Rekik, Khouloud Boukadi, Hanane Ben-Abdallah, Rekik. Mona, and H. Ben-Abdallah. "BPMN metamodel extension with deployment and security information". In: *13th International Arab Conference on Information Technology (ACIT).* 2012, pp. 611–617. DOI: `10.3390/app10144981`.

[189] Ralf H. Reussner, Steffen Becker, Jens Happe, Robert Heinrich, Anne Koziolek, Heiko Koziolek, Max Kramer, and Klaus Krogmann. *Modeling and Simulating Software Architectures – The Palladio Approach.* Cambridge, MA: MIT Press, Oct. 2016. 408 pp.

[190] Alfonso Rodriguez, Eduardo Fernandez-Medina, and Mario Piattini. "A BPMN Extension for the Modeling of Security Requirements in Business Processes". In: *IEICE - Trans. Inf. Syst.* E90-D.4 (Mar. 2007), pp. 745–752. DOI: `10.1093/ietisy/e90-d.4.745`. URL: `http://dx.doi.org/10.1093/ietisy/e90-d.4.745`.

[191] Alfonso Rodriguez, Ignacio Garcia-Rodriguez de Guzman, Eduardo Fernandez-Medina, and Mario Piattini. "Semi-Formal Transformation of Secure Business Processes into Analysis Class and Use Case Models: An MDA Approach". In: *Information and Software Technology* 52.9 (2010), pp. 945–971. DOI: `10.1016/j.infsof.2010.03.015`.

[192] Per Runeson et al. *Case Study Research in Software Engineering: Guidelines and Examples.* John Wiley & Sons, Inc., 2012.

[193] Muhammad Qaiser Saleem, Jafreezal Jaafar, and Mohd Fadzil Hassan. "A Domain-Specific Language for Modelling Security Objectives in a Business Process Models of SOA Applications". In: *International Journal on Advances in Information Sciences and Service Sciences (AISS)* 4 (Jan. 2012), pp. 353–362. DOI: `10.4156/aiss.vol4.issue1.45`.

[194] Mattia Salnitri, Fabiano Dalpiaz, and Paolo Giorgini. "Designing Secure Business Processes with SecBPMN". In: *Softw. Syst. Model.* 16.3 (July 2017), pp. 737–757. DOI: `10.1007/s10270-015-0499-4`.

[195] Mattia Salnitri, Fabiano Dalpiaz, and Paolo Giorgini. "Modeling and Verifying Security Policies in Business Processes". In: *Enterprise, Business-Process and Information Systems Modeling.* Springer Berlin Heidelberg, Jan. 2014, pp. 200–214. DOI: `10.1007/978-3-662-43745-2_14`.

[196] K. S. Sang and B. Zhou. "BPMN Security Extensions for Healthcare Process". In: *IEEE International Conference on Computer and Information Technology; Ubiquitous Computing and Communications; Dependable, Autonomic and Secure Computing; Pervasive Intelligence and Computing.* 2015, pp. 2340–2345. DOI: `10.1109/CIT/IUCC/DASC/PICOM.2015.346`.

[197] Jürgen Schlegelmilch and Ulrike Steffens. “Role Mining with ORCA”. In: *Proceedings of the Tenth ACM Symposium on Access Control Models and Technologies*. SACMAT ’05. ACM, 2005, pp. 168–176. DOI: 10.1145/1063979.1064008.

[198] Christian Schmidt and Peter Buxmann. “Outcomes and success factors of enterprise IT architecture management: empirical insight from the international financial services industry”. In: *European Journal of Information Systems* 20.2 (2011), pp. 168–185. DOI: 10.1057/ejis.2010.68.

[199] Maxim Schnjakin, Michael Menzel, and Christoph Meinel. “A Pattern-Driven Security Advisor for Service-Oriented Architectures”. In: *Proceedings of the 2009 ACM Workshop on Secure Web Services*. SWS ’09. ACM, 2009, pp. 13–20. DOI: 10.1145/1655121.1655126.

[200] M. Schumacher. *Security engineering with patterns: origins, theoretical models, and new applications*. Springer-Verlag, 2003.

[201] Markus Schumacher, Eduardo Fernandez-Buglioni, Duane Hybertson, Frank Buschmann, and Peter Sommerlad. *Security Patterns: Integrating Security and Systems Engineering*. Wiley, 2006.

[202] Stephan Seifermann, Robert Heinrich, and Ralf H. Reussner. “Data-Driven Software Architecture for Analyzing Confidentiality”. In: *IEEE International Conference on Software Architecture, ICSA 2019, Hamburg, Germany, March 25-29, 2019*. IEEE, 2019, pp. 1–10. DOI: 10.1109/ICSA.2019.00009.

[203] Stephan Seifermann, Dominik Werle, and Mazen Ebada. “Mapping Data Flow Models to the Palladio Component Model”. In: *Proceedings of the 10th Symposium on Software Performance (SSP)*. Softwaretechnik Trends. Gesellschaft für Informatik, 2019, pp. 41–43.

[204] V. Seppanen, J. Heikkila, and K. Liimatainen. “Key Issues in EA-Implementation: Case Study of Two Finnish Government Agencies”. In: *IEEE Conference on Commerce and Enterprise Computing*. IEEE, 2009, pp. 114–120. DOI: 10.1109/CEC.2009.70.

[205] C. Simons. “CMP: A UML Context Modeling Profile for Mobile Distributed Systems”. In: *40th Annual Hawaii International Conference on System Sciences (HICSS’07)*. IEEE, 2007, pp. 289–299. DOI: 10.1109/HICSS.2007.125.

[206] G. Sindre and A. L. Opdahl. “Eliciting security requirements by misuse cases”. In: *Requirements Engineering* 10.1 (2005), pp. 34–44. DOI: 10.1109/TOOLS.2000.891363.

[207] John Slankas, Xusheng Xiao, Laurie Williams, and Tao Xie. “Relation Extraction for Inferring Access Control Rules from Natural Language Artifacts”. In: *Proceedings of the 30th Annual Computer Security Applications Conference (ACSAC)*. ACM, 2014, pp. 366–375. DOI: 10.1145/2664243.2664280.

[208] Andre R. R. Souza, Bruno L. B. Silva, Fernando A. A. Lins, Julio C. Damasceno, Nelson S. Rosa, Paulo R. M. Maciel, Robson W. A. Medeiros, Bryan Stephenson, Hamid R. Motahari-Nezhad, Jun Li, and Caio Northfleet. "Incorporating Security Requirements into Service Composition: From Modelling to Execution". In: *7th International Joint Conference on Service-Oriented Computing*. Springer Berlin Heidelberg, 2009, pp. 373–388. DOI: 10.1007/978-3-642-10383-4_27.

[209] J.F. Sowa and John A. Zachman. "Extending and Formalizing the Framework for Information Systems Architecture". In: *IBM Systems Journal* 31.3 (1992).

[210] International Organization for Standardization (ISO) and der International Electrotechnical Commission (IEC). *ISO/IEC 27000:2018*. Accessed: May 31, 2021. 2018. URL: https://www.iso.org/standard/73906.html.

[211] National Institute of Standards and Technology (NIST). *Collection of Role Based Access Control Case Studies*. Accessed: May 31, 2021. URL: https://csrc.nist.gov/projects/role-based-access-control/rbac-case-studies.

[212] Harald Störrle. "How Are Conceptual Models Used in Industrial Software Development?: A Descriptive Survey". In: *Proceedings of the 21st International Conference on Evaluation and Assessment in Software Engineering*. EASE'17. ACM, 2017, pp. 160–169.

[213] Harald Störrle. "How Are Conceptual Models Used in Industrial Software Development?: A Descriptive Survey". In: *Proceedings of the 21st International Conference on Evaluation and Assessment in Software Engineering*. EASE'17. ACM, 2017, pp. 160–169.

[214] Carolyn Strano and Qamar JRehmani. "The role of the enterprise architect". In: *Information Systems and e-Business Management* 5.4 (2007), pp. 379–396. DOI: 10.1007/s10257-007-0053-1.

[215] Luis Jesus Ramon Stroppi, O. Chiotti, and P. Villarreal. "A BPMN 2.0 Extension to Define the Resource Perspective of Business Process Models". In: *Iberoamericano Conference on Software Engineering (CIbSE)*. 2011, pp. 1–14.

[216] Haiyang Sun, Jian Yang, Xin Wang, and Yanchun Zhang. "A Verification Mechanism for Secured Message Processing in Business Collaboration". In: *Advances in Data and Web Management*. Springer Berlin Heidelberg, 2009, pp. 480–491. DOI: 10.1007/978-3-642-00672-2_42.

[217] Hassan Takabi and James B.D. Joshi. "StateMiner: An Efficient Similarity-Based Approach for Optimal Mining of Role Hierarchy". In: *Proceedings of the 15th ACM Symposium on Access Control Models and Technologies*. SACMAT '10. ACM, 2010, pp. 55–64. DOI: 10.1145/1809842.1809853.

[218] D. Thomsen, D. O'Brien, and J. Bogle. "Role based access control framework for network enterprises". In: *Proceedings 14th Annual Computer Security Applications Conference*. IEEE, 1998, pp. 50–58.

[219] D. Thomsen, R. O'Brien, and C. Payne. "Napoleon: Network Application Policy Environment". In: *Proceedings of the Fourth ACM Workshop on Role-Based Access Control.* ACM, 1999, pp. 145–152.

[220] Salah Triki, H. Ben-Abdallah, J. Feki, and N. Harbi. "Modeling Conflict of Interest in the Design of Secure Data Warehouses". In: *International Conference on Knowledge Engineering and Ontology Development (KEOD).* HAL CCSD, 2010, pp. 445–500.

[221] K. Tuma, R. Scandariato, and M. Balliu. "Flaws in Flows: Unveiling Design Flaws via Information Flow Analysis". In: *IEEE International Conference on Software Architecture (ICSA).* IEEE, 2019, pp. 191–200. DOI: 10.1109/ICSA.2019.00028.

[222] European Union. *General Data Protection Regulation (GDPR).* Accessed: May 31, 2021. 2016. URL: http://eur-lex.europa.eu/legal-content/EN/TXT/PDF/?uri=CELEX:32016R0679&from=EN.

[223] Jaideep Vaidya, Vijayalakshmi Atluri, and Qi Guo. "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 '07. ACM, 2007, pp. 175–184. DOI: 10.1145/1266840.1266870.

[224] Jaideep Vaidya, Vijayalakshmi Atluri, Qi Guo, and Nabil Adam. "Migrating to Optimal RBAC with Minimal Perturbation". In: *Proceedings of the 13th ACM Symposium on Access Control Models and Technologies.* SACMAT '08. ACM, 2008, pp. 11–20. DOI: 10.1145/1377836.1377839.

[225] Jaideep Vaidya, Vijayalakshmi Atluri, and Janice Warner. "RoleMiner: Mining Roles Using Subset Enumeration". In: *Proceedings of the 13th ACM Conference on Computer and Communications Security.* CCS '06. ACM, 2006, pp. 144–153. DOI: 10.1145/1180405.1180424.

[226] A. J. Varela-Vaca, R. M. Gasca, and A. Jimenez-Ramirez. "A Model-Driven engineering approach with diagnosis of non-conformance of security objectives in business process models". In: *5th International Conference on Research Challenges in Information Science (RCIS).* IEEE, 2011, pp. 1–6. DOI: 10.1109/RCIS.2011.6006844.

[227] Belen Vela, C. Blanco, E. Fernandez-Medina, and E. Marcos. "A practical application of our MDD approach for modeling secure XML data warehouses". In: *Decision Support Systems* 52.4 (2012), pp. 899–925.

[228] Belen Vela, Eduardo Fernandez-Medina, Esperanza Marcos, and Mario Piattini. "Model Driven Development of Secure XML Databases". In: *ACM SIGMOD Record* 35.3 (2006), pp. 22–27. DOI: 10.1145/1168092.1168095.

[229] V. Venkatesh, Hillol Bala, S. Venkatraman, and J. Bates. "Enterprise Architecture Maturity: The Story of the Veterans Health Administration". In: *MIS Quarterly Executive* 6 (2007), pp. 79–90.

[230] Alain Wegmann. "On the Systemic Enterprise Architecture Methodology (SEAM)". In: *Proceedings of the 5th International Conference on Enterprise Information Systems.* Vol. 3. Infoscience EPFL, 2003, pp. 483–490.

[231] Roel J. Wieringa, H. M. Blanken, M. M. Fokkinga, and P. W. P. J. Grefen. “Aligning Application Architecture to the Business Context”. In: *Proceedings of the 15th International Conference on Advanced Information Systems Engineering*. Springer Berlin Heidelberg, 2003, pp. 209–225.

[232] Robert Winter and Joachim Schelp. “Enterprise Architecture Governance: The Need for a Business-to-IT Approach”. In: *Proceedings of the 2008 ACM Symposium on Applied Computing*. SAC ’08. ACM, 2008, pp. 548–552. DOI: 10.1145/1363686.1363820.

[233] Patrick Woche. “Entwicklung von Geschäftsprozessen zur Analyse eines Transformationsalgorithmus für die Erhebung von Zugriffsberechtigungen eines rollenbasierten Zugriffskontrollmodells”. Bachelor’s Thesis. University of Applied Sciences for Engineering and Economics Berlin, 2017.

[234] Christian Wolter and Christoph Meinel. “An Approach to Capture Authorisation Requirements in Business Processes”. In: *Requirements Engineering* 15.4 (2010), pp. 359–373. DOI: 10.1007/s00766-010-0103-y.

[235] Christian Wolter, Michael Menzel, and Christoph Meinel. “Modelling Security Goals in Business Processes”. In: *Modellierung*. Gesellschaft für Informatik, 2008, pp. 197–212.

[236] Christian Wolter and Andreas Schaad. “Modeling of Task-Based Authorization Constraints in BPMN”. In: *Business Process Management*. Springer Berlin Heidelberg, 2007, pp. 64–79. DOI: 10.1007/978-3-540-75183-0_5.

[237] Christian Wolter, Andreas Schaad, and Christoph Meinel. “Task-Based Entailment Constraints for Basic Workflow Patterns”. In: *Proceedings of the 12th ACM Symposium on Access Control Models and Technologies*. SACMAT ’08. ACM, 2008, pp. 51–60. DOI: 10.1145/1377836.1377844.

[238] Felix Wortmann. “Entwicklung einer Methode für die unternehmensweite Autorisierung”. PhD Thesis. University of St. Gallen, 2006.

[239] Xusheng Xiao, Amit Paradkar, Suresh Thummalapenta, and Tao Xie. “Automated Extraction of Security Policies from Natural-Language Software Documents”. In: *Proceedings of the ACM SIGSOFT 20th International Symposium on the Foundations of Software Engineering*. FSE ’12. ACM, 2012. DOI: 10.1145/2393596.2393608.

[240] Zhongyuan Xu and Scott D. Stoller. “Algorithms for Mining Meaningful Roles”. In: *Proceedings of the 17th ACM Symposium on Access Control Models and Technologies*. SACMAT ’12. ACM, 2012, pp. 57–66. DOI: 10.1145/2295136.2295146.

[241] John A. Zachman. “A Framework for Information Systems Architecture”. In: *IBM Systems Journal* 26.3 (1987).

[242] Karim Zarour, Djamel Benmerzoug, Nawal Guermouche, and Khalil Drira. “A BPMN Extension for Business Process Outsourcing to the Cloud”. In: *New Knowledge in Information Systems and Technologies*. Springer International Publishing, 2019, pp. 833–843. DOI: 10.1007/978-3-030-16181-1_78.

[243] Dana Zhang, Kotagiri Ramamohanarao, and Tim Ebringer. “Role Engineering Using Graph Optimisation”. In: *Proceedings of the 12th ACM Symposium on Access Control Models and Technologies.* SACMAT ’07. ACM, 2007, pp. 139–144. DOI: 10.1145/1266840.1266862.

[244] Dana Zhang, Kotagiri Ramamohanarao, Steven Versteeg, and Rui Zhang. “Graph Based Strategies to Role Engineering”. In: *Proceedings of the Sixth Annual Workshop on Cyber Security and Information Intelligence Research.* CSIIRW ’10. ACM, 2010. DOI: 10.1145/1852666.1852694.

[245] Z. Zhang, F. Hong, and J. Liao. “Modeling Chinese Wall Policy Using Colored Petri Nets”. In: *The Sixth IEEE International Conference on Computer and Information Technology (CIT’06).* IEEE, 2006, pp. 162–162. DOI: 10.1109/CIT.2006.123.