Identifying Needs for a Holistic Modelling Approach to Privacy Aspects in Enterprise Software Systems

Sascha Alpers, Roman Pilipchuk, Andreas Oberweis and Ralf Reussner
FZI Forschungszentrum Informatik, Haid- und Neustraße 10-14, 76131 Karlsruhe, Germany

Keywords: Business Architecture, Software Architecture, Modelling, Privacy.

Abstract: Modelling is a common method for both Business Architecture Management and for Software Architecture Management. In general, there is a gap in the model continuity between business models and software models. Especially when modelling compliance driven requirements like privacy traceability is important for compliance checks and helps to build the models in an efficient way. In this paper, approaches for modelling privacy from business and software engineering perspective are examined. A key finding is that there is currently no comprehensive modelling approach covering the needed aspects and perspectives.

1 INTRODUCTION

Many companies, especially large companies, model their organisational processes and software systems. The reason is to improve them, identify and reduce flaws and save costs by implementing correct workflows. However, business and software system experts typically use different modelling languages. There exist many languages for modelling business processes. BPMN, a semi-formal notation, is the most prominent one. Petri nets provide a formalised view on processes. Transformations exist which establish mappings between BPMN and Petri nets. In the following, we focus primarily on Petri net (Reisig, 2013) models and consider BPMN only marginally. The state-of-the-art modelling language for software systems is UML (OMG, 2017). As neither business process modelling languages nor UML have elements capable for modelling privacy, extension mechanisms exist for introducing additional symbols to model various aspects of privacy. Additionally, security is relevant because privacy is related to some security goals like confidentiality or integrity. Both security and privacy are becoming increasingly important for example, due to the upcoming General Data Protection Regulation (GDPR) (European Union, 2017). An example of organisations that are affected by the GDPR are those that build upon platform-based business models. The business case of such organisation is hosting digital platforms to connect producers and consumers in specific sectors, such as mobility and energy. Monetary turnover is produced through either access fees, transaction fees, premium services or targeted advertisement. To realise such a business idea, a close and flawless collaboration between the business level and information technology (IT) is needed because IT implements the technical foundation (the digital platform), which is at the heart of the overall business idea. On this platform, various stakeholders will operate and conduct their business. Whether it is customer to customer, business to customer or business to business, the privacy of individuals, organisations and especially of sensible data is critical on digital platforms. Therefore, the need for a current and comprehensive modelling approach to privacy between business processes and software models is critical.

Although, there are many approaches to extend business process modelling notations and UML to cover security and other aspects, there is no common and generally accepted approach for modelling privacy. A broad variety of approaches exists for introducing additional symbols to model privacy directly or indirectly through security elements; however, the extent to which privacy can be modelled by every proposal varies. Additionally, modelling approaches are missing, which support transformations from business process models to software design to keep business process models like Petri nets and software models like UML consistent with each other. Due to these reasons, we analysed the capabilities of existing architecture oriented

Alpers, S., Pilipchuk, R., Oberweis, A. and Reussner, R.
Identifying Needs for a Holistic Modelling Approach to Privacy Aspects in Enterprise Software Systems.
DOI: 10.5220/0006606200740082
In *Proceedings of the 4th International Conference on Information Systems Security and Privacy (ICISSP 2018)*, pages 74-82
ISBN: 978-989-758-282-0

Table 1: Overview of Architecture oriented approaches.

No.	Paper	Diag. Type	Ext. Through	To Model
1	Engineering Privacy for Big Data Apps with the Unified Modelling Language	Use Case	Super container	Privacy specifications
2	Towards a UML Profile for Privacy-Aware Applications	Various	UML profile	Privacy policies
3	UMLsec: Extending UML for Secure Systems Development **(+2)**	Various	UML profile	Security requirements / primitives / management and threat scenarios
4	Supporting Confidentiality in UML: A Profile for the Decentralised Label Model	Class	UML profile	Decentralized label model
5	Towards the Engineering of Security of Information Systems (ESIS): UML and the IS Confidentiality	Sequence	UML profile	Access control and information flow control
6	A UML Profile for Requirements Analysis of Dependable Software	Class	UML profile	Problem frames (e.g., confidentiality, integrity)
7	Extending UML for Designing Secure Data Warehouses **(+2)**	Class	UML profile	Security classes and separation of duty
8	Weaving Security Aspects into UML 2.0 Design Models	Class and Sequence	UML profile	Security requirements and aspect-oriented solutions
9	CMP: A UML Context Modelling Profile for Mobile Distributed Systems	Class	UML profile	Privacy restrictions

and business process oriented modelling approaches to model privacy aspects. We analysed, how privacy can be modelled and tried to understand the possibility and need for a comprehensive modelling language in the field of privacy to cover business processes and software systems. We selected these approaches according to their abilities to model privacy aspects directly or indirectly through security aspects. The selected approaches were analysed and compared with each other to identify their similarities and differences. This was done to understand the need for a comprehensive model of privacy aspects and to explore how it could be realised beginning from a business process model and then leading to a software architecture model. For this, we categorised the approaches and identified two criteria, namely "security mechanisms" and "different views". "Security mechanisms" describe, by which elements and mechanisms the approach supports privacy modelling. The second criteria, "different views", groups approaches according to the view of the stakeholder for whom the approach is intended. Our results show that only a few approaches actually introduce elements to model privacy principles. Section 2 presents the business process-based approaches. Software architecture based approaches are presented in Section 3. Section 4 discusses similarities and differences between both approaches. The paper ends with some concluding remarks in Section 5.

2 ARCHITECTURE ORIENTED APPROACHES

This chapter introduces the architecture oriented approaches for modelling privacy. The first section introduces the de facto standard modelling language in the field of software engineering and the second section introduces the architecture-based approaches in the context of privacy and confidentiality.

2.1 Modelling Language

The de facto standard for modelling architecture in software engineering is the Unified Modelling Language (short: UML). UML is a general-purpose modelling language that is standardised by the Object Management Group (short: OMG). It comprises 14 diagrams divided into two major diagram types: structure diagrams and behaviour diagrams (OMG, 2017). While structure diagrams represent the static structure of a system, behaviour diagrams represent its dynamic part. The use case diagram visualises functional requirements, as well as actors and their relationships, while the sequence diagram shows the chronological flow of messages between objects. In the class diagram classes, associations, methods, and attributes are described. A detailed explanation can be found in the UML specification (OMG, 2017).

2.2 Analysis of Architecture Oriented Approaches

This section surveys the architecture-based approaches. Table 1 summarises all papers, the UML diagram types which they extend, whether they extend through UML profile or not, and what the extension allows to be modelled.

(Jutla et al., 2013) propose an extension to the UML use case diagram for representing privacy specifications like pseudonymization, anonymization, and consent in an easily understandable way (see Table 1 No. 1). The extension is not based on the UML profile extension mechanism. Instead, a

Microsoft Visio extension ribbon is created that offers the needed elements. All possible privacy requirements and specifications can be expressed due to the usage of free text fields. The extension works by introducing a ‘super container’ in-between actors and use cases of a use case diagram. Privacy control classes and obligations are stated inside the super container. This extension allows modelling of all kinds of privacy principles but also other security principles like confidentiality.

(Basso et al., 2015) introduced a UML profile, which is capable of expressing different privacy concepts through privacy policies incorporated in various UML diagrams (see Table 1 No. 2). Privacy policies are composed by one or more statements, which describe the rules specified in the privacy policy. Besides that, they specify the purpose for data collection, management, and prerequisites that need to be met. Private data and actions performed on it can be expressed through stereotypes, for example, to whom private data is allowed to be disclosed, the period, and how it will be used. Several other stereotypes describe how the data is provided and managed, either by a user or by a system. In both cases, the UML profile allows the design of privacy-aware applications through modelling the application’s privacy policy and keeping track of the elements responsible for enforcing them. It allows not only modelling of access control on private data but also of privacy principles like consent, data security, and purpose limitation.

(Jürjens, 2002) proposed a UML profile, called UMLsec, for expressing security-relevant information within various UML diagrams (see Table 1 No. 3). This should allow persons which are not experts in security to express their security needs easily. UMLsec enables software engineers to express basic security requirements including security concepts, security primitives, security management and threat scenarios. This allows modelling confidentiality of information and information flows. Furthermore, it is possible to check whether the constraints associated with the stereotypes are fulfilled by a given specification and, by this, indicate possible vulnerabilities (Jürjens, 2005).

(Heldal et al., 2004) present a UML profile incorporating the decentralized label model into the UML class diagrams to model confidentiality at design time (see Table 1 No. 4). The so-called UMLs profile allows the specification of confidential information flow in a fine-grained manner. Different stereotypes defining owners and users are used to annotate classes, attributes, operations, parameters, errors, and return types. These labels are used to decide whether the information flow is permitted or not. Declassification of information is realised with the *authorityConstraint*, which models the weakening of the confidentiality of information coming from higher confidential sources. This is necessary for operations processing confidential data but providing less confidential results. The approach is presented for class diagrams, but it is extendable to other diagram types like interaction, use case, and activity diagrams.

The work of Goudalo et al. (Goudalo and Seret, 2008) elaborates on modelling security aspects of information systems (see Table 1 No. 5). The proposed UML profile is an example of how to properly encapsulate security knowledge during design time. This is shown in the context of confidentiality. Confidentiality of information and information flow can be modelled in sequence diagrams by defining stereotypes modelling confidentiality levels of resources, subjects, and subsystems. In summary, the UML profile enables software engineers to model confidentiality in various manners.

The work of Hatebur et al. (Hatebur and Heisel, 2010) builds upon a UML profile for expressing problem frames in UML class diagrams (see Table 1 No. 6). Problem frames are patterns used to define problem classes by their contexts and characteristics. This UML profile is extended to express dependability requirements. In the context of security, the traditional goals of confidentiality, availability, and integrity can be expressed. They are modelled as stereotypes, including specifications like the data to be secured, the attacker, and the stakeholder of data. This allows the expression of arbitrary confidentiality requirements via the use of problem frames. The authors mention the main advantage of their approach, namely the ability to express dependability requirements without the anticipation of a solution. This clearly separates the problem space from the solution space. In addition, it is easy to visually distinguish between different security requirement classes.

The approach of (Fernandez-Medina et al., 2004), called SECDW, allows modelling confidentiality aspects in UML class diagrams (see Table 1 No. 7). SECDW is an extension intended for the domain of Data Warehouses. The approach introduces a UML profile that allows the specification of security classes for information and users. By using tuples composed of security classifications, sets of user compartments (classification of users in department like structures), and user roles, it is possible to specify constraints about which users are allowed to read certain information. The extension proposed by Triki et al. (Triki et al., 2010) (SECDQ+) introduces the ability to model leaks of confidential information, e.g., health information or company turnover, that are due to access to combinations of data that would impose no information leakage if isolated. This problem is known as *conflict of interest* (Triki et al., 2010).

(Mouheb et al., 2009) propose a UML profile capable of both capturing security requirements and specifying security solutions (see Table 1 No. 8). This is done by waving security aspects into UML class and sequence diagrams in an aspect-oriented modelling manner. This approach allows the separation of security concerns from software functionalities. Security experts can specify security solutions as aspects in the UML model and model their points (where they are implemented) in UML sequence diagrams and, by this, provide an easily understandable solution for non-security experts.

(Simons, 2007) presents a UML profile to model privacy restrictions in UML class diagrams (see Table 1 No. 9). The profile was developed for the context of mobile distributed systems, but it can be used in other contexts as well. The main idea is to bind access rights to context information. This is done by formulating privacy restrictions on context information. These restrictions consist of the source and validity of the context information and the access rights in the form of confidentiality levels. In Simons's UML profile, constraints are used to validate the model. This is done by imposing restrictions on the defined stereotypes to enforce the correct use of the profile.

3 BUSINESS PROCESS ORIENTED APPROACHES

Privacy is not an end in itself. Privacy and security are business requirements, and, therefore, privacy as well as security requirements in future will be included in enterprise modelling more often. This can be achieved in different ways:

- Via models of privacy and security aspects using normal enterprise modelling languages,
- in the form of annotations,
- or with the help of more-or-less formalised privacy/security notation add-ons for existing modelling languages.

For business processes as one component of enterprise modelling, we analysed 'Petri Nets' and 'Business Process Model and Notation (BPMN)'. Therefore, we performed systematic literature reviews using the method described by Kitchenham and Charters (2007). Two students executed the reviews in their master's theses. Gholam Hassan Sahabi focused on information security when using Petri Nets (a modelling language with mathematical foundation), and Daniel Tewolde focused on information security when using Business Process Model and Notation (BPMN). To obtain comparable results, we used the same methodology, and the reviews were conducted for the same publication period. The theses' supervisors were used as experts to score random parts of the publications. The scores were compared to the scores that the students had assigned to the paper, and the variances were analysed and discussed.

3.1 Analysis of Petri Net-based Approaches

There are plenty of approaches using Petri Nets for modelling information security aspects, particularly information confidentiality. They can be used to model privacy requirements as well, but special privacy model extensions are not common today. The problem is also that some of the approaches only focus on the technical level, which generally means that they are discussing problems like algorithms, protocols or technical architecture using Petri nets for visualisation but omit the business process perspective.

Huang and Kirchner have introduced a formal method to verify whether the compositions of sub-policies fulfil the required general policies of a company (Huang and Kirchner, 2013). They used coloured Petri Nets and Petri Net-based properties like completeness, termination, consistency and confluence. One use case is the verification as to whether a set of policies fulfils a general policy like GDPR. Therefore, the requirements of the GDPR must be transformed into a model.

(Mixia et al., 2005) extended Object Petri Nets by using modules to define security services like the de-/en-cryption of data. This could be interesting for data protection because encrypted data must not be protected itself as long as the key is strong and kept secret. (Akbarzadeh and Azgomi, 2010) defined a framework for the assessment of security protocols. They used coloured stochastic activity nets and implemented probabilistic model checking. In addition, (Bouroulet et al., 2008) analysed security protocols and a Petri Net extension called S-net, which is designed such that the terms of the Security Protocol Language (Crazzolara and Winskel, 2001) can be used. Other Petri Net based approaches aim at building models for special concepts. For example, (Zhang et al., 2006) modelled the Chinese Wall policy with coloured Petri Nets; afterwards, they used a coverability graph to analyse the guarantees of the Chinese Wall policy. (Henry et al., 2010) used coupled Petri Nets for the risk analysis of computer networks. Sun et al. published a 'Verification Mechanism for Secured Message Processing in Business Collaboration' (Sun et al., 2009). They used the role-based access control (RBAC) mechanism and hierarchical coloured Petri Nets to detect conflicts in message access within collaboration process instances with the role-based policy. A similar approach from (Lai et al., 2008) focussed on the confidentiality of information exchanges between organisations and therefore has special places in coloured activity nets for incoming and outgoing information. Chinese Wall and interorganizational information exchange are also relevant for privacy protection questions. As shown, many approaches use Petri Nets for modelling security aspects but focus on a technical level or only cover one single aspect. Therefore, these approaches are not suitable for business process experts to model their security requirements and discuss them with technical experts.

In addition, some approaches use Petri Nets for modelling or analysing security aspects of business processes. Accorsi and Wonnemann developed InDico (Accorsi and Wonnemann, 2011), an information-flow analysis method for labelling Petri Net-based business process models. InDico focusses on 'information propagation throughout the systems (end-to-end) rather than mere data access (point to point)' (Accorsi and Wonnemann, 2011). Accorsi et al. (Accorsi et al., 2015) published an extension of InDico for analysing information-flow effects during process execution. They used security levels (here called 'levels of confidentiality') but reduced them to two, and analysed structural interferences between them. It is impossible to express different levels of confidentiality for the same place in one business process scheme, e.g., different information, or more than two levels of confidentiality for the whole business process scheme. Li et al. (Li et al., 2009) described a coloured Petri Net extension for detecting confidentiality problems in information flow models. They use security levels and add the concrete security level as attributes of the tokens. Li et al. did not focus on resources handling the information. Knorr (Knorr, 2001), who also used security levels, presented a method to verify multilevel security policies in workflow models, but he modelled control and information flow as different arcs in his workflow Petri Nets. Atluri and Huang (Atluri and Huang, 1996), who have also used Petri Nets, presented a multilevel security approach with security levels for places and tokens. They later extended their approach with more concepts like separation of duty and role-based access using a coloured, timed Petri Net (Atluri and Huang, 2000). They did not consider resources or the possibility of reducing the security level of a token, e.g., when information is truncated.

The high number of approaches for modelling security aspects using (high-level) Petri Nets shows that the integration and processing of confidential information in Petri Net-based business process models is currently a major challenge. This is one reason why we think, Petri Nets fit well for privacy questions as well. Other reasons in favour of Petri Nets are their mathematical foundation and the availability of a broad range of analysis methods. Especially for analysis functionality, formal Petri Nets are necessary.

3.2 Analysis of BPMN-based Approaches

Extensions of the Business Process Model and Notation for modelling security requirements exist for all of the three classic security objectives: confidentiality, integrity, and availability. Leitner et al. (Leitner et al., 2013) have published a systematic literature review on 'Security Aspects in the Business Process Model and Notation'. Therefore, we do not provide a detailed overview here. In summary, some publications use BPMN for security questions without new extensions. In (Meland and Gjaere, 2012), Meland and Gjaere argue that in most cases there is no need for new BPMN extensions. Several other approaches extend the BPMN notation, e.g., with new symbols to create a faster overview about security issues for the model users (Wolter and Meinel, 2010). Focussing on privacy as part of security, (Mülle et al., 2011) used BPMN for introducing privacy in business process models while Labda et al. (Labda et al., 2013) extended BPMN to model privacy aware BPMN. They focussed not only on modelling privacy aspects but also proposed a methodology for transferring them into the implementation.

4 COMPARING APPROACHES

We identified two criteria in which the architecture oriented and business process oriented approaches can be compared.

Security Mechanisms: this criterion describes how and through which security and privacy mechanisms privacy can be expressed by the particular approach. We identified two characteristics:

- **Information flow and access control:** this characteristic establishes privacy by introducing concepts that restrict the information flow or the access to information, functions or system parts by imposing rights. These approaches introduce concepts of confidentiality in various ways and to different degrees. The introduced concepts are used either directly or can be used to express privacy in a certain way. Examples are Chinese Wall policy and confidentiality levels. The following approaches contribute to this characteristic [(Jürjens, 2002), (Heldal et al., 2004), (Goudalo and Seret, 2008), (Simsons, 2007), (Fernandez-Medina et al., 2004), (Zhang et al., 2006), (Sun, et al., 2009), (Lai et al., 2008), (Accorsi and Wonnemann, 2011), (Accorsi et al., 2015), (Li et al., 2009), (Knorr, 2001), (Atluri and Huang, 2000), (Mülle et al., 2011)].
- **General structures:** approaches with these characteristics use abstract structures to express either several or a particular security and privacy principle. For example, problem frames used in (Hatebur and Heisel, 2010) give the ability to express a problem and, through this, express an actual security principle. Another example, common in the security area, is policies. We identified the following approaches contributing to this characteristic: [(Jutla et al., 2013), (Basso et al., 2015), (Hatebur and Heisel, 2010), (Mouheb et al., 2009), (Huang and Kirchner, 2013), (Mixia et al., 2005), (Akbarzadeh and Azgomi, 2010), (Bouroulet et al., 2008), (Henry et al., 2010), (Atluri and Huang, 2000)].

Each approach is assigned to one characteristic. The approaches we reviewed focus either on confidentiality to express privacy or on introducing various other structures through which privacy can be expressed. The first are grouped under the characteristic 'information flow and access control' and the latter ones under the characteristic 'general structures'. We determined that nearly half of the reviewed architecture oriented and business process oriented approaches contribute to the first characteristic. They all introduce elements to model confidentiality. Some of them use confidentiality mechanisms to establish privacy in a certain way [(Fernandez-Medina et al., 2004), (Zhang et al., 2006), (Sun et al., 2009), (Lai et al., 2008), (Accorsi and Wonnemann, 2011), (Accorsi et al., 2015), (Li et al., 2009), (Knorr, 2001)]. The others only introduce modelling elements for confidentiality and not directly for the purpose of privacy [(Jürjens, 2002), (Heldal et al., 2004), (Goudalo and Seret, 2008), (Simsons, 2007), (Mülle et al., 2011)]. The other half of the reviewed approaches utilises various other mechanisms to model privacy. [(Julta et al., 2013)] introduce new structures like super containers and problem frames to express privacy. Some others use policies [(Basso et al., 2015), (Huang and Kirchner, 2013)].

Different views: this criterion describes the view of the model for which the approach is developed. As there are various stakeholders with different concerns to express, different views arise that can be specialised for the specific needs of a stakeholder. Typical examples from the field of security are the attacker view and security specialist view. While the attacker view introduces model elements showing how the attacker could intrude into the system, the security specialist view would highlight the security measures in place.

The criterion 'different views' divides the approaches according to their use by stakeholders. Common views are:

- **Attacker view:** models the attacker with the attacks, threats, and vulnerabilities of a system, or analyses the given model for flaws in the information flow [(Jürjens, 2002), (Akbarzadeh and Azgomi, 2010), (Bouroulet et al., 2008), (Henry et al., 2010), (Accorsi and Wonnemann, 2011), (Accorsi et al., 2015), (Li et al., 2009), (Atluri and Huang, 2000)].
- **Requirements & Implementation view:** introduces elements to express requirements pertaining to security and privacy aspects and elements, which model security and privacy solutions [(Julta et al., 2013), (Basso et al., 2015), (Heldal et al., 2004), (Goudalo and Seret, 2008), (Hatebur and Heisel, 2010), (Simsons, 2007), (Mouheb et al., 2009), (Fernandez-Medina et al., 2004), (Mixia et al., 2005), (Zhang et al., 2006), (Sun et al., 2009), (Lai et al., 2008), (Atluri and Huang, 1996), (Atluri and Huang, 2000), (Mülle et al., 2011)].
- **Verification view:** allows users to check whether a model fulfils certain requirements by checking them against the model. This is realised, for example, with constraints, which are checked for correct implementation or the

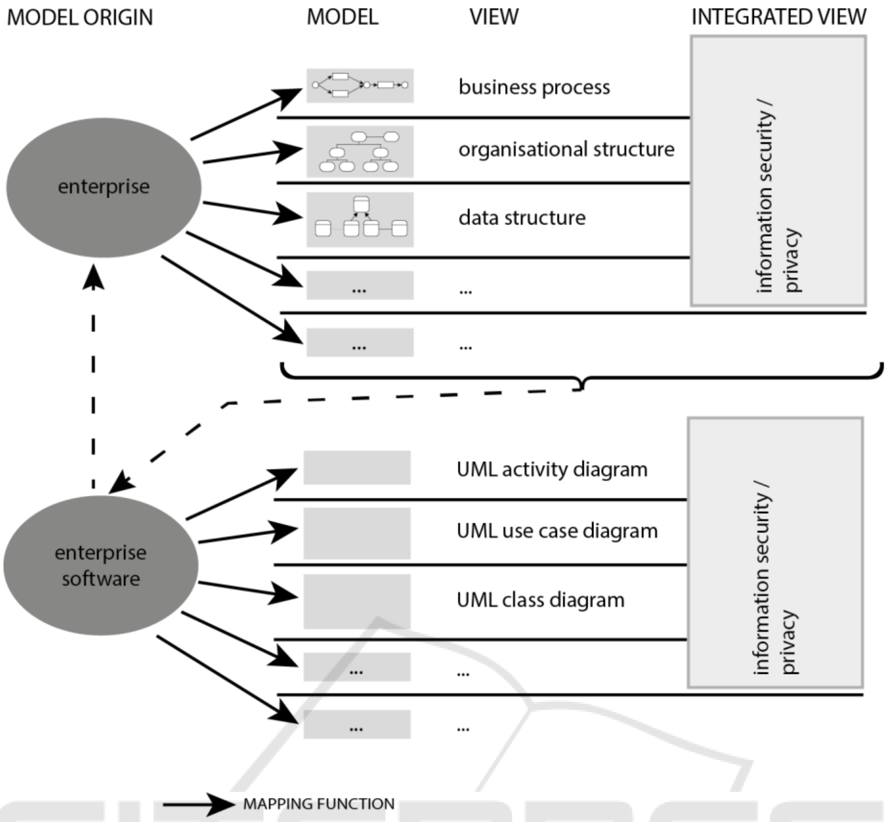

Figure 1: Holistic modelling approach.

verification of policies [(Basso et al., 2015), (Jürjens, 2002), (Heldal et al., 2004), (Fernandez-Medina et al., 2004), (Huang and Kirchner, 2013), (Zhang, Hong and Liao, 2006), (Accorsi et al., 2015), (Li et al., 2009), (Knorr, 2001), (Atluri and Huang, 1996)].

In the architecture oriented approaches, the ‘attacker view’ is realised by introducing an attacker and his capabilities. We found only one approach of this type (Jürjens, 2002). The business process oriented side identifies flaws in the information flow and, thus, privacy breaches. The ‘requirements & implementation view’ is represented in both approaches. Here, elements are introduced to express security and privacy requirements or solutions. The difference between the approaches is in the degree of abstraction. While the business process oriented approaches typically are on a less technical and more abstract level, the architecture-based approaches introduce both a non-expert view and sometimes a more technical, expert view. In both architecture oriented approaches and business process oriented approaches, we identified the intention of verifying whether the implementation or model is correct with respect to certain requirements. This is the ‘verification view’. While architecture oriented approaches verify the correctness of modelled solutions, business process oriented approaches try to identify and verify security policies against a given model. In general, we recognised that in the reviewed approaches, the architecture-based approaches tended to model requirements or design solutions more often as well as verify whether the model fulfils them, while the business process-based approaches focussed more on the identification of flaws and the verification of policies.

In summary, only a few approaches we reviewed introduced elements to model actual privacy principles [(Julta et al., 2013), (Basso et al., 2015), (Atluri and Huang, 2000)]. Most of them introduce privacy more by way of establishing confidentiality and the restriction of access to information.

5 CONCLUSIONS

As we have shown, there are some approaches for systematically modelling security and/or privacy protection aspects of organisations in a specific view. However, no comprehensive approach integrates all aspects such as process, structure organization, and data. Such approaches must be developed further. For the enterprise that is shown in the upper part of Figure 1, the enterprise software is shown in the lower section of the figure 1. As illustrated in figure 1, integrated views are combining different other views of an organisation and enrich them by the additional integration of privacy aspects.

Important links are represented by the dotted lines. Requirements of the enterprise models must be transferred into the software models to be implemented later. This is especially true for organisations in which the main business idea depends on the realisation of a privacy-sensitive complex IT infrastructure as well as those that are either building digital platforms or working on the automation of business processes. We therefore suggest an automated model transformation from enterprise to software modelling. Continuous modelling is a prerequisite for the traceability of the requirements. Therefore, it must be possible to transfer business requirements modelled in Petri Nets to software requirements modelled in UML. The other edge shows the link between the enterprise software and the enterprise.

We are focusing on integrating privacy into enterprise software by looking at the privacy within underlying business processes and the architecture of Enterprise Software. Additionally, it is possible to use either software or frameworks for privacy management. Some examples are the Privacy Management Platform (Stach and Mitschang, 2014), the AVARE project, which is for the privacy and self-protection of citizens (Alpers et al., 2017), and the Context-Aware Privacy Protection System (Pingley et al., 2009).

ACKNOWLEDGEMENTS

This work has been partially financed by the Baden-Württemberg Stiftung GmbH within the project ‘AVARE’.

This research was also partially sponsored by the German Ministry for Research and Economics (BMBF) as a part of the ‘Forschungscampus Initiative‘, specifically ‘Forschungscampus Mobility2Grid‘, under the grant number 03SF0522A.

REFERENCES

Accorsi, R.; Wonnemann, C.; 2011: InDico: Information flow analysis of business processes for confidentiality requirements. In *Security and Trust Management, Springer, pp. 194–209.*

Accorsi, R.; Lehmann, A.; Lohmann, N.; 2015: Information leak detection in business process models: Theory, application, and tool support. In *Inf. Syst., vol. 47, pp. 244–257.*

Akbarzadeh, M.; Azgomi, M. A.; 2010: A framework for probabilistic model checking of security protocols using coloured stochastic activity networks and PDETool. In *5th International Symposium on Telecommunications (IST), pp. 210–215.*

Alpers, S.; Pieper, M.; Wagner, M.; 2017: Herausforderungen bei der Entwicklung von Anwendungen zum Selbstdatenschutz. *In Informatik 2017, pp. 1061-1072.*

Atluri, V.; Huang, W.-K.; 1996: An extended Petri net model for supporting workflows in a multilevel secure environment. In *Database Security Tenth International Conference on Database Security, Como, Italy, pp. 240–258.*

Atluri, V.; Huang, W.-K.; 2000: A Petri net based safety analysis of workflow authorization models. In *J. Comput. Secur., vol. 8, no. 2, 3, pp. 209–240.*

Basso, T.; Montecchi, L.; Moraes, R.; Jino, M.; Bondavalli, A.; 2015: Towards a UML Profile for Privacy-Aware Applications. In *IEEE International Conference on Computer and Information Technology, pp. 371-378.*

Bouroulet, R.; Devillers, R.; Klaudel, H.; Pelz, E.; Pommereau, F.; 2008: Modeling and analysis of security protocols using role based specifications and Petri nets. In *Applications and Theory of Petri Nets, K. M. van Hee and R. Valk, Eds. Springer Berlin Heidelberg, pp. 72–91.*

Crazzolara, F.; Winskel, G.; 2001: Events in security protocols. In *Proceedings of the 8th ACM conference on Computer and Communications Security, pp. 96–105.*

European Union; 2017: General Data Protection Regulation, *http://eur-lex.europa.eu/legal-content /EN/ TXT/PDF/?uri=CELEX:32016R0679&from=EN*, last *accessed 24.10.2017.*

Fernandez-Medina, F.; Trujillo, J.; Villaroel, R.; Piattini, M.; 2004: Extending UML for Designing Secure Data Warehouses. In *Conceptual Modeling - ER 2004: 23rd International Conference on Conceptual Modeling, pp. 217-230.*

Goudalo, W.; Seret, D.; 2008: Toward the Engineering of Security of Information Systems (ESIS): UML and the IS Confidentiality. In: *2nd International Conference on Emerging Security Information, Systems and Technologies, pp. 248-256.*

Hatebur, D.; Heisel, A.: A UML Profile for Requirements Analysis of Dependable Software. In *Computer Safety, Reliability, and Security: 29th International Conference, SAFECOMP 2010, pp. 317-331.*

Henry, M. H.; Layer, R. M.; Zaret, D. R.; 2010: Coupled Petri nets for computer network risk analysis. In *Int. J. Crit. Infrastruct. Prot., vol. 3, no. 2, pp. 67–75.*

Huang, H.; Kirchner, H.; 2013: Secure interoperation design in multi-domains environments based on colored Petri nets. *In Inf. Sci., vol. 221, pp. 591–606.*

Jürjens, J.; 2002: UMLsec: Extending UML for Secure

Systems Development. In *Proceedings of the 5th International Conference on The Unified Modeling Language (UML ‘02), pp. 412-425.*

Jürjens, J.; 2005: Model-Based Security Engineering with UML. In *Foundations of Security Analysis and Design III: FOSAD 2004/2005 Tutorial Lectures, pp. 42-77.*

Jutla, D. N.; Bodorik, P.; Ali, S.; 2013: Engineering Privacy for Big Data Apps with the Unified Modeling Language. In *IEEE International Congress on Big Data, pp. 38-45.*

Kitchenham, B.; Charters, S.; 2007: Guidelines for performing Systematic Literature Reviews in Software Engineering. EBSE Technical Report, https://www.elsevier.com/__data/promis_misc/525444systematicreviewsguide.pdf, last *accessed 24.10.2017.*

Knorr, K.; 2001: Multilevel security and information flow in Petri net workflows. In *Proceedings of the 9th International Conference on Telecommunication Systems, pp. 613–615.*

Labda, W.; Mehandjiev, N.; Sampaio, P.; 2013: Privacy-aware business processes modeling notation (prvbpmn) in the context of distributed mobile applications. In *Trends in Mobile Web Information Systems. Springer, pp. 120-134.*

Lai, H.; Hong, J.; Jeng, W.; 2008: Model E-contract update by coloured activity net. In *IEEE Asia-Pacific Services Computing Conference. APSCC ’08, 2008, pp. 488–493.*

Leitner, M.; Miller, M.; Rinderle-Ma, S.; 2013: An Analysis and evaluation of security aspects. In *The Business Process Model and Aotation, pp. 262–267.*

Li, W.; Wu, R.; Huang, H.; 2009: Colored Petri nets based modeling of information flow security; *In Second International Workshop on Knowledge Discovery and Data Mining, pp. 681–684.*

Meland; P. H.; Gjaere, E. A.; 2012: Representing Threats. In *BPMN 2.0, pp. 542–550.*

Mixia, L.; Qiuyu, Z.; Dongmei, Y.; Hong, Z.; 2005: Formal security model research based on Petri-net. In *IEEE International Conference on Granular Computing, vol. 2, pp. 575–578.*

Mouheb, D.; Talhi, C.; Lima, V.; Debbabo, M.; Lang, L.; Pourzandi, M.; 2009: Weaving Security Aspects into UML 2.0 Design Models. In *Proceedings of the 13th Workshop on Aspect-oriented Modeling, pp. 7 – 12.*

Mülle, J,; Stackelberg, S. v.; Böhm, K.; 2011: Modelling and transforming security constraints in privacy-aware business processes. In *2011 IEEE International Conference on Service-Oriented Computing and Applications (SOCA), pp. 1–4.*

OMG; 2013: Business Process Model and Notation (BPMN) v2.0.2, *http://www.omg.org/spec/BPMN/2.0.2/ PDF, last accessed 24.10.2017.*

OMG; 2017: Unified Modeling Language v2.5, *http://www.omg.org/spec/UML/2.5/PDF, last accessed 24.10.2017.*

Pingley, A.; Yu, W.; Zhang, N.; Fu, X.; Zhao, W.; 2009: CAP: A Context-Aware Privacy Protection System for Location-Based Services. In *29th IEEE International Conference on Distributed Computing Systems, pp. 49-57.*

Heldal, R.; Schlager, S.; Bende, J.; 2004: Supporting Confidentiality in UML: A Profile for the Decentralized Label Model. In *Proceeding Workshop on Critical Systems Development with UML, pp. 56-70.*

Reisig, W.; 2013: Understanding Petri Nets: Modeling Techniques, Analysis Methods, Case Studies, Springer, New York.

Simons, C.; 2007: CMP: A UML Context Modeling Profile for Mobile Distributed Systems. In *40th Annual Hawaii International Conference on System Sciences, pp. 289-299.*

Stach, C.; Mitschang, B.; 2014: Design and Implementation of the Privacy Management Platform. In *Proceedings of the 2014 IEEE 15th International Conference on Mobile Data Management, pp. 69-72.*

Sun, H.; Yang, J.; Wang, X.; Zhang, Y.; 2009: A verification mechanism for secured message processing in business collaboration. In *Advances in Data and Web Management, Springer, pp. 480–491.*

Triki, S.; Ben-Abdallah, H.; Feki, J., Harbi, N.; 2010: Modeling Conflict of interest in the Design of Secure Data Warehouses. In *KEOD 2010 - International Conference on Knowledge Engineering and Ontology Development, pp. 445-500.*

Wolter, C.; Meinel, C.; 2010: An approach to capture authorisation requirements. In *Business Processes. Requir. Eng., vol. 15, no. 4, pp. 359–373.*

Zhang, Z.-L.; Hong, F.; Liao, J.-G.; 2006: Modeling Chinese Wall Policy Using Colored Petri Nets. *In The Sixth IEEE International Conference on Computer and Information Technology, pp. 162–162.*