Methods for Designing Privacy Aware Information Systems: A review

Christos Kalloniatis¹, Evangelia Kavakli¹, Stefanos Gritzalis²

¹Cultural Informatics Laboratory, Dept. of Cultural Technology and Communication University of the Aegean, Harilaou Trikoupi & Faonos Str., 81100 Mytilene, Greece
²Laboratory of Information and Communication Systems Security, Dept. of Information and Communications Systems Engineering, University of the Aegean, 83200 Samos, Greece

{ch.kalloniatis, kavakli}@ct.aegean.gr, sgritz@aegean.gr

Abstract

A major challenge in the field of software engineering is to make users trust the software that they use in their everyday professional or recreational activities. Trusting software depends on various elements, one of which is the protection of user privacy. Protecting privacy is about complying with user’s desires when it comes to handling personal information. It can also be defined as the right to determine when, how and to what extend information about them is communicated to others. The aim of this paper is to bring forward the modern practices for ensuring privacy during the design of information systems. To this end, it provides an overview of recent requirements engineering approaches which focus on the elicitation and management of privacy requirements. Comparative analysis of these approaches based on a systematic framework reflects on current research trends as well as open issues that provide a foundation for further research.

1. Introduction

In the new digital world online users hold a number of different data sets in order to be able to access various e-services and take part in specific financial and social transactions. Such data sets require special consideration since they may convey personal data, sensitive personal data, employee data, credit card data etc. Recent surveys [1,2] have shown that people feel that their privacy is at risk from identity theft and erosion of individual rights and this has a negative effect on the degree of trust they feel with respect to the information systems they use. One of the main criteria that determine users’ trust is the way that their privacy is protected. This is especially crucial in e-applications, since the greater collection and storage of personal data, the lower the trust of users using the specific applications.

As a result, privacy as a design criterion has received much attention in recent years by researchers and practitioners alike. Initially, privacy protection efforts have focused on technological solutions for assuring user privacy during software implementation (referred to as privacy enhancing technologies or PETs). However, focusing on software solutions irrespective to the organisational context in which the system will operate makes it difficult to determine which software solution best fits the organizational needs. Understanding the relationship between user needs and the capabilities of the supporting software systems is of critical importance. This has led to the development of a number of security-oriented requirement engineering methodologies which consider security issues (including privacy) during the early stages of system development and this is the subject matter of this paper. In particular, this paper provides an overview of privacy-oriented methods within the context of RE (section 2) and presents a critical analysis of these methods based on a systematic framework (section 3). Finally, section 4 concludes the paper with a few observations of the current state of the art and pointers for future developments.

2. Privacy-oriented Requirements Engineering

Privacy as a social and legal issue, has traditionally, been the concern of social scientists, philosophers and lawyers. However, the extended use of various software applications in the context of basic e-services sets additional technology-related requirements for protecting the electronic privacy of individuals. Most e-services rely on stored data for identifying customers, their preferences and previous record of transactions. Combining such data constitutes in many cases, an invasion of privacy. The development of a global information society results in the rapid development of new information infrastructures that span among various countries. This creates additional threats regarding privacy protection of the users using these global resources.

In today’s networked society privacy is endangered and cannot be protected solely by laws and regulations. Developers and information system specialists must consider privacy as a main technical concern which has to be considered early in the system development lifecycle as a separate design criterion. In fact, privacy is not a
single requirement but can be further specialised into more specific requirements, such as authentication, anonymity, unlinkability, unobservability and so on [3, 4, 5, 6]. Depending on the intended system usage and user needs, one or more of the aforementioned requirements are considered during system development. The majority of IS development methods manage privacy as one of the system’s non-functional requirements. No specific techniques are proposed especially for identifying and implementing privacy requirements. However, the increasing importance of system security in general and user privacy in particular, has resulted in a number of methods that adopt concepts from the field of IS security engineering and use them in order to explicitly represent security requirements (which also include privacy requirements) and they define the way that these requirements can be transformed in specific policies for the system under construction. We term all these efforts, listed below, privacy oriented approaches.

- The NFR (Non-Functional Requirement Framework) method [7]
- The i* method [8]
- The Tropos method [9]
- The KAOS method [10]
- The GBRAM (Goal-Based Requirements Analysis Method) method [11]
- The RBAC (Role-Based Access Control) method [12]
- The M-N (Mofett-Nuseibeh Framework) method [13]
- The B-S (Bellotti-Sellen Framework) method [14]
- The STRAP (STRuctured Analysis for Privacy) method [15]
- The PriS (Privacy Safeguard) method [16]

In most cases privacy itself is not the main focus of the approach however, explicit identification and management of privacy requirements is an integrated part of its method and plays an essential role in achieving the aims of the approach. An overview of these methods is presented in the following sections 2.1 – 2.10.

2.1. The NFR Method

NFR provides a systematic way of representing and recording the system design process in terms of interdependent design decisions on how to best satisfy (satisfice) non-functional requirements such as security, accuracy, performance and cost [7, 17]. Non-functional requirements are treated as softgoals to be achieved, i.e., they are special type of goals which need to be clarified, disambiguated, prioritized, elaborated upon, etc. Although the NFR method might be applied in all phases of system development (requirements analysis, conceptual design, software specification) it has mostly been used in the early stages of system requirements specification.

The NFR operation can be seen as the incremental and interactive construction, elaboration and revision of a softgoal interdependency graph (SIG). The graph records the developer’s consideration of softgoals and shows the interdependencies among softgoals. Interdependencies show refinements of general softgoals downwards into more specific (operational) softgoals. They also show the contribution of softgoals upon meeting other softgoals which may be at the same or higher level. To determine whether softgoals are achieved an evaluation procedure which considers interdependencies as well as developer claims. Finally, operationalised softgoals are linked to the system functional requirements thus, showing how they constrain the system implementation.

An important aspect of the NFR method is that the construction of the softgoal graph can be guided by previous experience recorded in the form of softgoal type classification catalogues (guiding softgoal refinement), correlation catalogues (guiding the identification of negative/positive contributions among softgoals) and method catalogues (guiding the identification of appropriate operationalisations). Specific NFR catalogues have been constructed for security requirements also considering privacy (as part of the confidentiality security requirement), which makes NFR a useful tool for defining privacy requirement and identifying possible design alternatives.

The functionality of the NFR method is based on a formal model. In addition, a case tool namely the Organisation Modelling Environment (OME) has been developed, which offers a graphical interface for building NFR models and guides developers in each step of the NFR way-of-working.

2.2. The i* Method

The i* method focus on the early stages of system design and aims on mapping organisation’s logic and context [8, 17, 18]. It is agent-oriented in the sense that it focuses on systems agents and their social interdependencies. i* was initially designed as tool for modelling, analyzing and redesigning organisation processes. Recently, it has been used for modelling security and privacy requirements, as well [19, 20, 21, 22]. Similarly to the NFR method, modelling security requirements in i* is based in the notion of softgoal. However, i* does not focus on overall system goals rather on individual goals of system actors (active entities of the system into consideration). In fact, system actors are interdependent since the accomplishment of their goals depends on tasks performed by other actors or on resources shared by other agents.

Following an initial construction of a domain model, in terms of the actors involved and their dependencies,
security analysis takes place using a number of analysis techniques. In particular, attacker analysis helps identify potential system abusers and their malicious intents (threats). Dependency vulnerability analysis helps detect vulnerabilities in terms of organizational relationships among stakeholders. Countermeasure analysis supports the dynamic decision-making process of addressing vulnerabilities and threats. This results into further refinement of actor softgoals. When all actors are considered an evaluation procedure is applied in order to decide whether the impact of threats and vulnerabilities has been eliminated to an acceptable level. Finally, role-based access control analysis further details actor roles in terms of the tasks they perform and the resources they depend upon thus, getting closer to a system design.

The $i^*$ method meta-model is formally defined and is supported by the OME tool described previously in NFR.

2.3. The Tropos Method

Tropos is a method for describing organisational systems and their environment [9, 23, 24]. It is applied in all stages of system development, adopting a homogeneous analysis from requirements analysis, to system design and implementation. Tropos organisational model is based on $i^*$ adopting its basic concepts such as actors, goals, tasks and social dependencies. Additionally, Tropos defines the term secure entity to describe any goals and tasks related to the security of the system. A secure goal represents the strategic interests of an actor with respect to security. A secure dependency introduces security constraint(s) on the relations between actors.

The security process in Tropos is one of analysing the security needs of the stakeholders and the system in terms of security constraints imposed to the stakeholders (early requirements analysis) and the system (late requirements analysis), identifying secure entities that guarantee the satisfaction of the security constraints, and assigning capabilities to the system (architectural design) to help towards the satisfaction of the secure entities. These capabilities are subsequently specified in AUMl notation -- a UML extension of actor modelling (detailed design).

The Tropos method is supported by formal models. In addition, a software tool (namely ST-Tool) has been developed for designing the method’s models as well as for formally analyzing these models.

2.4. The KAOS Method

KAOS is a goal-oriented framework, which focuses on the elaboration and specification of the system requirements [11, 25-30]. To this end, it provides a specification language for defining system requirements, as well as an elaboration method for eliciting such requirements.

From a methodological perspective, KAOS uses a goal elaboration method for capturing security requirements from high-level organizational goals. Firstly the goal AND/OR structure is developed by defining goals and their refinement links until assignable goals are reached. Subsequently, the objects involved in goal formulation are defined (object capture). In a similar manner, operation capture identifies the objects state transitions that are meaningful to the goals and operationalisation derives pre/post/trigger conditions on operations ensuring that all terminal goals are met. Finally, responsibility assignment identifies all alternative responsibilities for terminal goals, making decisions among refinements and assigns operations to agents, guaranteeing the terminal goals selected.

During the elaboration process obstacles and anti-goals preventing security goals from being met are also taken into consideration and a refinement procedure is followed until these obstacles are overcome.

KAOS offers to system designers a formal representation language, requirements engineering strategies as well as support through a software tool which assists designers in defining requirements derived from more abstract-generic system goals.

2.5. The GBRAM Method

GBRAM [11] is another goal-oriented method. It provides a methodical approach for identifying and refining the goals that software systems must achieve, managing trade-offs among the goals, and converting them into operational requirements.

GBRAM has also been used to analyze privacy policies of e-applications and to systematically extract the requirements and goals underlying organizations’ privacy practices. This process, called ‘goal-mining’, makes use of a privacy taxonomy that classifies the extracted requirements as either protection goals or vulnerabilities. Protection goals express the effort declared by an organization to honor/respect its customers’ privacy. Vulnerabilities reflect potential threats to customer privacy as derived from current organization practices such as information collection, storage and transfer. The privacy goals are subsequently operationalised into system requirements, using a number of techniques including scenario analysis, identification of goal obstacles and constraints and refinement strategies via heuristics, guidelines and recurring question types.

Finally, the degree of compliance between requirements and policy statements is assessed and possible conflicts and ambiguities are resolved, thus aligning privacy requirements to organisations’ privacy policies.

GBRAM is not supported by formal models. A software tool called SMaRT (Scenario Management and
Requirements Tool) has been developed in order to support the GBGRAM goal analysis. The tool also incorporates the privacy taxonomy.

2.6. The RBAC Method

He and Antón present an agent oriented framework for modelling privacy requirements as the as the contexts and obligations of Role-Based Access Control (RBAC) entities and relationships, thus linking privacy requirements to organisational access control policies [12]. It includes a context-based data model for representing roles that have permissions to access data objects and privacy elements linked to these objects. Three privacy elements are recognised namely purpose for accessing a data object, conditions that must be satisfied before a data access request can be permitted, and obligations i.e., actions that must be carried out if a request to access data is granted.

In addition, it provides a goal-driven role engineering process for eliciting and modelling these privacy elements. This process consists of two phases: Role Permission Analysis (RPA) and Role Permission Refinement (RPR). RPA uses goal-oriented, scenario analysis to identify the tasks to be performed by each role and associated RBAC permissions. In particular, the events of each scenario are modeled as RBAC permissions and the actors of the events are modelled as RBAC roles. The set of roles and permissions candidates generated from is further refined during the RPR phase in order to identify associated privacy elements.

RBAC is not supported by formal models. A guiding framework has been defined for expressing the relations between the roles and their restrictions on organisation’s assets. Finally, a software tool has been developed but it does not cover the whole method since it does not support role analysis.

2.7. The M-N Method

The objective of this method is to elicit and analyse security requirements during the early stages of system design. It provides a meta-model in the spirit of the KAOS conceptual model [13] but suggests an alternative way-of-working for dealing with security requirements. In addition, it adopts the security engineering concept of assets (anything valuable for the organisation, e.g. the backup files, the main server, etc.) together with threats of harm to those assets. Security goals aim to protect assets from those threats and are operationalised into security requirements.

The M-N way of working includes two steps. In the first step security risk analysis and management techniques are applied in order to identify and evaluate the risks to a system and make the decisions for the appropriate security measures. This step identifies assets and their valuation against relevant threats for those assets. In the second step, high-level security goals are defined in the sense of “what we are aiming to achieve”. The determination of each security goal comes from each threat defined in step 1. Refinement of these security goals into security constraints is accomplished using KAOS goal refinement method.

M-N method is not supported by formal models nor software tools.

2.8. The B-S Method

Belloti and Sellen [14] developed the method B-S, which main objective is the elicitation of privacy requirements during system design phase and specifically during requirements specification. Privacy requirements in B-S are represented as privacy criteria which designers should follow in order to define organisation’s vulnerabilities so as to be able to suggest a number of possible solutions for overcoming the identified vulnerabilities.

B-S specifies privacy requirements in the following way. Firstly, the designers evaluate the organisation based on a list of privacy criteria defined by the method. Designers can also ask a number of questions (also defined by the method) to organisation’s employees in order to acquire the relevant knowledge and understand the organisation’s status in a faster and more reliable way. When designers receive the answers from the above mentioned process, the vulnerabilities of the system are noted and the designers decide how they will represent those vulnerabilities.

The method proposed in Hong, Lederer and Landay [31] extends B-S by increasing the number and the quality of questions being asked as well as the criteria list on which the proposed solutions are based. The B-S method is not supported formally. Also it does not offer any software tool for assisting developers during late design phases.

2.9. The STRAP Method

The STRAP method (STRuctured Analysis of Privacy) [15] was developed with the objective of eliciting and analysing privacy requirements during system design phase. In STRAP privacy requirements are represented as vulnerabilities. STRAP builds a goal-model which represents all functional requirements and vulnerabilities have the form of obstacles between the goals and the subgoals in the goal-model.

The method’s way of working comprises of the following steps: a) Analysis, b) Refinement, c) Evaluation and d) Iteration. During the analysis phase, firstly all the system goals are analysed. The result of this phase is the identification of all goals, the active entities and the basic
system components. Additionally, relevant information are gathered regarding the context of the developing system and the first set of privacy requirements are elicited and recorded. Like B-S method, STRAP uses a number of questions for every goal and subgoal identified by the previous step. The result of the questions leads to the identification of various system vulnerabilities regarding privacy protection. Vulnerabilities then are recorded in the goal-model as obstacles between the goals and the relevant subgoals. Moreover, the identified vulnerabilities are categorised based on the Fair Information Practices list [32]. Before the categorisation an analysis on the vulnerabilities set is conducted in order to eliminate similar ones thus narrowing the set and facilitating the solutions suggestion phase.

During the refinement phase, designers eliminate and narrow the set of vulnerabilities by deleting all those vulnerabilities for which a solution is easy to implement. Specifically, for the vulnerabilities belonging to this category, designers immediately suggest a specific solution and they delete them from the vulnerabilities set.

In the next phase, the evaluation, the suggested design scenarios for the developing system are evaluated. All these suggestions are evaluated based on a number of criteria. The first criterion evaluates the proposed suggestions by examining the way each suggestion overcomes the identified vulnerabilities. The best scenario is the one that eliminates the most vulnerabilities thus decreasing the risk and assuring the proper privacy protection. STRAP suggests a number of criteria that guide the evaluation phase.

Finally, in the iteration phase the aforementioned steps are repeated since the system needs to be redesigned for including all possible alterations conducted during the previous phases. Particularly, the goal structure is re-examined, the necessary alterations take place, the vulnerabilities are redefined, the new scenarios are evaluated etc. The iteration phase ends when no alterations take place in the previous three steps. STRAP is not supported by formal models. Also, no software tool has been implemented for guiding the designers and for the graphical representation of the method’s models.

2.10. The PriS Method

PriS [16] is a security requirements engineering method, which incorporates privacy requirements early in the system development process. PriS considers privacy requirements as organizational goals that need to be satisfied and adopts the use of privacy process patterns as a way to: (a) describe the effect of privacy requirements on business processes; and (b) facilitate the identification of the system architecture that best supports the privacy-related business processes.

PriS method models privacy requirements as a special type of goal, the privacy goal, which constraints (have impact on) the causal transformation of organisational goals into processes. Privacy goals are generated because of issues. In particular, eight types of privacy goals are recognised namely a) authentication, b) authorisation, c) identification, d) data protection, e) anonymity, f) pseudonymity, g) unlinkability and h) unobservability.

From a methodological perspective reasoning about privacy goals in PriS comprises of the following activities. The first step concerns the elicitation of the privacy goals that are relevant to the specific organisation. This task usually involves a number of stakeholders and decision makers who aim to identify the basic privacy concerns and interpret the general privacy requirements with respect to the specific application context into consideration. In addition, existing privacy requirements already forming part of the organisation’s goals are identified. The second step consists of two stages. In the first stage the impact of privacy goals on the organisational goals is identified and analysed. In the second stage, the impact of the privacy goals on the relevant processes that realise these goals is examined and the processes that realize the privacy-related goals are identified and characterized as privacy-related processes.

Having identified the privacy-related processes the next step is to model them, based on the relevant privacy process patterns. Business process patterns are usually generalised process models, which include activities and flows connecting them, presenting how a business should be run in a specific domain [33]. The last step is to define the system architecture that best supports the privacy-related process identified in the previous step. Once again, process pattern are used to identify the proper implementation technique(s) that best support/implement corresponding processes.

PriS assists in the application of privacy requirements in the organisational context as well as in providing a systematic way of locating a number of system architectures that can realise these requirements. PriS way of working assumes that privacy goals are generic-strategic organisational goals thus being mentioned high in the goal model hierarchy. PriS metamodel has a formal definition and graphical representation. Also, a software tool has been developed for guiding designers as well as for automating its way of working.

3. Analysis of Privacy-oriented Methods

This section provides a critical analysis of the different strands of privacy-oriented RE research, based on a common framework for understanding privacy-oriented approaches.
This takes into account a number of different views, as they are expressed in the following questions: in which stage of the RE process are they applied (usage); what type of privacy issues do they address (subject); what mechanisms do they offer for expressing privacy issues (representation); what kind of support do they provide to designers in applying proposed way-of-working (development).

Usage view
Despite the fact that there is no common definition of the RE process, three general tasks to be performed have been identified: requirements elicitation, requirements specification and requirements validation. All methods provide suitable concepts for specifying privacy requirements (softgoals, security goals, etc) as well as a number of relationships among them (AND/OR refinement, support / conflict, obstacles) and other organisational concepts (actors, tasks, objects, processes). Specification of privacy goals is greatly affected by work in the area of goal modelling. This is natural since privacy requirements are non-functional (quality) requirements, and goals have long been used as a way to handle NFRs in RE. A great number also deals with the elicitation of privacy goals and their relationships, offering a number of techniques, including privacy taxonomies, security ontologies (which might be generic or linked to a specific domain), privacy requirement catalogues risk analysis, scenario analysis. Few consider the requirements validation and mainly as qualitative goal-reasoning techniques aiming to evaluate the degree to which privacy goals are satisfied. They are based on the identification of conflicts between privacy issues or obstacles that hinder their achievement. However, few actually offer techniques for eliminating / resolving such obstacles (e.g., scenario analysis used in GBRAM).

Subject view
Most methodologies consider privacy requirements as overall system goals derived from business or organization goals, which constrain the system functional requirements. In i* and Tropos consider the privacy goals of individual system agents thus defining privacy is seen as a trade-off process between the aspirations of different agents. Few actually link these requirements to privacy related technologies, with the exception of NFR and PriS which use method catalogues and privacy process patterns respectively in order to identify relevant system implementation techniques. i* and RBAC use role-based access control analysis to define data access policies, but still these policies are requirement statements and not implementation techniques. GBRAM in addition, considers the relation between privacy requirement to business privacy policies, thus ensuring the compliance between system operation and business rules.

Representation view
Regarding the representation view, most methods use a graphical notation of representing their models. NFR, i*, Tropos, KAOS and PriS use decomposition techniques for representing the hierarchy among goals and their subgoals. Beside the graphical notation these methods also use a formal language for formally representing their models. NFR and i* use the Telos language. Tropos uses Formal Tropos specification language. Formal Tropos complements i* by defining a textual notation for i* models and allows the description of dynamic constraints among the different elements of the specification in a first order linear-time temporal logic. In KAOS security goals are formalized according to the pattern of behaviour they require, using temporal logic. PriS way of working has also been expressed formally. M-N framework and GBRAM do not use a formal language or a graphical notation for representing their models. M-N framework uses an informal textual approach for expressing its models. GBRAM defines tables where each goal is assigned with specific obstacles identified and scenarios for solving those obstacles. RBAC does not use any graphical notation or formal language. Boolean expressions are used only for modeling the conditions specified in a privacy policy.

Development view
Regarding the development perspective, most methods use modeling tools for helping the developer to impress the models generated or to manually make any changes. NFR framework with NFR Assistant, i* with its Organizational Modeling Environment, Tropos with graphical Tropos and KAOS with Observier provide modeling tools but they do not supply any guidance. On the other hand M-N framework, GBRAM and RBAC don’t present a modeling tool like Observier but they provide, in a pleasant degree, guidance through their models. This guidance helps the developer to resolve any possible conflicts between security goals or other security issues or even change the way the model is manipulating security requirements so far. NFR, i*, Tropos and KAOS are very restricted since the developer can only achieve a good representation of the model without getting any suggestions on how to resolve impending problems. PriS uses a software tool for assisting its designers on creating its models. Models are represented in tree view but users cannot change this view to a more graphic one. Additionally, PriS guides its users throughout the whole usage of the specific tool starting from the elicitation of privacy requirements and continuing until the suggestion of the relevant implementation techniques.

The comparison results are shown in table 1.
4. Conclusions

Privacy is recognised internationally as a human right which has to be protected from the societies in which people participate and interact. Every day, more and more people use Internet since the services it offers improve their way and quality of living. This rapid development of Internet users led service providers to increase the number of products and services they offer in order to improve the quality of services for the users and also to manage to remain in the market aiming for better profits. This paper provided an overview of selected privacy-oriented methods from the field of requirements engineering. The selection was based on the fact that these methods incorporate basic concepts for the clear representation of privacy requirements during system design and also they study ways for transforming these requirements into specific implementation alternatives.

The common theme in all these approaches is the realisation that security and privacy requirements must be analysed early during the system design phase and not later in the implementation phase since the time and cost needed for resolving any vulnerabilities in the later phase is much greater than in the design phase. However, as can be seen in Table 1, security and privacy issues are not considered with equal significance during the various phases of system design. Many methods stop the analysis before reaching the implementation phase thus failing to successfully guide the designer throughout the whole phases of system development.

The tendency for a holistic confrontation of security and privacy requirements from the early stages of system design through its implementation phase is expressed in the latest research methods (e.g. PriS, SecureTropos [9, 16, 23, 24]). Furthermore, most privacy models are qualitative in nature. The absence of parameters, inputs, initial conditions and generally of factors that are needed for testing such models greatly diminish their value as tools to understanding phenomena of the world. For without testing of the models it is impossible to comprehend their implications by merely observing, walking through, and debating about their contents. Nor is it always feasible to test them through observations from experimentation in the real world.

Techniques from the domains of risk analysis and secure and dependable systems (misuse cases [34, 35, 36], abuse cases [37], threat trees [38], attack trees [39]) can be used in order to quantify the impact of privacy threats and assist the decision making process. Another option is the use of simulation techniques such as scenario analysis. Simulation imposes rigorous testing that removes ambiguity, exposes alternatives to stakeholders and effectively removes any affectation towards the models driven by personal biases or political factors.
<table>
<thead>
<tr>
<th>Usage</th>
<th>NFR</th>
<th>$p^*$</th>
<th>Tropos</th>
<th>KAOS</th>
<th>GBRAM</th>
<th>RBAC</th>
<th>M-N</th>
<th>B-S</th>
<th>STRAP</th>
<th>PriS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Requirements Elicitation</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Requirements Specification</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Requirements Validation</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Subject</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Overall Business Privacy</td>
<td>✓</td>
<td></td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Requirements</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Privacy Goals of collaborating actors</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Privacy enabling technologies</td>
<td>✓</td>
<td></td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Privacy Policies</td>
<td>✓</td>
<td></td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Representation</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Graphical notation</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Formal language</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Development</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Guidance Processes</td>
<td>✓</td>
<td></td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Modeling Tools</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
</tbody>
</table>
5. References


