Requirements Eng (2008) 13:241–255
DOI 10.1007/s00766-008-0067-3

ORIGINAL ARTICLE

Addressing privacy requirements in system design: the PriS method

Christos Kalloniatis · Evangelia Kavakli · Stefanos Gritzalis

Received: 3 August 2007 / Accepted: 15 July 2008 / Published online: 7 August 2008

Abstract A major challenge in the field of software engineering is to make users trust the software that they use in their every day activities for professional or recreational reasons. 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. Users' privacy can also be defined as the right to determine when, how and to what extend information about them is communicated to others. Current research stresses the need for addressing privacy issues during the system design rather than during the system implementation phase. To this end, this paper describes PriS, a security requirements engineering method, which incorporates privacy requirements early in the system development process. PriS considers privacy requirements as organisational goals that need to be satisfied and adopts the use of privacy-process patterns as a way to: (1) describe the effect of privacy requirements on business processes; and (2) facilitate the identification of the system architecture that best supports the privacy-related business processes. In this way, PriS provides a holistic approach from 'high-level' goals to 'privacy-compliant' IT systems. The PriS way-of-working is formally defined thus, enabling the development of automated tools for assisting its application.

Keywords Requirements engineering · Privacy requirements · Formal methods · Privacy-process patterns · Privacy enhancing technologies · Goal-oriented approach · System design

C. Kalloniatis (✉) · E. Kavakli
Cultural Informatics Laboratory, Department of Cultural Technology and Communication, University of the Aegean, Harilaou Trikoupi and Faonos Str., 81100 Mytilene, Greece
e-mail: ch.kalloniatis@ct.aegean.gr

E. Kavakli
e-mail: kavakli@ct.aegean.gr

S. Gritzalis
Information and Communication Systems Security Laboratory, Department of Information and Communications Systems Engineering, University of the Aegean, 83200 Samos, Greece
e-mail: sgritz@aegean.gr

1 Introduction

Privacy as a social and legal issue, traditionally, has been the concern of social scientists, philosophers and lawyers [1]. 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 are relying on stored data for identifying customers, their preferences and previous record of transactions. Combining such data constitutes in many cases, an invasion of privacy. Protecting privacy is especially important in e-applications, since the greater collection and storage of personal data, the lower the trust of users using the specific applications.

Privacy-related issues are many and varied, as privacy itself is a multifaceted concept. Privacy comes in many forms, relating to what one wishes to keep private. Review of current research, highlights the path for user privacy protection in terms of eight privacy requirements namely *identification*, *authentication*, *authorisation*, *data protection*, *anonymity*, *pseudonymity*, *unlinkability* and *unobservability* [2–4]. The first three requirements are mainly security requirements but they are included due to their key role in the privacy protection. By addressing these requirements one aims to minimise or eliminate the collection of user identifiable data.

Research efforts aiming to the protection of user privacy fall in two main categories: security-oriented requirement engineering methodologies and privacy enhancing technologies. The former focus on methods and techniques for considering security issues (including privacy) during the early stages of system development and the latter describe technological solutions for assuring user privacy during system implementation. The main limitation of security requirement engineering methodologies is that they do not link the identified requirements with implementation solutions. Understanding the relationship between user needs and the capabilities of the supporting software systems is of critical importance. Privacy enhancing technologies, on the other hand, focus on the software implementation alone, irrespective of the organisational context in which the system will be incorporated. This lack of knowledge makes it difficult to determine which software solution best fits the organisational needs.

This paper describes PriS, a method for incorporating basic privacy requirements into the system design process. PriS models privacy requirements in terms of organisational goals and uses the concept of privacy-process pattern for describing the impact of privacy goals onto the organisational processes and the associated software systems supporting these processes. In addition, Formal PriS provides a formal definition of the Pris way-of-working, i.e., it formally defines the processes of: (1) analysing the impact of privacy requirement(s) on organisational goals, subgoals and process and (2) suggesting of appropriate sysstem implementation technique(s) for realising these requirements. The rest of this paper is structured as follows. Section 2 describes the e-voting system case study which is used throughout the paper. Section 3 presents the PriS conceptual framework and way of working. Also it introduces privacy-process patterns and explains how they can be used in order to identify appropriate privacy implementation techniques. Formal PriS presented in Sects. 4 and 5 discuss PriS in the context of related work. Finally, Sect. 6 concludes with pointers to future work.

2 The e-voting case

PriS method is demonstrated through an e-voting case study, regarding the transformation of an Internet based electronic voting system in order to accommodate the new legal framework regarding privacy protection.

The initial design of the electronic voting system was developed in the context of the European Project "E-Vote" by the University of Regensburg, in cooperation with the University of the Aegean, the Cryptomatic company, the Quality and Reliability company and the Athens University of Economics and Business and is described in [5]. According to this description, the main objective of the e-voting system is to provide eligible citizens the right to cast a vote over the Internet rather than visiting an election district, aiming to simplify the election processes thus increasing the degree of citizens' participation during elections. It is described by four main principles that form the four primary organisational goals namely: (1) Generality, (2) Equality, (3) Freedom and (4) Directness. Generality implies that all citizens above a certain age should have the right to participate in the election process. Equality signifies that both political parties—that participate in the election process—and voters have equal rights before, during and after the election process and neither the system nor any other third party is able to alternate this issue. Freedom implies that the entire election process is conducted without any violence, coercion, pressure, manipulative interference or other influences, exercised either by the state or by one or more individuals. Finally, directness means that no intermediaries chime in the voting procedure and that each and every ballot is directly recorded and counted.

A partial view of the system's current goal model is presented in Fig. 1. In the last line the doted boxes are the relevant processes that satisfy organisational goals.

As mentioned earlier, the system has to be re-designed in order to guarantee that user's privacy is not violated. To this end, PriS was applied by two teams of postgraduate students of the University of the Aegean that worked in parallel in order to:

1. to analyse the impact of privacy issues on the system's goals and processes and propose alternative system implementations (first team),
2. formally describe the above process and its deliverables (second team);
3. provide feedback regarding both difficulties encountered and recommendations or incorporation into the PriS method (both teams).

The students were computer science graduates and had knowledge of requirements engineering principles but no experience with the particular method. Work from this case study is reported in [6, 7]. The findings of this case study were cross checked with the ones of a second case study regarding the University of the Aegean Career Office System [8] which was conducted by two similar groups during the same period.

3 The PriS method

3.1 PriS conceptual framework

As mentioned above, privacy enhancing technologies focus on the software implementation alone. In other words,

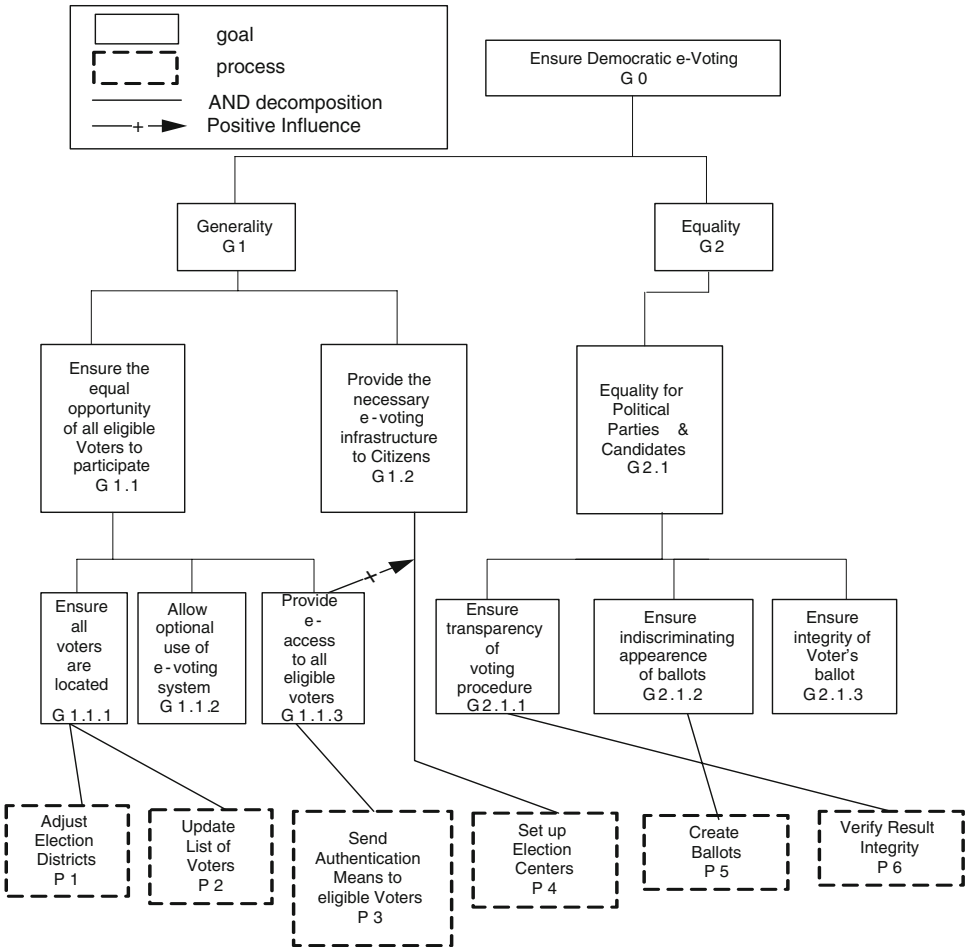

Fig. 1 Partial view of the e-voting system goal model

there is no obvious link between the organisational processes that are constrained by the privacy requirements and the supporting software systems. This lack of knowledge makes it difficult not only to determine which software solution best fits the organisational needs but also to evaluate alternatives.

To this end, PriS provides a set of concepts for modelling privacy requirements in the organisation domain and a systematic way-of-working for translating these requirements into system models. The conceptual model used in PriS, shown in Fig. 2, is based on the Enterprise Knowledge Development (EKD) framework [9, 10], which is a systematic approach for developing and documenting organisational knowledge. This is achieved through the modelling of: (1) organisational goals, that express the intentional objectives that control and govern its operation, (2) the 'physical' processes, that collaboratively operationalise organisational goals and (3) the software systems that support the above processes. In this way, a connection between system purpose and system structure is established.

In more detail, PriS models system requirements as *goals*, i.e., state of affairs that need to be attained. Typical goals are to 'ensure participation of all eligible voters' or 'ensure equality for all participating political parties'. Goals pertain to *stakeholders* e.g., voters, political parties, regulators, the constitution, system designers, etc. Goals are generated because of *issues*, such as the 'need to conform to the new legal framework concerning privacy'. Goals are realised by *processes*. To this end, high-level or 'strategic' goals may be decomposed in simpler, 'operational' goals forming *AND/OR* goal hierarchies. In addition, two goals (in different branches of the goal hierarchy), may *support* their mutual achievement or may be in *conflict*. Such conflicts should be made explicit and resolved through negotiation among the various stakeholders involved in the process. The negotiation task can be facilitated by conflict resolution techniques such as, requirement prioritisation, voting procedures, etc. During this process initial goals may get rephrased, some of them may be rejected and additional goals may be identified.

Privacy requirements as a special type of goal (*privacy goals*) which constraint (*have impact on*) the causal transformation of organisational goals into processes. In particular, eight types of privacy goals are recognised, corresponding to the eight privacy concerns identified in

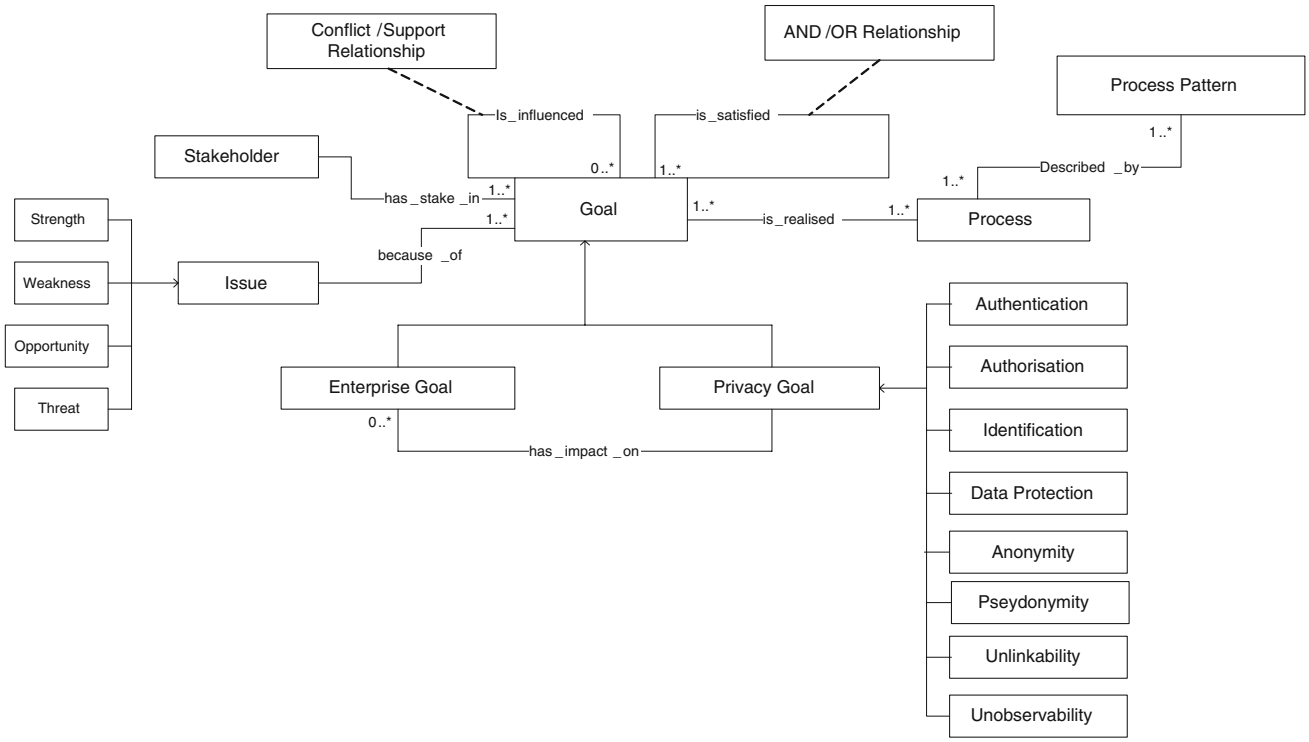

Fig. 2 PriS conceptual model

Sect. 1. Similar to other organisational goals, privacy goals may be decomposed in simpler goals or may support/conflict the achievement of other goals. In this case, negotiation techniques similar to the ones described earlier should be used in order to achieve a goal model which best satisfies stakeholder's needs. In fact, a number of security goals exist that influence privacy goals. This is the reason for including in the privacy goals set a number of security-related requirements, i.e., authentication, authorisation, identification and data protection.

Finally, processes are described by *process patterns*, i.e., generalised process models which include activities and flows connecting them, presenting how a business should be run in a specific domain [11]. In particular, PriS defines seven *privacy-process patterns* corresponding to the eight basic privacy goals.[1]

From a methodological perspective reasoning about privacy goals comprises of the following activities:

(a) Elicit privacy-related goals 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 (managers, policy makers, system developers, system users, etc.). Identifying privacy concerns is guided by the eight privacy goal types shown in Fig. 2. The aim is to interpret the general privacy requirements with respect to the specific application context into consideration. In the e-voting case two privacy goals were identified, namely: unlinkability and unobservability. The former refers to the voters' right to receive the respective authentication means (username and password) without others being able to reveal to whom the data are sent. Thus, even when a malicious third party is able to steal these data he/she will not be able to know neither the user nor the system where these data can be used. The latter concerns the voters' right to ensure the transparency of the e-voting procedure by verifying the results' integrity without other parties (either system users or malicious third parties which do not belong to the system) being able to observe the whole verification process.

In addition, existing privacy requirements already forming part of the organisation's goals are identified. For example, in the e-voting case the goal of authentication is 'hidden' in the current goal model and is realised by process 'P3: Send Authentication Means to eligible Voters'. It should be noted, that PriS assumes the existence of the organisation's current goal model. If not, a goal modelling method should be used for constructing the goal model prior to PriS's application [12].

(b) Analyse the impact of privacy goals on organisational processes The second step is to analyse the impact of privacy goals on processes and related support systems.

To answer this question, the first task is to identify the impact it may have on other organisational goals. This

[1] Since pseudonymity can be considered as part of anonymity, they are both addressed in one pattern.

impact may lead to the introduction of new goals or to the improvement/adaptation of existing goals. Introduction of new goals may lead to the introduction of new processes while improvement/adaptation of goals may lead to the adaptation of associated processes accordingly. Repeating this process for every privacy goal and its associated organisational goals leads to the identification of alternative ways for resolving privacy requirements. The result of this process modelled in the spirit of an extended AND/OR goal hierarchy. A summary of this process is shown in the Fig. 3.

For example, let us consider the privacy goal of unlinkability in the e-voting case. Guaranteeing voters' unlinkability will clearly impact the way that goal '$G_{1.1}$: Ensure the participation of all eligible voters' is realised. In particular, by applying unlinkability goal on $G_{1.1}$, this will have an impact on all subgoals that realise goal $G_{1.1}$. For every subgoal it is analysed which are the modifications that need to be done in order to satisfy the unlinkability goal. In the specific example, subgoals '$G_{1.1.1}$: Ensure all Voters are located' and '$G_{1.1.2}$: Update List of Voters' are maintained while goal '$G_{1.1.3}$: Provide e-access to all eligible Voters' needs to be adapted. Specifically, two new subgoals are introduced namely '$G_{1.1.1.1}$: Provide e-access' and 'Prevent others to reveal to whom the data are sent' as the result of the impact analysis. Finally, the process that realises these new subgoals is also adapted for accomplishing the realisation of the new privacy goal. The result of this analysis is graphically illustrated in Fig. 4.

(c) Model affected processes using privacy-process patterns Having identified the privacy-related processes the next step is to model them, based on the relevant privacy-process patterns. A detailed description of the seven privacy-process patterns can be found in Appendix 1.

Figure 5 presents the process pattern for addressing the unlinkability requirement, which describes the relevant activities needed to realise that process. The application of the unlinkability pattern on process 'P3: Send Authentication Means to eligible voters', which realises goals $G_{1.1.3.1}$ and $G_{1.1.3.2}$ as shown in Fig. 1, is presented next to the general pattern.

(d) Identify the technique(s) that best support/implement the above processes 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.

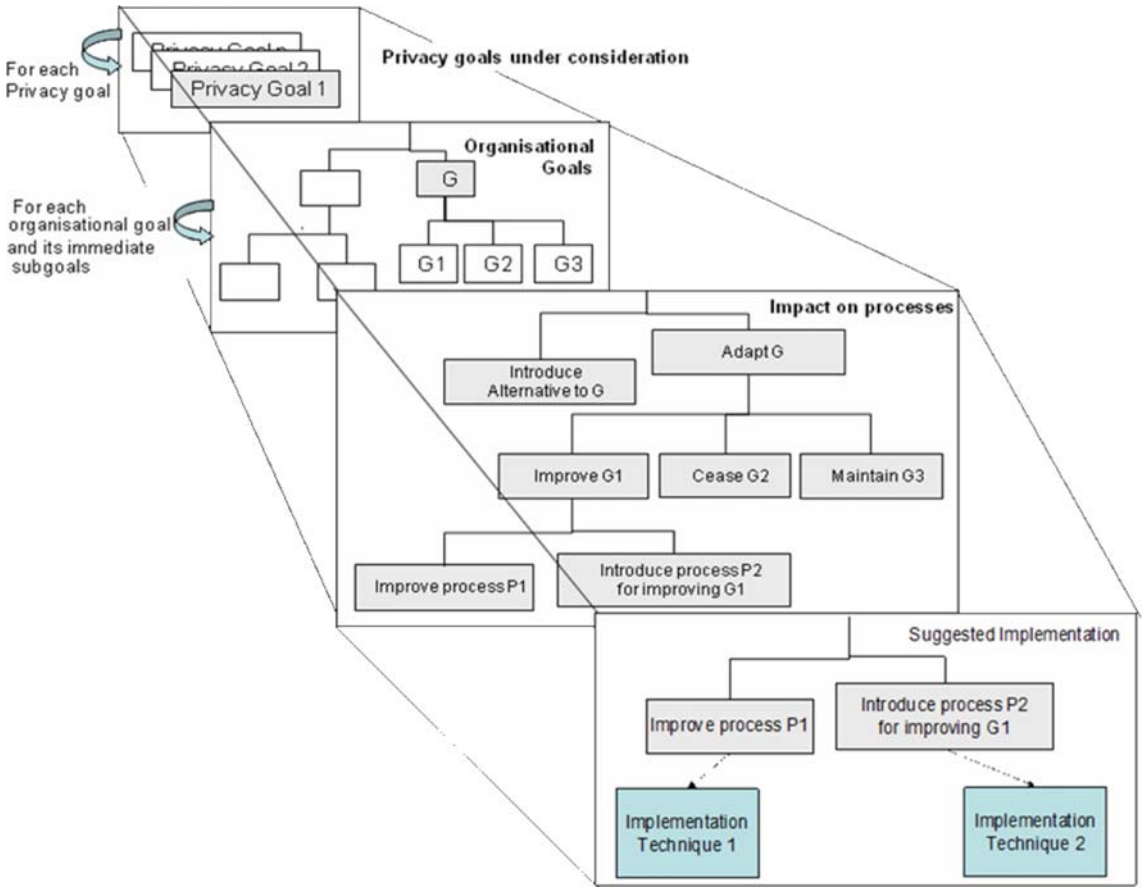

Fig. 3 Analyse the impact of privacy goals on business processes

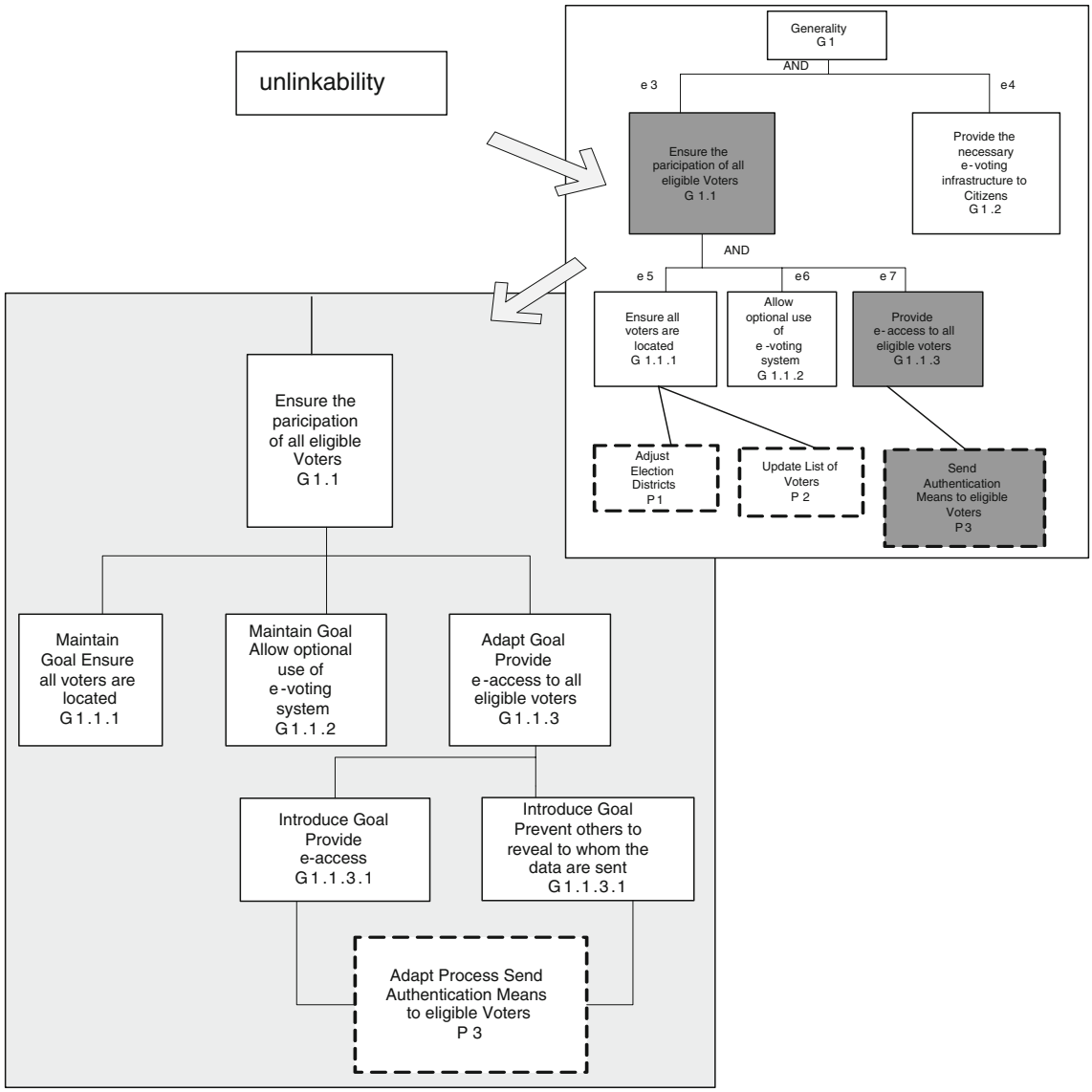

Fig. 4 Analyse the impact of unlinkability goal

In particular, every process pattern highlights the specific privacy-related activities that should be implemented thus, indicating the developer where the privacy implementation technique needs to be introduced in order to ensure that the process is privacy compliant. Naturally, the choice of the appropriate implementation technique depends on the privacy requirement(s) under consideration. The correspondence between privacy-process patterns and implementation tools is shown in Table 1.

As seen in Table 1, existing privacy implementation techniques are classified in six categories, namely: (1) Administrative tools, (2) Information tools, (3) Anonymizer products, services and architectures, (4) Pseudonymiser tools, (5) Track and evidence erasers, and (6) Encryption tools. An overview of these categories can be found in [13]. Each category includes a number of technologies-methodologies [13, 14]. For example, administrative tools include: Identity management, Biometrics, Smart Cards, Permission Management and Monitoring and Audit tools. Anonymizer tools include: Browsing Pseudonyms, Virtual E-mail addresses, Surrogate Keys as wells as a number of Privacy Enhancing Technologies (PETs) such as Crowds, Onion Routing, Gap, Tor, etc.

Different tools in each category implement specific privacy-process patterns. Using Table 1, a developer can choose for every process pattern which is/are the best implementation technique(s) among the ones available, always based on the privacy requirement(s) that needs to be realised, as well as the specific business context in which it will be implemented.

Therefore, instead of prescribing a single solution, PriS identifies a number of implementation techniques that best

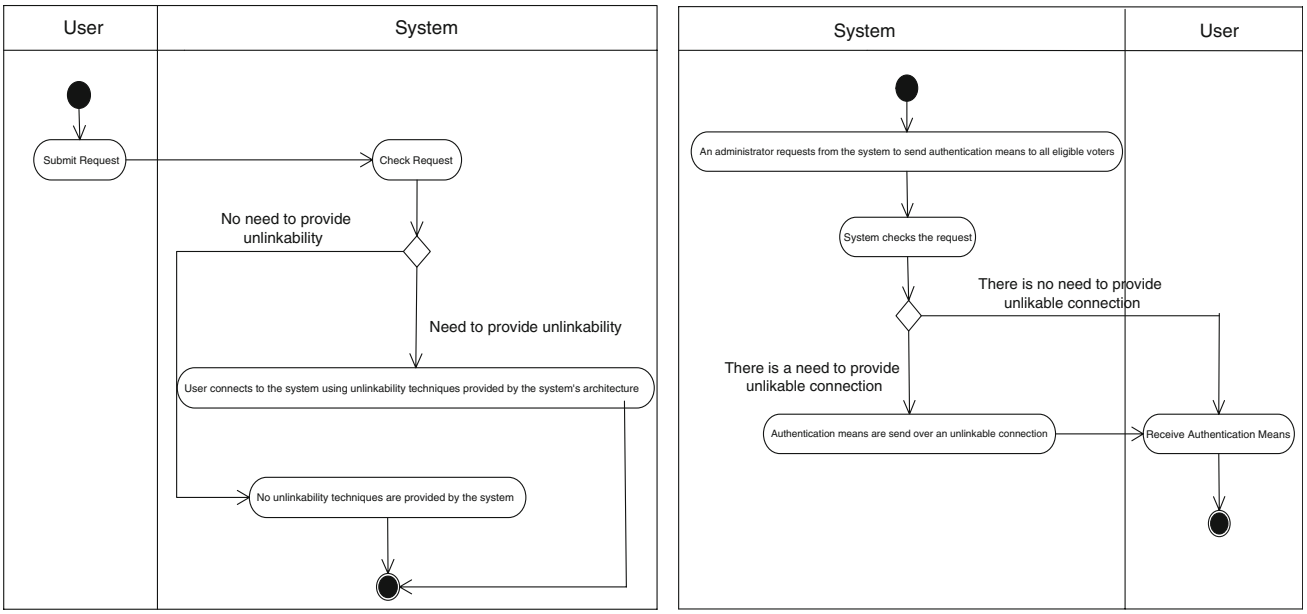

Fig. 5 Unlinkability pattern and its specialisation on the e-voting case

support the realisation of each privacy-related process letting the developer decide which architecture is best for the developing system based on organisation's priorities such as, cost, system efficiency, implementation complexity, etc.

In the e-voting case for the realisation of process 'P3: Send Authentication Means to eligible voters' the developer may choose the Tor architecture which satisfies the unlinkability pattern.

4 Formal PriS

Formal PriS aims to provide consistent, unambiguous and precise representations of all PriS concepts as well as to provide the basis for useful tool support for PriS activities. The following sections formally describe the four PriS activities described in Sect. 3.

4.1 Activity 1: elicit privacy-related goals

Let us start with the formal definition of the PriS goal model. The conceptual model of PriS uses a goal hierarchy structure and especially a goal graph structure since beside the AND/OR relationship, the CONFLICT/SUPPORT relationship exists which can be applied in goals belonging at the same level of the hierarchy. Thus, the goal model is defined as a directed acyclic graph as follows:

Definition 1 A directed acyclic graph $V = (G, E)$ is defined for representing the goal model. $V = (\{G_1, G_2, G_3,\ldots,G_{v-1}, G_v\}, \{E_1, E_2, E_3,\ldots,E_{m-1}, E_m\})$ where, $G_1,\ldots,G_n$ are the total of all system's goals and subgoals as they are defined by the system's stakeholders and $E_1,\ldots,E_m$ are the set of relationships between the identified goals.

The E set contains all the relationships between the goals of the hierarchy. Every relationship is defined by the pair of the connected goals and the type of their connection. Based on the conceptual model of PriS four types of connection exist: AND, OR, SUPPORT, and CONFLICT. Every relationship type is expressed by a number from 1 to 4. Number 1 represents the OR relationship, number 2 the AND, number 3 the SUPPORT and number 4 the CONFLICT. For example the relationship $e_i = (G_1, G_2, 2)$ defines an AND connection of goal G_1 with the goal G_2 and especially that G_1 is the more abstract goal and G_2 the more specific one. In a relationship, the more abstract goal is called *parent goal* where the more specific is called *child goal*. By defining the relationships among goals, the goal hierarchy is also defined since the more abstract goals belong in a higher level than their children.

Next we need to define which of the goals in the G set are affected by which privacy goal(s), (relationship HAS_IMPACT_ON). To this end, seven *privacy variables* are introduced namely PV1, PV2,…, PV7. Every privacy goal is expressed by a variable which can take two values, 0 and 1. Every goal G_i is assigned seven values which represent which privacy requirements have an impact on the specific goal and which do not.

If G_i is not an end goal (has child goals) then the privacy goals that affect goal G_i also affect all child goals of G_i regarding the type of relationship between them.

Table 1 Matching privacy patterns with implementation techniques

	Administrative tools				Information tools				Anonymizer products, services and architectures										
	Identity management	Biometrics	Smart cards	Permission management	Monitoring and audit tools	Privacy policy generators	Privacy policy readers	Privacy compliance scanning	Browsing pseudonyms	Virtual Email addresses	Trusted third parties	Surrogate keys	Crowds	Onion Routing	DC-nets	Mix-nets	Hordes	GAP	Tor
Authentication	X	X	X	X	X														
Authorization	X	X	X	X	X														
Identification	X	X	X	X	X														
Data protection	X	X	X	X	X	X	X	X											
Anonymity and/or pseudonymity	X	X	X	X					X	X	X		X	X	X	X	X	X	X
Unlinkability											X	X		X	X	X	X	X	X
Unobservability			X	X	X												X	X	X

	Pseudonymizer tools		Track and evident erasers				Encryption tools		
	CRM personalization	Application data management	Spyware detection and removal	Browser cleaning tools	Activity traces eraser	Harddisk data eraser	Encrypting Email	Encrypting transactions	Encrypting documents
Authentication									
Authorization									
Identification									
Data protection									X
Anonymity and/or pseudonymity	X	X	X			X			
Unlinkability	X	X	X	X	X	X			
Unobservability			X	X	X	X	X	X	X

Table 2 Adjacency Matrix representing the goal 'G1: Generality' and its subgoals

	G_1	$G_{1.1}$	$G_{1.2}$	$G_{1.1.1}$	$G_{1.1.2}$	$G_{1.1.3}$	$G_{1.1.3.1}$	$G_{1.1.3.2}$
G_1	0	2	2	0	0	0	0	0
$G_{1.1}$	0	0	0	2	2	2	0	0
$G_{1.2}$	0	0	0	0	0	0	0	0
$G_{1.1.1}$	0	0	0	0	0	0	0	0
$G_{1.1.2}$	0	0	0	0	0	0	0	0
$G_{1.1.3}$	0	0	3	0	0	0	2	2
$G_{1.1.3.1}$	0	0	0	0	0	0	0	0
$G_{1.1.3.2}$	0	0	0	0	0	0	0	0

4.2 Creating an adjacency matrix

An adjacency matrix is being used for representing the goal model. The first line and first column of the table consist of the goal names participating in the goal model. Every cell is assigned by one value between 0 and 4. The purpose of the matrix is to show which goals are being connected and their connection type. Thus, the goals in the lines represent the parent goals while the goals in the columns represent the child goals. When a cell contains the value of 0 indicates that there is no connection between the goal referred to the beginning of the line with the one referred to the beginning of the column. Otherwise, a number between 1 and 4 is assigned indicating that a connection between these goals does exist and the connection type is the one indicated by the number. An example of an adjacency matrix based on the goal graph of the e-voting case is shown in Table 2.

4.3 Activity 2: analyse the impact of privacy goals on organisational processes

First we need to identify and create a link between the privacy-related operationalised goals and the respective processes that realise these goals. At the end of this step two tasks are accomplished: the identification of privacy-related processes and the creation of the links between the privacy-related operationalised goals and these processes (relationship IS_REALISED in the conceptual model). For example, the link between the operationalised goal G1.1.3.2 and the process P3 will be expressed as follows:

$$\text{Match\_G\_P}(G_{1.1.3.2}) = (P_3).$$

Next we must identify which privacy-process patterns need to be applied not only for modelling these processes but also for relating them with the proper implementation techniques.

For the accomplishment of this purpose the concept of *process pattern variable* is introduced. Process pattern variables, PP1,…,PP7 share the same logic like privacy variables. In particular, every process is assigned seven values which are the values of the seven process pattern variables. On every process pattern variable, two values can be assigned: 1 and 0. The value 1 indicates that the respective process pattern will be applied on the specific process while value 0 indicates the opposite.

4.4 Activity 3: model affected processes using privacy-process patterns

As mentioned above, every process is assigned a number of process patterns variables, corresponding to the privacy goals affecting the process. Despite the fact that the values of privacy variables are assigned as one set, a classification among these variables exists. Specifically, the first four privacy goals are related with identification issues, while the last three have to do with anonymity issues. In other words, the first four privacy goals focus on protecting privacy by identifying each subject and granting privileges regarding the rights of this subject to the data that it tries to access, while the last three privacy goals focus on protecting the privacy of each subject by ensuring its anonymity or by preserving the reveal of its personal data by malicious third parties.

Based on this classification, the seven privacy variables' values of every operationalised subgoal are examined separately and different rules exist when selecting the proper privacy-process patterns. In particular, based on the privacy-process patterns' description the following statements are true: data protection > identification > authorisation > authentication and unobservability > unlinkability. The symbol ">" indicates that when an operationalised goal has two or more privacy requirements, the process patterns that will be selected are always the left in the equation. In other words, data protection process pattern involves the realisation of identification which involves the realisation of authorisation which involves the realisation of authentication. The same applies in the case between unlinkability and unobservability. Anonymity/pseudonymity is not involved in the realisation of any other process pattern. It should be mentioned that by the word involving it is meant that for the realisation of the identification process pattern, for example, the realisation of the authorisation process pattern is necessary. This is represented as identification > authorisation > authentication.

PriS combines the above cases and rules and returns as a result the values of the seven process pattern variables for every privacy-related process. Thus, for example, let assume that process P3 was realising, beside the unlinkability goal, the goals of authentication, authorisation and unobservability. Thus the operationalised goal would have the following privacy requirement values (1, 1, 0, 0, 0, 1, 1).

Based on the above rules the values of the seven privacy-process patterns applied on the specific process (P3) would have been (0, 1, 0, 0, 0, 0, 1).

As it was mentioned before, every process may realise more than one operationalised goals. In this case, before the selection of the proper process patterns that will be applied on the specific process, PriS identifies the maximum values between every privacy requirement variable of each subgoal and creates a virtual goal G′ that contains all seven maximum values. Thus, for example, if goal G_i with values (0, 1, 0, 0, 1, 0, 0) and goal G_t with values (1, 1, 0, 1, 1, 0, 1) are realised by the same process P_k, a new virtual goal G′ will be created with values (1, 1, 0, 1, 1, 0, 1). The selection of the proper process patterns will be based on the seven values of the G′ goal. In this way, process P_k will satisfy both goals, G_i and G_t.

Definition 2 $\forall G_i \in G$, and are realised by process P_k, a new goal G′ is created and is defined as follows:

$$G' = G^i \vee G^j \vee \cdots \vee G^k$$

$$PV'_l = \left[PV^i_l \vee PV^i_l \vee \ldots \vee PV^k_l\right]$$

where, $k =$ the number of operationalised goals realised by one process, $l = 1, 2,\ldots,7$ (seven privacy variables for every goal)

Based on the above definition, PriS takes the maximum value of every operationalised goal's privacy variable and creates G′ which constitutes the maximum values of every privacy variable.

4.5 Activity 4: identify the technique(s) that best support/implement privacy-related processes

For describing which implementation techniques realise which patterns, seven variables are assigned to every technique following the same logic as before.

Specifically, every implementation technique is assigned seven values, which represent which process patterns it realises according to Table 1. For example, for the Tor architecture the seven values describing it are (0, 0, 0, 0, 1, 1, 1) meaning that the specific architecture implements the anonymity and pseudonymity process pattern, the unlinkability and the unobservability process pattern.

PriS checks the privacy-process patterns that are applied on every process and for every pattern, it suggests a number of implementation techniques according to their respective values. PriS can either suggest a number of implementation techniques separately for every process pattern, or can suggest a number of techniques for all the identified process patterns. In the case where the combination of process patterns does not lead to a specific implementation technique, PriS suggests the techniques that realise most of the privacy-process patterns. It should be mentioned that PriS does not choose the best technique out of the suggested ones. This is done by the developer who has to consider other factors like cost, complexity etc. PriS guides the developer by suggesting a number of implementation techniques that satisfy the realisation of the privacy-process patterns identified in the previous step.

5 Discussion

A number of requirement engineering methodologies have been proposed for managing security issues during system design including NFR [15, 16], Tropos [17–19], KAOS [20], *i** [21], RBAC [22], M-N framework [23], GBRAM [24, 25]. The above methodologies do not address privacy specifically, but think of it as part of system security. As such they do not offer specific techniques for identifying privacy issues. Furthermore, the majority of the proposed methodologies (with the exception of GBRAM) focus on the elicitation of security requirements from business goals but neither do address how these requirements are translated into system components, nor do they suggest any relevant implementation techniques. RBAC is the only method, which considers the generation of system policies based on the elicited security requirements. However, it does not suggest a systematic way for eliciting and managing these requirements.

Bellotti and Sellen [26] developed a framework for privacy-aware design in the field of ubiquitous computing. This framework proposes a procedure; designers may follow through a set of questions in order to evaluate a system. The evaluation is accomplished by identifying a set of new requirements, which must be implemented by the developers. A recent variation of this framework is proposed by Hong et al. [27]. In spite of the fact, that these frameworks are inexpensive to use and relatively fast to implement, a number of disadvantages exist. First, they do not address/suggest any implementation techniques for realising the identified requirements. A gap between design and implementation exists since they do not suggest a way for guiding the developer from the design to the implementation level. Also these frameworks produce a static set of vulnerabilities (which the current system must overcome) and leave the designer to re-evaluate the entire system since they do not take iteration into account as part of the design process. Changing one part in the system's design may affect other multiple parts in terms of privacy. Based on the aforementioned vulnerabilities, these frameworks are more likely to be employed once at the end of the design cycle rather than become a part of the design process.

The STRAP framework proposed in [28] takes a further step compared to the previous frameworks. Specifically, it is based on the above frameworks while borrowing methods from requirements engineering and goal-oriented analysis. In particular, at the beginning STRAP performs a goal-oriented analysis of the system for identifying the relevant actors, goals and major system components. Then a list of vulnerabilities is produced by asking a number of questions similar to the ones proposed in [26, 27] on every goal and sub-goal. Vulnerabilities are categorised based on the four Federal Information Practices presented in [14]. Once vulnerabilities are identified the steps of refinement, evaluation and iteration follow. While STRAP successfully combines goal-oriented analysis and heuristic-based frameworks for addressing privacy vulnerabilities, it does not take the next step of discovering/suggesting the relevant implementation techniques needed for eliminating these vulnerabilities.

Alongside the research on requirements engineering methodologies, a number of technological solutions (architectures, tools and protocols) have been designed for protecting user's privacy. Specifically, Anonymizer [29] is a third-party web site, which acts as a middle layer between the user and the site to be visited providing user's anonymity. Crowds is an agent that has been designed also for protecting user's anonymity. It is based on the idea that people can be anonymous when they blend into the crowd [30, 31]. Onion Routing is a general-purpose infrastructure for private communications over a public network. It provides anonymous connections that are strongly resistant to both eavesdropping and traffic analysis [32, 33]. DC-Net (Dining Cryptographers Network) proposed in [34, 35] allows participants to send and receive messages anonymously in an arbitrary network. It can be used for providing perfect sender anonymity. Mix-Networks is another technique introduced in [36] and further discussed in [37]. It realises unlinkability of sender and recipient as well as sender anonymity against recipient and optionally recipient anonymity. Hordes is a protocol designed for utilising multicast communication for the reverse path of anonymous connections, achieving not only anonymity but also sender unlinkability and unobservability. A detailed description of Hordes is given in [38]. GAP (GNUnet's Anonymity Protocol) presented in [39] achieves anonymous data transfers. However, GAP is customised to the functionality of a peer-to-peer network. Finally, Tor, presented in [40] is an architecture based on the Onion Routing architecture with an improved way of working.

Unlike the above RE methodologies, PET's are usually addressed either directly at the implementation stage of the system development process or as an add-on long after the system is used by individuals. However, PET's focus on the software implementation alone, irrespective of the organisational context in which the system will be incorporated.

PriS method focuses on bridging the gap between the design and the implementation phases. As an input PriS uses the current goal model constructed by any of the aforementioned RE methodologies. In this paper, the EKD method is used for the construction of the goal model. However, different goal modelling methodologies may be used since, as discussed in [12], there is a high degree of compatibility between existing goal modelling methodologies. In order to identify privacy goals PriS uses as a guide the eight basic privacy concerns (authentication, authorization, identification, data protection, anonymity, pseudonymity, unlinkability and unobservability). Moreover, the activity of privacy goal elicitation could benefit from a number of other techniques such as: threat trees [41], attack trees [42], abuse cases [43], misuse cases [44–46], security use cases [45] and abuse frames [48]. The specific techniques can assist decision makers to locate and accurately define the organization's privacy goals that need to be realised by the system into consideration.

To assess PriS's applicability, the proposed method has been tried in two case studies concerning an e-voting system [6, 7, 49], as well as the University of the Aegean career office system [8]. The results indicate that PriS can be used to effectively link organizational privacy needs to alternative system implementations that satisfy these needs and can guide designers to make informed decisions regarding the choice of the most suitable technological solution. PriS's application also showed that there are a great number of repetitive tasks (e.g., the assignment of privacy variables to different goals in the goal model) and thus the need for automated tool support.

Another issue that emerged from the two case studies is related to the selection of implementation techniques. PriS deals with implementation techniques as solutions that either, satisfy the realisation of a privacy goal, or not. However, in many situations one has to take into consideration the degree to which every implementation technique realises every privacy need, in the specific organizational context, which can make selection of a solution very complicated. The complexity is increased by the fact that there are a number of privacy needs that should be satisfied at the same time. Defining a number of context-dependent selection criteria, as well as classifying implementation technologies using fuzzy logic techniques in order to calculate the degree to which each technology contributes to the satisfaction of a particular privacy requirement is a possible solution and is the subject of our current research efforts. In this way, instead of proposing solutions which either satisfies a specific set of privacy goals or not, alternative solutions can be ranked depending on the degree of the privacy protection they achieve.

6 Conclusions

This paper presents PriS, a method for incorporating privacy user requirements into the system design process. PriS considers privacy requirements as business goals in the organisation domain and provides a methodological framework for analysing the effect of privacy requirements onto the organisational processes using specific privacy-process patterns. Using these patterns, PriS accelerates the modelling of privacy-related business processes indicating where the application of privacy implementation techniques is needed, also suggesting a list of specific implementation techniques that can realise each privacy-related process. Therefore, PriS provides an integrated way-of-working from high-level organisational needs to the IT systems that realise them.

A formal description of the PriS methodological framework is also presented in this paper. Formal PriS is used for the technical expression of the PriS way of working. It provides a set of expressions based on which the whole process, from the goal level to the selection of the appropriate implementation techniques, is accomplished in a more directed way.

Future work includes the development of a case tool that will automatically identify the impact of privacy goal in the goal-process structure, based on PriS formal definition. The tool will also provide developers with a description of each implementation technique, as well as guidelines for applying the selected technique. In addition, as discussed in the above section we are currently working on improving the method of implementation technologies selection using fuzzy modelling.

Appendix 1

Figure 6 presents the process pattern for addressing the authentication requirement, which describes the relevant activities needed to realise that process. Every time a user submits a request to the system, the system should check that request and if authentication is needed the user should provide the proper authentication data based on which access is granted or denied.

The authorisation process pattern is presented in Fig. 7. According to the authorisation requirement, user's private data should only be accessed by authorised users. Therefore, initially when a user submits a request to a system the nature of the request should first be checked since it is not legal for example to ask from that user to login for a service that identification is not needed. If the user requests specific services or access to data that need authorisation then he/she should pass the authentication process and

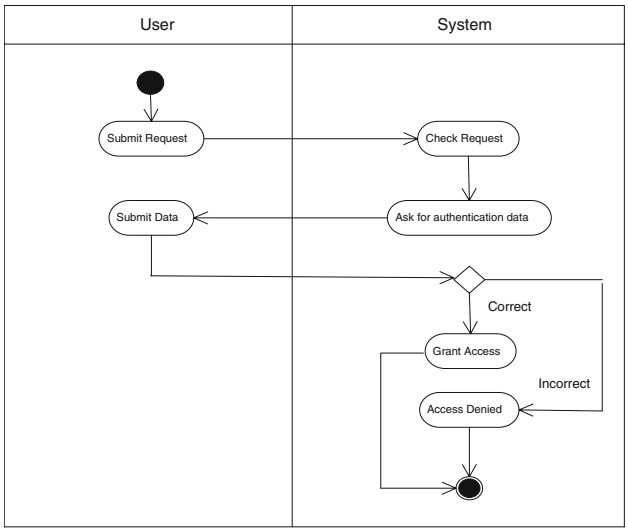

Fig. 6 Authentication pattern

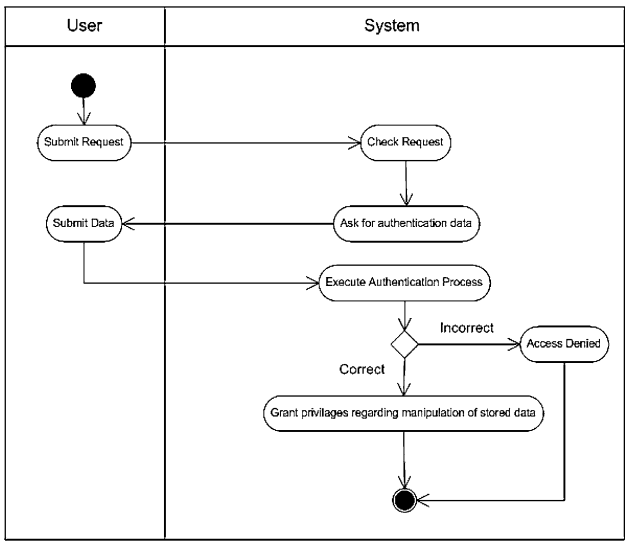

Fig. 7 Authorisation pattern

then, according to his/her rights, get the privileges for accessing or not the specific service or data.

The pattern corresponding to the identification requirement is presented in Fig. 8. The role of identification is twofold; first to protect both the user that accesses a resource or service and the user's data that are stored in the system and second to allow only authorised people to access them.

As shown in Fig. 8, when a user submits a request the identification process checks whether identity is required or not. If identity is not needed the system returns the information requested to the user without asking any kind of digital identity. If the request is related to accessing private

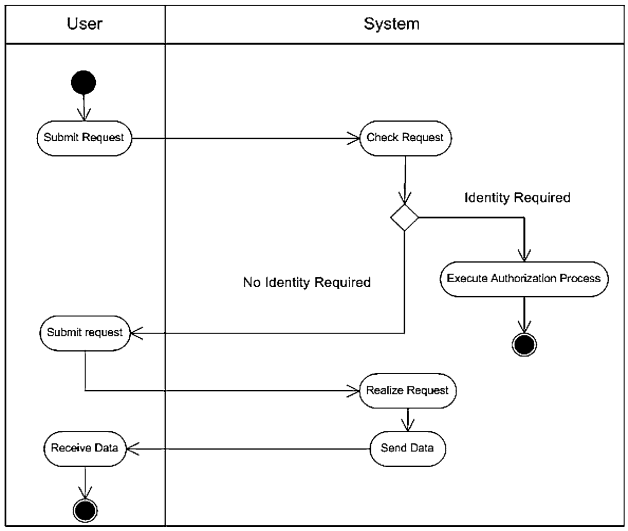

Fig. 8 Identification pattern

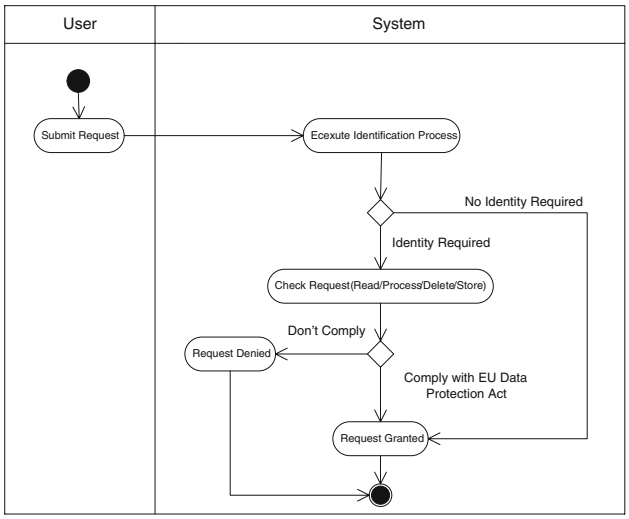

Fig. 9 Data protection pattern

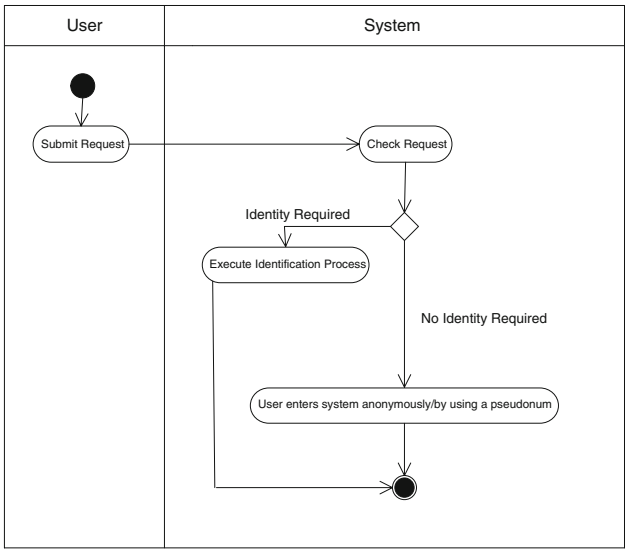

Fig. 10 Anonymity and pseudonymity pattern

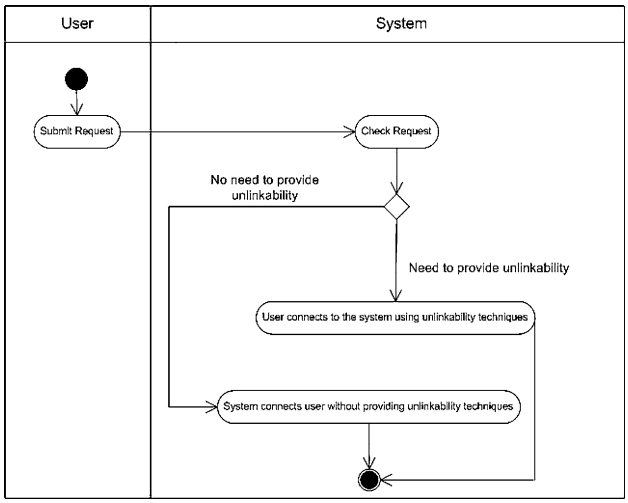

Fig. 11 Unlinkability pattern

information or accessing personalised services, then the process of authorisation is triggered. It should be noted that user anonymity is not ensured since this is not an anonymity service, just a transaction without providing identities. If anonymity is also required then the relevant process pattern described below, should also be applied.

Figure 9, presents the data protection process pattern. The aim here is to ensure that every transaction with personal data is realised according to the system's privacy regulations and Directive 95/46/EU [49] regarding the processing of personal data and the free movement of such data.

When a user tries to access private data, an identification process is triggered for identifying the user and for granting him/her with the rights of reading, processing, storing, or deleting private data. Subsequently, if the user asks to perform any of the above tasks the system checks whether this complies with the privacy regulations and the request is either granted or denied, accordingly. Thus, there are two intermediate "inspections" before actually a user is able to perform various tasks on other users' private data.

The next pattern (Fig. 10), addresses the anonymity and pseudonymity requirements. These two are addressed in one pattern since pseudonymity could be considered as part of anonymity.

As shown in Fig. 10, first, the user's request is checked in order to decide whether or not identity is needed. If there is a need for knowing user's identity, the identification process is triggered. If not, the user not only receives his/her information without providing any personal data, but

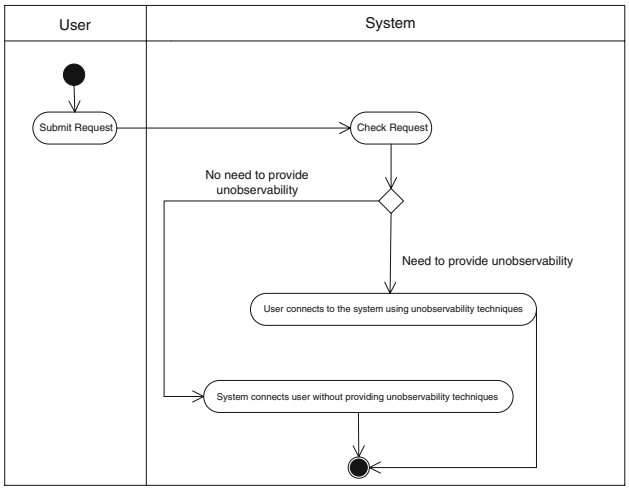

Fig. 12 Unobservability pattern

also specific techniques for protecting his/her anonymity are realised. Thus, identification may be a subpart of anonymity depending on whether or not specific data of user's identity are asked for processing. On the other hand, anonymity is a privacy requirement that needs to be protected and specific technologies should be used to realise user's anonymisation while he/she accessing the system and also during the whole communication. Pseudonymity is used when anonymity cannot be provided but again for the purpose of protecting user's anonymity.

Finally, the patterns for unlinkability and unobservability requirements (Figs. 11, 12, respectively) are presented below. The two patterns have a similar structure. User asks for a request. Based on system's requirements if one or both of these requirements need to be realised, then, appropriate unlinkability or unobservability techniques are used for connecting the user to the system.

References

1. Lunheim R, Sindre GS (1994) Privacy and computing: a cultural perspective. Security and control of information technology. In: Sizer R (ed) A Society (A-43)/x. Elsevier, North Holland, pp 25–40
2. Fischer-Hübner S (2001) IT-security and privacy, design and use of privacy enhancing security mechanisms. Lect Notes Comp Sci, vol. 1958. Springer, Berlin
3. Cannon JC (2004) Privacy, what developers and IT professionals should know. Addison-Wesley, Reading
4. Koorn R, van Gils H, ter Hart J, Overbeek P, Tellegen R Privacy Enhancing Technologies, White paper for Decision Makers. Ministry of the Interior and Kingdom Relations, the Netherlands, December 2004
5. University of the Aegean, E-Vote: an Internet-based electronic voting system. University of the Aegean, Project Deliverable D 7.6, IST Programme 2000#29518, 21 October 2003, Samos
6. Kavakli E, Gritzalis S, Kalloniatis C (2007) Protecting privacy in system design: the electronic voting case. Transf Gov People Process Policy 1(4):307–332. doi:10.1108/17506160710839150
7. Kavakli E, Kalloniatis C, Loucopoulos P, Gritzalis S (2006) "Incorporating Privacy Requirements into the System Design Process: The PriS Conceptual Framework", Internet research, special issue on privacy and anonymity in the digital era: theory. Technol Pract 16(2):140–158
8. Kalloniats C, Kavakli E, Gritzalis S (2005) Dealing with privacy issues during the system design process, 5th IEEE International Symposium on Signal Processing and Information Technology, 18–21 December 2005, Athens, Greece
9. Loucopoulos P, Kavakli V (1999) Enterprise knowledge management and conceptual modelling. LNCS, vol. 1565. Springer, Berlin, pp 123–143
10. Loucopoulos P (2000) From information modelling to enterprise modelling. In: Information systems engineering: state of the art and research themes. Springer, Berlin, pp 67–78
11. Kalloniatis C, Kavakli E, Gritzalis S (2007) Using privacy process patterns for incorporating privacy requirements into the system design process, Workshop on Secure Software Engineering (SecSe 2007) in conjunction with the International Conference on Availability, Reliability and Security (ARES 2007), April 2007, Vienna, Austria
12. Kavakli V (2002) Goal oriented requirements engineering: a unifying framework. Req Eng J 6(4):237–251. Springer, London
13. META Group Report v1.1 (2005) Privacy Enhancing Technology. March 2005
14. Code of Fair Information Practices (The) (1973), US Department of Health, Education and Welfare
15. Chung L (1993) Dealing with Security Requirements during the development of Information Systems, CaiSE '93, The 5th International Conference of Advanced Information System Engineering. Paris, France, pp 234–251
16. Mylopoulos J, Chung L, Nixon B (1992) Representing and using non-functional requirements: a process oriented approach. IEEE Trans Softw Eng 18:483–497. doi:10.1109/32.142871
17. Liu L, Yu E, Mylopoulos J (2003) Security and privacy requirements analysis within a social setting, 11th IEEE International Requirements Engineering Conference (RE'03), Monterey Bay, California, USA, pp 151–161
18. Mouratidis H, Giorgini P, Manson G (2003) An ontology for modelling security: the Tropos project, Proceedings of the KES 2003 Invited Session Ontology and Multi-Agent Systems Design (OMASD'03), UK, University of Oxford, Palade V, Howlett RJ, Jain L (eds) Lecture Notes in Artificial Intelligence 2773, Springer 2003, pp 1387–1394
19. Mouratidis H, Giorgini P, Manson G (2003) Integrating Security and Systems Engineering: towards the modelling of secure information systems, CAiSE '03, LNCS 2681. Springer, Berlin, pp 63–78
20. van Lamsweerde A, Letier E (2000) Handling obstacles in goal-oriented requirements engineering. IEEE Trans Softw Eng 26:978–1005. doi:10.1109/32.879820
21. Liu L, Yu E, Mylopoulos J (2002) Analyzing security requirements as relationships among strategic actors, (SREIS'02), e-proceedings available at http://www.sreis.org/old/2002/finalpaper9.pdf, Raleigh, North Carolina
22. He Q, Antón IA (2003) A Framework for modelling privacy requirements in role engineering, International Workshop on Requirements Engineering for Software Quality (REFSQ), 16–17 June 2003, Austria Klagenfurt/Velden, pp 115–124
23. Moffett DJ, Nuseibeh AB (2003) A framework for security requirements engineering. Report YCS 368, Department of Computer Science, University of York
24. Antón IA (1996) Goal-based requirements analysis, ICRE '96 IEEE Colorado Springs, Colorado, USA, pp 136–144
25. Antón IA, Earp BJ (2000) Strategies for developing policies and requirements for secure electronic commerce systems. 1st ACM

Workshop on Security and Privacy in E-Commerce (CCS 2000), 1–4 November 2000, unnumbered pages
26. Bellotti V, Sellen A (1993) Design for privacy in ubiquitous computing environments. In: Michelis G, Simone C, Schmidt K (eds) Proceedings of the Third European Conference on Computer Supported Cooperative Work—ECSCW 93, pp 93–108
27. Hong JI, Ng J, Lederer S, Landay JA (2004) Privacy risk models for designing privacy-sensitive ubiquitous computing systems, Designing Interactive Systems, Boston MA
28. Jensen C, Tullio J, Potts C, Mynatt DE (2005) STRAP: a structured analysis framework for privacy, GVU Technical Report
29. Anonymizer, available at www.anonymizer.com
30. Reiter KM, Rubin DA (1998) Crowds: anonymity for web transactions. ACM Trans Inf Syst Secur 1(1):66–92. doi:10.1145/290163.290168
31. Reiter KM, Rubin DA (1999) Anonymous web transactions with crowds. Commun ACM 42(2):32–38. doi:10.1145/293411.293778
32. Reed M, Syverson P, Goldschlag D (1998) Anonymous connections and Onion Routing. IEEE J Sel Areas Comm 16(4):482–494. doi:10.1109/49.668972
33. Goldschlag D, Syverson P, Reed M (1999) Onion Routing for anonymous and private Internet connections. Commun ACM 42(2):39–41. doi:10.1145/293411.293443
34. Chaum D (1985) Security without identification: transactions systems to make Big Brother Obsolete. Commun ACM 28(10):1030–1044. doi:10.1145/4372.4373
35. Chaum D (1988) The dining cryptographers problem: unconditional sender and recipient untraceability. J Cryptol 1(1):65–75. doi:10.1007/BF00206326
36. Chaum D (1981) untraceable electronic mail, return addresses, and digital pseudonyms. Commun ACM 24(2):84–88. doi:10.1145/358549.358563
37. Pfitzmann A, Waidner M (1987) Networks without user Observability. Comput Secur 6(2):158–166
38. Shields C, Levine NB (2000) A protocol for anonymous communication over the Internet. In: Samarati P, Jajodia S (eds) Proceedings of the 7th ACM Conference on Computer and Communications Security. ACM Press, New York, 33–42
39. Bennett K, Grothoff C (2003) GAP-Practical Anonymous networking. Proceeding of the Workshop on PET2003 Privacy Enhancing Technologies. Available at http://www.citeseer.nj.nec.com/bennett02gap.html
40. Dingledine R, Mathewson N, Syverson PT (2004) The second-generator Onion Router. Proceedings of the 13th USENIX Security Symposium, San Diego, CA, USA
41. Amoroso EG AT&T Bell Laboratories (1994) Fundamentals of computer security technology. P.T. R. Prentice Hall, ISBN 0-13-108929-3
42. Schneier B (1999) Attack trees 21–29. Dr Dobb's J Softw Tools 24 12(12):21–29
43. John MCF (1999) Using abuse case models for security requirements analysis, 15th Annual Computer Security Applications Conference (ACSAC '99), pp 55
44. Sindre G, Opdahl AL (2005) Eliciting security requirements with misuse cases. Requir Eng 10(1):34–44. doi:10.1007/s00766-004-0194-4
45. Sindre G, Opdahl AL (2002) Templates for misuse case description. In: Proceedings of the Seventh International Workshop on Requirements Engineering: Foundations for Software Quality—REFSQ'2001, Camille BA, et al (eds) Essener Informatik BeitrÃge, University of Essen, Germany, pp 125–136
46. Alexander I (2003) Use/misuse case analysis elicits non-functional requirements. Comput Contr Eng J 14(1):40–45. doi:10.1049/cce:20030108
47. Firesmith D (2003) Security use cases. J Object Technol 2(1):53–64
48. Lin L, Nuseibeh B, Ince D, Jackson M, Moffett JD (2003) Introducing abuse frames for analysing security requirements. Requirements Engineering 2003, 11th IEEE International Conference on Requirements Engineering (RE 2003), 8–12 September 2003, Monterey Bay, CA, USA. IEEE Computer Society 2003, pp 371–372
49. European Parliament and the Council: Directive 95/46/EC on the protection of individuals with regard to the processing of personal data and of the free movement of such data. October 1995