Applied Soft Computing 11 (2011) 4341–4348

Contents lists available at ScienceDirect

Applied Soft Computing

journal homepage: www.elsevier.com/locate/asoc

A soft computing approach for privacy requirements engineering: The PriS framework

Christos Kalloniatis [a,*], Petros Belsis [b], Stefanos Gritzalis [c]

[a] Cultural Informatics Laboratory, Department of Cultural Technology and, Communication University of the Aegean, Harilaou Trikoupi & Faonos Str., 81100 Mytilene, Greece
[b] Technological Education Institute of Athens, Department of Marketing, Agiou Spyridonos Street, 12210 Aigaleo, Athens, Greece
[c] Information and Communication Systems Security Laboratory, Department, of Information and Communications Systems Engineering, University of the Aegean, 83200 Samos, Greece

ARTICLE INFO

Article history:
Received 28 April 2010
Received in revised form
29 September 2010
Accepted 3 October 2010
Available online 30 October 2010

Keywords:
Privacy requirements engineering
Fuzzy logic
Case study
Soft computing
PriS method
Security requirements

ABSTRACT

Soft computing continuously gains interest in many fields of academic and industrial domain; among the most notable characteristics for using soft computing methodological tools is the ability to handle with vague and imprecise data in decision making processes. Similar conditions are often encountered in requirements engineering. In this paper, we introduce the PriS approach, a security and privacy requirements engineering framework which aims at incorporating privacy requirements early in the system development process. Specifically, 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 of PriS uses a goal hierarchy structure. Every privacy requirement is either applied or not on every goal. To this end every privacy requirement is a variable that can take two values [0,1] on every goal meaning that the requirements constraints the goal (value 1) or not (value 0). Following this way of working PriS ends up suggesting a number of implementation techniques based on the privacy requirements constraining the respective goals. Taking into account that the mapping of privacy variables to a crisp set consisting of two values [0,1] is constraining, we extend also the PriS framework so as to be able to address the degree of participation of every privacy requirement towards achieving the generic goal of privacy. Therefore, we propose a fuzzification of privacy variables that maps the expression of the degree of participation of each privacy variable to the [0,1] interval. We also present a mathematical framework that allows the concurrent management of combined independent preferences towards the necessity of a privacy measure; among the advantages of the presented extended framework is the scalability of the approach in such a way that the results are not limited by the number of independent opinions or by the number of factors considered while reasoning for a specific selection of privacy measures.

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.

Nowadays, protecting privacy is focused on reducing the information collected and stored to a minimum, and deleting the information as soon as it has served its purpose. Most of today's e-services are relying on stored data, identifying the customer, his preferences and previous record of transactions. However, combining such data will in many cases constitute an invasion of privacy.

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*, *authorization*, *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 minimize 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

* Corresponding author.
E-mail addresses: ch.kalloniatis@ct.aegean.gr (C. Kalloniatis), pbelsis@cs.teiath.gr (P. Belsis), sgritz@aegean.gr (S. Gritzalis).

1568-4946/$ – see front matter © 2010 Elsevier B.V. All rights reserved.
doi:10.1016/j.asoc.2010.10.012

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 organizational context in which the system will be incorporated. This lack of knowledge makes it difficult to determine which software solution best fits the organizational needs. A review on a number of well-known privacy requirements engineering methods can be found in Ref. [5].

To this end, PriS, a new security requirements engineering methodology, has been introduced aiming to incorporate privacy requirements early in the system development process. PriS models privacy requirements in terms of business goals and uses the concept of privacy process patterns for describing the impact of privacy goals onto the business processes and the associated software systems supporting these processes.

The conceptual model of PriS uses a goal hierarchy structure. Every privacy requirement is either applied or not on every goal. The representation of a privacy requirement that constraints a goal is achieved by the use of a variable which can take two values, zero and one. If one of the privacy requirements is applied on a specific goal the respective privacy variable will be assigned with the value of one otherwise will remain zero which was also its initial value. Thus, on every privacy-related goal seven privacy variables are applied and representing which privacy requirements constraint the goal and which not (since pseudonymity can be considered as part of anonymity, they are both addressed in one pattern). Following this way of working PriS ends up suggesting a number of implementation techniques based on the privacy requirements constraining the respective goals. While PriS successfully guides the developers through the implementation phase by suggesting a number of implementation techniques it fails to address the degree of participation of every privacy requirement for achieving the generic goal of privacy.

The contributions of this paper are the following: We present the PriS conceptual framework for privacy management in requirements engineering, along with a formal representation of PriS; we extend also the presented framework using a soft computing approach that enables the expression of preferences from independent participants in the system design process and their combined management using fuzzy metrics; the presented approach has also the advantage that it is not limited by the number of factors considered while evaluating a metric nor by the number of preferences considered.

The rest of this paper is structured as follows: Section 2 describes the PriS conceptual framework and way of working. Formal PriS is presented in Section 3. Section 4 presents the fuzzy extension of PriS. Finally, Section 5 concludes with pointers to future work.

2. The PriS method

2.1. PriS conceptual framework

PriS [6] 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.

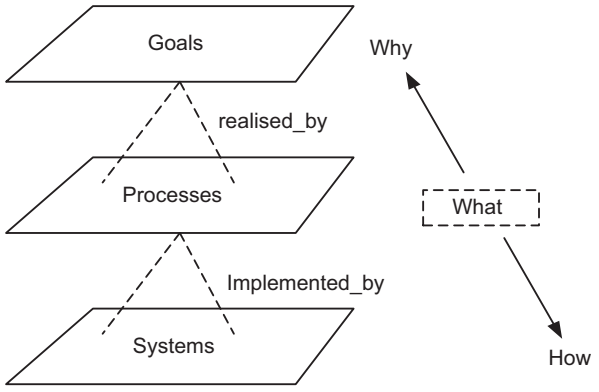

Fig. 1. The EKD schema.

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 is based on the Enterprise Knowledge Development (EKD) framework [7,8], which is a systematic approach to developing and documenting organisational knowledge. This is achieved through the modelling of: (a) organisational goals that express the intentional objectives that control and govern its operation, (b) the 'physical' processes, that collaboratively operationalise organisational goals and (c) the software systems that support the above processes.

The EKD generic schema is shown in Fig. 1. As shown in Fig. 1, processes represent WHAT needs to be done, goals justify WHY the associated processes exist, while systems describe HOW processes can be implemented in terms of appropriate system architectures.

In this way, a connection between system purpose and system structure is established.

Based on this schema, PriS models privacy requirements as a special type of goal (privacy goals) which constraint the causal transformation of organisational goals into processes. From a methodological perspective reasoning about privacy goals comprises of the following activities: (a) elicit privacy-related goals, (b) analyse the impact of privacy goals on business processes, (c) model affected processes using privacy process patterns and (d) identify the technique(s) that best support/implement the above processes. The PriS way-of-working is described in the following section.

2.2. The PriS way of working

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 [9]. 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.

3. Formal PriS

The following sections formally describe the four activities mentioned in Section 2.1.

3.1. Elicit privacy related goals

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.

$$V = (\{G_1, G_2, G_3, \ldots\ldots, G_{v-1}, G_v\}, \{E_1, E_2, E_3, \ldots\ldots, E_{m-1}, E_m\})$$

whereby, $G_1 \ldots G_n$ 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. 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.

The goal model is represented by an adjacency matrix. 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.

3.2. Analyse the impact of privacy goals on business 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).

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, indicates whether the respective process pattern will be applied on the specific process or not.

3.3. 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 revelation 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. Authentication process pattern is applied only in the cases where the specific request is supposed to be accessed only by authenticated users. Authorisation is applied when different categories of data exist where authenticated users need certain rights for accessing certain services. The identification pattern which realises the respective requirement has a twofold role. Firstly to protect both the user that accesses a resource or service and the user's data that are stored in the system and secondly to allow only authorized people to access them. Specifically, 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 information or accessing personalized services then the process of authorization 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, should also be applied. Regarding the data protection process pattern the aim is to ensure that every transaction with personal data is realized according to the system's and European's privacy regulations. 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. 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 identification for example the realisation of authorisation is necessary and for the authorisation the realisation of authentication. 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.

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.

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

$$G' = G^i \vee G^j \vee \ldots \vee G^k$$
$$PV'_l = [PV^i_l \vee PV^j_l \vee \ldots \vee PV^k_l]$$

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 of the maximum values of every privacy variable.

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

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.

PriS has been applied in an e-voting case study [6] and the main drawback that arises is the non-flexible way that the method suggests the implementation techniques which satisfy the identified process patterns. Our main issue is to expand PriS by using fuzzy theory for overcoming this drawback.

4. Soft computing for requirements evaluation

4.1. Preliminary concepts on sets and fuzzy set theory

In this section, we present in brief the basic concepts and notations of set theory and fuzzy set theory. We use upper-case notation to denote sets for example A denotes a set. In order to denote that x is an element of a set A we use the notation $x \in A$. For every crisp set X there exists a characteristic membership function f that maps elements of X to the set $\{0,1\}$. For example, $f(x) = 0$ means that x does not belong to A where $f(x) = 1$ means that x is a member of A.

If every member of set A is also a member of set B ($\forall x \in A \Rightarrow x \in B$) then A is called a subset of B and we denote this using the notation $A \subseteq B$. If we have $A \subseteq B$ and A, B are not equal or mathematically written $A \neq B$, then we call A to be a proper subset of B, or alternatively $A \subset B$. We denote the empty set, or in other words the set that contains no elements with the symbol $\emptyset$. The union of sets A and B is a set that contains all the elements that belong to any of the two sets. We use the symbol $\cup$ to denote the union of two sets; more specifically we have $A \cup B = \{X | x \in A \text{ or } x \in B\}$. The generalization of the union operation to more than two sets can be defined as $\bigcup_{k \in K} A_k = \{k | k \in A_k \text{ for some } k \in K\}$. In a similar manner, we denote as intersection $\cap$ the set that comprises by all the common elements of two or more sets. We use the notation $A \cap B = \{X | x \in A \text{ and } x \in B\}$. The generalized intersection for more than two sets can be written in a similar manner as previously: $\bigcup_{k \in K} A_k = \{k | k \in A_k \text{ for any } k \in K\}$.

Fuzzy set theory provides us with the possibility to represent one form of uncertainty. For non-infinite sets which we use in our approach, the following definition holds [10]:

Definition 3. Given a universal set X and a nonempty family $\wp$ of subsets of X, a fuzzy measure on $\langle X, \wp \rangle$ is a function g: $\wp \rightarrow [0,1]$ that satisfies the requirements:

(1) $g(\emptyset) = 0$ and $g(X) = 1$
(2) for all A, B, $\in \wp$, if $A \subseteq B$ then $g(A) \leq g(B)$

The first requirement states that always the element that we examine does not belong to the empty set and always belongs to the universal set. The second requirement expresses the fact that the evidence of an element being part of a set must be at least as great as the evidence that the element belongs to any subset of it. It is obvious that fuzzy measures are generalizations of probability measures as expressed in their classical form; such generalizations are useful in cases of uncertainty and will be used in the following sections.

4.2. Methodology

In order to avoid the non-flexible nature that characterizes crisp sets, fuzzy set theory provides an alternative means to express situations with uncertainty. It is hard to believe that in most cases the importance of a requirement can be explicitly determined; instead, it is more realistic to expect that we will be able to our willingness to incorporate some feature on a given scale. To this end, fuzzy theory provides the theoretical tools to handle such uncertainty, expressing a degree of satisfaction of a given requirement expressed on a given scale expressed in the [0,1] interval.

Fuzzy measures help interpret vague, imprecise and in general data that may not follow some well expected behaviour; as such we may also consider data that are associated with subjective opinions, for example an evaluation from a human expert. In that case the metrics under consideration need to be interpreted on grounds of statistical or other appropriate methodological tools such as those provided by evidence theory [10].

We present the mathematical background and expressions and apply these theoretical tools to help us overcome the impreciseness that is the outcome of cooperation between different partners with different needs and experiences in a software project. In such a case we argue that the determination of different priorities while

assessing the requirements of a given implementation often leads to a conflicting situation due to the presence of different estimations; such a conflict may be overcome by means of an overall evaluation of each participant's preferences and the related support on its usefulness.

Therefore we argue that using fuzzy measures and evidence theory, we can determine the degree of support towards a given fact; thus, considering that different participants determine their preference towards a given security requirement, we can determine a joint degree of satisfaction and determine the most critical to implement while others that do not have a strong joint preference may be neglected.

In evidence theory, of major importance are belief measures, which can be defined as a function mapping a given set to the [0,1] interval: $Bel{:}P(X) \rightarrow [0,1]$. The belief measure may be interpreted as the degree of confidence that a fact is true or that a given element belongs to a set. It is obvious that if X is the set for which its subjects are considered, then the following relations stand: $Bel(\emptyset) = 0$, $Bel(X) = 1$, $Bel(A_1 \cup A_2 \cup \ldots \cup A_n) \geq \sum_i Bel(A_i) - \sum_{i<k} Bel(A_i \cap A_k) + \ldots + (-1)^{n+1} Bel(A_i \cap A_2 \cap \ldots \cap A_n)$ for all subsets A of X (1).

Considering the facts $A_1, A_2, ..., A_n$, are pair-wise disjoint the inequality (1) requires that the belief required with the union of the sub-sets is no smaller than the sum of belief pertaining to each individual set.

Theorem. *The inequality relation supports the monotonicity requirement necessary for all fuzzy functions.*

Proof. Consider the inequality relation for $n = 2$.

If: $B = A_1 - A_2$, then $A_1 = B \cup A_2$ and $Bel(A_2) \cap Bel(A_1) \geq Bel(A_2) + Bel(B) - Bel(A_2 \cap B)$ and because $Bel(A_2 \cap B) = Bel(\emptyset) = 0$ we have $Bel(A_1) \geq Bel(A_2) + Bel(B) \geq Bel(A_2)$ which proves the monotonicity property for two independent sets. □

The Belief metric can be represented by a function m: $P(X) \rightarrow [0,1]$, such that $m(\emptyset) = 0$ and $\sum m(A) = 1$. Function $m(A)$ expresses the proportion to which available evidence supports the claim that a particular element belongs to A. In other words the relation between the metric and the supporting function can be expressed as:

$$Bel(A) = \sum_{B|B \subseteq A} m(B). \tag{2}$$

The utility of the aforementioned measures is considerable in case that the evidence comes from independent sources (for example from independent evaluators). We are interested in estimating the joint estimation $m_{1,2}$ by considering in our calculations the independent assignments to values m_1, m_2 from two independent sources.

Therefore in order to calculate $m_{1,2}$ for the set A considering the evidence that focuses on subset $B \in P(X)$ and on the subset $C \in P(X)$ the following sum of products needs to be calculated:

$$\sum_{B \cap C = A} m_1(B) \cdot m_2(C) \tag{3}$$

for all $A \neq \emptyset$. Since $m_{1,2}(\emptyset)$ should equal to 0, we need to exclude the following sum of products of these subsets who's intersection results in the empty set: $\sum_{B \cap C = \emptyset} m_1(B) \cdot m_2(C)$. Since $\sum_{A \in P(X)} m(A) = 1$, the combined evidence we are seeking is calculated if we subtract the value $\sum_{B \cap C = \emptyset} m_1(B) \cdot m_2(C)$ from 1 resulting in: $1 - \sum_{B \cap C = \emptyset} m_1(B) \cdot m_2(C)$. For normalization purposes the final result for the combined evidence $m_{1,2}(A)$ is given by the following equation:

$$m_{1,2} = \frac{\sum_{B \cap C = A} B \cap C = \emptyset}{\sum_{B \cap C = \emptyset} B \cap C = \emptyset}. \tag{4}$$

We need at this point to remark that while the value $m(A)$ characterizes the degree of evidence that the specific element under consideration belongs to a specific set A, the belief metric $Bel(A)$ characterizes the total evidence or belief that the element under consideration belongs to set A or to any of the specific subsets of A; this is apparent from the fact that the values for the belief metric are extracted based on the estimation of the m metric, according to Eq. (2).

4.3. Applicability in the software design process

The aforementioned metrics may be utilized in the system design process as a need to handle the uncertainty and different estimations related with either the nature of software projects or with the different perceptions of project contributors in respect to the necessity and applicability of patterns that implement certain goals. We seek to formalise a calculation of a joint combination of two independent sources that reason over the usefulness of a measure, based on different estimations that take into account different factors. Then, using Eq. (1) we make a joint calculation. The combined evidence facilitates the selection of appropriate technologies that realize the necessary process patterns. We consider that we have two different parties that contribute to the selection process.

In a given project we consider that a given set of requirements is implemented by a integrating a given number of measures. As X we may consider the universal set of measures that implement a specific requirement. We consider next the subsets N, A and C that: (a) the first subset N includes the measures that are by presumptive evidence essential in implementing a specific requirement, (b) the set A includes the measures that are cost-efficient (affordable) and provide a value for money and (c) is the set of measures that their complexity is such that allows their integration into a given software project.

We consider a project that will be implemented by different partners, each one of which evaluates with a different priority the necessity for specific integration of appropriate technologies that implement a given target.

For example, we consider the authentication process pattern: From the relative table of related technologies we select the following:

- Identity management
- Biometrics
- Smart cards
- Permission management
- Monitoring and auditing tools

For each of the available technologies the two parties will evaluate based on the three criteria we previously described: the necessity of a measure, the cost for its implementation and the complexity for its development. For each of the two parties we will consider that each one assigns a value that shows to which degree it belongs to the given set (ex the degree that it is not expensive in respect to its efficiency) and the belief towards this estimation. Then we combine the evidence from the two sources according to Eq. (4) so that the outcome produces the combined evidence. The same process can be applied iteratively when more factors are considered for a given project.

We will show the applicability of our approach by evaluating one of the technologies that implement the authentication process pattern: the use of biometrics.

Considering the three different subsets N, A, C that include the necessary, affordable (cost-effective) and complex-low technologies, two independent evaluations are considered; therefore, we assign a value that shows the degree to which the biometrics technology falls into one of the independent categories or their

Table 1
Combined evidence from independent sources while implementing a specific privacy measure (e.g. use of biometrics) in PriS.

Elements	Partner 1		Partner 2		Combined value	
	m_1	Bel_1	m_2	Bel_2	$m_{1,2}$	$Bel_{1,2}$
N	0.05	0.05	0.10	0.10	0.20	0.20
A	0.01	0.01	0.02	0.02	0.03	0.03
C	0.04	0.04	0.08	0.08	0.07	0.07
$N \cup A$	0.20	0.26	0.10	0.22	0.12	0.35
$A \cup C$	0.20	0.25	0.30	0.40	0.24	0.34
$N \cup C$	0.20	0.29	0.10	0.28	0.22	0.49
$N \cup A \cup C$	0.30	1	0.30	1	0.12	1

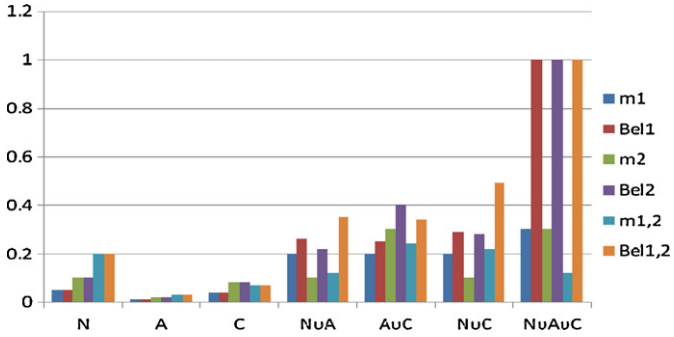

Fig. 2. Graphical representation of combined evidence values for biometrics.

combination. We need now to calculate the normalization factor, represented as the denominator $1 - \sum_{B \cap C = \varnothing} m_1(B) \cdot m_2(C)$. Thus, we have:

$$K = \sum_{B \cap C = \varnothing} m_1(B) \cdot m_2(C) = m_1(N)m_2(\mathrm{A}) + m_1(N)m_2(C) + m_1(N)\, m_2 \times (A \cup C) + m_1(A)m_2(N) + m_1(A)m_2(C) + m_1(A)m_2(N \cup C) + m_1 \times (C)m_2(N) + m_1(C)m_2(A) + m_1(C)m_2(N \cup A). \quad (5)$$

For our case $K = 0.023$.

The joint calculation for $m_{1,2}(N)$ gives:

$$m_{1,2}(N) = \frac{(m_1(N)\, m_2(N) + m_1(N)\, m_2(N \cup A) + m_1(N)\, m_2(N \cup C) + m_1(N)\, m_2(N \cup A \cup C) + m_1(N \cup A)\, m_2(N) + m_1(N \cup A)\, m_2(N \cup C) + m_1(N \cup C)\, m_2(N) + m_1(N \cup C)\, m_2(N \cup A) + m_1(N \cup A \cup C)\, m_2(N))}{(1 - K)} \quad (6)$$

Using Eq. (4) in a similar manner we have the following equations for the joint calculation of the combined values for the variables under consideration.

$$m_{1,2}(A) = \frac{(m_1(A)\, m_2(A) + m_1(A)\, m_2(N \cup A) + m_1(A)\, m_2(A \cup C) + m_1(A)\, m_2(N \cup A \cup C) + m_1(N \cup A)\, m_2(A) + m_1(N \cup A)\, m_2(A \cup C) + m_1(A \cup C)\, m_2(A) + m_1(A \cup C)\, m_2(N \cup A) + m_1(N \cup A \cup C)\, m_2(A))}{(1 - K)} \quad (7)$$

$$m_{1,2}(C) = \frac{(m_1(C)\, m_2(C) + m_1(C)\, m_2(A \cup C) + m_1(C)\, m_2(N \cup C) + m_1(C) m_2(N \cup A \cup C) + m_1(A \cup C)\, m_2(C) + m_1(A \cup C)\, m_2(N \cup C) + m_1(N \cup C)\, m_2(C) + m_1(N \cup C)\, m_2(A \cup C) + m_1(N \cup A \cup C)\, m_2(C))}{(1 - K)} \quad (8)$$

$$m_{1,2}(N \cup C) = \frac{(m_1(N \cup C)\, m_2(N \cup C) + m_1(N \cup C)\, m_2(N \cup A \cup C) + m_1(N \cup C \cup A)\, m_2(N \cup C) +)}{(1 - K)} \quad (9)$$

$$m_{1,2}(N \cup A) = \frac{(m_1(N \cup A)\, m_2(N \cup A) + m_1(N \cup A)\, m_2(N \cup A \cup C) + m_1(N \cup C \cup A)\, m_2(N \cup A) +)}{(1 - K)} \quad (10)$$

$$m_{1,2}(N \cup A \cup C) = \frac{((m_1(N \cup A \cup C)\, m_2(N \cup A \cup C))}{(1 - K)} \quad (11)$$

The results are summarised in Table 1.

By examining Table 1 and the graphical representation of these data at Fig. 2 we see that although there is some support towards the use of this technology by all the participating in the evaluation process parties, there are concerns about the cost effectiveness of this measure as a means to achieve the specific target. For example, we see that although the values of N from each party and combined are not negligible, still the values assigned for the A variable are very low for both parties and their combined values also. Comparing also the values assigned for subsets as unions of independent events, we see that the joint value $m_{1,2}(N \cup A)$ has a relatively lower value than the value for $m_{1,2}(A \cup C)$ or the joint value $m_{1,2}(N \cup C)$. This expresses the concern about whether there is a strong need to implement such a measure in respect to the cost required; primarily this values is affected by the low values all the parties assigned independently to the A variable. The values for the union of the subsets have as expected considerably higher values than the values for each individual variable considered independently; this was expected since each technology has higher possibilities to belong to either one of the various categories, than to a single one. For example each technology has more chances of being either cost-effective either non-complex either necessary for the implementation of a specific privacy requirement than the possibility that this technology has to an adequate degree one of the above characteristics. In addition the possibility that one candidate solution belongs to any of the previous subsets to some extent is equal to 1 which is expressed by the corresponding value for the Belief variable. If this was not the case, then the technology under consideration would not have any of the characteristics of interest to us and thus there would be no point to be considered for the achievement of a specific privacy requirement.

For reasons of completeness, we will provide another similar example so that from the comparison of the two approaches the application of our approach will be clearer. We have performed an evaluation by asking the participating parties to evaluate the use of an alternative solution, more specifically the use of smart cards. This technique in general is considered as more cost effective and does not require the use and storage of sensitive personal data as it was the case in the previous example. We have recorded the values assigned by the evaluating parties for the variables under consideration and calculated the combined values using Eqs. (5) and (11). The results are summarised in Table 2.

Table 2
Combined evidence from independent sources for the evaluation of smart cards as a privacy measure in PriS.

Elements	Partner 1		Partner 2		Combined value	
	m_1	Bel_1	m_2	Bel_2	$m_{1,2}$	$Bel_{1,2}$
N	0.03	0.03	0.05	0.05	0.05	0.08
A	0.04	0.04	0.03	0.03	0.03	0.07
C	0.03	0.03	0.02	0.02	0.2	0.08
$N \cup A$	0.1	0.17	0.15	0.23	0.15	0.27
$A \cup C$	0.1	0.17	0.18	0.23	0.18	0.24
$N \cup C$	0.2	0.26	0.2	0.27	0.21	0.26
$N \cup A \cup C$	0.3	1	0.37	1	0.37	1

By observing Table 2 or the graphical representation of the recorded values at Fig. 3 we see that there is a stronger support in respect to the cost variable in the previous example (we see that the combined value for A is more than double than in the previous example). This was expected since there is in general a higher cost when implementing a strong security measure such as biometrics; smart cards on the other hand are considered cheaper as a means to implement security. We also notice that there is stronger evidence for the values of $m_{1,2}(N \cup A \cup C)$ which means that there is stronger evidence towards implementing the second solution (smart cards) than implementing the first (biometrics).

4.4. Discussion and further analysis

Software requirements engineering is a discipline where many parameters have to be dealt with and this is also apparent by the continuous evolving of new methodologies trying to tackle with the variety of problems. The difficulty in capturing and implementing the requirements is due to different non measurable factors such as the participation of clients in the process of requirements gathering, the different non functional requirements (such as the cost, the implementation platforms available, the methodological and software tools often imposed to the developers), as well as the different perceptions that each developer has that makes difficult to make all the different opinions align to a solid solution. Taking all these into account the PriS method provides a solid methodological approach that facilitates incorporation of privacy requirements early in the system development process. The extended PriS framework proposed in this paper attempts to tackle with the issue of managing uncertainty in the presence of differentiating opinions from different developers, an issue that was not addressed in previous versions of PriS. It is essential to point out that the PRiS framework allows the developer to grade each solution that satisfies a requirement in such a way that it is difficult to be formulated. For example if a solution satisfies more than one requirement, although it might be expensive than it has greater

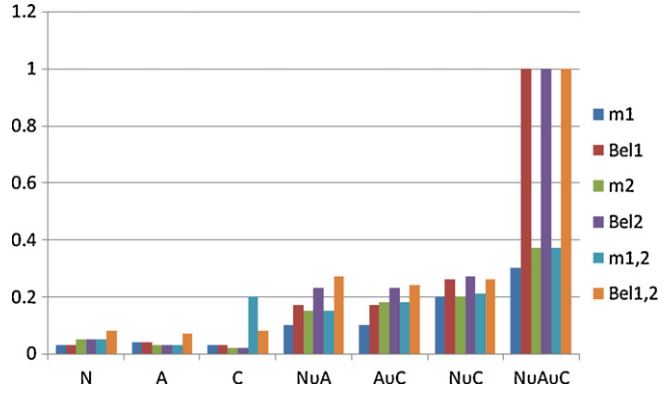

Fig. 3. Graphical representation of combined evidence values for smart cards.

importance and it can be rated as such; this is an important factor that allows the proposed solutions to be graded according to different criteria according to the expert's opinion. But introducing uncertainty as a parameter complicates things and demands an approach that allows handling this issue appropriately; the problem of dealing with uncertainty in expert's opinions has been an important issue in many fields of research, were different factors need to be approximated based on judgment due to the fact that certain parameters are not measurable. Various methods often used in relevant literature able to deal with uncertainty are Bayesian approaches, prediction expansion, model set expansion and as in our case, Dempster–Shafer theory. In the probabilistic approach, when an expert does not have adequate information to provide fixed values of the probabilities of a model being correct or about the validity of a parameter in forming a specific result it is preferable that this uncertainty is expressed in numerical intervals in terms of a subjective probability distribution $f(p_1, p_2, ..., p_{n-1})$. For example, in the case of two possible alternatives M_1, M_2, the expert, instead of assigning a fixed value of p_1, may provide a probability density function expressing his/her epistemic uncertainty; if, for example, the expert believes that the value of p_1 is located somewhere between 0.3 and 0.6, with no preference for any value within the range, this uncertainty could be represented by a uniform probability density function in [0.3, 0.6]. The uncertainty regarding the probabilities p_l described by the distribution f is often referred to as second-order uncertainty.

We should note however that there are specific limitations [11] when it comes to characterizing uncertainty using single probability values or probability; more specifically when assigning a value for a combined set of models (features in our case), and one of them does not contribute as the other it is not possible to make the distinction and promote the contribution of one of the two. On the other hand, the Dempster–Shafer method has proved to be a proper framework for effectively representing the uncertainty on the correctness of the different hypotheses by means of two limiting values, belief and plausibility, overtaking the aforementioned difficulties. The added value of the approach is particularly evident in the case in which some possible parameters do not contribute to the model for any reason, and that may lead to counterintuitive results if addressed probabilistically. The comparison between Dempster–Shafer theory and probabilistic or Bayesian approaches are out of the scope of this paper and the interested reader may refer to [11,12] for more concrete examples.

Another issue worth noting is that by introducing, in our approach, the normalization factor $(1 - K)$ at the denominator we normalise the values and consider the appearance of strongly conflicting evidence as unlikely, associating thus such conflicts with the null set; in relevant literature there has been a lot of discussion on managing conflicts in evidence theory. In [13], Zadeh presents an example with a medical scenario in presence of conflicting evidence from different medical experts; it is shown that the combined evidence for unlikely events with high degrees of belief towards this unlikelihood, may result in a case where these not so probable events are given priority (due to the high support values on this unlikelihood). We need to clarify that in our examples, due to the nature of the software development process and due also to the fact that the variables have been specified from the beginning and are not assigned ad hoc by the evaluators, we do not consider that two different experts in the field will give conflicting evidence while examining the same parameters for the same technology. For example, the use of biometrics is considered by most of the people as more expensive than the alternative technologies; the opinions about its necessity may also may vary but not to such an extent – under usual circumstances – so as to disturb the results as in the medical scenario presented in Ref. [13], where it was also possible

for a doctor to introduce a totally different diagnosis as a parameter under examination. Still, we need to mention that if during the application of our methodology such conflicts are encountered, they can be handled – if they arise – by adjusting appropriately Eq. (4) using Yager's modified rule [14] which does not considers the normalization factor, or the Ianagaki's unified combination rule [15,16]. However, we need to notice that a complete negotiation of management of conflicts in evidence theory in not within the scope of this paper per se; we need to emphasize though that the methodological tools are present already if similar cases of conflicting evidence arise and appearance of such conflicts even unlikely does not consist a burden to our approach.

We have thus provided methodological tools that help the developers estimate the most appropriate solutions by considering combined opinions from independent sources while developing privacy measures in the software design process. The aforementioned method enables also to tackle a serious problem of estimating the combined opinions in a formal manner. Also it is important to note that the method is not limited by the number of independent evaluations nor by the number of subsets (factors) considered prior to making the decision.

5. Conclusions

A number of Privacy Enhancing Technologies (PETs) have been developed for realizing privacy. The purpose of PETs is to protect the privacy of individuals, while still enabling them to interact with other parties in a modern society, using electronic communications. Examples of PETs include the Anonymizer [17], Crowds [18,19], Onion Routing [20,21], Dc-Nets [22,23], Mix-Nets [24,25], Hordes [26], GAP [27], and Tor [28]. Nevertheless, 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.

From a software systems perspective, a number of security oriented technologies and architectures have been proposed in the literature [29]. These architectures consider privacy requirements earlier in the systems development process, at the design level. However, they focus only on specific privacy issues without providing an intergraded solution for meeting all basic privacy requirements. As far as we know, none of the existing methodologies present a holistic approach for addressing the specific privacy requirements and their relationship with the respective implementation techniques that realise these requirements. Also most of these architectures do not offer any software tool for assisting the developer in realizing the elicited privacy requirements and analyzing their impact on organisation's goals and processes.

To this end, PriS, a new security requirements engineering methodology, has been introduced aiming to incorporate privacy requirements early in the system development process. Decision making in software design process is not always straightforward; often the implementation of specific privacy related countermeasures depends on the evaluation of different factors for which often opinions vary among the project partners. We have presented a fuzzy approach that allows the combination of different estimations in a formal manner. The combined evidence allows better decision making since the resulting values from independent sources may produce reliable results that can be evaluated on the [0,1] scale. The process is extendable not only to the number of parameters examined, but also to the number of evaluators, allowing thus better and detached results.

Acknowledgements

We would like to thank the anonymous reviewers for their insightful comments on previous versions of this paper.

References

[1] R. Lunheim, G. Sindre, et al., Privacy and computing: a cultural perspective. Security and control of information technology, in: R. Sizer (Ed.), Society (A-43), Elsevier Science B.V, North Holland, 1994, pp. 25–40.
[2] S. Fischer-Hübner, IT-security and privacy Design and Use of Privacy Enhancing Security Mechanisms. Lecture Notes in Computer Science, vol. 1958, Springer-Verlag, Berlin, 2001.
[3] J.C. Cannon, Privacy, in: What Developers and IT Professionals Should Know, Addison-Wesley, 2004.
[4] R. Koorn, H. van Gils, J. ter Hart, P. Overbeek, R. Tellegen, Privacy enhancing technologies, in: White Paper for Decision Makers. Ministry of the Interior and Kingdom Relations, The Netherlands, December, 2004.
[5] C. Kalloniatis, E. Kavakli, S. Gritzalis, Methods for designing privacy aware information systems: a review, in: V. Chrysikopoulos, N. Alexandris, C. Douligeris, S. Sioutas (Eds.), Proceedings of the PCI 2009 13th Pan-Hellenic Conference on Informatics, September, Corfu, Greece, IEEE CPS Conference Publishing Services, 2009, pp. 185–194.
[6] C. Kalloniatis, E. Kavakli, S. Gritzalis, Addressing privacy requirements in system design: the PriS method, Requirements Engineering 13 (3) (2008) 241–255.
[7] P. Loucopoulos, V. Kavakli, Enterprise Knowledge Management and Conceptual Modelling. LNCS, Vol.1565, Springer, 1999, 123-143.
[8] P. Loucopoulos, From Information Modelling to Enterprise Modelling. Information Systems Engineering: State of the Art and Research Themes, Springer-Verlag, Berlin, 2000, 67-78.
[9] E. Kavakli, S. Gritzalis, C. Kalloniatis, Protecting privacy in system design: the electronic voting case, Transforming Government People Process and Policy 1 (4) (2007) 307–332.
[10] G. Klir, B. Yuan, Fuzzy Sets and Fuzzy Logic, Prentice Hall, 1995.
[11] P. Baraldi, E. Zio, A comparison between probabilistic and Dempster–Shafer theory approaches to model uncertainty analysis in the performance assessment of radioactive waste repositories, Risk Analysis 30 (7) (2010) 1139–1156.
[12] E.L. Droguett, A. Mosleh, Bayesian methodology for model uncertainty using model performance data, Risk Analysis 28 (5) (2008).
[13] L.A. Zadeh, Review of books: a mathematical theory of evidence, The AI Magazine 5 (3) (1984) 81–83.
[14] R. Yager, Quasi-associative operations in the combination of evidence, Kybernetes 16 (1987) 37–41.
[15] K. Sentz, S. Ferson, Combination of Evidence in Dempster–Shafer Theory, Sandia National Laboratories, Technical Report SAND 2002-0835, Albuquerque, New Mexico, 2002.
[16] T. Inagaki, Interdependence between safety-control policy and multiple-sensor. Schemes via Dempster–Shafer theory, IEEE Transactions on Reliability 40 (2) (1991) 182–188.
[17] Anonymizer, 'Anonymizer Tool', 2008. Available from: http://www.anonymizer.com.
[18] K.M. Reiter, D.A. Rubin, Crowds: anonymity for web transactions, ACM Transactions of Information and System Security 1 (1) (1998) 66–92.
[19] K.M. Reiter, D.A. Rubin, Anonymous web transactions with crowds', Communications of the ACM 42 (2) (1999) 32–38.
[20] M. Reed, P. Syverson, D. Goldschlag, Anonymous connections and onion routing, IEEE Journal on Selected areas in Communications 16 (4) (1998) 482–494.
[21] D. Goldschlag, P. Syverson, M. Reed, Onion routing for anonymous and private Internet connections, Communications of the ACM 42 (2) (1999) 39–41.
[22] D. Chaum, Security without identification: transactions systems to make big brother obsolete, Communications of the ACM 28 (10) (1985) 1030–1044.
[23] D. Chaum, The dining cryptographers problem: unconditional sender and recipient untraceability, Journal of Cryptology 1 (1) (1988) 65–75.
[24] D. Chaum, Untraceable electronic mail, return addresses, and digital pseudonyms, Communications of the ACM 24 (2) (1981) 84–88.
[25] A. Pfitzmann, M. Waidner, Networks without user observability, Computers and Security 6 (2) (1987) 158–166.
[26] C. Shields, N.B. Levine, A protocol for anonymous communication over the internet, in: P. Samarati, S. Jajodia (Eds.), Proceedings of the 7th ACM Conference on Computer and Communications Security, ACM Press New York NY, 2000.
[27] K. Bennett, C. Grothoff, GAP-Practical Anonymous networking, in: Proceeding of the Workshop on PET 2003 Privacy Enhancing Technologies, Dresden, Germany, 2003.
[28] R. Dingledine, N. Mathewson, P. Syverson, Tor: the second-generator onion router, in: Proceedings of the 13th USENIX Security Symposium, San Diego, CA, USA, 2004.
[29] C. Kalloniatis, E. Kavakli, S. Gritzalis, Security requirements engineering for e-government applications Proceedings of the DEXA EGOV'04 Conference, LNCS, vol. 3183, Springer, Zaragosa, Spain, 2004, pp. 66–71.