MCIS 2009 Proceedings

Mediterranean Conference on Information Systems (MCIS)

2009

Pris Tool: A Case Tool For Privacy-Oriented Requirements Engineering

Christos Kalloniatis
Dept. of Cultural Technology and Communication,University of the Aegean, ch.kalloniatis@ct.aegean.gr

Evangelia Kavakli
. of Cultural Technology and Communication, University of the Aegean, vkavakli@ct.aegean.gr

Efstathios Kontellis
Dept. of Cultural Technology and Communication, University of the Aegean, ctmb06012@ct.aegean.gr

PRIS TOOL: A CASE TOOL FOR PRIVACY-ORIENTED REQUIREMENTS ENGINEERING

Kalloniatis, Christos, Dept. of Cultural Technology and Communication,University of the Aegean, Harilaou Trikoupi & Faonos Street, 81100 Mytilene-Lesvos, Greece, ch.kalloniatis@ct.aegean.gr

Kavakli, Evangelia, Dept. of Cultural Technology and Communication, University of the Aegean, Harilaou Trikoupi & Faonos Street, 81100 Mytilene-Lesvos, Greece, vkavakli@ct.aegean.gr

Kontellis, Efstathios, Dept. of Cultural Technology and Communication, University of the Aegean, HArilaou Trikoupi & Faonos Street, 81100 Mytilene-Lesvos, Greece, ctmb06012@ct.aegean.gr

Abstract

PriS is a security and privacy requirements engineering method which aims on 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. In this paper we present the PriS conceptual framework and a case tool that assist PriS way of working the PriS-Tool. Specifically, PriS-Tool assist developers by offering design capabilities of the organization’s goal-process model, helps them to monitor the impact of privacy requirements on organisation’s goals and processes, suggests them a set of implementation techniques for the realization of the privacy related processes and offers guidance throughout this process.

Keywords: *Privacy, Requirements Engineering, Case Tool, Security, Software Engineering*

1 INTRODUCTION

In the online world every person has to hold a number of different data sets so as to be able to have access to various e-services and take part in specific economical and social transactions. Such data sets require special consideration since they may convey personal data, sensitive personal data, employee data, credit card data etc. Recent surveys have shown that people feel that their privacy is at risk from identity theft and erosion of individual rights. The result is that privacy violation is becoming an increasingly critical issue in modern societies.

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.

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 [Fischer-Hübner, S. (2001), Cannon, J.,C. (2004), Ronald, K., et.al (2004)]. The first three requirements are basically 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.

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 [Anonymizer], Crowds [Reiter, K.M. and Rubin, D.A.(1998), Reiter, K.M. and Rubin, D.A.(1999)], Onion Routing [Reed,M. et al (1998), Goldschlang, D. et al (1999)], Dc-Nets [Chaum, D.

(1985), Chaum, D. (1988)], Mix-Nets [Chaum, D. (1981), Pfitzmann, A. and Waidner, M. (1987)], Hordes [Shields, C. and Levine, N.B. (2000)], GAP [Bennett, K. and Grothoff, C. (2003)], and Tor [Dingledine, R. et al (2004)]. 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 [Kalloniatis, C. et. al (2004)]. 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. 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. In this paper, we extend the PriS method by introducing a prototype tool that supports PriS way of working. The paper is structured as follows. Section 2 presents a brief description of PriS conceptual framework and its way of working. In section 3 an in-depth analysis of the Formal PriS is conducted based on which the PriS-Tool was constructed. A description of the PriS-Tool is presented in section 4. Finally, concluding remarks are mentioned in section 5.

2 THE PRIS METHOD

2.1 The PriS Conceptual Framework

PriS [Kalloniatis, C. et. al (2008)] is a security requirements engineering method, which incorporates privacy requirements early in the system development process. PriS considers privacy requirements as organizational goals that need to be satisfied and adopts the use of privacy process patterns as a way to: (a) describe the effect of privacy requirements on business processes; and (b) facilitate the identification of the system architecture that best supports the privacy-related business processes.

PriS 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 [Loucopoulos, P., and Kavakli, V. (1999), Loucopoulos, P., (2000)], 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 figure 1. As shown in figure 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.

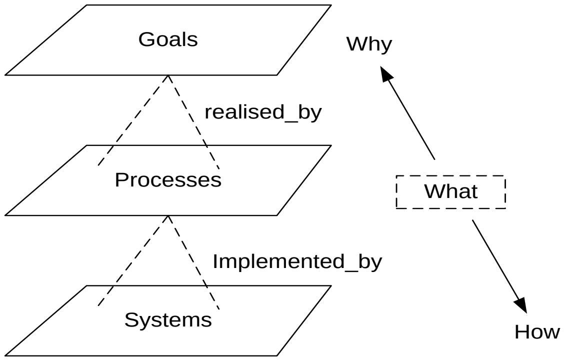


Figure 1. The EKD Schema.

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 [Kavakli, V. (2007)]. 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.

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

whereby, $G_1 \dots G_n$ are the total of all system's goals and subgoals as they are defined by the system's stakeholders and $E_1 \dots E_m$ 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. 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 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: $\forall G_i \in G$, and are realised by process P_k, a new goal G' is created and is defined as follows:

$$G' = G^{i} \vee G^{j} \vee \dots \vee G^{k}$$

$$PV'_{l} = [PV_{l}^{i} \vee PV_{l}^{j} \vee \dots \vee PV_{l}^{k}]$$

where,

k = the number of operationalised goals realised by one process

l = 1,2,…,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.

4 THE PRIS TOOL

The Pris-Tool was developed to support the application of the PriS method. It is a CASE tool that incorporates both the functions and the rules of this method. It consists of about 4.000 lines of Java code and 23 classes. The basic characteristics of the tool are:

- It has a user-friendly interface
- •Except from the scrollbar, it introduces a dynamic way to represent the tree model by hiding/ showing it's sub-trees
- It includes a real-time refreshed information panel where the data of tree nodes appear
- It contains a mechanism that provides error recognition and displays the appropriate warnings
- It is platform independent
- All output files produced have the .txt format

4.1 PriS-Tool Architecture

The tool mainly uses two types of data structures. The Java Hashtable and Vector. The most important part of the tool is the DynamicTree class which includes the data structures where the tree information is being stored. Furthermore, this class incorporates another smaller class, in order to display properly every node of the tree. The nodes of the tree are either goals or processes. The information about the type of the node is stored into a Hashtable that maps the name of every node to its type. The name of every goal-node is unique so the use of the specific data structure does not cause any dysfunctions. On the other hand the name of a process may appear two or more times. As only one entry is required for a specific process name the specific exception does not cause any dysfunction.

Furthermore, the DynamicTree class includes a Hashtable where the type of the link between every node and its child is being stored (link or / and). The types of the links are indexed by goal names. So it is possible for the information panel to present the type of the link of a particular goal with its children and parent goals. Another major data structure that is included in that class is the Hashtable, which stores which privacy requirements are affecting which goals and which process patterns are applied to which processes. The type of the link is the same too. The name of the goal or process indicates which requirements or patterns have been applied to each one. Nevertheless, when the process reappears, there is a special treatment before its insertion in the data structure.

The last data structure of the specific class is another Hashtable where the influences among goals, either positive or negative, are being stored. This information consists of three parts: a) the name of the node that influences, b) the name of the node that is being influenced and c) the type of the influence. However, a node name cannot be used as the key in the Hashtable on its own, as is was implemented in the previous structures. For the proper storage of influences between goals in the specific hashtable both node names are stored thus producing a complex key which uniquely leads to the influence type of these nodes.

The "Tree Panel" contains the Dynamic Tree class and everything relative to that. It also plays the role of the contactor between the program and the user by carrying out some particular features. One of them is the right click for the definition of the requirements. The "Tree Panel" is being hosted by the “Program Panel” in order to represent the tree. Furthermore the “Program Panel” constitutes the user interface of the most functions that are being offered to the user.

The architecture described above is graphically represented in figure 2.

4.2 Tool’s Function

In this section it is described how PriS Tool realises the four activities of the PriS method.

4.2.1 4.2.1 Elicit Privacy Related Goals

PriS tool offers a design area where users are able to create a new goal hierarchy or load an existing one. As it was mentioned before, the goal – process diagrams are stored as independent text files. Figure 3 presents a screenshot of a beta goal-process model created for the scope of this paper. Users are able to create new nodes (either goals or processes), to remove nodes, to define positive or negative influences or to eliminate these influences. The AND/OR relationship between each parent goal and its child goals is defined when the new node is created and prior to its insertion in the model. In the bottom area of the window an information panel exists on which the user can receive relevant information regarding the node that has been selected in the model. If it is a goal he/she can see its name, which privacy requirements are constrained it, the type of connection that has with its child goals and with its parent goals (except if it’s the root goal where no parent goal exists) as well as the type of positive/negative influences that has with other goals. In case the node is of a process type then the user can see its name, the privacy process patterns applied on that process as well as the goal(s) that is (are) realized by this process.

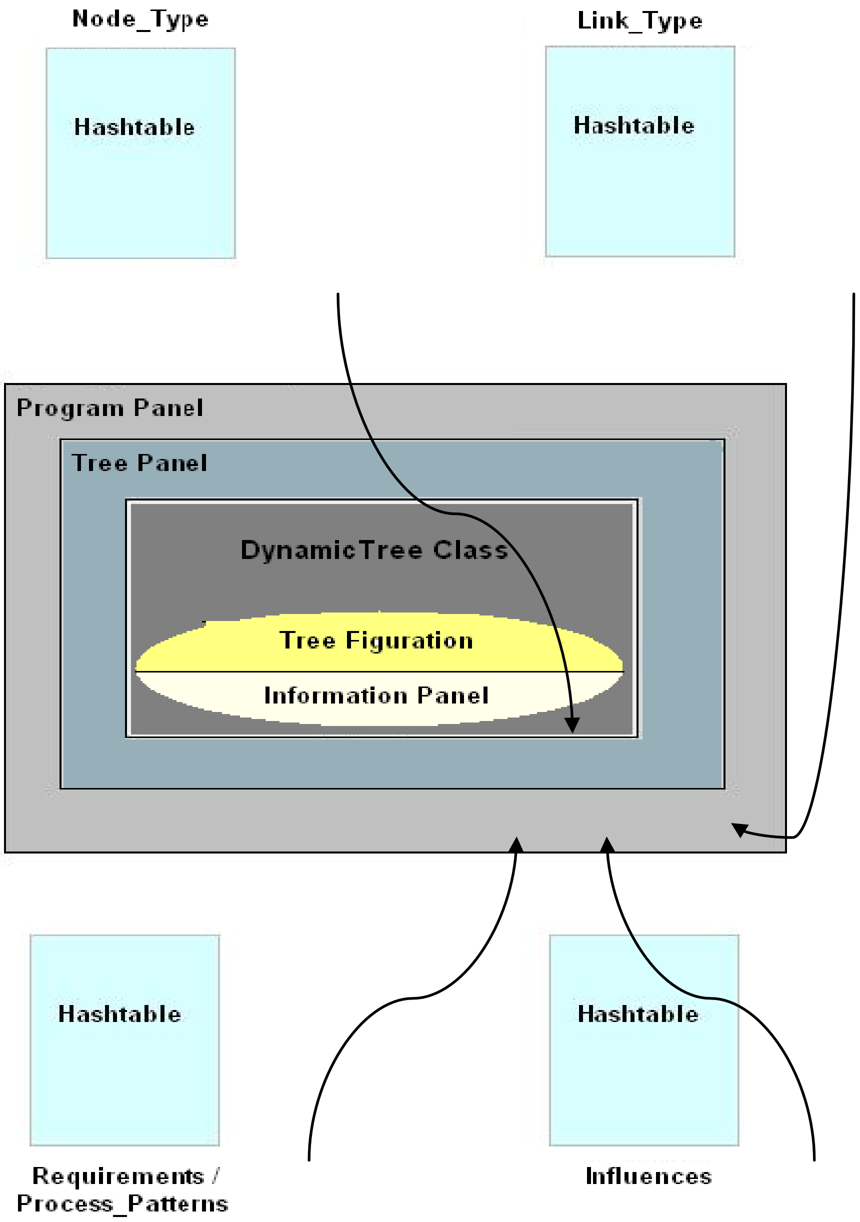
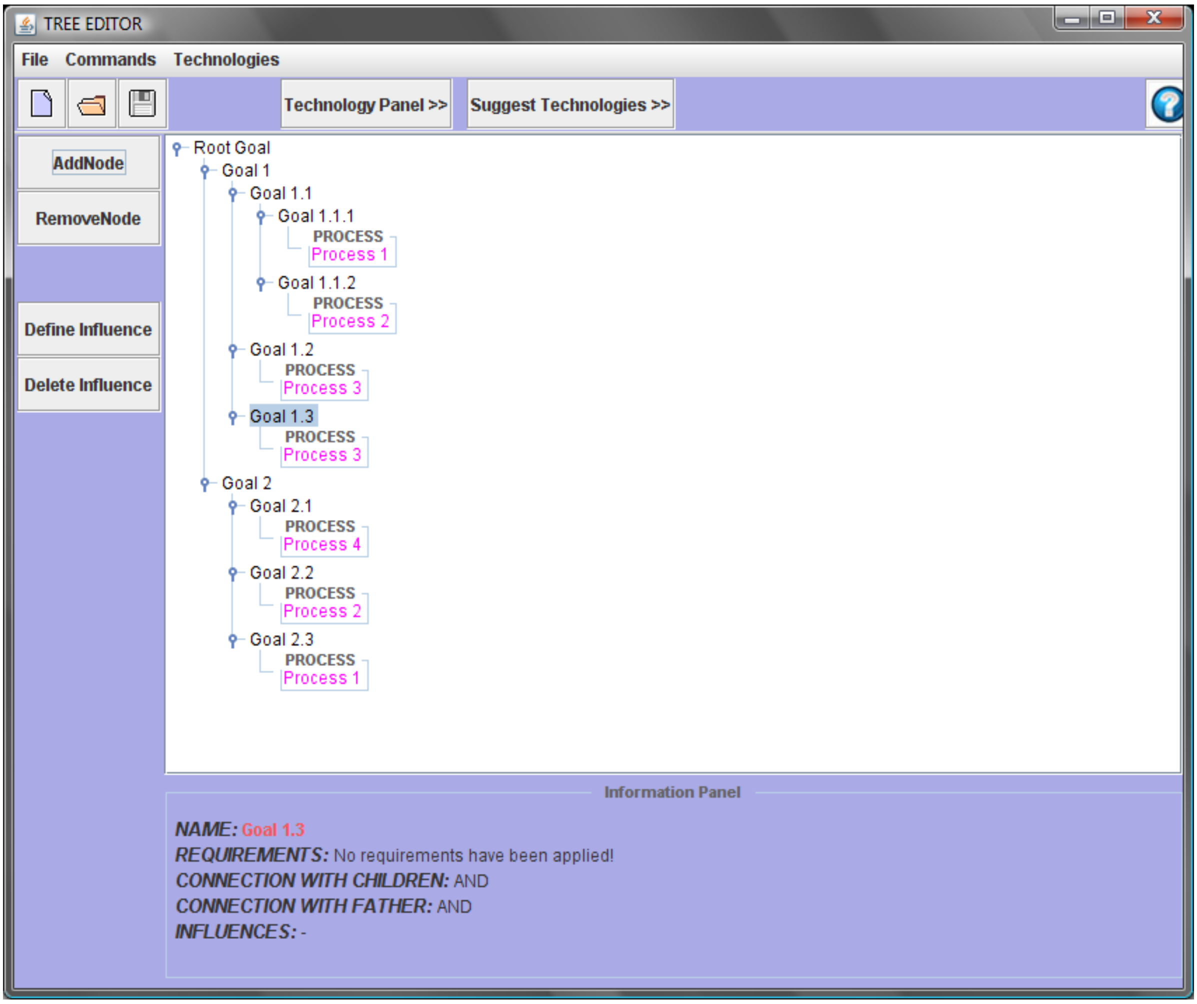


Figure 2. Architecture of PriS Tool



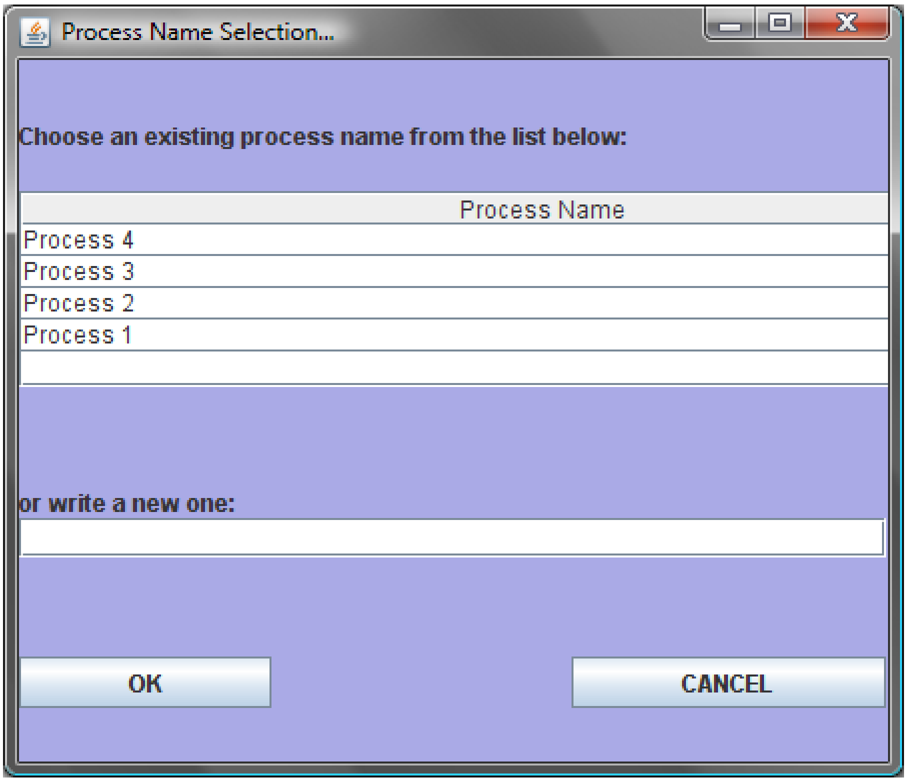
NAME: Goal 1.3
REQUIREMENTS: No requirements have been applied!
CONNECTION WITH CHILDREN: AND
CONNECTION WITH FATHER: AND
INFLUENCES: -

Figure 3. A goal-process model

When a user wishes to insert a node, the tool asks if it's a goal or a process node. In case of a goal node, user enters its name and its relationship type with its children. However, when a user wishes to insert a process node, the names of the existing processes are being showed; so that the user has the opportunity to define a process which can realize more than one goal or to insert a new one. Figure 4 presents this case.

4.2.2 4.2.2 Analyse the impact of privacy goals on business processes

After the construction of the goal-process model, users are able to define the privacy requirements that constrain the goals in the hierarchy. For achieving that, the tool offers the ability to right click on every goal and select which privacy requirements constrain that specific goal. In the case where a parent goal is constrained by a specific privacy requirement users cannot remove this requirement from the child goals. However, they can add new privacy requirements that constraint solely the child goals and not the parent one. An example of this case is shown in figure 5. Specifically, the privacy requirements of authentication and authorisation have been applied in Goal 1.1. When the user tries to

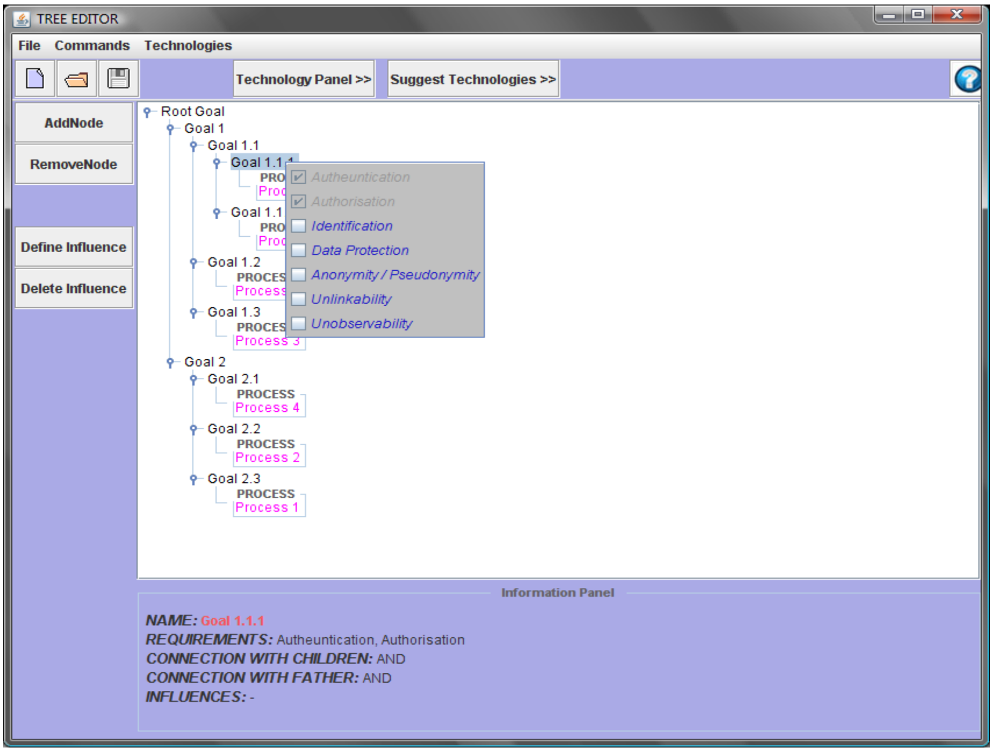


Choose an existing process name from the list below:

Process Name
Process 4
Process 3
Process 2
Process 1

or write a new one:

Figure 4. Inserting a process



☑ Autheuntication
☑ Authorisation
□ Identification
□ Data Protection
□ Anonymity / Pseudonymity
□ Unlinkability
□ Unobservability

NAME: Goal 1.1.1
REQUIREMENTS: Autheuntication, Authorisation
CONNECTION WITH CHILDREN: AND
CONNECTION WITH FATHER: AND
INFLUENCES: -

Figure 5.Applying privacy requirements on goals

add a new privacy requirement in Goal 1.1.1 the first two privacy requirements are not available since they are inherited by the parent goal to its child goals.

Through this process users are able to define the privacy requirements on every privacy related goals.

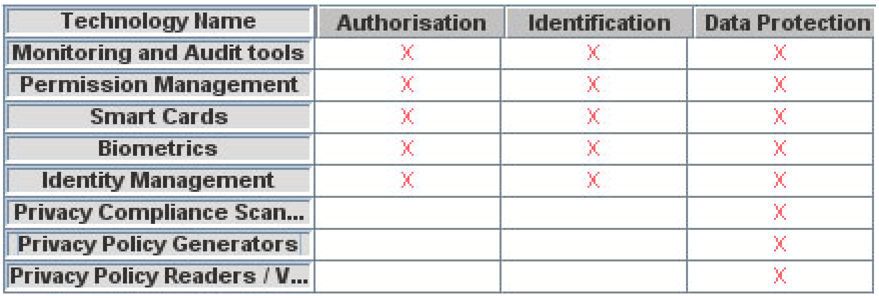
4.2.3 4.2.3 Model affected processes using privacy process patterns

Every time the user adds or deletes a privacy requirement from an organisational goal the tool automatically adds or eliminates this requirement from the processes that implement this goal. Additionally, the process patterns applied on every privacy related process are defined dynamically every time a new privacy requirement is applied or deleted. Thus the user is not required to press a button or finish drawing the model for applying the respective privacy requirements. From the time that a part of the tree is being constructed users can add privacy requirements on goals and see in the same time how the child goals are affected, which processes are affected and which privacy process patterns are applied on every process thus assisting its realization.

The aforementioned flexibility is based on the tool's dynamic design and implementation as it was presented in the previous section.

4.2.4 4.2.4 Identify the technique(s) that best support/implement the above processes

The tool offers an easy and user friendly interface for the management of the available technologies. The tool contains a number of technologies that can implement the privacy process patterns defined in PriS. Thus, a table is being constructed with the privacy process patterns and the available technologies that realize these patterns. In that table, part of which is presented in Figure 6, the technologies are organized in groups. For every technology a number of red "X" symbols dedicate the correlation with the appropriate process patterns. That table is implemented by a Vector.



Technology Name	Authorisation	Identification	Data Protection
Monitoring and Audit tools	X	X	X
Permission Management	X	X	X
Smart Cards	X	X	X
Biometrics	X	X	X
Identity Management	X	X	X
Privacy Compliance Scan...			X
Privacy Policy Generators			X
Privacy Policy Readers / V...			X

Figure 6.Partial view of the table that joins process patterns with implementation techniques

The interface also offers the user the capabilities below:

- Insertion of a new technology either in an existing group or a new one
- Update of the process patterns that realizes an available technology and its description
- Deletion of a technology
- Independent view of a technology's description

The aim of the PriS-Tool is to suggest the appropriate technologies for realizing the identified privacy process patterns thus realizing the privacy goals of the system under construction. When the privacy requirements have been defined, the program calls the routine that chooses the process patterns which have to be applied to the appropriate processes and stores them to the *Hashtable Requirements/Process_Patterns*. The next step is to suggest the respective technologies. The combination between the information stored in the *Hashtable Requirements/Process_Patterns* and the *Vector* of correlation between process patterns and technologies, leads to the appropriate technologies per process pattern.

Thus, when the user presses the *"Suggest Technologies"* button (see figure 3) the tool suggests a number of implementation techniques for every privacy process pattern. Firstly, the technologies that best match the process patterns applied on every process are suggested and then the technologies that partially realize the privacy process patterns of a process are mentioned.

5 CONCLUSIONS

In this paper a new tool for realizing privacy requirements was presented. Specifically, the tool realises PriS way of working. Using the PriS tool users have the ability to draw their models in a tree-view mode and then apply the privacy requirements on the respective organisational goals. Then, the tool guides the users in monitoring the impact of these requirements on the respective processes and ends up by suggesting a number of implementation techniques that realize the privacy related processes. Thus a holistic approach is presented starting from the early design phase and concluding prior to the implementation phase where the developers are then responsible for the selection and realization of the privacy process patterns taking into consideration the tool's suggestions. The PriS tool has been used for modeling a case study of the application of the PriS method on an e-voting system.

References

Fischer-Hübner, S. (2001). IT-Security and Privacy, Design and Use of Privacy Enhancing Security Mechanisms. Lecture Notes in Computer Science, Vol. 1958. Springer-Verlag, Berlin

Cannon, J.,C. (2004). Privacy, What Developers and IT Professionals Should Know. Addison-Wesley.

Ronald, K., Herman van G., Joris ter H., Overbeek, P., and Tellegen, R (2004). Privacy Enhancing Technologies, White paper for Decision Makers. Ministry of the Interior and Kingdom Relations, the Netherlands.

Anonymizer, available at www.anonymizer.com

Reiter, K.M. and Rubin, D.A.(1998). Crowds: Anonymity for Web Transactions. ACM Transactions of Information and System Security, Vol. 1, No. 1, p.p. 66-92.

Reiter, K.M. and Rubin, D.A. (1999). Anonymous Web Transactions with Crowds. Communications of the ACM, Vol. 42, No. 2, p.p. 32-38.

Reed, M., Syverson, P. and Goldschlag, D. (1998). Anonymous connections and Onion Routing. IEEE Journal on Selected areas in Communications, Vol. 16, No. 4, p.p. 482-494.

Goldschlag, D., Syverson, P. and Reed, M. (1999). Onion Routing for anonymous and private Internet connections. Communications of the ACM, Vol. 42,No. 2, p.p. 39-41.

Chaum, D. (1985). Security without identification: Transactions Systems to make Big Brother Obsolete. Communications of the ACM, Vol. 28, No.10, p.p. 1030-1044.

Chaum, D. (1988). The Dining Cryptographers Problem: Unconditional Sender and Recipient Untraceability. Journal of Cryptology, Vol. 1, No. 1, p.p. 65-75.

Chaum, D. (1981). Untraceable Electronic Mail, return Addresses, and Digital Pseudonyms. Communications of the ACM, Vol. 24, No. 2, p.p. 84-88.

Pfitzmann, A. and Waidner, M. (1987). Networks without user Observability. Computers & Security, Vol. 6, Issue 2, p.p. 158-166.

Shields, C. and Levine, N.B. (2000). A protocol for anonymous communication over the Internet. In: Samarati, P. and Jajodia, S. (eds.): Proceedings of the 7th ACM Conference on Computer and Communications Security. ACM Press New York NY, p.p. 33-42.

Bennett, K. and Grothoff, C. (2003). GAP-Practical Anonymous networking. Proceeding of the Workshop on PET2003 Privacy Enhancing Technologies (2003), also available at http://citeseer.nj.nec.com/bennett02gap.html.

Dingledine, R., Mathewson, N. and Syverson, P. (2004). Tor: The Second-Generator Onion Router. Proceedings of the 13th USENIX Security Symposium, San Diego, CA, USA.

Kalloniatis, C., Kavakli, E. and Gritzalis, S. (2004). Security Requiremenets Engineering for e-Government Applications, Proceedings of the DEXA EGOV'04 Conference, LNCS Vol. 3183. Springer, p.p. 66-71.

Kalloniatis, C., Kavakli, E. and Gritzalis, S. (2008). Addressing privacy requirements in system design: The PriS method, Requirements Eng. Vol 13, No 3, pp. 241-255, Springer.

Loucopoulos, P. and Kavakli, V. (1999). Enterprise Knowledge Management and Conceptual Modelling. LNCS Vol. 1565. Springer pp. 123-143.

Loucopoulos, P. (2000). From Information Modelling to Enterprise Modelling. In: Information Systems Engineering: State of the Art and Research Themes. Springer-Verlag, Berlin, pp. 67-78.

Kavakli, E., Gritzalis, S and Kalloniatis, C. (2007). Protecting Privacy in System Design: The Electronic Voting Case. Transforming Government: People, Process and Policy Vol. 1, Issue 4, pp. 307-332.