FROM HIPPOCRATIC DATABASES TO SECURE TROPOS: A COMPUTER-AIDED RE-ENGINEERING APPROACH

FABIO MASSACCI† and JOHN MYLOPOULOS‡ and NICOLA ZANNONE§

Department of Information and Communication Technology, University of Trento, Italy

†*massacci@dit.unitn.it*

‡*jm@dit.unitn.it*

§*zannone@dit.unitn.it*

Received 18 March 2005
Revised 3 January 2006
Accepted 25 April 2006

Privacy protection is a growing concern in the marketplace. Yet, privacy requirements and mechanisms are usually retro-fitted into a pre-existing design which may not be able to accommodate them due to potential conflicts with functional requirements.

We propose a procedure for automatically extracting privacy requirements from databases supporting access control mechanisms for personal data (hereafter Hippocratic databases) and representing them in the Tropos modeling framework where tools are available for checking the correctness and consistency of privacy requirements. The procedure is illustrated with a case study.

Keywords: Security Engineering, IS Re-engineering, Agent-Oriented Technologies.

1. Introduction

Interest in privacy and privacy-aware information and communication technologies is growing and many countries have promulgated new privacy legislation. Regulations in the US and in the EU are largely based on the idea of "Fair Information Practices". These practices stem from a set of principles established in 1980 by the Organization for Economic Cooperation and Development (OECD). Most of these laws incorporate rules governing collection, use, storage and distribution of personally identifiable information. Enterprises that do not correctly manage customer personally identifiable information chance some risks. It is up to an organization to ensure that data processing operations comply with legislative requirements. Actually, an enterprise can be sued if it uses customer personal data for purposes other than those they were collected.

Central to the concept of privacy are transparency and fairness principles. Transparency means that organizations should make known to individuals which information has been collected about them, and how it is used. Fairness means that information should be used only for the purpose for which it is collected. If the organization wants to use personal information for other purposes, it must previously obtain consent from its owners.

In recent years, researchers have recognized that traditional access control models, such as discretionary access control[14], mandatory access control[8] and role-based access control[26], fail in protecting privacy (see Ref. 6, 10, 25). Actually, privacy policies focus on the correct use of data (i.e., the use of data in compliance with the fairness principle), rather than authorization decisions based on who is accessing the data and which action is performed on them (see Ref. 10). Therefore, defining and enforcing privacy policies require extensions to traditional access control models. In particular, data subject[a] and the purpose for which data items can be collected and used should be considered in the model (see Ref. 25); also, mechanisms for recording consent[25], and enforcing minimal disclosure[3,20,22] and limited retention[3] should be provided by technologies developed for privacy protection.

Many privacy-aware technologies propose new purpose-based access control models for enforcing compliance with privacy policies (see Ref. 6, 11, 19). Among them, Agrawal et al. propose Hippocratic databases[3] that use *purpose* as a central concept around which privacy protection is built. Their aim is to negotiate the information exchanged by consumers and enterprises and enforce enterprise privacy policies. Policies would specify the purpose for which information is collected, who can receive it, the length of time it can be retained and the authorized users who can access it. Looking at such policies customers would have the choice to accept or deny them. However, the proposed framework defines privacy rules at a logical, rather than conceptual schema. As such, it is not possible to check the set of privacy rules supported by an organization for consistency and completeness.

The objective of this paper is to propose a re-engineering approach and algorithms for automatically extracting privacy requirements from policy statements stored in existing Hippocratic databases. These are then represented in a Requirements Engineering framework[16] where tools are available for formal analysis. Specifically, we aim to re-model privacy concerns captured in Hippocratic databases in Secure Tropos and check for their consistency. This approach has two advantages. Firstly, it provides a representation of the enterprise privacy policy in a modeling framework where formal tools are available for model checking (see Ref. 16). Secondly, it offers a unifying view of systems built using a structured Requirements Engineering methodology such as Tropos/i* or KAOS and systems directly implemented as Hippocratic databases. Thus, different design decisions can be compared at a level suitable for the designer.

Our approach is illustrated through a case study that is a somewhat more complex than the one presented in Ref. 3. This allows us to compare these representations and identify possible limits in the Hippocratic DB approach. We are currently applying our methodology to a large case studies for capturing technical and administrative security measures required by the Italian privacy legislation for Public Administrations (see Ref. 23). We are also developing a tool, called ST-Tool,[b] sup-

[a]Data subject is the natural or legal person to whom the personal data are related.
[b]ST-Tool is for Secure Tropos Tool. It is available on the web at *http://sesa.dit.unitn.it/sttool.*

porting the entire methodology and automatic verification of the correctness and consistency of functional, security and privacy requirements.

The remainder of the paper is structured as follows. Section 2 introduces a scenario used throughout the paper as running example. Section 3 introduces the Secure Tropos methodology and describes the basic concepts and diagrams used to model privacy requirements. Section 4 digests the Hippocratic DB approach, and Section 5 identifies and discusses some of its limitations. Section 6 proposes algorithms to map Hippocratic databases into the Secure Tropos framework and applies those algorithms to our scenario. Finally, Section 7 discusses related work, and Section 8 concludes the paper and outlines future work.

2. Scenario

Fineco is an Italian on-line bank that needs to obtain minimum personal information from customers to perform its services. This information includes name, address, and email. Fineco offers to its clients three methods of delivering bank statements: by email, by dedicated courier services, and by post. To execute the last two methods, it relies on a delivery company and the post office, respectively. In particular, the bank relies on DHL for courier delivery of bank statements, and therefore needs to re-delegate its costumer information to such a delivery company. The customer gives his personal information to the bank in order to receive bank statements. Accordingly, it is up to the bank to choose how to perform this task, but the customer may opt in or out a specific delivery method. Fineco also offers to its customers credit card services for which relies on Credit Union, a credit card company.

Credit Union is a US company offering a variety of products that provide customers with choice in the way they pay. However, Credit Union has privacy concerns since the United States enforces different measures to protect privacy than those taken by the European Union. The United States adopts a sectoral approach based on a mix of legislation and regulation. On the other hand, the EU privacy protection is based on a comprehensive legislation that requires the creation of government data protection agencies and prior approval before personal data processing may be performed. The European Commission has also published a Privacy Directive[c] that prohibits the disclosure of personal data to non-European Union organizations if they do not meet the European standard for privacy protection. Thus, Credit Union fears interruption in business with its EU partners and prosecution by European authorities under European privacy laws. To avoid these experiences, the U.S. Department of Commerce in consultation with the European Commission developed

[c]Directive 95/46/EC of the European Parliament and of the Council of 24 October 1995 on the protection of individuals with regard to the processing of personal data and on the free movement of such data.

the Safe Harbor.[d] This provides guidelines for US organizations to comply with the EU Privacy Directive. Certifying to the Safe Harbor, Credit Union assures Fineco and its EU customers that it enforces privacy protection in accordance with the EU Directive.

Bob is a Fineco customer who does not care how the bank delivers bank statements. Moreover, he is a frequent traveler and needs a credit card. Alice, on the other hand, does not want a credit card and prefers that Fineco delivers bank statements by regular mail because she thinks that delivering them through email is not secure and through dedicated couriers is too expensive.

3. Secure Tropos for Privacy

Secure Tropos[16] is an agent-oriented methodology extending the i*/Tropos framework[9] intended to model and analyze security and privacy requirements at individual and social levels. This framework uses the concepts of actor, service (goal, task, resource) and social relationships for defining objectives, entitlements and capabilities of actors. Actors are intentional entities that perform actions to achieve goals and represent agents (at individual level) and roles (at social level). A goal represents some strategic interest of an actor. A task represents a way of doing something. A resource represents a physical or an informational entity.

Social relationships could be functional dependencies, delegations of permission, trust relations, ownership, or provisioning. A functional dependency between two actors indicates that one actor depends on another to accomplish a goal, execute a task, or deliver a resource. A delegation of permission models a formal passage of authority (e.g., a signed piece of paper, or a digital credential) occurring in the domain of analysis. The basic idea of ownership is that an actor has full authority concerning access and disposition of his own services. "Owning a goal" means that the owner can decide who can fulfill the goal or can provide for its fulfillment. The distinction between owning and provisioning makes it clear how to model situations where, for example, a customer is the legitimate owner of his personal data and a Web Service provider who stores customer personal data, provides access to his data. In this setting, the provider needs the consent of the data owner for maintaining[e] his data, that is, the Web Service provider can maintain customer data only if the customer has previously delegated the permission to him. Finally, trust relations refer to the belief of clients that the provider will not misuse their personal data.

This framework essentially consists of three models that are defined at both social and individual levels (see Ref. 17).

Trust and ownership requirements model represents ownership and trust

[d] *http://export.gov/safeharbor*

[e] In accordance with United States Privacy Act of 1974, the term "maintain" includes maintain, collect, use or disseminate.

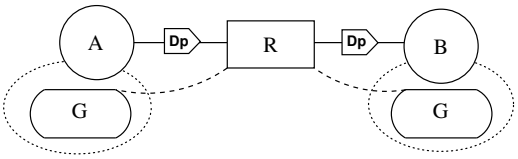


Fig. 1. Permission Model

relations among actors and services.

Functional requirements model represents functional dependencies among actors and services. Further, for every actor the services they pretend and the services they can provide are shown.

Trust management implementation represents delegations of permission among actors and services.

Once the stakeholders have been identified along with their objectives, entitlements, capabilities and social relations, the analysis proceeds in order to enrich the model with further details. Goal refinement rests on the analysis of goals, conducted from actor perspective using AND/OR decomposition.

In Ref. 24, the authors have given some hints on how to model privacy in Tropos, but no answer has been given on how to link permission to purpose in Tropos diagrams. Actually, Tropos lacks constructs for capturing this feature.

Our proposal is to introduce a link between a permission and the purpose for which the permission has been granted. An example of this model is given in Figure 1: actor A delegates the permission on resource R to actor B only to fulfill goal G. From an actor perspective, goal G is the purpose for which the permission is delegated. This relation is different from RBAC role-permission assignment (which maps a role onto a set of permissions); rather it is closer in the intuition to a delegation step in Trust Management languages (see Ref. 15, 16). We use the notation $A \xrightarrow{Dp} G/R \xrightarrow{Dp} B$ to represent delegations of permission link, $A \xrightarrow{Df} G \xrightarrow{Df} B$ to represent functional dependencies, and $A \xrightarrow{o} R$ to represent ownership.

We assume that the delegator delegates not only a purpose, but also the purpose sub-tree below it, to the delegatee. In this way, we distinguish the case in which one delegates the root purpose and the case where only a sub-purpose is delegated. This allows to model scenarios where an enterprise wants to outsource some parts of its business process without disclosing the end-goal of the business process itself. Moreover, this solution allows an actor to force the delegatee to comply with a specific plan in order to achieve the assigned purpose. On the other side, if the delegatee refines the delegated purpose, also the delegator knows the refinement. In agreement with the transparency principle, who delegates the permission on his data to somebody else wants to know which data are disclosed to a third party and how they are used.

In Ref. 16, it is shown how one can use Datalog[1] and the DLV system[13] to model

check the correctness of the previous three models or the consistency among them.

4. Hippocratic Databases

Hippocratic databases[3] are built upon the principles rooted in the privacy regulation from US legislation.[f] Accordingly, before storing a data item, the DB system must take into account

purpose specification, i.e., the functional goal for which the data item can be collected;
consent, i.e., the permission of data owners to maintain the data item.

Furthermore, DB actions are constrained by the following privacy principles:

limited collection: the DB system can collect only that information strictly necessary to fulfill the purpose;
limited use: the DB system can answer only queries for which the purpose is equal to one stored within the required data;
limited disclosure: the DB system cannot disclose data for purposes different from those for which the data owner has previously given the consent;
limited retention: the DB system can maintain data only for the time needed to fulfill the purpose for which data are stored.

Example 1. Table 1 shows the schemata for the tables *customer*, *account* and *transaction* forming the Fineco's database.

Table 1. Database Schema

table	attributes
customer	purpose, customer-id, name, address, email
account	purpose, customer-id, account-id, number, amount
transaction	purpose, customer-id, account-id, transaction-id, payment

For each purpose and for each data item stored in databases, one defines:

- *external-recipients*: the classes of users to whom the data item may be disclosed;
- *retention-period*: the period in which the data item can be maintained in the database;
- *authorized-users*: the users entitled to access the data item.

Purpose, external recipients set, authorized users set, and retention period are stored in the database with respect to the metadata schema in Table 2 (see

[f] United States Privacy Act of 1974. EU rules are tighter. US companies processing information of EU citizen subject them self to the Safe Harbor Agreement.

Table 2. Privacy Metadata Schema

table	**attributes**
privacy-policy	purpose, table, attribute, {external-recipients}, retention
privacy-authorization	purpose, table, attribute, {authorized-users}

Ref. 3). This schema is composed of the two tables: *privacy-policy table* and *privacy-authorization table*. Privacy-policy table contains the enterprise privacy policy, while privacy-authorization table contains the access control policy that implements such a privacy policy.

Privacy-policy table describes how the enterprise will use the collected information at organizational level. Here, the enterprise only specifies generic actors (representing classes of users) to who customer information is outsourced. For instance, a bank may state in its policies that it will give customer information to a "credit card company" for credit card services. However, the bank may make agreement with different credit card companies for that purpose. External recipients essentially correspond to classes of users. Then, the privacy policy is matched with customer preferences. Here, customers can choose a particular instance of the external recipient. Such an instance is the authorized user.

Moreover, a customer can specify how long the enterprise can maintain his data. If such a period is shorter than the period required by the enterprise to fulfill the purpose, the customer information will not collected since the purpose cannot be achieved. This points out that retention period is necessary at policy level.

In summary, the additional information stored by Hippocratic DB systems is split as follows: external-recipients and retention period are in the privacy-policy table, while the authorized-users in the privacy-authorization table. The purpose is stored in both of them.

Example 2. Fineco's privacy policy is shown in Table 3. The bank can keep customer data for 3 years in order to fulfill purpose *credit card service*. Moreover, it can outsource such data to a credit card company. The credit card company refines this purpose into *issuing credit card* and *credit assessment*. To guarantee the transparency principle, we assume that Fineco stores such sub-purposes into its privacy-policy table. For *issuing credit card*, customer information has a retention period of 1 month, that is, the time needed to check customer financial credentials. For *credit assessment*, it has a retention period of at most 40 days since the payment is due by the 10th of the following month. Finally, the bank can keep customer data for 1 week to achieve purpose *delivering bank statements (BS)* and all its sub-purposes.

Hippocratic DB systems rely on the *Privacy Constraint Validator*[3] for verifying whether the customer agrees with the enterprise privacy policy by matching it with customer preferences.

Table 3. Privacy-Policy Table

purpose	table	attributes	external-recipients	retention
credit card service	customer	name	{credit-card-company}	3 years
credit card service	customer	address	{credit-card-company}	3 years
credit card service	account	number	{credit-card-company}	3 years
credit card service	account	amount	{credit-card-company}	3 years
issuing credit card	customer	name	{credit-card-company}	1 month
issuing credit card	customer	address	{credit-card-company}	1 month
issuing credit card	account	number	{credit-card-company}	1 month
issuing credit card	account	amount	{credit-card-company}	1 month
credit assessment	customer	name	{credit-card-company}	40 days
credit assessment	account	number	{credit-card-company}	40 days
credit assessment	account	amount	{credit-card-company}	40 days
delivering BS	customer	name	{delivery-company, post-office}	1 week
delivering BS	customer	address	{delivery-company, post-office}	1 week
delivering BS	customer	email	*empty*	1 week
delivering BS	account	number	{delivery-company, post-office}	1 week
delivering BS	account	amount	{delivery-company, post-office}	1 week
delivering BS by email	customer	name	*empty*	1 week
delivering BS by email	customer	email	*empty*	1 week
delivering BS by email	account	number	*empty*	1 week
delivering BS by email	account	amount	*empty*	1 week
delivering BS by hand	customer	name	{delivery-company}	1 week
delivering BS by hand	customer	address	{delivery-company}	1 week
delivering BS by hand	account	number	{delivery-company}	1 week
delivering BS by hand	account	amount	{delivery-company}	1 week
delivering BS by post	customer	name	{post-office}	1 week
delivering BS by post	customer	address	{post-office}	1 week
delivering BS by post	account	number	{post-office}	1 week
delivering BS by post	account	amount	{post-office}	1 week

Example 3. Alice's preferences would be to opt out of everything except *delivering bank statements by post*, and she may have a constraint that personal information should not be kept for more than 2 weeks. Otherwise, if Alice defines a retention period of 3 days, the DB system will reject the customer. On the contrary, the bank privacy policy is fully acceptable for Bob.

Once it is verified that the privacy policy does not violate customer preferences, data are disclosed by the customer and stored into the privacy-authorization table.

Example 4. Tables 4 and 5 show the authorizations derived by matching the bank privacy policy with Bob's preferences. Credit Union is authorized to access customer information for credit card service. Similarly, DHL, post-office and customer service are authorized to access customer information for delivering bank statements. However, customer service is not authorized to access customer address, while DHL and post-office are not authorized to access customer email.

Table 4. Privacy-Authorization Table for Credit Card Service

purpose	**table**	**attributes**	**authorized-users**
credit card service	customer	customer-id	*all*
credit card service	customer	name	{Credit Union}
credit card service	customer	address	{Credit Union}
credit card service	account	customer-id	*all*
credit card service	account	account-id	*all*
credit card service	account	number	{Credit Union}
credit card service	account	amount	{Credit Union}
credit card service	transaction	customer-id	*all*
credit card service	transaction	account-id	*all*
credit card service	transaction	transaction-id	*all*
credit card service	transaction	payment	{Credit Union}
issuing credit card	customer	customer-id	*all*
issuing credit card	customer	name	{Credit Union}
issuing credit card	customer	address	{Credit Union}
issuing credit card	account	customer-id	*all*
issuing credit card	account	number	{Credit Union}
issuing credit card	account	amount	{Credit Union}
credit assessment	customer	customer-id	*all*
credit assessment	customer	name	{Credit Union}
credit assessment	account	customer-id	*all*
credit assessment	account	account-id	*all*
credit assessment	account	number	{Credit Union}
credit assessment	account	amount	{Credit Union}
credit assessment	transaction	customer-id	*all*
credit assessment	transaction	account-id	*all*
credit assessment	transaction	transaction-id	*all*
credit assessment	transaction	payment	{Credit Union}

Users can submit queries to the database as part of their duties. In this setting, queries should contain the purpose for which the returned records will be used. A query is allowed by the *Attribute Access Control*[3] only if the user who issued the query occurs in the authorized users field and the purpose belongs to the set of purposes stored in the privacy-authorization table. For any query, the *Record Access Control*[3] discloses only data items whose purposes matches the purpose expressed in the query.

To enforce the limited retention principle, Hippocratic DB systems use the *Data Retention Manager*[3]. This module deletes data items whose retention period is expired. The same data item may be stored for more than one purpose. In this case, the data item is maintained in the database for the period of the purpose with the longer retention period.

5. Beyond Hippocratic Databases

Sometimes it is necessary to decompose a generic purpose into more specific ones. In Hippocratic DB systems, the attributes stored for a purpose cannot be changed, but *"this limitation can be circumvented by splitting a conceptual purpose into multiple*

Table 5. Privacy-Authorization Table for Delivering Bank Statement

purpose	table	attributes	authorized-users
delivering BS	customer	customer-id	*all*
delivering BS	customer	name	{DHL, post-office, customer-service}
delivering BS	customer	address	{DHL, post-office}
delivering BS	customer	email	{customer-service}
delivering BS	account	customer-id	*all*
delivering BS	account	account-id	*all*
delivering BS	account	number	{DHL, post-office, customer-service}
delivering BS	account	amount	{DHL, post-office, customer-service}
delivering BS by email	customer	customer-id	*all*
delivering BS by email	customer	name	{customer-service}
delivering BS by email	customer	email	{customer-service}
delivering BS by email	account	customer-id	*all*
delivering BS by email	account	account-id	*all*
delivering BS by email	account	number	{customer-service}
delivering BS by email	account	amount	{customer-service}
delivering BS by hand	customer	customer-id	*all*
delivering BS by hand	customer	name	{DHL }
delivering BS by hand	customer	address	{DHL}
delivering BS by hand	account	customer-id	*all*
delivering BS by hand	account	account-id	*all*
delivering BS by hand	account	number	{DHL}
delivering BS by hand	account	amount	{DHL}
delivering BS by post	customer	customer-id	*all*
delivering BS by post	customer	name	{post-office}
delivering BS by post	customer	address	{post-office}
delivering BS by post	account	customer-id	*all*
delivering BS by post	account	account-id	*all*
delivering BS by post	account	number	{post-office}
delivering BS by post	account	amount	{post-office}

database purposes.[3] " Unfortunately, this approach hides the nature of the relationship between a goal and its subgoals (e.g., AND or OR).

Another solution is proposed by Karjoth et al. (see Ref. 19) who consider purposes as strings that identify the intentions for which an operation can be executed. In their approach, purposes are ordered in a hierarchical manner with a directory-like notation. In this setting, if an operation is allowed for a given purpose, it is also allowed for all sub-purposes. Yet, it is not possible to distinguish if a sub-purpose is obtained through an AND or OR decomposition, thereby limiting reasoning about the fulfillment of the root purpose.

Our proposal is to introduce a *purpose-hierarchy table*. This table stores for each purpose, its parent; also, whether it is derived through AND or OR refinement. With this information, one can re-construct the purpose hierarchy at each node of the hierarchy.

Example 5. The bank may refine *delivering bank statements* into *delivering bank statements by email*, *delivering bank statements by hand*, and *delivering bank statements by post* (see Table 6).

Table 6. Purpose-Hierarchy Table

purpose	**up-level purpose**	**AND/OR**
credit card service		
issuing credit card	credit card service	AND
credit assessment	credit card service	AND
delivering BS		
delivering BS by email	delivering BS	OR
delivering BS by hand	delivering BS	OR
delivering BS by post	delivering BS	OR

Customers can opt in or out part of their information. The enterprise can then choose on its own among available alternatives on the basis of customer preferences; alternatively, customers, like Alice, could opt in or out a specific alternative.

Finally, in Hippocratic databases the functional requirement model is implicit, while the notion of trust is not considered. Further, the Hippocratic DB approach defines only objectives and responsibilities of actors, but does not identify who is really able to provide services. Consequently, one cannot capture availability and *need-to-know* requirements. This may lead to "hidden" clashes with privacy principles and, specifically, with the limited collection principle.

6. Modeling Hippocratic Databases with Secure Tropos

Our objective is to take an existing Hippocratic DB system and automatically derive from it Secure Tropos models which represent more directly and naturally privacy requirements of the information system. A sketch of the procedure is described here. The functional requirements model and trust management implementation at social level are derived from the privacy-policy table, while these models are instantiated by the privacy-authorizations table.

The first activity for the acquisition of requirements model involves the actor modeling phase. This phase consists of identifying and analyzing application domain stakeholders along with their intentions and entitlements.

- Actors
  - Client
  - Database Owner
  - One actor for each entity that occurs in the *external-recipients* field in privacy-policy table and in the *authorized-users* field (except for internal users) in privacy-authorization table.
- Services
  - Goals = Purposes
  - Resources = Data items

We also need to structure goals and resources. To this intent, we build goal hierarchies based on the purpose-hierarchy table. Also data items can be organized

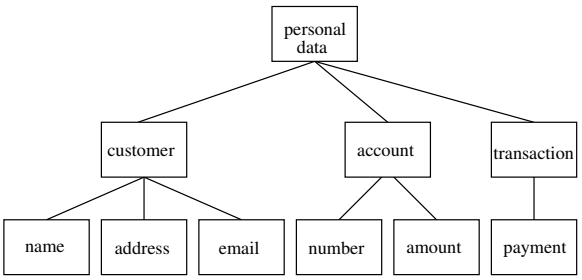


Fig. 2. Hierarchy of customer personal data

Procedure Actor_Modeling(PPT: Privacy-Policy-Table, PAT: Privacy-Authori-Table);
$C =$ new *actor*; % Client
for each {data-item} $d \in \pi_{attributes}(PPT) \cup \pi_{attributes}(PAT) \setminus *\text{-}id$ **do**
add $C \xrightarrow{o} d$;
end-do
$DO =$ new *actor*; % Database-Owner
for each {purpose} $p \in \pi_{purpose}(PPT) \cup \pi_{purpose}(PAT)$ **do**
add p in rationale of DO;
AND/OR_decomposition; % order goals wrt purpose-hierarchy table
end-do
for each {actor} $A \in \pi_{external\text{-}recipient}(PPT) \cup \pi_{authorized\text{-}users}(PAT)$ **do**
$A =$ new *actor*;
end-do

Fig. 3. Actor Modeling

hierarchically, for example, with respect to the DB schema. In our scenario, the hierarchy of personal data can be defined with respect to Table 1 as shown in Figure 2. Leaves represent data items and their parents the table where they are stored. We do not consider identifiers since we assume that they are anonymous data, that is, they do not reveal information about customers. For making Tropos diagrams more readable, we use the name of the table for representing all the data items stored in such a table. The procedure for actor modeling is shown in Figure 3.

The next step aims at identifying the relations among actors and services. The trust and ownership model is very simple: an ownership link is added for each client and data item. No trust relation is drawn since, as we have already said, Hippocratic DB approach do not support the notion of trust.

Algorithms for building the functional requirements models at social and individual levels are shown in Figure 4. At social level, for each row in the privacy-policy table, we add a functional dependency link among client/purpose/DB-owner, since clients agree with enterprise privacy policy. Furthermore, for each the row in privacy-policy table and for each actor in the external-recipients field, we add a functional dependency link among DB-owner/purpose/external-recipient. The in-

Procedure FRM_SL(PPT: Privacy-Policy-Table, DO: DB-owner, C: client);
for each {row} $R \in PPT$ **do**
$p = \pi_{purpose}(R)$;
add $C \xrightarrow{Df} p \xrightarrow{Df} DO$;
for each {actor} $A \in \pi_{external\text{-}recipients}(R)$ **do**
add $DO \xrightarrow{Df} p \xrightarrow{Df} A$;
end-do
end-do

Procedure FRM_IL(PAT: Privacy-Authorizations-Table, DO: DB-owner);
for each {row} $R \in PAT$ **do**
$p = \pi_{purpose}(R)$;
for each {actor} $A \in \pi_{authorized\text{-}users}(R)$ **do**
add $DO \xrightarrow{Df} p \xrightarrow{Df} A$;
end-do
end-do

Fig. 4. Functional Requirements Modeling at Social and Individual Level

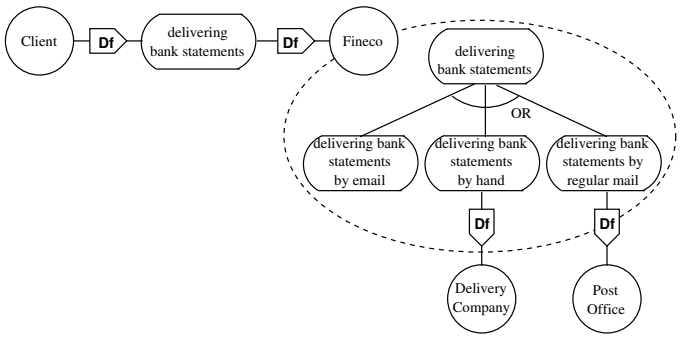


Fig. 5. Functional Requirements Model at Social Level

dividual level is similar except that dependency links are between the DB owner and authorized users. For each row in the privacy-authorization table and for each actor in the authorized-users field, we add a functional dependency link among DB-owner/purpose/authorized-user.

Figures 5 and 6 show functional requirements models for purpose *delivering bank statements* at social and individual levels, respectively. In such diagrams, we represent functional dependency links as edges labeled by **Df**. At social level, the client depends on the bank for delivering bank statements, and in turn the bank depends on delivery companies for achieving this purpose by hand and on the post office by regular mail. Further, diagrams reveal that the bank should have the capability to deliver bank statements by email by itself. At individual level, the generic delivery company is instantiated with DHL.

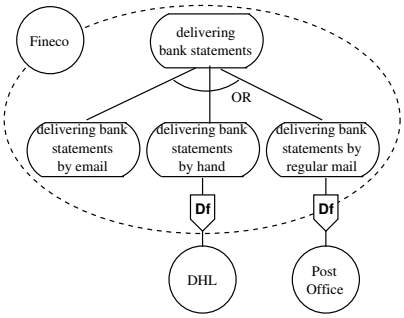


Fig. 6. Functional Requirements Model at Individual Level

Procedure TMI_SL(PPT: Privacy-Policy-Table, DO: DB-owner, C: client);
for each {row} $R \in PPT$ **do**
$p = \pi_{purpose}(R)$;
$d = \pi_{attributes}(R)$;
add p in rationale of C;
add $C \xrightarrow{Dp} p/d \xrightarrow{Dp} DO$;
for each {actor} $A \in \pi_{external\text{-}recipients}(R)$ **do**
add p in rationale of A;
add $DO \xrightarrow{Dp} p/d \xrightarrow{Dp} A$;
end-do
end-do

Procedure TMI_IL(PAT: Privacy-Authorizations-Table, DO: DB-owner);
for each {row} $R \in PAT$ **do**
$p = \pi_{purpose}(R)$;
$d = \pi_{attributes}(R)$;
for each {actor} $A \in \pi_{authorized\text{-}users}(R)$ **do**
add p in rationale of A;
add $DO \xrightarrow{Dp} p/d \xrightarrow{Dp} A$;
end-do
end-do

Fig. 7. Trust Management Implementation at Social Level

Algorithms for building trust management implementations at social and individual levels are shown in Figure 7. At social level, for each row in the privacy-policy table, we add a delegation of permission link as we have done for the functional requirements model with the only notice that we use delegation of permission instead of functional dependency. For each row in the privacy-policy table and for each actor in the external-recipients field, we add a delegation of permission link among DB-owner/data-item/purpose/external-recipient. If the external-recipients field is empty, no link is added. The individual level is similar except that delegation of permission links are between the DB owner and authorized users. For each row in the

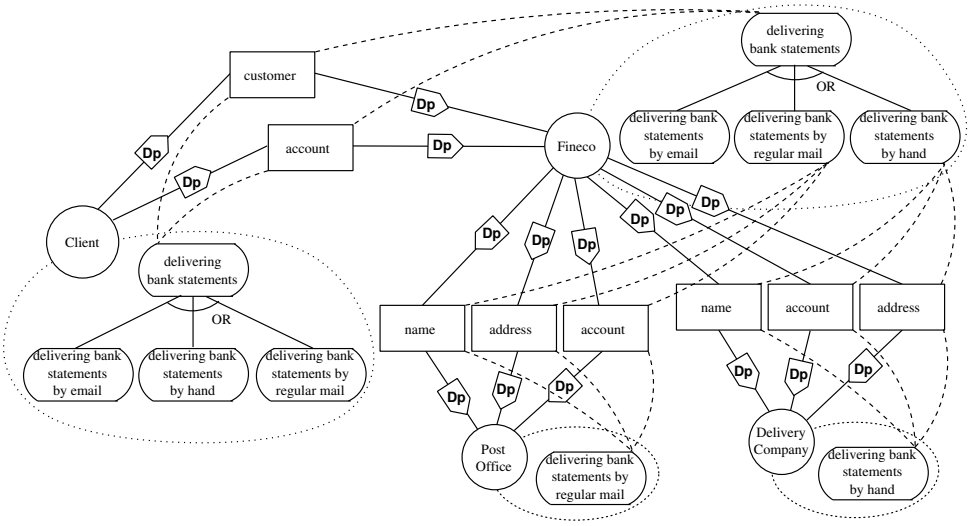


Fig. 8. Trust Management Implementation at Social Level

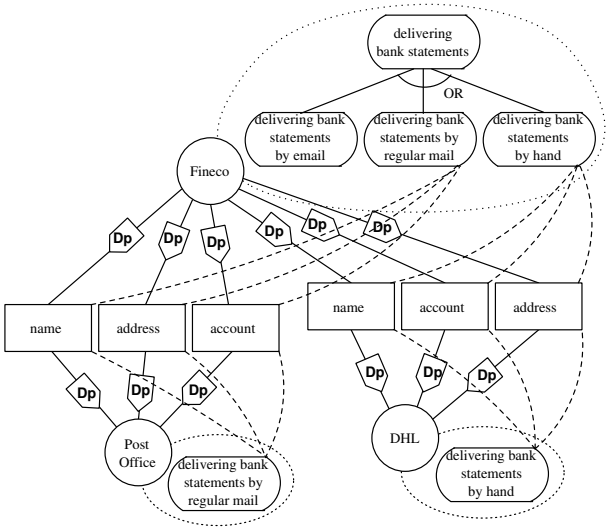
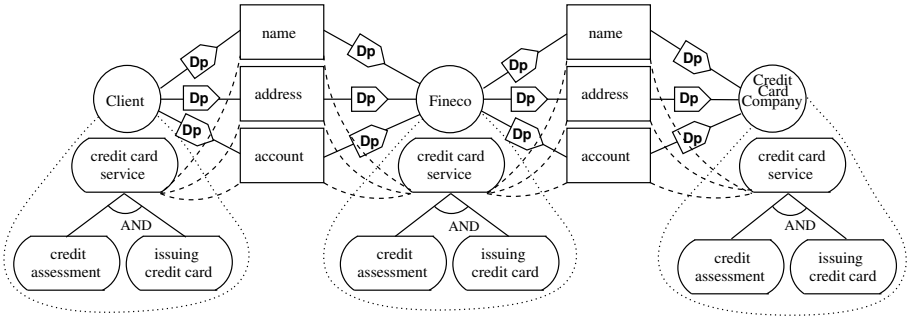


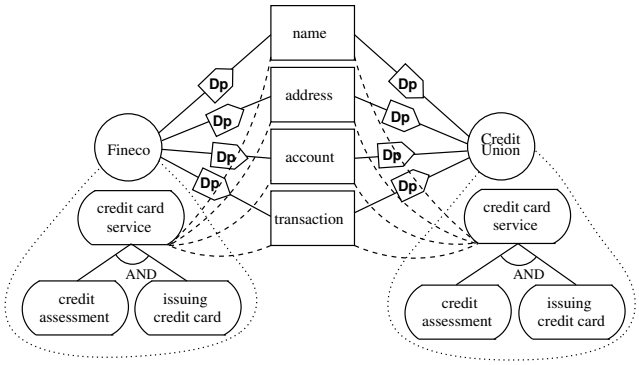
Fig. 9. Trust Management Implementation at Individual Level

privacy-authorization table and for each actor in the authorized-users field, we add a delegation of permission link among DB-owner/data-item/purpose/authorized-user.

Figures 8 and 9 show the trust management implementations for purpose *delivering bank statements* at social and individual levels, respectively. In these diagrams, we represent delegation of permission links as edges labeled by **Dp**. At social level,



(a) Social Level



(b) Individual Level

Fig. 10. Trust Management Implementation

we have taken into consideration relationships among clients and the DB owner, since clients agree with the enterprise privacy policy. On the other hand, in models at individual level we do not consider such relations since clients have no access to the privacy-authorization table in Hippocratic DB approach.

Once we have mapped privacy requirements of Hippocratic DB systems into the Tropos framework, we can use the Secure Tropos formal framework[16] to automatically check the correctness and consistency of such requirements. Applying the analysis to our scenario, an inconsistency concerning *credit card service* is detected. The trust management implementation at social level (Figure 10(a)) and the one at individual level (Figure 10(b)) show that the bank delegates the permission on a data item, *transaction*, for which it has no permission. In other words, the bank outsources customer data without the consent of the data subject.

7. Related Work

Privacy was studied under the name of security (e.g., for statistical databases[2] in the 1970s and 1980s), while now there is a clear distinction between security and

privacy research. Indeed, Alan Westin has defined privacy as *"the right of individuals to determine for themselves when, how, and to what extent information about them is communicated to others"*.

The last decades have seen an increasing awareness that privacy plays a key role in organizations. The World Wide Web Consortium (W3C) developed the P3P standard[12] to aid users to protect their personally identifiable information when they visit web sites, and web sites to formalize their privacy policies in a standard format that can be easily understood by users. Karjoth et al. propose Platform for Enterprise Privacy Practices[19] (E-P3P) that establishes how an enterprise should manage and exchange customer personal information. Also Enterprise Privacy Authorization Language[6,7] (EPAL), developed by IBM, allows an enterprise to enforce its privacy policy. This language supports enterprises for formalizing their privacy promises into policies and associating a policy to each information stored into the database. Byun et al. propose a purpose-based access control extending RBAC. In particular, they introduce the notion of purpose together with purpose hierarchies and a purpose management model for reasoning on access control (see Ref. 11). However, their hierarchies are based on the principles of generalization and specialization, and so they are not expressive enough to support the complex strategies that can be defined by enterprises.

What is still missing in these privacy-aware technologies is a procedure for checking the consistency of privacy requirements. This limitation has led to a number of research proposals that try to incorporate privacy in the software engineering process. Liu et al. propose an agent-oriented modeling framework for dealing with security and privacy requirements (see Ref. 21). In Ref. 27, it is presented a Requirement Engineering approach extending the KAOS framework, for modeling security and privacy goals and anti-goals, and for deriving attack trees automatically through antigoal refinement. These methodologies, however, are different from ours since they do not use purpose as foundation to model privacy requirements.

In Ref. 5, general taxonomies for privacy are established. These can serve as a general knowledge repository for a knowledge-based goal refinement process. He et al. present a goal-driven framework for modeling privacy requirements in the role engineering process (see Ref. 18). Antòn et al. propose a process to abstract privacy requirements from security and privacy policies, that is, a sort of re-engineering methodology (see Ref. 4).

8. Conclusions

Enterprise privacy policies define the rules that control access to customer personal information. The main contribution of this paper is a procedure for extracting privacy requirements from existing Hippocratic DB architectures and mapping them into Secure Tropos models where the correctness and consistency of such requirements can be checked. To this end, we have extended Secure Tropos in order to support the notion of purpose since it is fundamental in privacy-aware technolo-

gies. Another advantage of this approach is that it offers a unifying view of systems directly implemented as Hippocratic databases using a structured requirements engineering methodology. This allows system designers to compare different design decisions at a level that is suitable for them.

There are a number of issues left for future work:

- Find the minimum set of customer data needed to fulfill a purpose.
- Specify privacy requirements in Secure Tropos during requirements analysis, and then generate from these a Hippocratic DB system that implements these requirements.
- Derive enterprise-wide privacy policies by looking at several Hippocratic DB systems within an enterprise and merging them into a single Secure Tropos model.
- Introduce an actor hierarchy to model the hierarchical nature of organizational actors (e.g., company-division-department-group-individual worker).
- Develop trust conflict resolution techniques for conflicts between social and individual levels. For instance, this is needed when a bank relies on a delivery company that is not trusted by the customer.

Acknowledgments

This work was partly supported by the projects RBNE0195K5 FIRB-ASTRO, 016004 IST-FP6-FET-IP-SENSORIA, 27587 IST-FP6-IP-SERENITY, 2003-S116-00018 PAT-MOSTRO, 1710SR-B/P PAT-STAMPS.

References

1. S. Abiteboul, R. Hull, and V. Vianu. *Foundations of Databases*. Addison-Wesley, 1995.
2. N. R. Adam and J. C. Worthmann. Security-control methods for statistical databases: a comparative study. *CSUR*, 21(4):515–556, 1989.
3. R. Agrawal, J. Kiernan, R. Srikant, and Y. Xu. Hippocratic Databases. In *Proc. of VLDB'02*, pages 143–154. Morgan Kaufmann, 2002.
4. A. I. Antón, D. Bolchini, and Q. He. The Use of Goals to Extract Privacy and Security Requirements from Policy Statements. Technical Report TR-2003-17, NCSU Computer Science, September 2003.
5. A. I. Antón and J. B. Earp. A requirements taxonomy for reducing Web site privacy vulnerabilities. *Requirements Eng. J.*, 9(3):169–185, 2004.
6. M. Backes, G. Karjoth, W. Bagga, and M. Schunter. Efficient comparison of enterprise privacy policies. In *Proc. of the 2004 SAC*, 2004.
7. M. Backes, B. Pfitzmann, and M. Schunter. A Toolkit for Managing Enterprise Privacy Policies. In *Proc. of ESORICS'03*, volume 2808 of *LNCS*, pages 162–180. Springer, 2003.
8. D. E. Bell and L. J. LaPadula. Secure Computer System: Unified Exposition and MULTICS Interpretation. Technical Report MTR-2997 Rev. 1, The MITRE Corporation, Bedford, MA, 1976.
9. P. Bresciani, P. Giorgini, F. Giunchiglia, J. Mylopoulos, and A. Perini. TROPOS: An Agent-Oriented Software Development Methodology. *JAAMAS*, 8(3):203–236, 2004.

10. J.-W. Byun, E. Bertino, and N. Li. Purpose-Based Access Control for Privacy Protection in Relational Database Systems. Technical Report 2004-52, Purdue University, 2004.
11. J.-W. Byun, E. Bertino, and N. Li. Purpose Based Access Control of Complex Data for Privacy Protection. In *Proc. of SACMAT'05*, pages 102–110. ACM Press, 2005.
12. L. Cranor, M. Langheinrich, M. Marchiori, and J. Reagle. The Platform for Privacy Preferences 1.0 (P3P1.0) Specification. W3C Recommendation, Apr. 2002.
13. T. Dell'Armi, W. Faber, G. Ielpa, N. Leone, and G. Pfeifer. Aggregate Functions in Disjunctive Logic Programming: Semantics, Complexity, and Implementation in DLV. In *Proc. of IJCAI'03*, pages 847–852. Morgan Kaufmann, 2003.
14. D. Downs, J. Rub, K. Kung, and C. Jordan. Issues in Discretionary Access Control. In *Proc. of Symp. on Sec. and Privacy*, pages 208–218. IEEE Press, 1985.
15. P. Giorgini, F. Massacci, J. Mylopoulos, and N. Zannone. Filling the gap between Requirements Engineering and Public Key/Trust Management Infrastructures. In *Proc. of EuroPKI'04*, volume 3093 of *LNCS*, pages 98–111. Springer, 2004.
16. P. Giorgini, F. Massacci, J. Mylopoulos, and N. Zannone. Requirements Engineering meets Trust Management: Model, Methodology, and Reasoning. In *Proc. of iTrust'04*, volume 2995 of *LNCS*, pages 176–190. Springer, 2004.
17. P. Giorgini, F. Massacci, J. Mylopoulos, and N. Zannone. Modelling Social and Individual Trust in Requirements Engineering Methodologies. In *Proc. of iTrust'05*, volume 3477 of *LNCS*, pages 161–176. Springer, 2005.
18. Q. He and A. I. Antón. A Framework for Modeling Privacy Requirements in Role Engineering. In *Proc. of the 9th Int. Workshop on Requirements Eng. : Found. for Software Quality*, pages 137–146, 2003.
19. G. Karjoth, M. Schunter, and M. Waidner. Platform for Enterprise Privacy Practices: Privacy-enabled Management of Customer Data. In *Proc. of PET'02*, volume 2482 of *LNCS*, pages 69–84. Springer, 2002.
20. K. LeFevre, R. Agrawal, V. Ercegovac, R. Ramakrishnan, Y. Xu, and D. J. DeWitt. Limiting Disclosure in Hippocratic Databases. In *Proc. of VLDB'04*, pages 108–119. Morgan Kaufmann, 2004.
21. L. Liu, E. S. K. Yu, and J. Mylopoulos. Security and Privacy Requirements Analysis within a Social Setting. In *Proc. of RE'03*, pages 151–161. IEEE Press, 2003.
22. F. Massacci, J. Mylopoulos, and N. Zannone. Hierarchical Hippocratic Databases with Minimal Disclosure for Virtual Organizations. *The VLDB J.*, 2006.
23. F. Massacci, M. Prest, and N. Zannone. Using a Security Requirements Engineering Methodology in Practice: The compliance with the Italian Data Protection Legislation. *Comp. Standards & Interfaces*, 27(5):445–455, 2005.
24. F. Massacci and N. Zannone. Privacy is Linking Permission to Purpose. In *Proc. of the 12th Int. Workshop on Sec. Protocols*, 2004.
25. C. S. Powers, P. Ashley, and M. Schunter. Privacy promises, access control, and privacy management. Enforcing privacy throughout an enterprise by extending access control. In *Proc. of the 3rd Int. Symp. on Electronic Commerce*, pages 13–21. IEEE Press, 2002.
26. R. S. Sandhu, E. J. Coyne, H. L. Feinstein, and C. E. Youman. Role-based access control models. *IEEE Comp.*, 29(2):38–47, 1996.
27. A. van Lamsweerde, S. Brohez, R. De Landtsheer, and D. Janssens. From System Goals to Intruder Anti-Goals: Attack Generation and Resolution for Security Requirements Engineering. In *Proc. of RHAS'03*, pages 49–56, 2003.