Vol 5. No. 5, Sept 2012 ISSN 2006-1781
African Journal of Computing & ICT

www.ajocict.net

A Hippocratic Privacy Protection Framework for Relational Databases

[1]Oberholzer Hendrik H.J.G., [2]Ojo Sunday O & [3]Olugbara Oludayo O
Information Sciences Department/Unit
Tshwane University of Technology
Nelspruit Campus
[1]oberholzerhjg@tut.ac.za [2]ojoso@tut.ac.za [3]oludayoo@dut.ac.za

ABSTRACT
Individuals are not comfortable when disclosing their personal information to corporate organisations and are becoming increasingly concerned. Decision criteria needed for privacy protection are more complex than those that apply to access control when managing security. A typical problem in this context concerns giving individuals better control over their personal information, while at the same time allowing the organisation to process its transactions on the same personalised information. To address this difficulty, we consider extending the Hippocratic principles and model them in our Hippocratic Privacy Protection (HPP) framework that is based on the concept of privacy contracting. A prototype of the proposed HPP framework was constructed to serve as a proof of concept in order to demonstrate the developed HPP framework as an applicable and efficacious model for solving privacy problems. Based on this prototype, we afford individuals more control over their personal information. The prototype that we developed is validated against a proposed PET evaluation framework.

Keywords: Hippocratic database, privacy, contract, UML, OCL.

1 INTRODUCTION

Privacy violations against their personal data disturb individuals, especially with regard to websites that gather personal information and store it in databases for use without their consent. Government frequently introduces an ever-increasing number of new privacy laws and industrial privacy regulations. All these matters have complicated the protection of the privacy of individuals. To curb the effect of these problems, the past 15 years have witnessed the development of numerous Privacy Enhancing Technologies (PETs). Examples include technologies such as Hippocratic databases (HD), PRIME/PrimeLife and EnCoRe. All these technologies acknowledge that the individual must be in control of his/her personal information by specifying privacy preferences.

We have proposed in [1] the concept of a privacy contract as an extension of privacy policies to alleviate the various problems relating to privacy policies [2, 3]. Subsequent work entailed the incorporation of these privacy contracts in a Hippocratic privacy protection framework that enforces personalised privacy in relational databases [2]. This framework extended the fundamental principles of Hippocratic databases [3] from ten to a set of sixteen extended Hippocratic privacy principles. In this paper, we formalise this proposed HPP framework by modelling the framework using conventional modelling languages like the Unified Modelling Language (UML) and Object Constraint Language (OCL). We construct a prototype of the proposed HPP framework to serve as a proof of concept in order to demonstrate the developed HPP framework as an applicable and efficacious model for solving privacy problems. Lastly, a framework for evaluating PETs in general based on acknowledged positive characteristics of PETs sourced from literature is proposed.

We evaluated the prototype against other PETs to determine the efficacy of our prototype in giving individuals control over their personal information.

The next section briefly presents the architecture and a discussion of the HPP framework in order to present an overview of the context of our work. Section 3 elaborates on the methodology of the study using UML and OCL. We discuss and illustrate the implementation of an HPP prototype in section 4 and use screen-shots to demonstrate the main components of the framework using a scenario and linking the functionalities of the prototype to real life examples. Section 5 briefly touches on related work and subsequently evaluates the prototype as an experiment performed to test the efficacy of the HPP framework. The paper concludes with some perspectives on future work.

2 HPP framework architecture

The significance of individuals having control over their personal information is obvious: first at the lowest level of control, most remedies currently available only offer opt-in or opt-out choices on the use or disclosure of the personal data of individuals. For example, individuals may have an option to consent that organisations use their e-mail addresses, without being able to specify a purpose for the use of this data. At the next level of control, the specification of purposes allows individuals more control. Instead of opting-in or opting-out on the use of their personal data in general, individuals can now specify that organisations use their personal data for a specific purpose, for example, marketing.

Inspired by the Hippocratic Oath, [5] suggest that databases need to be re-designed to include the responsibility for the privacy of data as a fundamental tenet and that those databases that take responsibility for privacy protection are called Hippocratic databases. Hippocratic databases allow individuals to specify their privacy preferences in terms of purposes at a granular level of detail, while the database takes responsibility for the protection of the individual's privacy as a fundamental tenet. The individual is now at liberty to specify that the organisation may use his/her e-mail address for marketing purposes only. However, if willing, individuals may consent to more than one purpose relating to one or more attributes of their personal information.

Previous work [1] proposed an alternative, user-based approach to protecting the privacy of individuals in terms of privacy contracts. The concept of privacy contracts, incorporated in an HPP framework, allows the organisation to specify their privacy policies. Individuals will consequently be able to amend and incrementally add privacy agreements to conclude their privacy contracts. The HPP framework then enforces the privacy policies and privacy contracts when performing transactions. These privacy contracts consist of privacy agreements. As the privacy contract is a legal document between the organisation and the individual, in the event of a breach of privacy, the individual will then be able contest the contract in court. If the individual can prove that the organisation has breached the contract, the court might award penalties against the organisation.

Unlike the Hippocratic approach, which is data-centred, the HPP framework proposes a user-based and process-centred approach in terms of transactions. Individuals may consent to the transaction, the personal data used and the purpose of the transaction. Individuals consent to these transactions in terms of privacy agreements. Privacy agreements and transactions relate to each other and individuals might consent to these agreements at different levels of privacy. Individuals have to consent to a minimum set of mandatory transactions. In addition, individuals might also consent to optional transactions. The individual has control over what information may be used and for what purpose when the organisation performs a specific transaction. Although the enforcement of the privacy contract resides with the organisation, individuals exercise control over their personal information as they consent to specific transactions performed on their personal data for specific purposes.

The architecture of our HPP framework can be seen in Figure 1. It is based on three components at the external level, namely: to manage privacy policies (MPP), manage privacy contracts (MPC), and enforce privacy policies and privacy contracts (EPPC). The reader interested in a more elaborate discussion on the HPP framework as such, may refer to [1]. At the conceptual level, the framework is modelled using UML and OCL. The database is implemented at the internal level using the Oracle database management system.

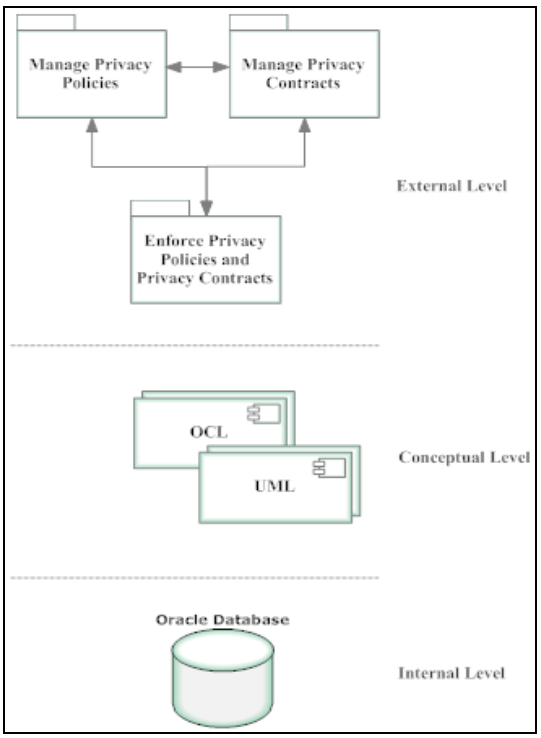

Fig. 1. HPPF database architecture

3 HPP model

From a modelling approach, a modeller needs a combination of diagrammatic and formal languages. Within UML, the Object Constraint Language (OCL) is an add-on to UML and the standard for specifying expressions that add vital information to object-oriented models [6]. As the HPP framework is innovative, we use the Unified Modelling Language (UML) to model the framework using class diagrams. Class diagrams are more applicable for modelling than Entity Relationship Diagrams (ERD); class diagrams not only include classes (entities) and attributes, but model the methods or operations applied to the classes and objects as well. However, these diagrams do not convey rules on derivation, limits or ranges, or constraints. The paper needs to describe the HPP framework formally to give it precise and unambiguous meaning.

OCL has the characteristics of an expression language, a modelling language and a formal language. OCL expressions rely on the UML-defined types and therefore include at least some aspects of UML. The latest version of OCL (OCL version 2.0) adds information to UML diagrams that UML cannot express in a diagram. Lastly, OCL is a modelling language that forms part of the UML specification and all its constructs have a formally defined meaning. Therefore, we incorporate UML combined with OCL 2.0 to build the HPP framework as a model with a high level of abstraction, independent of any implementation technology.

The HPP framework offers certain desirable characteristics. First, the HPP framework must allow an organisation to implement its privacy policies. Secondly, the HPP framework must enable the individual the opportunity to specify how the organisation can store, use and disclose his/her personal information stored in the relational database. Thirdly, the organisation must be able to perform transactions as required to run its business while the database must enforce the privacy policies of the organisation and privacy preferences of the individual. The HPP framework partitions these three main components into three packages as indicated in Figure 1.

The UML class diagrams following next do not show operations and business routines, which we are able to model using the OCL, thereby offering a robust modelling of the HPP framework. However, the UML class diagrams include all other operations. The sections that follow give a textual description that defines the OCL context rules as invariants, initial values, pre- and post conditions, body specifications and operational definitions. The framework stores and maintains all these classes in object relational database table structures. We replicated some of the classes on the UML diagrams that follow for clarity, as we cannot discuss the rest of the classes outside the context of the other UML diagrams. In addition, the operations to add, update and delete a class do not apply to every class or association class. Some operations do not appear on the UML class diagrams, as we define these operations using the OCL 'define' clause.

For every UML diagram that follows, a brief textual description of the operations belonging to some of the classes is given. Based on the extent of the UML diagrams, only the most significant classes will be discussed, followed by the OCL definitions of the class, defining its context, pre- and post conditions and where applicable the body of the operations.

3.1 Manage privacy policies

From an organisational point of view, an organisation normally publishes a general privacy policy stating how they will handle the personal or private information of individuals. In contrast to publishing a general privacy policy, we propose an interactive and customised manner through which the organisation can install and maintain its privacy policies. Firstly, the chief privacy officer (CPO) together with other stakeholders of the organisation (henceforth referred to as the privacy policy team or PPT) has to define all the transactions and the data items that every transaction needs to process into meaningful information. The PPT also has to define for every transaction and data item the purposes why the organisation needs to store, access, process and disclose this data. In addition, the PPT also has to define the privacy laws and regulations that apply to every transaction in order to adhere to these laws and regulations.

The package named Manage Privacy Policies implements the privacy policies defined by the PPT. This package consists of two components: namely, Create Privacy Metadata Tables and Maintain Privacy Policies. We will not discuss these components separately as the framework has integrated them together. It is the responsibility of the database administrator (DBA) to create and maintain the metadata tables and the responsibility of the PPT to dedicate staff to capture and maintain the metadata of the privacy policies.

Creation of the privacy metadata tables consists of an automated script that the DBA runs to create the relational database tables that will store the data. This is an automated script to create all the necessary tables and do the inserts of all rows of data. If needed, the PPT of an organisation may change the script to adapt to their specific kind of business. The two statements list examples of the Oracle Structured Query Language (SQL) (see Figure 2). The *CREATE TABLE* statement creates the metadata table named transaction and the *INSERT* statement adds one row to the transaction table.

```
CREATE TABLE transaction
(trCode          CHAR(3)
        PRIMARY KEY
          CHECK trCode = UPPER(trCode),
trName           VARCHAR2(40),
trDescription    VARCHAR2(71),
trType           CHAR(1)
                   CHECK trType IN ('B','I','M'),
trMandatory      BOOLEAN,
trDate           DATE);
```

```
INSERT INTO transaction VALUES
('PCC', 'Perform Credit Check',
'A Mandatory Transaction required to CHECK Credit References',
'I',
'True,
TO_DATE('09-AUG-2011','dd-mon-yyyy'));
```

Fig. 2. Oracle SQL Create Table and Insert statement

The UML class diagram in Figure 3 models the package to manage privacy policies. The diagram models transactions, transactional purposes, purposes, transactional attributes, attributes, tables, transactional users, users, access kinds, transactional user access, transactional laws, privacy laws and privacy breach resolutions.

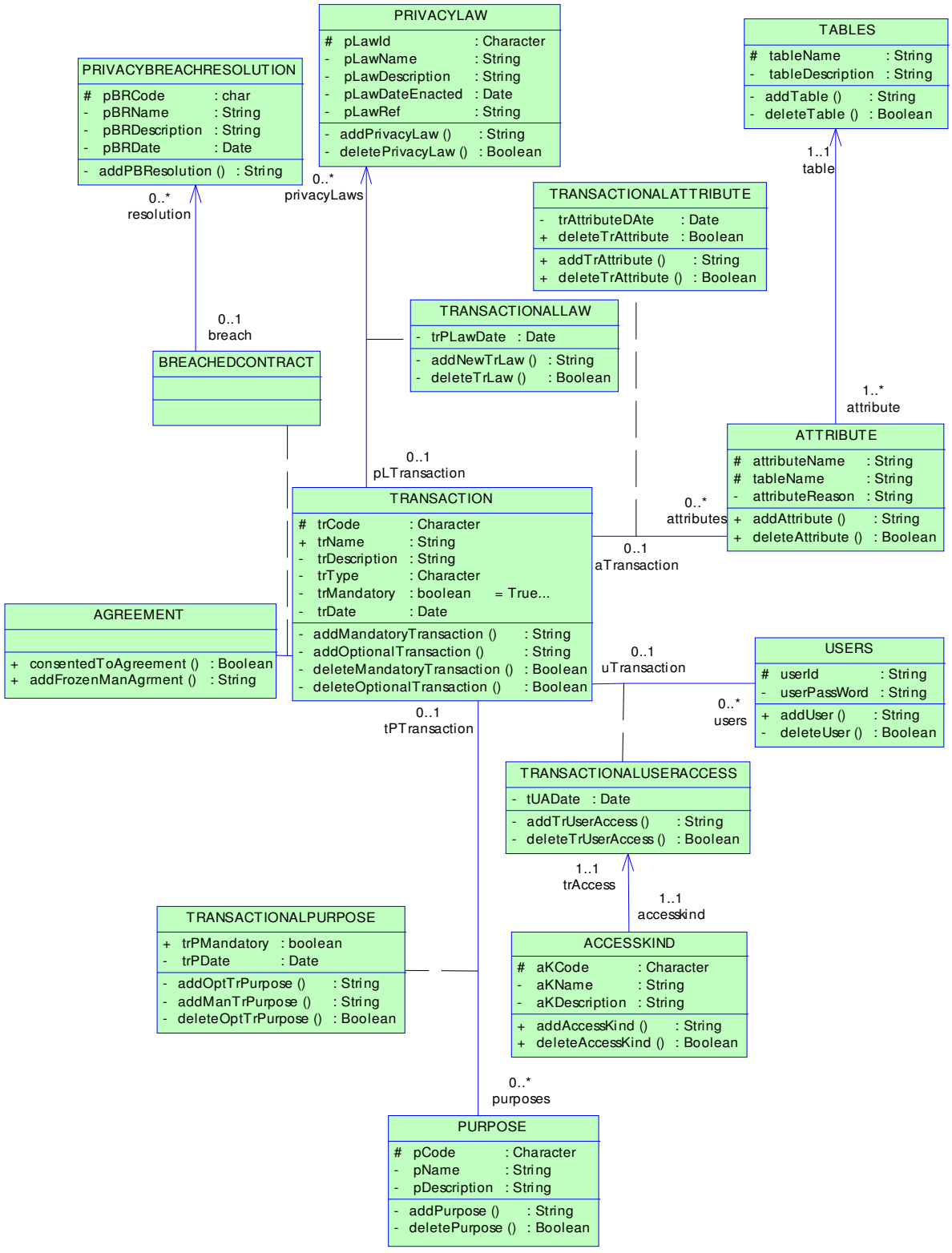

Fig. 3. UML class diagram representing package: Manage Privacy Policies

Maintain a transaction

When maintaining transactional operations, careful attention is necessary and it is critical to distinguish between mandatory and optional transactions, because their operations are contextually different. The context of class transaction defines several operations of which the framework lists their rules below.

/* The context of the following operation adds a mandatory transaction to table transaction after the framework has become operational. The rest of the transactional operations are defined in the package to manage privacy contracts */
context Transaction::addMandatoryTransaction(
newManTrCode:Character,

newManTrName:String,

newManTrDescription:String,

newManTrType:Character,

newManTrMandatory:Boolean,

newManTrDate:Date):Transaction
def: newManTrMandatory:Boolean = true
newManTrDate:Date = sysdate -- *the system date*
pre: self.Transaction → excludes(newManTrCode)
post: self.Transaction → includes(newManTrCode)

/* The context of the following operation ensures that the newly added mandatory transaction is added to all privacy agreements, but these agreements must be frozen as the individuals have not yet consented to this mandatory transaction. Although class agreement fits better to the package that manages privacy contracts, the framework includes the context to add a frozen mandatory agreement to this package because it logically fits better here */
context Agreement::addFrozenManAgrment(cNo:Integer,

manTrCode:Character,

manAgreeDate:Date,

manAgreeLevel:Integer,

manFrozen:Boolean): Agreement
def: manAgreeDate:Date = sysdate
manAgreeLevel:Integer= 0
manFrozen:Boolean = true
pre: self.Agreement → excludes(ManTrCode)
post: self.Agreement → includes(ManTrCode)

/* The context of the following operation adds an optional transaction to table transaction */
context Transaction::addOptionalTransaction(
newOptTrCode:Character,

newOptTrName:String,

newOptTrDescription:String,

newOptTrType:Character,

newOptTrMandatory:Boolean,

newOptTrDate:Date):Transaction
def: newOptTrMandatory:Boolean = false
newOptTrDate:Date = sysdate
pre: self.Transaction → excludes(newOptTrCode)
post: self.Transaction → includes(newOptTrCode)

/* The context of the following operation deletes a mandatory transaction from table transaction and ensures that no privacy agreements, no transactional attributes and no transactional purposes relating to this transaction exist */
context Transaction::deleteMandatoryTransaction(
delManTrCode:Character):Boolean
inv -- if no agreements exist for this transaction, no level 2 or level 3 agreements exist
self.agreements → excludes(delManTrCode)
pre: self.Transaction → includes(delManTrCode)
post: self.Transaction → excludes(delManTrCode)
self.attributes → excludes (delManTrCode)
self.purposes → excludes (delManTrCode)
self.agreements → excludes (delManTrCode)

Updating a transaction as a policy element is not straightforward. If the framework updates a mandatory transaction, the modification affects all existing privacy contracts because all the existing contract owners had to consent to this mandatory privacy agreement. However, this modification does not affect all new privacy contracts as the new contract owners have not consented to this specific privacy agreement yet. The obligation to update a mandatory privacy agreement requires that the framework freeze all existing privacy contracts until the contract owners have consented to the specific mandatory privacy agreement. Therefore, it is not advisable to update existing privacy contracts. The best solution to this dilemma will be to define a new, alternative mandatory transaction.

To conclude, ICT developers are not legal professionals and might find it difficult to interpret and implement all the legal and regulatory requirements required by law to protect personal information. Therefore, teams of privacy professionals and legal administrators should collaborate with ICT developers to embed the most applicable privacy legislation and regulations in the definition of privacy policies and statements that the HPP framework has to enforce when applying privacy rules to business applications and transactions.

3.2 Manage privacy contracts

The second package named Manage Privacy Contracts enables the creation and manipulation of a single privacy contract for every individual that needs to interact with the organisation. Privacy contracts allow every individual customisation of his/her privacy

contract through customised privacy agreements, related to a specific transaction, chosen by the individual according to his/her unique privacy preferences. It must be noted that whenever the organisation makes any changes to a privacy policy, all affected privacy agreements stay as they were until the individual has consented to the new or changed privacy agreement. The issue of changing privacy policies is handled by the component that manages privacy policies (see section 3.1) and the act of getting new or renewed consent, is handled by the package that enforces privacy policies and contracts (see section 3.3). The management of privacy contracts consists of three components, namely: Authenticate Privacy Contracts, Create Privacy Contracts and Maintain Privacy Contracts. We will not discuss these components separately as the framework has integrated them together.

The UML class diagram listed in Figure 4 models the package to manage privacy contracts. The diagram models the classes representing credit references, privacy contracts, agreements, privacy agreements at level two, privacy agreements at level three and an audit log on agreements. The following OCL rules state the initial value rules of the classes before the paper addresses the contexts of each of these classes.

context CreditReference::verified
init: false
context Contract::active
init: false
context Agreement::frozen
init: false
context Contract::contractDate
init: sysdate
context Contract::lastUpdate
init: sysdate
context AgreementAuditLog::trDate
init: sysdate

Create a privacy contract

When an individual enters into a contract with the organisation, the framework has to create a new contract for the individual. The framework generates a new contract number which is one greater than the previously created contract number. Unique identifiers like student numbers, account numbers and more might serve as contract numbers. The post condition of the createContract operation ensures that the new contract owner is included in the list of existing contract owners.

/* The context describing the following operation creates a new contract */
context Contract::createContract(newContract:Contract):Contract
inv self.Contract.contractNo → isUnique(contractNo)
pre: newContract = ''
post: contractNo = contractNo@pre + 1
Contract = Contract@pre → including(newContract)

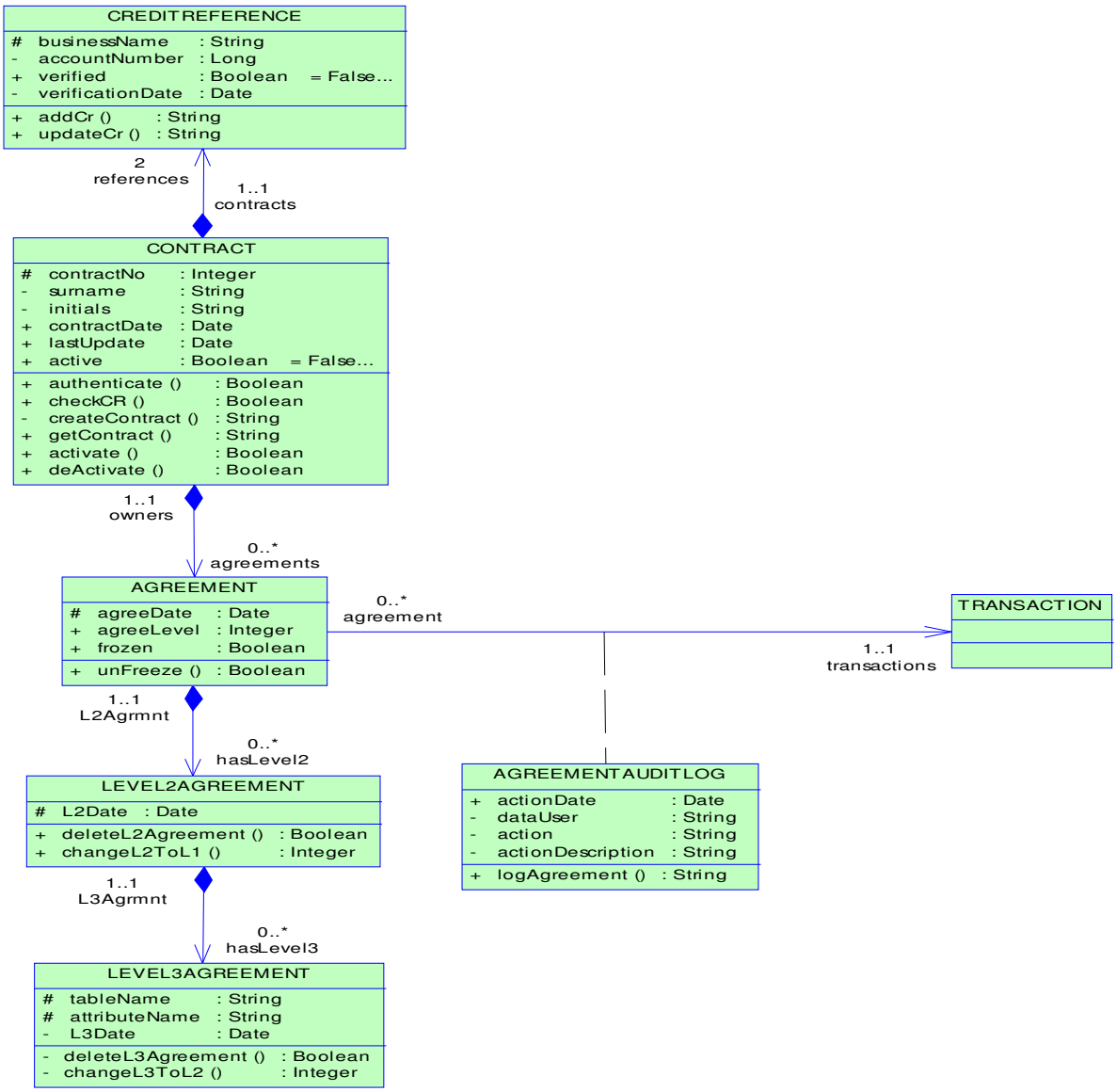

Fig. 4. UML class diagram representing package: Manage Privacy Contracts

Consent to mandatory privacy agreements at privacy level zero

One of the reasons why individuals do not bother to read privacy policies is that they are boring to read [3] and very extensive wording hides a considerable amount of policy fine print [2]. Instead of consenting to a long list of policy statements that cover the whole spectrum on how the organisation will protect the personal information of the individual, the framework allows individuals to consent to a small number of mandatory agreements initially in order to activate their privacy contracts. Individuals have to consent to mandatory privacy agreements at level zero.

These mandatory agreements at level zero represent the mandatory transactions. When the individual has consented to all the mandatory agreements, the framework can activate the privacy contract belonging to the specific individual and the organisation may use all the data of the individual relating to the specific transactions for the purposes consented to by the individual.

```
/* The context describing the following operation requires the individual to consent to the set of mandatory transactions at level zero */
context   Agreement
def:      consentToMandatoryTransactions( ): Set( Transaction ) =
                    self.transactions → iterate     (
                                                                t
          : transaction;

          resultSet  : Set( Transaction ) = Set{ } |
                                                                if
t.trMandatory = true then

          resultSet.including( t )
                                                                else

          resultSet
                                                                endif
                                                          )

pre:      self.owners.active = false
post:     self.owners.active = true
          self.agreeDate = sysdate
          self.agreeLevel = 0
          self.owners.lastUpdate = sysdate
          Agreement = Agreement@pre → including( t )
```

A possible challenge might arise when the PPT has to add a mandatory transaction to the transaction class at a later stage. In this case, all existing privacy agreements based on the specific transaction have to be frozen. The reason for freezing all these privacy agreements is that individuals with existing privacy contracts must first consent to the new mandatory transaction (policy) that the PPT has added, before the framework allows the organisation and the individuals to perform the specific transaction. If the individuals do not consent to the new mandatory transaction, the specific privacy agreement will stay frozen, otherwise their privacy agreements will be unfrozen when they have given their consent and they would then be able to commence with the transaction against the database. The organisation will only be able to handle data according to privacy agreements that are not frozen.

Activate a contract

Active contracts are privacy contracts to which the individual has consented, against which the organisation may process database transactions. When a contract is inactive, the framework allows two transactions against the contract only, namely: reactivating the contract and performing an audit against the contract.

As soon as the framework has created the new privacy contract, verified the credit references and the individual has consented to the mandatory privacy agreements (transactions), the framework has to activate the contract. Individuals must consent to all mandatory transactions, whether they are business, management or personal transactions.

```
/* The following context defines the operation to activate a contract */
context   Contract::activate( cNo:Integer):Boolean
inv:      self.references.verified = true
pre:      self.active = false
post:     self.active = true
```

Consent to an optional privacy agreement at privacy level one

An alternative way of stating "Consenting to an optional transaction at privacy level one" would be to state, "Adding a privacy agreement at privacy level one". Only when the individual has consented to all the mandatory agreements (transactions) at level zero and the framework has activated his/her privacy contract, may he/she add optional privacy agreements at levels one, two, or three to which he/she might consent. Optional transactions or privacy agreements at level one imply that although the transaction is optional, the transaction has mandatory purposes to which individuals have to consent and the organisation may use the data related to the specific transaction for the consented mandatory purposes only, otherwise neither the organisation nor the individual can process the specific transaction. If the individual has not consented to the specific optional transaction, this implies that the organisation may not use the personal information that relates to the specific transaction for this individual. As soon as the individual has consented to the privacy agreement at level one, the framework has to add the privacy agreement to the privacy contract of that specific individual.

```
/* The following context defines the operation to consent to an optional privacy agreement at level one – the purpose is mandatory */
context   Agreement
inv       isNotFrozen:          self.frozen = false,
          -- active = true in the next rule implies mandatory agreements were consented to
          isActive:  self.owners.active = true
def:      consentToOptionalAgreementAtLevel1(
                    transactions.optTrCodeL1:Character)     :     Set(
                    TransactionalPurpose                )           =
                    self.transactions.transactionalPurpose →
                    iterate     (
                                t           : transaction;
                                tP          : transactionalPurpose;
                                resultSet   : Set( TransactionalPurpose )
= Set{ } |

                                if    t.trMandatory     =     false     AND
                    tP.trPMandatory = true then
                                            resultSet.including( tP )
                                else
                                            resultSet
                                endif
                                )
pre:      self.Agreement → excludes self.Agreement( optTrCodeL1 )
post:     self.agreeDate = sysdate,
          self.agreeLevel = 1
          agreement.owners.lastUpdate = sysdate
          Agreement = Agreement@pre → including( t )
```

Log an agreement

As soon as an individual has consented to a non-existing agreement, that is, one that did not exist previously, the framework has to insert a record into class agreementAuditLog for auditing purposes.

/* The context describing the following operation logs an agreement that was consented to in the agreementAuditLog */
context AgreementAuditLog::logAgreement(cNo:Integer,

trCode:Character,

actionDate:Date,

dataUser:String,

action:String,

actionDescription):agreementAuditLog
pre: AgreementAuditLog → AgreementAuditLog → excluding(cNo)
AgreementAuditLog → AgreementAuditLog → excluding(trCode) AgreementAuditLog → AgreementAuditLog → excluding(actionDate)
post: AgreementAuditLog → AgreementAuditLog → including(cNo)
AgreementAuditLog → AgreementAuditLog → including(trCode) self.actionDate = sysdate

3.3 Enforce privacy policies and privacy contracts

The third package, named Enforce Privacy Policies and Contracts, protects the personal information of individuals through enforcement of a customised privacy contract that belongs to a specific individual, while at the same time enforcing the privacy policies of the organisation. This package defines operations that the framework uses when it needs to process personal or business transactions and perform the obligations when updating a privacy policy of the organisation. [7] require that every electronic contract has the ability to verify and enforce non-violation of the terms of the contract by the parties involved. Conflicts in privacy practices occur when individuals have to reveal personal information, while at the same time wishing to preserve their privacy. In order to open an account, the individual not only has to supply personal information like a home address, but also very private information like banking details and gross income. The organisation might need this information in order to approve or deny the application. Eventually, the individual still has the choice either to disclose all the required information or not to have the convenience of an account.

Figure 5 presents the UML class diagram of the package that enforces privacy policies and contracts. The diagram models the following classes: 'CustomerDetail', 'Invoice', 'Product', 'AgreementAuditLog' and 'BreachedContract'.
The purpose of this package is to enforce the privacy policies of the organisation and the privacy contracts of individuals while interacting through transactions. The package first has to determine whether to manage privacy policies or contracts, or to process database transactions (enforcing policies and contracts). When opting for the third option the data user must first choose between performing business transactions or personal transactions and quitting the system.

1) 3.3.1 Verification operations

We define the context of operations to perform verification of various aspects of the framework before the framework can proceed to perform a transaction next.

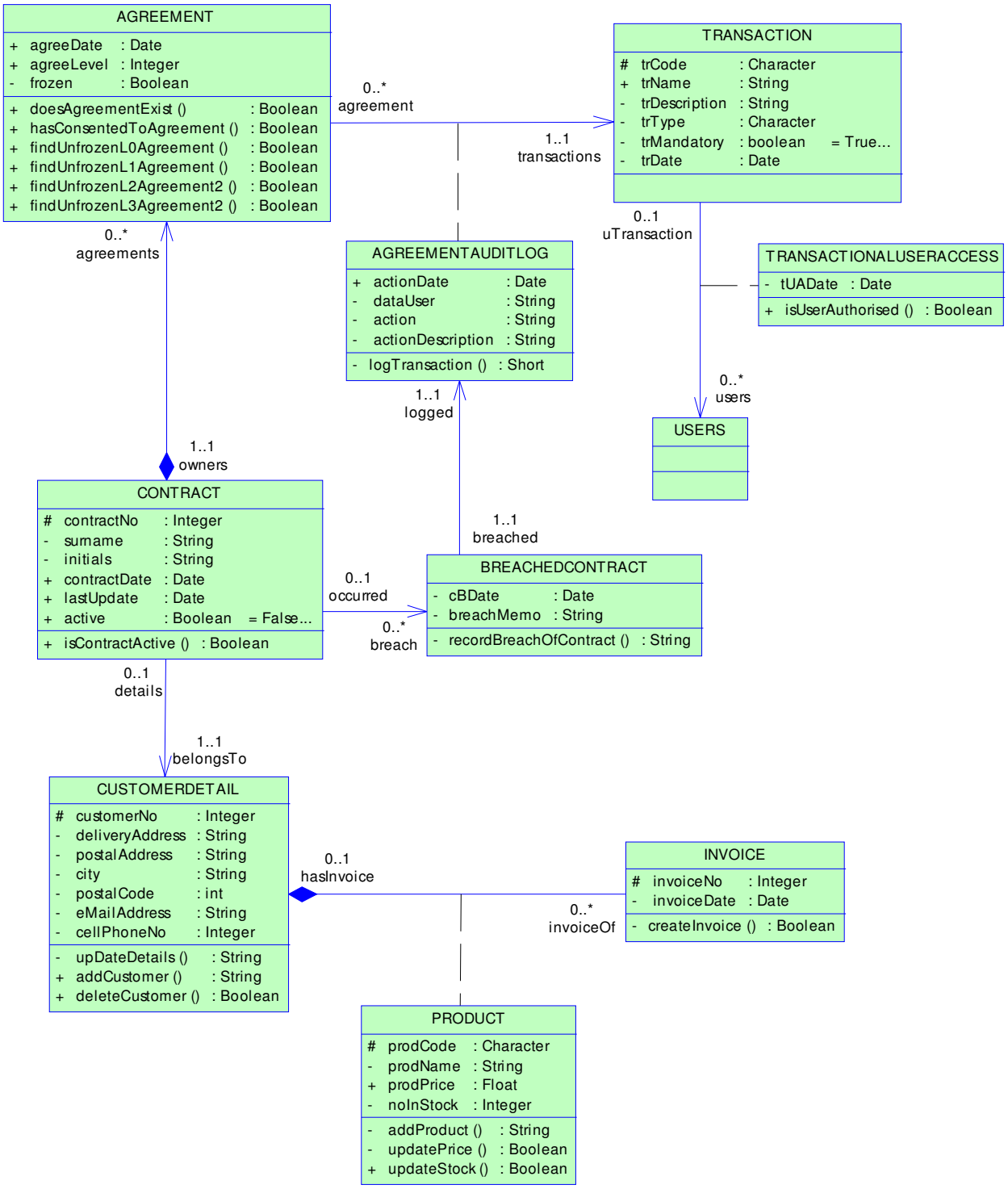

Fig. 5. UML class diagram for package: Enforce Privacy Policies and Contracts

Verify whether the data user has been authorised to perform the specific transaction

When a data user initiates a transaction, the data user has to indicate whether the transaction type is a group or individual type of transaction. In addition, the user must select the specific transaction and specify the kind of access that he/she requires. The framework checks the applicable privacy policy to ensure that the data user is authorised to perform the specific transaction.

/* The following context defines the operation to verify whether the data user has been authorised to perform the specific transaction */
context TransactionalUserAccess::isUserAuthorised(
trCode:Character,

trUAUserId:String,

trUAAKCode):Boolean
inv: self.TransactionalUserAccess → includes(trCode) AND
self.TransactionalUserAccess → includes(trUAUserId) AND
self.TransactionalUserAccess → includes (trUAAKCode)

Verify that the privacy contract is active

As soon as the framework has verified that the data user is authorised to perform the specific transaction, the framework has to verify that the privacy contract of the individual, on whose data it is required to perform a specific transaction is active.

/* The following context defines the operation to verify that the privacy contract of the specific individual is active */
context Contract::isContractActive(cNo:Integer) : Boolean
inv: isActive: self.active = true

Verify whether the individual has consented to such a privacy agreement

Whenever an individual requests the organisation to process a particular transaction, or the organisation needs to process the data of one or more individuals, the first step is to verify whether a privacy agreement related to the specific transaction does exist in the contract of the individual concerned and whether the agreement is not frozen. If such a privacy agreement does not exist, the individual has the option of first consenting to the necessary privacy agreement before the organisation can perform the transaction.

/* The following context defines the operation to verify whether the individual has consented to such an agreement */
context Agreement::hasConsentedToAgreement(
cNo:Integer,

trCode:Character):Boolean
inv: self.frozen = false
Agreement → includes(trCode) AND
Agreement → includes(cNo)

We make an important distinction between business transactions and personal transactions when it comes to the enforcement of privacy policies and privacy contracts. Business transactions are transactions that the organisation or a data user employed by the organisation initiates while a personal transaction is initiated by the individual that requests that a data user performs the specific transaction on his/her personal data.

Business Transactions

We distinguish between group business transactions and individual business transactions. Group business transactions are transactions that the organisation may perform on the data of a group of individuals according to the personal privacy agreements defined in the privacy contracts of the individuals. However, individual business transactions are transactions that the organisation may perform on the data of a single individual according to the personal privacy agreements defined in the privacy contract of the specific individual. A data user can perform a group business transaction relating to a common privacy agreement on the data of a group of individuals, while the data user can perform an individual business transaction for one individual at a time. However, the individuals themselves cannot request the organisation to perform a business transaction as they can only request personal/individual transactions. Only data of those individuals who have consented to the common privacy agreement are included in the transaction. The framework logs the business transaction for every individual whose data are included in the group processing for auditing purposes. The paper has already addressed the context of authorising whether the organisation has granted the data user the right to perform the specific business transaction. As soon as the framework has verified that the data user is authorised to perform the specific transaction according to the kind of access the data user requested, the framework allows the data user to perform the business transaction on the data of those individuals who concluded privacy agreements relating to the specific transaction.

Log a transaction

/* The context describing the following operation logs a transaction to class agreementAuditLog */
context AgreementAuditLog::logTransaction(cNo:Integer,

trCode:Character,

actionDate:Date,

dataUser:String,

action:String,

actionDescription):agreementAuditLog
pre: AgreementAuditLog → AgreementAuditLog → excluding(cNo)
AgreementAuditLog → AgreementAuditLog → excluding(trCode) AgreementAuditLog → AgreementAuditLog → excluding(actionDate)

post: AgreementAuditLog → AgreementAuditLog → including(cNo)
AgreementAuditLog → AgreementAuditLog → including(trCode) self.actionDate = sysdate

4 Real life prototype implementation

This section presents a prototype implementation of our HPP framework as proof of the concept to demonstrate that it is possible to implement the framework and to demonstrate its efficacy. The prototype illustrates some of the more significant concepts referring to the three packages discussed in the previous section.

The first screen of the HPP application displays the application's main window listing the main options to manage privacy policies, manage privacy contracts, enforce privacy policies and privacy contracts (see Figure 6). The application requests the data user to login. The application authenticates the credentials of the user and verifies whether the framework has authorised the user for the requested type of access and the transaction the user wants to perform. The application displays an informative alert if the user fails one of these verification tests.

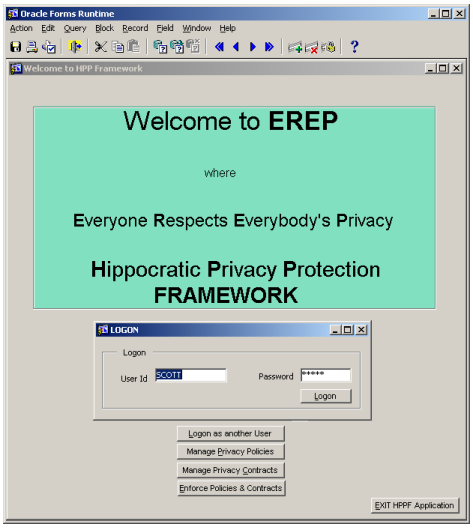

Fig. 6. HPP Application Logon screen

4.1 Manage privacy policies

The main interface to manage privacy policies offers the data user menu options to manage table and attribute objects, transactional purposes, transactional columns, user transactions, privacy laws and transactions and privacy resolutions. The following paragraphs describe the management of transactional purposes and transactional users. When the user chooses to manage the transactional purposes, the application displays the window as listed in Figure 7.

This screen-shot illustrates a subset of the transactions and their associated purposes that are available to the data user. The frame at the top of the figure displays the transaction codes, their associated names and descriptions, whether the transaction is mandatory (T), or optional (F) and the type of transaction, indicating whether the user performs the transaction for management purposes (M), for individual/personal purposes (I), or for using in bulk/groups (B). The screen-shot also displays the date when the responsible user (appointed by the PPT) added the transaction to the privacy policy database.

On viewing transaction MCR (Manage Credit References), for example, it is clear that this transaction is mandatory. When managing privacy contracts (see next section), the individual will have no choice but to consent to this transaction. The frame at the bottom of the figure illustrates that the user may use this transaction for any of the three purposes, namely, to check credit references, manage credit references, or to open an account, depending on the consent given by the specific individual and whether the purpose is mandatory or optional. This table also indicates whether the specific purpose relating to this transaction is mandatory (T) or optional (F), as individuals may also consent to privacy agreement levels one to three for the purposes that are optional.

When the individual consents to an optional transaction at privacy agreement level one, the individual does not have a choice regarding when the purpose of the specific transaction is mandatory (level one). See the mandatory purposes CCR and MCR, for example. However, at privacy agreement level two, the individual has the choice of whether to consent to the additional optional purposes as to why the specific transaction may be performed by the user or not. This means that when the individual consents to transaction MCR, which is mandatory, s/he will have to consent to the purpose CCR (Checking Credit References) and MCR (Managing Credit References); however, s/he may choose to consent to the optional purpose OAC (Opening an Account) or not. This specific window illustrates that the framework categorised transactions in such a manner as to afford that individuals a wide choice when consenting to the use of these transactions.

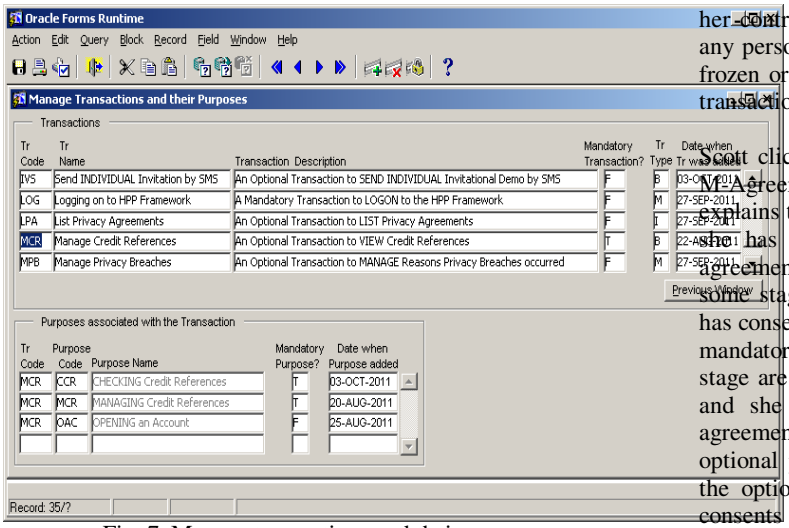

Fig. 7. Manage transactions and their purposes

4.2 Manage privacy contracts

As Alice is a very popular individual, cited in many privacy scenarios and literature case studies, she will interact with the prototype to illustrate the privacy contract concepts in a scenario.

Recently, while undergoing a procedure at a state hospital, she contracted HIV/AIDS during a blood transfusion. Fortunately, antiretroviral medicine is available, but the government directed that patients should purchase this medication on account using a prescription only. In addition, they want to know the age and gender of every HIV patient. The HPP application can retrieve this information from the social security number (I.D. number) of every individual. It is mandatory that customers provide this information to pharmacies without which the life-prolonging medication cannot legitimately be dispensed to them. Alice is also an internationally renowned researcher, of which privacy is one of her favourite research interests. She quickly realises that she has no option but to disclose some personal details and open an account or do without the medicine.

The pharmacy also needs either her postal address or e-mail address to which her statement must be sent at the end of the month. She does not really have a choice and realises that she has to consent to the use of her personal data as required. As soon as Alice requests the pharmacy to open an account, the pharmacy refers her to the data user, Scott, who will be assisting her with opening an account and dispensing her medication. Scott informs Alice that in order for her to open an account at EREP (the pharmacy where "Everybody Respect Everybody's Privacy"), she needs to enter into a privacy contract with EREP. The contract should put her mind at ease that EREP will respect her personal information and use her information only according to the privacy agreements to which she consents. He tells her that she has to remember the number of her privacy contract, as she will need to provide this unique number every time she wants to add, modify, or delete some of her privacy agreements. She also has to provide her contract number to EREP when requesting EREP to process any personal transactions in future. The contract is currently not frozen or active, as Alice has not yet consented to the mandatory transactions or agreements (see Figure 8).

Scott clicks on the button to list the mandatory agreements (List M-Agreements in Figure 8) to which she has to consent and explains the mandatory transactions to Alice. He also tells her that she has to consent to every mandatory transaction or privacy agreement; otherwise, he will not be able to create her contract. At some stage, Scott has activated Alice's privacy contract and she has consented to the mandatory privacy agreements. However, the mandatory privacy agreements that she has consented to at this stage are a limited subset of agreements that are available to her and she now has the option to consent to optional privacy agreements based on optional transactions. Concluding an optional privacy agreement requires the individual to consent to the optional transaction at any time, as long as the individual consents before he/she or the organisation processes the optional transaction. Every individual has the opportunity to define his/her own preferred privacy level for every optional transaction. Optional transactions are available at privacy levels one, two, or three.

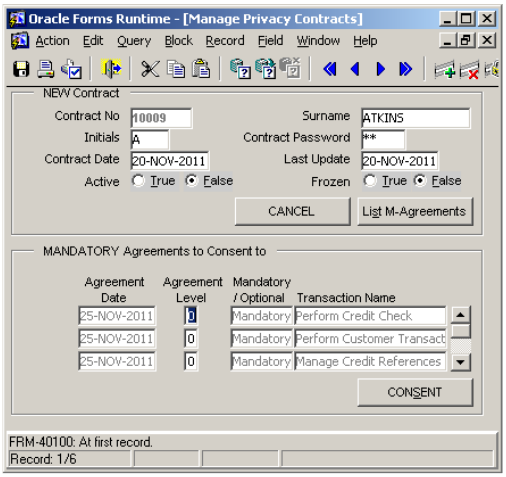

Fig. 8. Creating a privacy contract for Alice Atkins

The framework associates optional transactions consented to at privacy level one with mandatory purposes and mandatory data items relating to mandatory purposes only. No optional purposes associated with an optional transaction are available when consented to at level one. No opt-in or opt-out choices are available, except with regard to the optional transaction relating to the specific mandatory purpose. Thus, level one provides a take-it-as-it-is-or-not option for every optional transaction associated with its mandatory purposes and mandatory data items at level one. For example, Alice might decide that she needs EREP to deliver her medicine at home, as she values her privacy too much to fetch her HIV medication from one of the dispensary counters of EREP. Therefore, she will have to consent to the optional transaction at level one to deliver her medicine at home. However, this optional transaction has a mandatory purpose named 'delivering medicine at home' (see Figure 9). This means that EREP will require her home address. Alice does not have any opt-in or opt-out choices to choose from at level one, except consenting to the optional transaction or not. However, if Alice wants EREP to deliver her medication, then she has to consent to this optional transaction with its mandatory purpose.

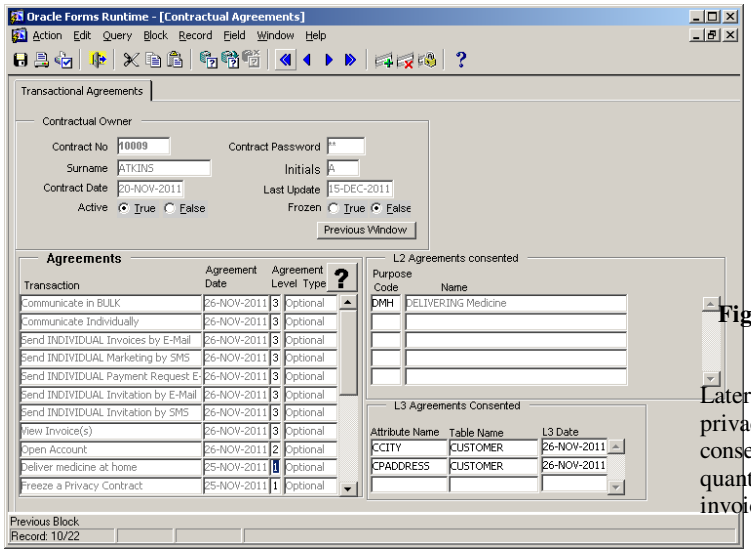

Fig. 9. Privacy agreement at level one - mandatory purpose to deliver medicine at home

We may present more screen-shots similar to these figures for privacy agreement levels two and three, but based on the extent of this work, this should suffice. The following section demonstrates how the prototype enforces the privacy policies while at the same time adhering to the privacy preferences set and consented to by individuals through their privacy contracts.

4.3 Enforce privacy policies and privacy contracts

The prototype has implemented a fictitious pharmacy called EREP where Alice consented to privacy agreements at various levels of privacy, which constituted her privacy contract. This section illustrates some simulated transactions between Alice and EREP, namely purchasing on account, viewing invoices and querying EREP on how the pharmacy used her personal information. On the 12th of December 2011, Alice visits EREP to submit her prescription for antiretroviral medicine. Scott welcomes her. She hands over her prescription to Scott and requests him to dispense the medicine on account. He asks her for her contract number after which she enters her password secretly. The application authenticates her and checks that her contract is active but not frozen. He enters her prescription on the system, saves the invoice and hands over the medicine (see Figure 10).

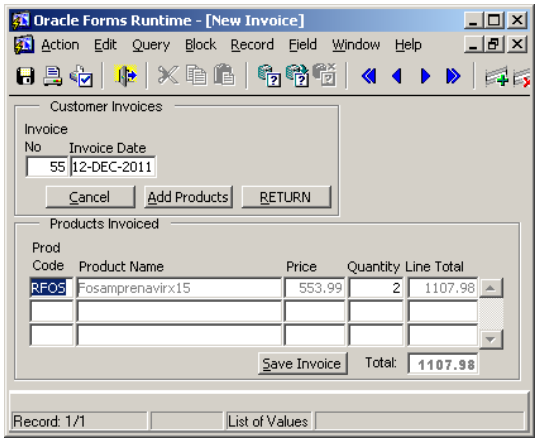

Fig. 10. Invoice number 55 - Alice purchasing antiretroviral medicine on account

Later Scott attempts to view Alice's invoices. However, as a privacy fundamentalist, Alice, in order to protect her privacy, consented that a data user may see only her invoice numbers, the quantity of every item purchased and the total price of every invoice (see Figure 11).

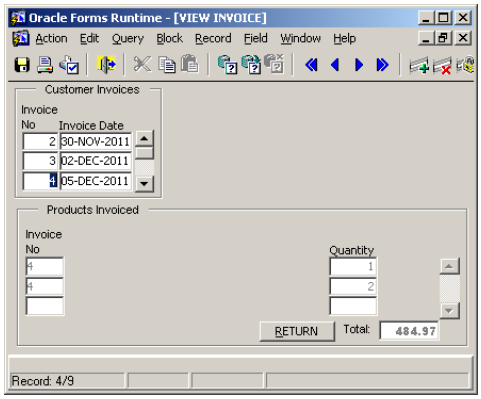

Fig.11. Viewing invoice number 55 sensitive information hidden

ISSN 2006-1781
African Journal of Computing & ICT

www.ajocict.net

Alice now requests Scott to show her how EREP has used her data. Scott navigates to the menu option to view the log of Alice's transactions. The application displays the window confirming the transactions that EREP performed against Alice's privacy agreements. The log shows the code of the transactions that user Scott performed. One of the actions shows that Scott sent an individual communication by SMS to request the payment of an account (see Figure 12).

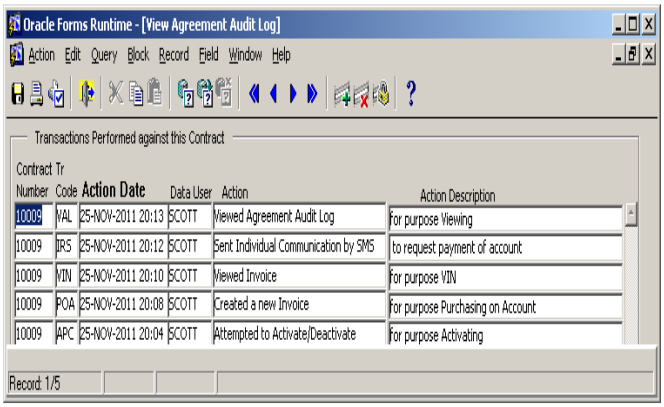

Fig. 12. Transaction log showing EREP's use of Alice's data

Scott previously performed the logged transaction highlighted in Figure 12 by clicking on the push button to send a payment request by SMS. Figure 13 depicts all the individual transactions available on the system. The enabled buttons are those individual transactions that Alice has consented to and that Scott may perform.

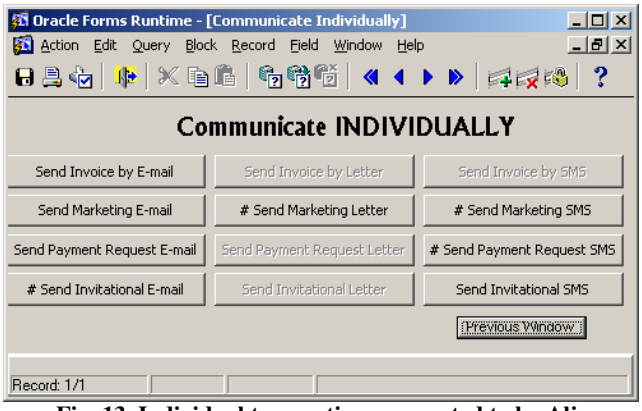

Fig. 13. Individual transactions consented to by Alice

The study implemented group business transactions in a similar way. One example should suffice. When Scott performs the group transaction to send requests for payment in bulk using e-mail, he clicks on the appropriate push button to begin the transaction (see Figure 14 for the transactions available to Scott to perform in bulk).

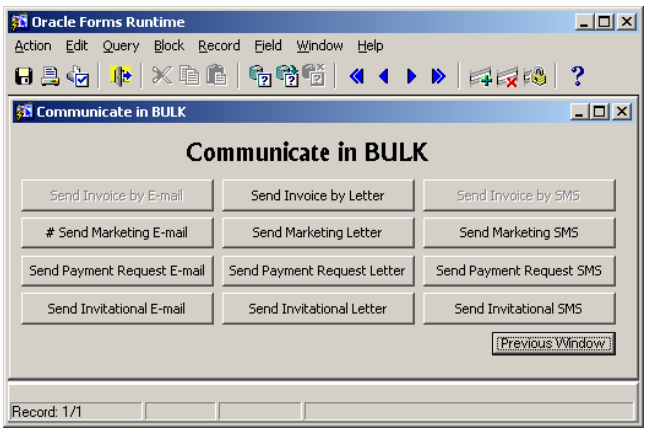

Fig. 14. Using e-mail to send requests for payment as a group transaction

Scott receives feedback from the system as listed in Figure 15. It shows all the e-mail addresses to where he sent the request for payment using e-mail. This section has clearly proved and illustrated through the implementation of the prototype, that the framework has merits. As a rule, the organisation can define their privacy policies through the PPT clearly and non-ambiguously. The developers implement these rules in the database as metadata in the form of transactions, purposes and other related tables. Individuals then enter into privacy contracts with the organisation. The framework then enforces the privacy policies of the organisation and the privacy contracts of the individuals. The organisation cannot perform any transaction against the contract of an individual if the individual has not consented to the use of the transaction for the specific purpose using that individual's specific data. As such, individuals should have improved control over the use of their personal information, be more able to trust the organisation that it will protect their personal information and as a result, the organisation itself will ultimately benefit.

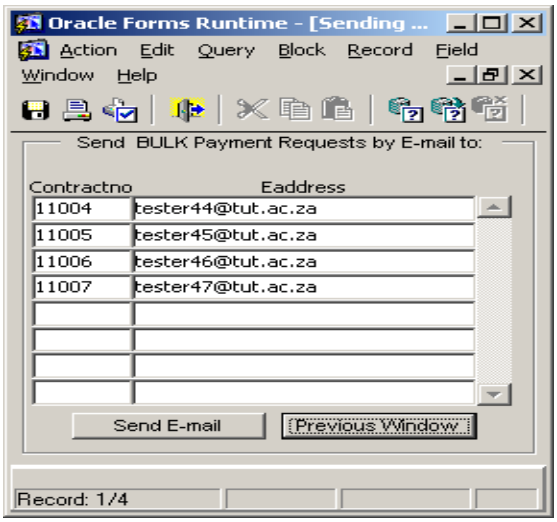

Fig. 15. Confirmation of bulk e-mail sent

The next section briefly discusses related work and the evaluation of the framework.

5 Related work and experimentation

The past 15 years have witnessed the development of numerous Privacy Enhancing Technologies (PETs). Examples include technologies such as Hippocratic databases (HD), PRIME/PrimeLife and EnCoRe. All these technologies acknowledge that the individual must be in control of his/her personal information by specifying privacy preferences. A discussion on these technologies follows next. The rest of the section reports on an experimental evaluation of the HPP framework conducted, benchmarking it against these other PETs.

Inspired by the Hippocratic Oath, [5] have redesigned database systems to include privacy as a fundamental principle. They call such databases Hippocratic databases (HD). Their Hippocratic databases abide by ten fundamental principles. The central concern in Hippocratic databases revolves around purpose. The Hippocratic database system bases its architecture on a straw man design that comprises privacy metadata, a data collection manager, query manager, retention manager and various offline tools. The main advantage of this architecture is that it does not require customisation of existing applications of the company, resulting in easier installation, less customisation and less overhead and maintenance costs [8]. However, although the concept of the Hippocratic database is an innovative idea to take responsibility for the protection of the private information of individuals, it presents certain challenges as raised by [5].

The first challenge concerns their privacy policy language namely the Platform for Privacy Preferences (P3P) standard. It does not have the capability to ensure that websites adhere to their privacy policies. In addition, there are advocates who are not in favour of P3P [9]. The second challenge that arises is whether database systems can afford the additional cost of checking privacy preferences and enforcing these preferences. The third challenge concerns their principle of limited disclosure. [5] address this principle only in terms of external recipients, but seem to ignore the principle of limiting what information the HD discloses to whom. The question that arises is whether the individual can for example specify the attributes to disclose during a specific transaction.

The last challenge concerns their principle of auditing compliance. Logging all accesses to the information of customers might incur a cost in terms of database performance and data storage space. According to [5], the challenge is to provide every customer with logs of all accesses to his/her data, without paying a performance penalty. The organisation does not need to make these auditing logs available to every customer, but only to those customers who request them. The Community's Sixth Framework Program (Contract No. 507591) of the European Union and the Swiss Federal Office for Education and Science funded the PRIME (Privacy and Identity Management for Europe) project [10].

The project aims to help individuals manage all their partial identities through a privacy-enhanced Identity Management (IDM) system. The IDM system enables individuals to control what personal information, based on the role of the data user or organisation, the system reveals to them. A strong design principle of PRIME requires that the design must start from maximum privacy. The individual maintains a complete view of the privacy policies of all his/her transaction partners. The individual can select a privacy policy that states how his/her personal information should be handled. After negotiation with the organisations or transaction partners, the individual can conclude an agreement that forms contractual provisions (similar to a privacy contract) on the privacy rights of the parties involved. [11] make a very strong statement about privacy contracts when they state that "Such agreements serve as legal contracts that must be fulfilled". The European Commission's seventh Framework Programme funded PrimeLife (Privacy and Identity Management in Europe for Life) from 2008 until 2011. PrimeLife is a research project aimed at bringing sustainable privacy and identity management to future networks and services. PrimeLife proposes that websites display privacy policies using easy-to-grasp symbols and people-friendly summaries. One more proposal is that the browser could warn the individual if a website is greedy for his/her data. They are also concerned about personal privacy and working on giving individuals more control over their personal information.

EnCoRe (Ensuring Consent and Revocation) is a research project undertaken by the United Kingdom (UK) industry and academia to give individuals more control over their personal information. Their work aims at giving an individual greater means of controlling what happens to the personal information that the individual discloses to an organisation. Different trust models enable individuals to retain control over their personal information in a number of different ways. However, the personal data management operations of the organisations do not really justify the trust that individuals place in them because of a variety of legal, regulatory, process-related, economical and technical reasons. It defines privacy choices as the actual choices the individual made based on the available options [12]. The technical design of EnCoRe is a block-level design. This means the technical system comprises functional blocks of software and service components and the data flows between them. The technical architecture specifies the flexible expression of privacy preferences (choices), privacy-aware access control, obligation and sticky policies, logging, auditing and compliance checking. In addition, they also formally specify fine-grained privacy preferences driven by graphical interfaces or by individuals setting specific data values. Lastly, supporting their privacy compliance checking capabilities, organisations must be able to log and monitor access, management, processing and disclosure of personal data.

In order to evaluate the effectiveness of various PETs, [13] propose an evaluation framework that analyses solutions along the dimensions of high-level privacy principles and privacy concerns. In addition to the HD principles addressed in this paper, they also address what they call anonymity principles such as anonymity, pseudonymity (not identifiable, but still traceable through an alias),

unobservability, unlinkability and deniability (by denying some characteristics or actions). They further propose what they call other desirable principles for privacy enhancement. These are user preference, negotiation, seclusion (right to be left alone), ease of adoption, ease of compliance, usability and responsiveness. Some of their so-called other desirable principles together with other acknowledged criteria are adapted for use in this study, as defined in Table 1.

Table 1. PET Evaluation Criteria

Criteria	Definition	Ref
Comprehensiveness	Comprehensiveness of the privacy solution that sets clear rules for access to, use of and disclosure of personal information and includes accountability.	[14]
Usability	Efforts required by individuals to use the PET should be reasonable.	[13]
User preference	Every individual can tailor his/her privacy preferences to personal choices.	[13]
Enforcement	Data protection rights need to be actually enforced to be effective.	[15]
Responsiveness / flexibility	The ability of the solution to promptly respond according to changes in the preferences of the individual.	[13]
Ease of compliance	Ease of PET to fulfil legal requirements.	[13]
Ease of adoption	The solution does not rely on other proprietary technology.	[13]
Auditing capability	The PET includes auditing functionalities.	[5]
Expressive GUI	Completeness of GUI – simplicity, effectiveness.	[16]

Table 2 lists whether the evaluated PETs answer to each of the criteria or not. For every criterion, the individual PETs have been ranked from 4 (highest) to 1 (lowest) or 0 (if no literature is available regarding the criterion for the specific PET). At the end of the process, all the points for each PET were summarised. Eventually, the four PETs were ranked according to their totals in order to determine which PET is the most effective. A brief discussion on each of the PETs with regard to the specific criterion follows Table 2 so as to motivate the awarding of the points.

Table 2. PETs evaluation results

Criteria	Hippocratic database	PRIME / PrimeLife	EnCoRe	HPP framework
Comprehensiveness	3	1	2	4
Usability	2	3	1	4
User preference	2	1	3	4
Enforcement	3	1	2	4
Responsiveness	3	3	3	4
Compliance	4	3	0	3
Ease of adoption	4	0	0	3
Auditing capability	4	1	2	3
Expressive GUI	1	4	3	2
Total points	26	17	19	31
Rank	2	4	3	1

6 Future Work

This paper offers an opportunity for further research in a number of respects. Future research might include the implementation of the HPP framework as a full-scale development to obtain user input regarding the effectiveness of the framework and to determine to what extent individuals measure the effectiveness and accept the developed application. A more thorough practical evaluation of the HPP framework using more rigorous evaluation criteria, with regard to its practical applicability, usefulness, effectiveness and performance is required.

In addition, the evaluation of the HPP framework as a PET indicated that the framework needs improvement regarding the ease in which existing systems can adopt the principles of the HPP framework, issues regarding auditing features, compliance and lastly, the GUI of the HPP framework.

7 Conclusion

Every individual has the right to privacy and hence, the need for the protection of his/her personal information. At the same time, every organisation has the right to insist on protecting its business interests through access to needed information. The apparent conflict of interests in this requires striking an appropriate balance in PET provisioning. Based on the results of this study and the fact that individuals hold different views on privacy, it has been demonstrated that such rights can be guaranteed by providing a PET that enables defining at what level of privacy they require organisations to process their personal information.

The main contribution of the study is the development of an innovative, transaction-based, process-driven privacy enhancing technology (PET) that contributes to system optimality and usability. In addition, the formal specification of the framework using conventional modelling languages like UML and OCL provides a basis demonstrating feasibility of the practical realisation and application of the framework. The developed prototype shows the theoretical and empirical validity of the framework thus contributing to both the theory and practice of PETs. Giving increased control over their personal information to individuals enable them to be more comfortable about disclosing their information to corporate organisations in the knowledge that they will manage the personal information according to fair information privacy practices.

REFERENCES

1. OBERHOLZER, H.J.G & OLIVIER, M.S. Privacy contracts as an extension of privacy policies. In: **Proceedings of the International Workshop on Privacy Data Management (PDM 2005)**, 11-19, Tokyo, Japan, April 2005.

2. CATE, F.H. 2007. The failure of fair information practice principles [Online]. Available from: www.hunton.com/files/tbl_s47Details/FileUpload265/1248/Failure_of_Fair_Information_Practice_Principles.pdf [Accessed: 12/06/2004].

3. GENG, J., LIU, L. & BRYANT, B.R. 2010. Towards a personalized privacy management framework. In: *Proceedings of the 2010 ICSE Workshop on Software Engineering for Secure Systems,* May 2010. ACM:58-64.

4. OBERHOLZER, H.J.G & OLIVIER, M.S. 2006. Privacy contracts incorporated in a privacy protection framework. In: **Computer Systems Science and Engineering**, 21, 1, 5-16, 2006.

5. AGRAWAL, R., KIERNAN, J., SRIKANT, R. & XU, Y. 2002. Hippocratic databases. In: *Proceedings of the 28th VLDB Conference,* 20-23 August 2002, Hong Kong, China.

6. WARMER, J. & KLEPPE, A. 2003. *The object constraint language*. 2nd ed. Addison-Wesley.

7. EFRAIMIDIS, P.S., DROSATOS, G., NALBADIS, F. &TASIDOU, A. 2008. Towards privacy in personal data management [Online]. *Technical Report*, Democritus University of Thrace, Greece, June 2008. Available from: http://polis.ee.duth.gr [Accessed: 13/05/2010].

8. IBM Almaden Research Center. 2004. Intelligent information systems: Hippocratic database [Online]. *IBMAlmaden Research Center*. Available from: http://www.almaden.ibm.com/software/quest/Projects/hippodb/activeenf/ [Accessed: 19/01/2004].

9. BOWERS, F. 2004. Dept admits privacy breach [Online]. *IrishHealth*, Jun. 6. Available from: http://www.irishhealth.com/?level=4&id=5992 [Accessed: 27/10/2004].

10. PETTERSSON, J.S., FISCHER-HüBNER, S., DANIELSSON, N., NILSSON, J., BERGMANN, M., CLAUSS, S., KRIEGELSTEIN, T. & KRASEMANN, H. 2005. Making PRIME usable. In: *Proceedings of the International Symposium on Usable Privacy and Security (SOUPS),* July 2005. Pittsburgh, Pennsylvania: 53-64.

11. CAMENISCH, J., SHELAT, A., SOMMER, D., FISCHER-HÜBNER, S., HANSEN, M., KRASEMANN, H., LACOSTE, J., LEENES, R. & TSENG, J. 2005. Privacy and identity management for everyone. *ACM DIM'05,* 11 November 2005, Fairfax, Virginia, USA.

12. MONT, M.C., SHARMA, V., PEARSON, S., SAEED, R & FILZ, M. 2011. Technical architecture arising from the third case study. *EnCoRe Project Deliverable,*18 November 2011.

13. WANG, Y. & KOBSA, A. 2009. *Handbook on Research on social and organizational liabilities in information security.* 1st ed. Hershey.

14. McGRAW, D., DEMPSEY, J.X., Harris, L. & GOLDMAN. 2009. Privacy as an enabler, not an impediment: building trust into health information exchange [Online]. *HealthAffairs*. Mar/Apr. 09. Available from: http://content.healthaffairs.org/content/28/2/416.short [Accessed: 07/06/2012].

15. REDING, V. 2011. *Your data, your rights: Safeguarding your privacy in a connected world* [Online]. Available from: http://europa.eu/rapid/pressReleasesAction.do?reference=SPEECH/11/183 [Accessed: 04/06/2012].

16. AUTILI, M. & PELLICCIONI, P. 2008. Towards a graphical tool for refining user to system requirements. In: *Proceedings of the Fifth International Workshop on Graph Transformation and Visual Modeling Techniques (GT-VMT 2006)*, 28 April 2008, 211 (C): 147-157.