Eliciting Confidentiality Requirements in Practice

Seda Gürses[a], Jens H. Jahnke[b], Christina Obry[b], Adeniyi Onabajo[b], Thomas Santen[c], Morgan Price[d]

[a] Department of Information Systems, Humboldt University, Berlin, Germany
[b] Department of Computer Science, University of Victoria, Victoria BC,Canada
[c] Softwaretechnik, FR 5-6, TU Berlin, Berlin, Germany
[d] Department of Family Practice, University of British Columbia, Vancouver BC, Canada

Abstract

Confidentiality, the protection of unauthorized disclosure of information, plays an important role in information security of software systems. Security researchers have developed numerous approaches on how to implement confidentiality, typically based on cryptographic algorithms and tight access control. However, less work has been done on defining systematic methods on how to elicit and define confidentiality requirements in the first place. Moreover, most of these approaches are illustrated with simulated examples that do not capture the richness of real world experience. This paper reports on our experiences eliciting confidentiality requirements in a real world project in the health care area. The method applied originates from the M.Sc. thesis of one of the authors and is still considered work in progress. Still, valuable insight into issues of confidentiality requirements engineering can be gained from this case study and we expect that its publication will become a basis for discussion and the definition of a further research agenda in this area.

1 Introduction

The vast majority of today's software applications are developed for distributed and network-centric platforms. Hence, security requirements have gained great importance in the development and operation of network based software/software running on networks. Security requirements are part of what is typically considered non-functional requirements. During systems development, they are often documented in informal documents using natural language. Still, the media reports about increasing numbers of security violations almost every day. Many of these violations can be traced back to errors made during requirements analysis, particularly with respect to capturing and integrating security requirements with functional features of systems and their subsequent implementation. There is lack of systematic methods for specifying security require-

ments and their consistent integration with functional system specifications.

This paper specifically focuses on *confidentiality* requirements. Confidentiality is an aspect of security that we consider most difficult when it comes to requirements elicitation. The method presented in this paper is based on one of the authors' M.Sc. thesis [7], but is still considered work in progress. This paper is an experience report on applying this method in a real-world project in the domain of health information management, called TAPAS (Technology Assisted Practice Application Suite). TAPAS is an open-source initiative currently funded from a western Canadian province's Ministry of Health in collaboration with two of six health authorities and two local universities to provide primary care physicians with information technologies to improve quality of care and assist them in their practice. At the time of writing this paper, its first version is about to be rolled out to a call group of physicians in Vancouver, called the North Shore Mobile Health Network.

The contribution of this paper is not a fully developed method that provides all answers on eliciting and engineering confidentiality requirements. Rather, the authors wrote this paper because there is a vacuum of real-world case studies and experience reports on how confidentiality requirements are dealt with in practice. We strongly believe that such case studies are needed and will play a pivotal role for identifying research agendas of practical relevance and to further the development of an effective methodology in this important area.

The rest of this paper is structured as follows: The next section will sketch an overview of the main steps involved in the applied method, which we call *CREE* for *Confidentiality Requirements Elicitation and Engineering*. Section 3 will flesh out details on these steps and report on their application to the TAPAS case study. Section 4 relates our approach to other work. Finally, we summarize our experiences in order to crystallize a research agenda in Section 5.

2 The *CREE* method - an overview

The purpose of this section is to give an overview on the *CREE* method and approach taken for engineering confidentiality requirements in TAPAS. *CREE* is based on ideas developed in [7]. TAPAS has been the first industrial-scale, real-world project to try out these ideas and, naturally, we have gained many valuable insights that have been used to evolve and refine the method. The method introduced here has not only been applied to the project, but there is also feedback from the experiences made during the project to the method. Section 5 discusses these points in more detail.

2.1 Terminology Remarks

In requirements engineering, the term *stakeholder* refers to individuals or groups of individuals who have requirements on the system to be built. It is the task of the requirements engineering process to make the stakeholders' requirements explicit, resolve conflicts between contradictory requirements, and finally produce a complete and consistent set of system requirements.

As an IT security goal, confidentiality is the protection of unauthorized disclosure of information. *Confidentiality stakeholders* are the individuals who have confidentiality requirements. They have an interest in keeping certain information (often information about themselves) confidential. In the following, the term 'stakeholder' refers to the general concept, whereas a 'confidentiality stakeholder' has specific requirements relating to confidentiality.

We also distinguish stakeholders from *stakeholder roles*. The latter term refers to a predicate identifying groups of individuals with similar requirements, whereas the former refers to single persons (instances). This distinction is analogous to the one between an *actor class* and an *actor instance* in UML use case analysis. Although often blurred in its actual use, the language clearly distinguishes between the class *actor*, i.e., the group of individuals who may act in that role, and the instances of that class who communicate with the system in the

role of a particular actor. Individuals can use the functionality of the system in the role of different actors, i.e., they can be instances of different actor classes. Similarly, individuals can be members of different stakeholder roles, e.g., a medical doctor may not only have confidentiality requirements as a member of the confidentiality stakeholder role *physician* but also as a *patient*, which is another confidentiality stakeholder role of TAPAS.

Finally, we would like to point out that we purposefully avoid the otherwise popular term 'adversary' in our method. Rather, we speak of *counter-stakeholders* to denote individuals with interests in conflict to other stakeholders. There are two reasons why we make this distinction. Firstly, we believe that 'adversary' is too narrow a term, since it has a connotation of malicious intent, but many security violations are caused by individuals that do not fit in this category. Secondly, we believe that an absolute distinction between stakeholders and adversaries is neither appropriate nor realistic. Rather, confidentiality stakeholders may have conflicting interests and the question who can be considered an 'adversary' is very much a relative concept. Therefore, we use the term *counter-stakeholder* when we refer to individuals that may have interests that may be in conflict with the interests of a particular confidentiality stakeholder.

2.2 Method characteristics and process

The *CREE* method presented here has a number of characteristics to it.

- *CREE* is designed to find requirements of *multilaterally secure* systems [14]. This means it considers each stakeholder as a party with its own confidentiality goals, which first need to be understood and made explicit before refining them into a consistent set of confidentiality requirements for the system.

- It distinguishes between two ways of expressing a stakeholder's confidentiality requirements – in the form of negative constraints called confidentiality *goals*, or as positive concessions, called confidentiality *consents*. The method details the relevant attributes to describe a goal or a consent.

- It explicitly relates confidentiality requirements with functional requirements. Confidentiality requirements may be used to motivate variations of functional requirements. The other way around, functional requirements can imply confidentiality requirements. Sometimes the implied confidentiality requirements may be ambiguous, and this ambiguity may be resolved by introducing variations of the functional requirements.

- It calls for sorting out the relationship between (functional) actors, who actually use the system, and stakeholders, who may or may not use the system. Actors may represent numerous stakeholders, or a given stakeholder may play the role of many actors. Yet, not all stakeholders are actors.

- It puts system functionality into the social context of *episodes*, which are made up of clusters of system functionality. Variations of these episodes describe similar functionalities which only differ in the way they accommodate confidentiality goals of different stakeholders.

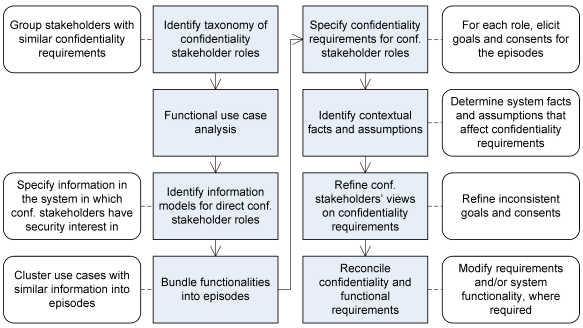

Figure 1: Method overview

Figure 1 illustrates the method, which consists of the following steps. Although the numbering of these steps might suggest otherwise, it is understood that there are strong dependencies between the results produced in each step, and that iterating the steps and adapting

their results will usually be necessary to produce a complete and consistent confidentiality requirements analysis.

1. *Identify taxonomy of confidentiality stakeholder roles.* Confidentiality stakeholder roles have confidentiality goals or consents related with the particular application at hand. We distinguish *direct* confidentiality stakeholder roles, i.e., stakeholder roles that may have some of their information processed by the projected system, from *indirect* confidentiality stakeholder roles (all others). Members of indirect stakeholder roles can be governments and organisations. Indirect stakeholder roles can also comprise individuals with malicious intent, such as hackers, who do not have a legitimate interest in the system. The stakeholder roles identified in the general requirements engineering process are natural candidates for confidentiality stakeholder roles. Other candidates are groups of individuals to which the early requirements documents refer, such as a user needs document. Those documents may not identify them as stakeholders because they only have a marginal interest in the system functionality as such. They may, however, have dedicated confidentiality requirements because the system processes data related to them. Furthermore, the grouping of individuals in confidentiality stakeholder roles may differ from the one for functional requirements because people with similar functional interests in the system may nevertheless have different confidentiality requirements, and vice versa.

2. *Provide a functional use case analysis.* Traditional use case analysis [8] with specified actor hierarchy.

3. *Identify information models for direct stakeholder roles.* The *stakeholder information models* are a basis for defining confidentiality goals and requirements. They model two kinds of information: first, information relating to the confidentiality stakeholders that is *exchanged with the system*; second, information about the stakeholders that may be *derived from the behavior* of the system.

4. *Identify episodes based on functional use cases.* Use cases may involve confidential information, either as data or as information derived from the particular interaction with the system. This step clusters use cases that are similar with respect to the information involved into *episodes*. The information requirement of each episode is defined based on the stakeholder information models. Defining an episode takes the stakeholders' interest in their information model into account. Thus, it relates functionality (use cases) with the stakeholders' social context within which the IT system will work. Setting up episodes relates the functional actor hierarchy with the confidentiality stakeholder role taxonomy: Considering the system functionality with the aim of finding confidentiality requirements must consider which stakeholders may actually use the system, and as instances of what actor classes.

5. *Specify stakeholder roles' confidentiality goals and consents.* We allow stakeholders to express their confidentiality requirements in form of negative constraints (called confidentiality *goals*) or as positive concessions (called confidentiality *consents*). Considering the interest of stakeholders in particular episodes and the other stakeholders involved in them aids in finding confidentiality goals and consents.

6. *Identify contextual facts and assumptions.* It is common knowledge in requirements engineering that requirements must be related to facts and assumptions about the environment of the system in order to make the requirements implementable [17]. This observation is particularly relevant for confidentiality requirements, because those address the flow of information– possibly mediated by the system – from stakeholders to other stakeholders or outsiders. It is obvious that the system to be built can only keep information confidential if it controls the flow of that information. Only facts ("There is

no relevant communication between those two stakeholders.") or assumptions ("This stakeholder does not disclose certain information.") make it possible to refine general confidentiality requirements to a consistent set of system confidentiality requirements that directly restrict the behavior of the system.

7. *Refine stakeholders' views to system confidentiality requirements.* The goals and consents of different confidentiality stakeholders will usually be mutually inconsistent, because each stakeholder has his or her own interest in the system and conflicts of interests naturally arise. Refining the goals and consents into a coherent whole may be solved in multiple ways. Either a compromise is found between the different interests of the stakeholders, in which case it is important to explicitly contrast that compromise with the original goals and consents. Or, mechanisms which allow the actors to negotiate their confidentiality requirements with their communication partners during run-time are introduced [16]. In both cases, it is important to document the rationale for the compromise or for the introduction of the confidentiality negotiation mechanisms.

8. *Reconcile confidentiality requirements with functional requirements.* System functionality may compromise confidentiality. Therefore, it is necessary to reconcile conflicting functional and confidentiality requirements. This may result in weakening confidentiality requirements, or in modifying envisaged system functionality. Negotiation mechanisms may also effect the functionality of the system and need to be reconciled with other functional requirements.

2.3 Denotational properties of confidentiality requirements

Confidentiality goals and consents are central to articulating the confidentiality requirements of the system. In the following we list attributes that describe confidentiality goals and consents in detail.

1. The *owner*. The owner is the subject of the goal or consent. The owner is a stakeholder role. Thus, if the owner of a confidentiality goal or consent is a 'patient', this comprises all patients (who have security concerns about the system).

2. The degree of *agreement* on the goal or consent between the individuals of a stakeholder role. The goal or consent is considered *unanimous* if all individuals of the stakeholder role share it, and it is *partial* if there are individuals who do not share that goal or consent but who also do not object to it.

3. The *kind* of requirement. Is the confidentiality requirement a *goal* or a *consent*?

4. The *counter-stakeholder*. The counter-stakeholder role is the role against whom the owner directs his/her concern.

5. The *strictness* of a goal. Strictness describes the required confidentiality for the information that a goal or a consent does not address explicitly. A *strict* goal means that the counter-stakeholder may obtain the information the goal does not explicitly require to be kept confidential from the counter-stakeholder. A strict consent requires any information to be kept confidential that the consent does not allow the counter-stakeholder to obtain. *Non-strict* goals or consents do not make a statement about the information they do not explicitly address.

6. The *information* to which the goal or consent refers. This is part of the owner's information model.

7. The owner's *rationale*. The rationale expresses the reason why the mentioned information must be kept confidential from the counter-stakeholder (for a goal), or may be made available to the counter-stakeholder (for a consent).

8. The *temporal range*. This states for how long the concern must (at least or at most) be considered. This may be expressed in absolute terms, or relative to certain events, e.g., the creation of particular data

entries in the system relative to an event or after a fixed amount of time.

9. The *context*. The context relates the goal or consent to the episode from which it emerged.

We assume non-strictness for confidentiality goals to avoid unnecessary disclosure of information. For consents, a strict interpretation is adequate if they are used as a means to describe a confidentiality goal that is hard to express directly. It may be easier to describe what information a counter-stakeholder may obtain and then require that 'nothing else' may be disclosed.

The degree of agreement is a way to avoid an overly fine-grained taxonomy of stakeholder roles. Individuals can share the same stakeholder role if they do not object to any confidentiality goal or consent associated to that stakeholder role. If, however, there is a group of individuals who contradict a particular goal, then it may be necessary to subdivide a stakeholder role into several new ones. Note that an individual always interprets a goal or consent from his/her perspective. Therefore, individuals who contradict a goal do not assume it for themselves. They do not necessarily object to other individuals sharing the goal in question.

Finally, a goal or consent can mention the same stakeholder role as its owner and as its counter-stakeholder. This is the case if the individuals of the stakeholder role have confidentiality goals against other individuals sharing their role, e.g., a physician may well wish to keep certain information confidential from other physicians.

3 *CREE* applied to the TAPAS project

The Oxford dictionary defines primary care as *health care at a basic rather than specialized level for people making an initial approach to a doctor or nurse for treatment.* In other words, TAPAS is not intended for specific acute care hospital situations but to support general practitioners and their support staff (nurses, medical office assistants, clinic managers, etc.) in their daily work.

The detailed characteristics of TAPAS are explained in the following subsections. There are several Electronic Medical Record (EMR) software systems available for primary care. The authors would like to point out two important differences between those EMR systems and TAPAS: Firstly, TAPAS is not meant to be a full implementation of an EMR system. EMR systems are very complex and the migration from a paper-based practice to an electronic practice requires significant investments in terms of change management, training and time. TAPAS's mandate is to become a stepping stone on the way from the paper-based office to computer-based medical practice. As such, it supports and augments a predominantly paper-based practice with several key functions provided by a typical EMR system. The second notable difference is that TAPAS is meant to be more ubiquitous than classical EMR systems in that it supports mobile clients as well as stationary clients. Mobility is particularly important because primary care physicians practice in many different places, including their clinic, the hospital, at patient homes, at home, and while on call.

3.1 Confidentiality stakeholders

As mentioned earlier, we distinguish between direct and indirect stakeholders. In the case of TAPAS, it is not difficult to conceive that patients belong to the set of direct stakeholders. However, it may be less obvious that clinicians such as physicians, nurses and medical office assistants also belong to the direct stakeholder category. It is important to follow a systematic and traceable process in order not to miss any relevant confidentiality stakeholder role. The following section describes how confidentiality stakeholders where identified in TAPAS.

In TAPAS, we have adopted (and adapted) the ReadySet software development documentation templates and method, which is available as an open source project at `http://readyset.tigris.org`. According to ReadySet the so-called *User Needs* document *explains the actual desires of stakeholders in roughly their own words.* It is created based on client interviews in preparation for the software requirements specification. The full TAPAS

User Needs document can be accessed online at `www.opentapas.org`. We used this document to derive confidentiality stakeholder roles. While analyzing the user needs document, it was important to understand the difference between confidentiality stakeholder roles and (functional) user roles. While the user needs document *mentioned* most confidentiality stakeholder roles, some of them did not appear as prominent as others. For example, studying the TAPAS user needs document, one notices that user roles like physician, nurse and medical office assistant (MOA) are discussed in prominent spots within separate paragraphs in a section entitled *Stakeholders / Actors*, while other important confidentiality stakeholder roles such as *patient* are just mentioned implicitly in the discussion of the user needs, e.g, *Physicians need to easily find information about patients...* . Due to our experience, this is a typical situation and the confidentiality requirements engineer has to carefully study the user needs document in order to identify all direct confidentiality stakeholders.

In addition to the direct confidentiality stakeholders identified based on the user needs document, later functional analysis may give rise to further direct stakeholder roles. This may be the case if the system realization requires additional roles, which are required solely for the purpose of operation and maintenance of its functions. An example would be a database administrator and an information auditor. Such roles may have to be added at a later stage of the process when functional requirements analysis has been performed.

Indirect confidentiality stakeholders are roles that do not disclose information about themselves to the system, yet still have confidentiality goals. They range from legal entities such as corporate organizations and health authorities, governments on various levels to roles that have traditionally been coined *adversaries* (hackers, information warriors, malicious spies etc.). In case of TAPAS, an important indirect stakeholder has been the BC government and the BC privacy commissioner whose role is to enforce the Personal Information Protection Act (PIPA) privacy act [13]. This act not only demands that personal information about patients must be kept secure and confidential, it also demands for an audit trail that identifies who has had access to this information and for what purpose. This is a particularly interesting example because adding the BC government as an indirect stakeholder indirectly led to adding clinicians to the direct stakeholder category. Why is this so? The reason for this iteration is that the requirement for a complete audit trail implies that personally identifiable information be kept about the usage profiles of individuals with access to patient information, i.e., clinicians. Hence, this is an example where the addition of an indirect stakeholder has resulted in the addition of a direct stakeholder. Of course, this may only become clear after the first iteration of applying the *CREE* method, after formally identifying the stakeholder goals.

Confidentiality roles should be arranged in a taxonomy in order to facilitate the definition of confidentiality requirements. The stakeholder taxonomy for TAPAS is summarized in Figure 2.

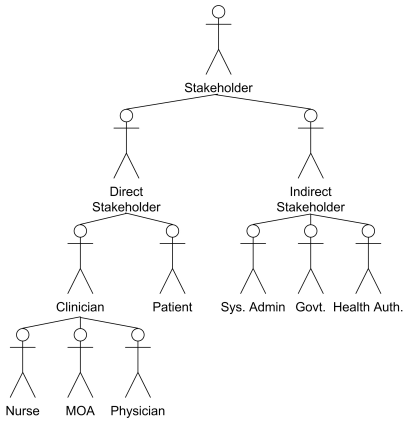

Figure 2: Confidentiality stakeholder role taxonomy

3.2 Functional Requirements Analysis

In this step, we analyzed the functional requirements for the TAPAS system. The regular process of eliciting functional system features based on a use cases analysis and a specification of an actor hierarchy was used here. Again, we used an adapted form of

the ReadySet templates for documenting the functional requirements. A detailed description of the TAPAS system function is beyond the scope and not needed for the contribution of this paper. The full software requirements specification for TAPAS can be accessed online at `www.opentapas.org/docs/srs.html`. Here is a concise summary of the functionality, in order to give the reader an impression of the features supported by the system: The system provides clinicians (nurses, physicians, MOAs) with functions for:

- Managing medical information for patients electronically, including a health history, allergies, and prescriptions;
- exchanging messages with other clinicians;
- managing a calendar showing the call schedule for on-call clinicians;
- and accessing all these functions from desktop computers as well as wireless, mobile PDAs.

From a confidentiality engineering point of view, it is important to point out that the actor hierarchy elicited in the functional analysis is different from the stakeholder hierarchy specified in Step 1 of *CREE*. While some of the functional actors also appear as confidentiality stakeholders (e.g., physician, nurse, MOA), others do not (e.g., the functional actor hierarchy specifies a system administrator). Furthermore, some important confidentiality stakeholders do not appear in the functional actor model, because they do not directly interact with the system, e.g., the role *patient*. Therefore, it is imperative to analyze the confidentiality roles separately from the functional roles.

As indicated earlier, each functional role that does not appear in the confidentiality stakeholder taxonomy should be investigated with respect to how they impact confidentiality roles and their requirements. We do this in the reconciliation Step 8 of *CREE*.

3.3 Stakeholder Information Model

A direct stakeholder is defined as a stakeholder whose information is exchanged in the system. Even if they are not directly interacting with the system or their information is not necessarily processed by the system, the fact that their information is collected and that it may be used in medical practise, makes them an important factor in specifying confidentiality requirements. Therefore, it is necessary to identify information models for direct stakeholders.

An information model describes all the data processed by the system in which a direct stakeholder may have security interests. It includes the data that is put into the system, that is created about the stakeholder or the data that a stakeholder uses. The system has to ensure that this data is not misused and that it is disclosed to only authorized people.

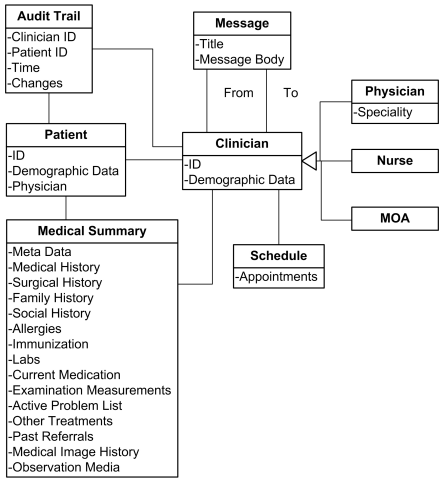

Figure 3: TAPAS System Information Model

In the TAPAS system there are two major direct stakeholder roles: patient and clinician. Figure 3 gives a brief overview of the TAPAS system information model including both stakeholders. Furthermore, this information model can be broken down to two separate information models, one for each stakeholder role. Figure 4 shows the patient's information model on the left side and the clinician's on the right side.

The patient information that is processed can be divided into the following categories:

- Identifying data such as health card number, name and other demographic data, which are captured in the Patient table in

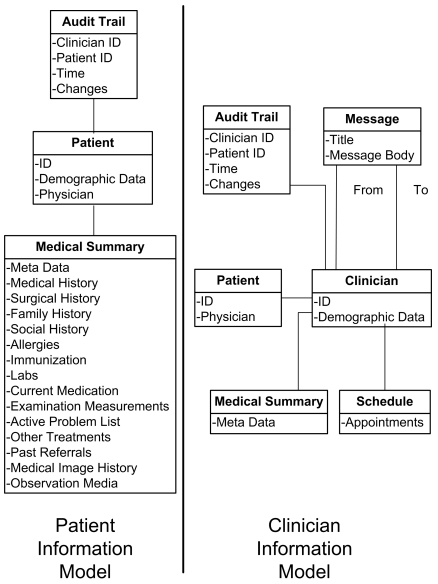

Figure 4: TAPAS Information Models

Figure 4.

- Medical summary, which includes information about the medical history, allergies, current medication, past referrals etc. The TAPAS project uses the e-MS (electronic medical summary) standard [1] to represent medical information. The e-MS standard, which is based on the HL7 Clinical Document Architecture (CDA) [6], is a subset of patient data suitable for communication amongst clinicians for the purpose of sharing the care of an individual patient.

- Audit trail data used to prevent and trace the abuse of personal data. The audit trails capture who changed which data at what time, a detailed list of the data items is shown in Figure 4.

Furthermore, the captured information that involves the clinicians can be divided into:

- Identifying data such as clinician ID, name and other demographic data, captured in the table Clinician in Figure 4.

- Messages that are used to communicate with other clinicians.

- Appointments that are captured in the schedule.

- Clinician's identifiers from the medical summary meta data.

- Audit trail data that includes the ID of the clinician, who changed the patient data.

As shown in Figure 4, Audit Trail and Medical Summary are included in both information models, this could be a potential cause of conflicts.

3.4 Episodes

Some of the use cases identified involve processing data that the direct stakeholders have confidentiality interests in. Identifying these use cases comes as a follow up to defining stakeholders' information models and the use case analysis described above. The initial use case analysis provides a high-level view of system functionality used in traditional software development processes. This view of what the system does with respect to the actors does not present the confidentiality stakeholders' perspective, which is to define confidentiality goals or consents for their data processed while realizing these functions.

Analyzing the information requirements for the use cases might yield sets with common information requirements, referred to as *episodes*. Identifying the episodes is a step in realizing the *CREE*'s characteristic of associating functional requirements to confidentiality requirements. We describe some of the episodes from TAPAS using the information model defined in Section 3.3:

Manage patient record excluding medical data (EI): This episode consists of use cases for managing patients' demographic data. The use cases involve creation, viewing, modification and archiving of demographic data by the actors, which are members of the clinician stakeholder role, for administrative and clinical purposes.

In addition, government legislations for protection of personal information handled by organizations, such as the afore-

mentioned PIPA are relevant to this episode. Elements for the episode are:

- confidentiality stakeholder: patient, clinician, government
- information requirement: patient identifying data

Manage medical data (EII): This episode consists of use cases for managing various sections of the clinical record. Implicitly, the patient confidentiality stakeholder's clinical information is a basic information requirement for this episode. Confidentiality stakeholders who provide care have to be indicated in the clinical record for traceability and audit purposes, hence the physician and nurse information models and confidentiality requirements need to be considered in analyzing the episode.

Elements for this episode are:

- confidentiality stakeholder: patient, clinician
- information requirement: patient medical data

Produce patient report (EIII): This episode includes uses cases for generating reports on patient care and contain both identifying and medical data. Similar to EI and EII, the patient confidentiality stakeholder's information model and confidentiality interests are important.

Elements for the episode are:

- confidentiality stakeholder: patient, clinician
- information requirement: patient medical data, patient identifying data.

Unlike episodes EI and EII above, access to both clinical and non-clinical data is required for these functions and is a potential for conflicts in the confidentiality requirements.

Specification of the confidentiality goals and consents of the concerned stakeholders, in particular the patient stakeholder, can result in modifications to the functionalities or negotiation of the confidentiality requirements.

3.5 Stakeholders Confidentiality Goals and Consents

Stakeholder confidentiality interests are specified as consents and goals. We define these in terms of the attributes listed above. The consents and goals are considered with respect to the episodes in order to reason about other stakeholders involved in the episodes. In addition, variations of the episodes, if required are based the requirements of the stakeholders involved.

We specified the stakeholders confidentiality interests using the properties described above (Section 2.3). For the episodes in Section 3.4 the requirements of the concerned stakeholders are shown in Tables 1 through 3. The requirements in the tables do not show the temporal range values, since these requirements are valid as long as the data exist in the system.

In Table 1, the patient confidentiality stakeholder role gives consent to the clinician stakeholder for its identifying information for administrative and care purposes (EI.1). This requirement applies to all members of the patient role (as indicated by the *unanimous* value for *degree of agreement*). The two scenarios for this consent are creation of record for a new patient or for updates to existing records. In addition, a partial goal (since all members of the role might not agree with this) is required for information shared for research purpose (EI.2). While medical data can be shared for research, identifying information must not be made available to government and health authorities.

Note that a confidentiality goal like this expresses a requirement, not an access control rule: This goal denies government (direct) access to patient identifying data, and, furthermore, it requires obscuring any information from government that would allow them to deanonymize population data. Providing adequate access control mechanisms is part of the design task. Further measures may be necessary to actually satisfy that confidentiality goal.

Identifier	*Owner*	*Degree of agreement*	*Kind*	*Counter-stakeholder*	*Strictness*	*Information*	*Context*	*Rationale*
EI.1	patient	unanimous	consent	clinician	strict	identifying data	new patient/update	administrative/ care
EI.2	patient	partial	goal	government, health authority	non-strict	identifying data	obtain population data	anonymity
EI.3	clinician	unanimous	consent	system admin., government	strict	audit trail data	audit data for data access	traceability/ investigation
EI.4	government	unanimous	goal	indirect stakeholder	non-strict	identifying data	research	not care related

Table 1: Requirements for Episode "Manage patient record excluding medical data"

Identifier	*Owner*	*Degree of agreement*	*Kind*	*Counter-stakeholder*	*Strictness*	*Information*	*Context*	*Rationale*
EII.1	patient	unanimous	consent	physician, nurse	non-strict	medical data	new patient /update	care
EII.2	patient	partial	consent	government, health authority	non-strict	medical data	obtain population data	partake in surveillance
EII.3	patient	partial	goal	MOA	non-strict	medical data	any	not care provider
EII.4	patient	unanimous	goal	system admin.	non-strict	medical data	any	not care provider
EII.5	clinician	unanimous	consent	system admin.	strict	audit trail data	audit data for data access	traceability

Table 2: Requirements for Episode "Manage medical data"

Identifier	*Owner*	*Degree of agreement*	*Kind*	*Counter-stakeholder*	*Strictness*	*Information*	*Context*	*Rationale*
EIII.1	patient	unanimous	consent	physician, nurse	strict	medical and identifying data	report for chart	administrative/care
EIII.2	clinician	unanimous	consent	system admin.	strict	audit trail data	audit data for data access	traceability

Table 3: Requirements for Episode "Produce patient report"

To facilitate the audit functionality of the system, the clinician confidentiality stakeholder role gives consent to its audit trail data (EI.3), which can be accessed during investigations into possible malpractice or for maintenance by the system administrator.

The requirement of the patient stakeholder to restrict access to medical data by the MOA (Table 2, EII.3) results in a conflict with TAPAS functionality requirement for the MOA stakeholder to generate reports, which contain both identifying and medical data. This highlights the need to explicitly relate the confidentiality and functional requirements. This conflict is addressed in Section 3.7.

3.6 Contextual Facts and Assumptions

Any software system is embedded in social processes and technical frameworks that are part of the world surrounding the system. System developers have to make assumptions about this embedding context when they design their system. Good requirements engineering practice dictates that these assumptions should be specified explicitly. This is particularly important when it comes to dealing with confidentiality as a system property. The following assumptions are an excerpt of the assumptions made in TAPAS.

- The calendar will only be used to manage the on-call schedule of clinicians. No patient data will be entered here.
- Clinicians will put no data about medical results, diagnosis or conditions into the subject line of messages. The subject line of messages will contain general information about the nature of the message (for example *referral*) and, potentially, a reference to a patient's identity.
- Physicians may leave their mobile PDAs unattended or may lose them.
- Physicians may want to access the TAPAS services from various, previously unknown computers. Other individuals may have administrator access to some of these computers.
- Printed medical records and messages will be kept in a safe place and disposed of safely (shredding) when they are no longer needed.
- The system administrator does not use his/her credentials to retrieve any information from the database, unless it is needed for his/her duty. Moreover, none of this information is communicated with other stakeholders.

3.7 Refinement of Stakeholder Confidentiality Requirements

There is a need to consider the confidentiality stakeholder roles' requirements from the various episodes. The listed goals and consents from the episodes constitute possible requirements which might be further refined in view of other episodes. For example, there exist a contradiction between the requirements of the patient stakeholder role to strictly allow members of the clinician stakeholder role to access its identifying data in EI.1 (Table 1) and the requirement to grant access to the same data (and medical data) strictly to the physician and nurse stakeholder roles in EIII.1 (Table 3). This contradiction can be addressed by limiting the consent listed for EIII.1 to medical data, since the concern for access to this by the MOA stakeholder role is addressed by the goal in EII.3 (Table 2).

As stated in Section 3.5 the patient's requirement for restricting access to medical data by non-care providers conflicts with system functions which allow the MOA to generate reports. While acknowledging the patient's desire to protect medical data, the functional need for access is essential for delivery of the needed care to the patient. To resolve this, the patient stakeholder is made aware of the functional perspective of the system. For example, the fact that TAPAS is used by general practitioners with small to medium support staff necessitates the need for MOA to perform this task. Attention is also drawn to measures to address this concern; including operational procedures, for example, handling of printed reports in TAPAS (Section 3.6). In addition, support staff can also be subject to non-disclosure agreements.

Although the patient stakeholder's requirement might not be satisfied with this approach, the compromise provides a degree of control on how the data is handled while providing care.

Multilateral security [14] calls for means for negotiating security conflicts. The example in the prior paragraph resolves a conflict during the security requirements engineering by delegating a conflict that cannot be solved by the system to the security policy of the system. Further possibilities for negotiating conflicts are mechanisms that allow users to balance their security requirements against those from others during run time [16]. This could either be done implicitly when users select a specific functionality of the system which fulfills their security requirements, or explicitly if such negotiations mechanisms are implemented, which allow users to negotiate their security preferences, e.g., through identity managers.

3.8 Reconcile confidentiality requirements with functional requirements

This reconciliation step involves two main substeps, i.e., (1) stakeholder hierarchy reconciliation and (2) use case chain evaluation. The first step requires comparing the actor hierarchy created in the functional analysis activity with the hierarchy of confidentiality stakeholder. Functional actors may give rise to additional stakeholder roles. In TAPAS, this would be the case if the identifiable audit trail requirement would extend even to database system administrators. Similarly, confidentiality stakeholders may give rise to additional functional actors (and system functions). For example, if the patient stakeholder role would attach a temporal validity 'until consent withdrawn' to its consent of releasing medical information to care givers, the system would need a use case 'withdraw consent'.

The second reconciliation step evaluates use cases and chains of use cases with respect to possible conflicts with regard to confidentiality requirements. Note that evaluating individual use cases may not reveal a conflict. However, chains of use cases may indirectly lead to violations of confidentiality requirements. An important precondition for this step is that all functional actors have been mapped to confidentiality stakeholder roles.

Further, if many or too few conflicts with regard to confidentiality requirements exist, reconsidering the granularity of the selected episodes may be useful. It may then be useful to break down an episode into several ones, or, in the contrary, to search for an overarching episode reducing the redundancy of several existing ones.

4 Related Work

Security concerns in general have received some attention in research on requirements engineering. There also are a number of studies that are explicitly related to medical information systems and health care.

Lui et al. [10, 11] emphasize the social aspects of security and privacy requirements, and consider security and privacy concerns in the design of an agent based health information system as a case study. They propose a framework based on the agent-oriented requirements modelling language i*. They use this framework to clarify the relationships among stakeholders and, in particular, to find vulnerabilities in their organizational relationships. This is meant to help in resolving conflicts between requirements and to make threats explicit. Common to our approach, Lui et al. point out the need to analyze the social context of stakeholders. But where we emphasize the multilateral point of view that each stakeholder has his or her own legitimate interests in the system, Lui et al. focus on attacker analysis, and ways a possible attacker might exploit vulnerabilities to compromise security. Thus, their analysis addresses the level of system requirements (the result of our Step 7), where the different views of stakeholders have already been refined to a coherent set of requirements, and the decisions that are necessary when designing a system based on those requirements.

Goal-Oriented Requirement Language (GRL) [2] is based on the i* modelling framework and it provides a language for supporting goal oriented modelling and reasoning of non-functional requirements. While *CREE* does not currently support a formal

representation, our next step we will include defining a representation which captures the properties of requirements (Section 2.3) and a reasoning mechanism for supporting analysis such as identifying inconsistencies of elicited requirements.

Similarly, Mouratidis et al. [12] adopt the i* modelling framework for their methodology Tropos. Tropos deals with non-functional requirements, including security requirements, throughout system requirements analysis, system design, and implementation in a homogenous way. The method analyses dependencies between actors, goals and tasks just as in Lui et al. [10, 11]. They mention that in the early requirements phase the security goals of the actors should be analyzed. Nevertheless, the authors offer no tools or descriptions of how this can be done. Steps 3 through 5 of our method attend to this question.

In their work extending Michael Jackson's problem frames, Nuseibeh et al. [9] introduce their Security Requirements Framework. The framework is based on security goals. Conflicts in security goals are resolved using threat analysis. Through threat analysis it is possible to prioritize goals. In addition, whenever a functional requirement is introduced it is contrasted with the existing security goals and appropriate security constraints are introduced. Although the authors emphasize the importance of analyzing the domain, a method for eliciting the conflicting security goals of the actors of the system among themselves is not considered.

Anderson [3] proposed a security policy for clinical information systems. As a security policy, it already reconciles the (assumed) goals of different stakeholders to a general framework that is more abstract than a specific set of requirements for a particular system to be built. It states 'principles' of access control, attribution (auditing), and design (trusted computing base) which must be followed to "reflect current best clinical practice".

Antón et al. [5, 4] discuss the differences between system requirements and security policies in detail. Policies are "meta-requirements" that are implemented by system requirements. They also point out that most approaches do not address the question whose goals a policy reflects – and to which extent it does so. The main objective of Antón et al. [5] is to align privacy and security policies with the functionalities of the system. Therefore, although the work emphasizes the importance of the different security (and privacy) interests of the stakeholders in delivering policies, the work does not focus on the acquisition of system requirements.

5 Experiences and Lessons Learnt

We have gained valuable insight by applying *CREE* to TAPAS. Apart from various minor changes in terms of method terminology, sequence of steps, and attributes for requirements, we would like to share the following thoughts with the reader:

- Software engineers seem to have a natural tendency to specify confidentiality requirements using an operational paradigm, e.g. stakeholder A is allowed to *read* information b. In this case, operation *read* would be one of the system functions. However, we believe that this operational paradigm of specifying confidentiality requirements is limiting and thus should be avoided.

- It may not be practical to involve all confidentiality stakeholders directly in the elicitation process. For example, patients can arguably be considered as one of the most important groups of confidentiality stakeholders. However, this group is so large and diverse that a direct involvement may not be practical nor economical. In the case of the TAPAS project, we have concentrated on studying literature and standards on patient information privacy concerns and produced their requirements on their behalf. This should ideally be reviewed from a representative patient association.

- Confidentiality requirements and functional requirements specifications cannot be developed separately since they impact each other significantly.

- The second step of reconciling functional and confidential requirements is quite la-

borious and error prone. This is because all possible use case chains have to be considered. Tool support is desirable.

- A more formal model for specifying confidentiality requirements would enable more intelligent analysis tools. However, even the semi-formal table format can be confusing to stakeholders in practice and, therefore, it is very important to provide a narrative translation for any semi-formal or formal notation.

- As confidentiality means to restrict information flow to a counter-stakeholder, and information inherently is a stochastic entity, confidentiality requirements can be satisfied in an absolute sense only in very restricted environments, which do not exist in the civil world. Therefore, it is worthwhile to explicitly express 'how confidential' certain information must be kept [15]. Moreover, the degree of confidentiality that can be achieved strongly correlates with the effort (cost) invested in achieving it. Building a 'secure' system in practice will always involve making compromises, but the 'degree' of confidentiality is typically not reflected in requirements specifications. This can lead to difficulties down the road during software design and implementation. We learned that leaving confidentiality degrees implicit can lead to extensive discussion between engineers disagreeing on the fit of a proposed solution to make a system 'secure'.

Acknowledgements

The TAPAS project has been funded by Vancouver Coastal Health and BC Ministry of Health. The authors would like to thank the many individuals that contributed to this open source project, including Joel Legris, Maike Dulk, Brad Barcley, Tim Cook and many others. Research on *CREE* has been funded in part by the Natural Science and Engineering Research Council of Canada. An IBFI seminar at Dagstuhl initiated the cooperation on *CREE*. Thomas Santen thanks the University of Victoria for hosting a most productive and pleasant stay.

References

[1] Electronic Medical Summary (e-MS). `http://www.e-ms.ca`. Acessed June 1st, 2005.

[2] GRL - Goal-oriented Requirement Language). `http://www.cs.toronto.edu/km/GRL/`. Acessed August 10, 2005.

[3] R. J. Anderson. A security policy model for clinical information systems. In *IEEE Symposium on Security and Privacy*, pages 30–43. IEEE Press, 1996.

[4] A. I. Antón, D. Bolchini, and Q. He. Using goals to extract privacy and security requirements from policies. Technical Report TR-2003-17, NCSU, 2003.

[5] A. I. Antón, J. B. Earp, C. Potts, and T. A. Alspaugh. The role of policy and stakeholder privay values in requirements engineering. In *Proc. 5th IEEE International Symposium on Requirements Engineering*, pages 138–145. IEEE Computer Society, 2001.

[6] R. H. Dolin, L. Alschuler, C. Beebe, P. V. Biron, S. L. Boyer, D. Essin, E. Kimber, T. Lincoln, and J. E. Mattison. The HL7 Clinical Document Architecture. *J Am Med Inform Assoc*, 8(6):552–569, 2001.

[7] S. F. Gürses. Security requirements elicitation in multi-laterally secure systems. Master's thesis, Humboldt University, Berlin, 2004.

[8] I. Jacobson. *Object-Oriented Software Engineering – A Use Case Driven Approach.* Addison-Wesley, 1992.

[9] L. Lin, B. Nuseibeh, D. Ince, M. Jackson, and J. Moffett. Introducing abuse frames for analysing security requirements. In *Proceedings of the 11th IEEE International Requirements Engineering Conference*, 2003.

[10] L. Liu, E. Yu, and J. Mylopoulos. Analyzing security requirements as relationships among strategic actors. In *Proc. 2nd Symposium on Requirements Engineering for Information Security (SREIS?02)*, 2002.

[11] L. Liu, E. Yu, and J. Mylopoulos. Security and privacy requirements analysis within a social setting. In *Proc. 11th IEEE Requirements Engineering Conference*, pages 151–161. IEEE Press, 2003.

[12] H. Mouratidis, P. Giorgini, and G. Manson. Integrating security and systems engineering: Towards the modelling of secure information systems. In *CAiSE 2003, LNCS 2681*, pages 63–78. Springer Verlag, 2003.

[13] Personal Information Protection Act. (PIPA). `http://www.oipcbc.org/`. Accessed June 6, 2005.

[14] K. Rannenberg, A. Pfitzmann, and G. Müller. IT security and multilateral security. In G. Müller and K. Rannenberg, editors, *Multilateral Security in Communications – Technology, Infrastructure, Economy*, pages 21–29. Addison-Wesley, 1999.

[15] T. Santen. Probabilistic confidentiality properties based on indistinguishability. In H. Federrath, editor, *Proc. Sicherheit 2005 – Schutz und Zuverlässigkeit*, Lecture Notes in Informatics, pages 113–124. Gesellschaft für Informatik, 2005.

[16] G. Wolf and A. Pfitzmann. Properties of protection goals and their integration into a user interface. *Computer Networks*, 32:685–699, 2000.

[17] P. Zave and M. Jackson. Four dark corners of requirements engineering. *ACM Transactions on Software Engineering and Methodology*, 6(1):1–30, 1997.