Designing for Privacy and Other Competing Requirements

Eric Yu[1] and Luiz Marcio Cysneiros[2]
[1]Faculty of Information Studies
yu@fis.utoronto.ca
[2] Department of Mathematics and Statistics
Information Technology Program
York University
cysneiro@mathstat.yorku.ca

Abstract. Privacy may be interpreted in different ways in different contexts, and may be achieved by means of different mechanisms. It is also frequently intertwined with security concerns. However, other requirements such as functionality, usability and reliability, must also be addressed since they often compete among each other. While the understanding of technical mechanisms for addressing privacy has been growing, systematic approaches are needed to guide software engineers to elicit, model and reason about privacy requirements and to address them during design. In a networked world, multi-agent systems have been emerging as a new approach. Each agent may have his own goals and beliefs and social relationships with each other. Each agent may have his own perspective concerning privacy. Perspectives from different agents may conflict with each other. Moreover, they may conflict with other requirements such as availability and performance. In this paper we present a framework to model the way agents interact with each other to achieve their goals. The framework uses a catalogue to guide the software engineer through alternatives for achieving privacy. Each alternative will be modeled showing how it contributes to privacy as well as to other requirements within this agent or in other agents. The approach is based on the ***i**** framework. Privacy is modeled as a special type of goal. We show how one can model privacy concerns for each agent and the different alternatives for operationalizing it. An example in the health care domain is used to illustrate.

1. Introduction

Privacy, security and trust are increasingly requiring attention in today's network-based systems, frequently demanding tradeoffs to be considered and requirements to be negotiated. They have to be taken into account at the earliest stages of the software development process. Some works such as [Antón 02] and [OECD 99] have helped to understand privacy from the viewpoint of providers and consumers, helping software engineers find solutions to privacy concerns. However, privacy is one kind of requirements among many. Thus, sometimes decisions may be made that put privacy at a lower priority level than other competing requirements. Consider the following examples:

- One of the goals of the Guardian Angel project [Szolovits 01] is to provide automated support to assess patients with chronic diseases. One solution could encompass several software agents working together such as: the patient's software agent, the physician's software agent and the hospital's software agent. Privacy is of course one of the major concerns here. Patients do not want their medical records to be seen by unauthorized third parties, especially health care insurance companies. Nevertheless, privacy assurance may lead to design decisions like the use of cryptography that can compromise performance aspects that might be considered critical by physicians and patients. Authentication can also pose challenges when considering a desired level of usability.
- In electronic payment systems, anonymous payment systems have been proposed as an approach to assure customer's privacy. The unfortunate side effect is it leaves room for illegal activities such as money laundering.

Understanding privacy as deeply as possible is undoubtedly a must, but we need also to understand, model and reason how these requirements interact with other requirements.

Privacy needs to be understood in terms of the social relationships that underlie the application domain. For example, physicians may be happy to meet patients' expectations for privacy on their medical records. However, the hospital software that physicians use may be provided by a third party not so eager to meet patients' expectations. Therefore, we need a way to model, understand and reason about the social relationships involved in the problem being addressed.

From the viewpoint of system development, aspects like privacy, security and usability are also known as non-functional requirements (NFRs). Functional requirements prescribe what functions the system should perform, while NFRs concern how well the functions delivered by the system are accomplished, e.g., a good response time (performance), how reliable is the software (reliability) or

how safe it is to use the system (safety). For lack of systematic approaches, NFRs are frequently poorly addressed or neglected during development, resulting in serious deficiencies in the final product. Errors due to omission of NFRs or not properly dealing with them are among the most expensive and most difficult to correct [Mylopoulos 92] [Ebert 97] [Cysneiros 01].

Like other NFRs, privacy can be interpreted in different ways leading to different possible solutions. A qualitative goal-oriented approach allows different interpretations to be accommodated during the early stages of requirements and design. Recent advances in requirements engineering offer systematic approaches for addressing this type of requirements and model the different alternatives that might arise during tradeoffs [Potts 94] [Chung 00] [VanLamsweerde 01]. However, they do not provide constructs to model and reason about the social dimension.

In today's networked world, we need to move towards an "agent-oriented" approach to modeling and analysis where agents can be humans, hardware and software interacting in complex ways to achieve shared or competing goals. An agent-oriented approach to requirements engineering extends the goal-oriented approach by introducing the social dimension. Agent abstractions are used to hide the detailed actions within the agent's discretion, thus allowing for local autonomy [Yu 97][Yu 01]. Strategically significant elements of work processes are described in terms of dependency relationships among agents. Complex social relationships can be modelled and analyzed. Each agent will have his own goals and beliefs as well as his own notion of privacy.

Despite their autonomy, agents interact in a social network requiring them to share information (raising privacy concerns), communicate with each other (raising security concerns) and to believe that goals will be achieved (raising trust concerns).

The ***i**** framework models relationships among social actors (agents) in a strategic way. Actors depend on each other intentionally (e.g. for achieving goals), thus forming a network of intentional dependencies. Examining this network, we can reason about opportunities and vulnerabilities.

In this paper we show how ***i**** supports ways for modelling and reasoning about non-functional requirements like privacy and security. Taking the patient-physician example mentioned above, we may express that a patient depends on the physician for having his expectations regarding privacy met. The physician depends on the hospital to provide him with software. The hospital in turn, depends on a software company to have the software installed and maintained. This way, one would be able to analyze specifically where privacy concerns arise, and how and by whom they could be addressed. Alternatives would be identified and modeled, so that one could assess which alternative would better meet privacy goals. Privacy and security as well as other NFRs can be modeled from the viewpoint of each stakeholder.

To illustrate the use of the catalogue and the ***i**** framework to deal with privacy aspects we use an example from the health care domain. The example provides a glimpse of the highly complex social relationships that require careful systematic analysis when addressing privacy requirements. The health care domain is also a good example because several aspects of privacy have recently been called into question by many organizations and governments [HealthPrivacy] [Berman 99].

2. Achieving Privacy and Security During System Design

In our approach, privacy is interpreted by refining it into subgoals and subsubgoals, eventually linking them to implementable mechanisms. Various subgoals and mechanisms may contribute to privacy to varying degrees. Each stakeholder's interpretation of privacy may lead to different goal refinements and mechanisms. The various interpretation of privacy can be collected and organized into a catalogue for reference during requirements elicitation, analysis and design.

There are a number of works presenting practices, techniques and technologies that can be used to implement and enforce privacy in networked environments. These different mechanisms are highly interrelated and may be used by anyone designing networked software systems. We here adopt the principles from the Organization for Economic Co-operation and Development guidelines [OECD 99]. Its categorization will guide us when decomposing the privacy goal. They divide privacy into six different categories:

- Allow Individual Participation
- Provide Openness of Purpose
- Limit Use and Disclosure of Data
- Accountability of Data Controller
- Educate Users and Private Sector
- Protect Privacy Through Transborder Data Flow Contract

A catalogue is used to capture knowledge on achieving privacy in many different situations. The knowledge may come from various sources ranging from the many existing works in the area to specific knowledge accumulated by individuals. Having this catalogue available, we can reuse its knowledge or add new knowledge to it. This knowledge will be represented using a primarily hierarchical structure to allow the representation of the organization's knowledge starting from the higher-level goals to achieve privacy. The catalogue also allows to represent different ways of achieving the same goal so one can choose the way that is best suited to the domain being analyzed. Along with the operationalizations for privacy, we also show possible correlations to other, possibly conflicting, requirements. This way we are able to show that one specific solution might satisfice privacy and contribute positively or negatively to other requirements such as usability and availability.

Figure 1 shows a catalogue of privacy with its refinements and operationalizations derived from the guidelines shown in [OECD 99] as a starting point to which we have added some knowledge from other sources. The catalogue was built using ***i**** constructs including: softgoals, goals, tasks, and beliefs. In Figure 1 can see for example, that Privacy is refined into Limit Use and Disclosure of Data, which is further, decomposed into Minimize Disclosure and Collection of Personal data and later decomposed, among other options, into Reduce Need for Personal Data. To satisfice the latter, we may find three options: Use Anonymous Payment, Use Digital Certificates and Use Anonymous Profile. These options can be used alone or together to achieve different needs for privacy. Notice that to Use Digital Certificates while contributing to Privacy will eventually hurt Maintainability since personal data change over time. The use of Public Key Cryptography can implement Digital Certificates but may also have negative impact on Performance.

The softgoal concept is used in ***i**** to express non-functional requirements. NFRs frequently interact with each other in complex ways. Qualitative reasoning can be carried out using contribution links among softgoals. The semantics of the links are based on the concept of satisficing [Chung 00]. The most common contribution types are Help/Hurt (positive/negative but not sufficient to meet the parental goal), Some+/Some- (positive/negative of unknown degree), whereas Make/Brake indicates positive/negative of sufficient degree. Although these distinctions are coarse grained, they are enough to decide whether we need further refinement and search for more specific softgoals and operationalizations or not. Contribution links allow one to decompose NFRs to the point that one can say that the operationalizations to this NFR have been reached (i.e., the goals are no longer "soft"). Operationalizations can be viewed as functional requirements that have arisen from the need to meet NFRs. This can explain why we frequently face doubts about if a requirement is functional or non-functional. Take for example a clinical analysis laboratory. We may have stated a requirement like: "Samples should be traceable so one can know, at different times, where this sample is". This may appear to be a functional requirement while, in fact, it is a functional requirement: "The software must handle samples" constrained by the NFR Traceability.

Operationalizations are typically specified as tasks, each indicating a particular way of doing something. All the subcomponents of a task (refined using the task decomposition link(↑)) must be carried out. If there is more than one way to accomplish something, then the state of affairs to be achieved is represented as a goal with means-end links(↑) linking to the alternatives.

Contribution links are the core of design decisions. By reasoning about how different operationalizations would contribute to satisfice a softgoal, one may decide which is the best alternative to purse. Based on the semantics of the contribution links [Chung 00], decision values are propagated from an offspring to its parents allowing one to visualize what impact would come from adopting one alternative instead of another. A prototype tool has been developed to support

propagating the contributions automatically but allowing interventions of the designer in case of conflicts or undecided situations arises.

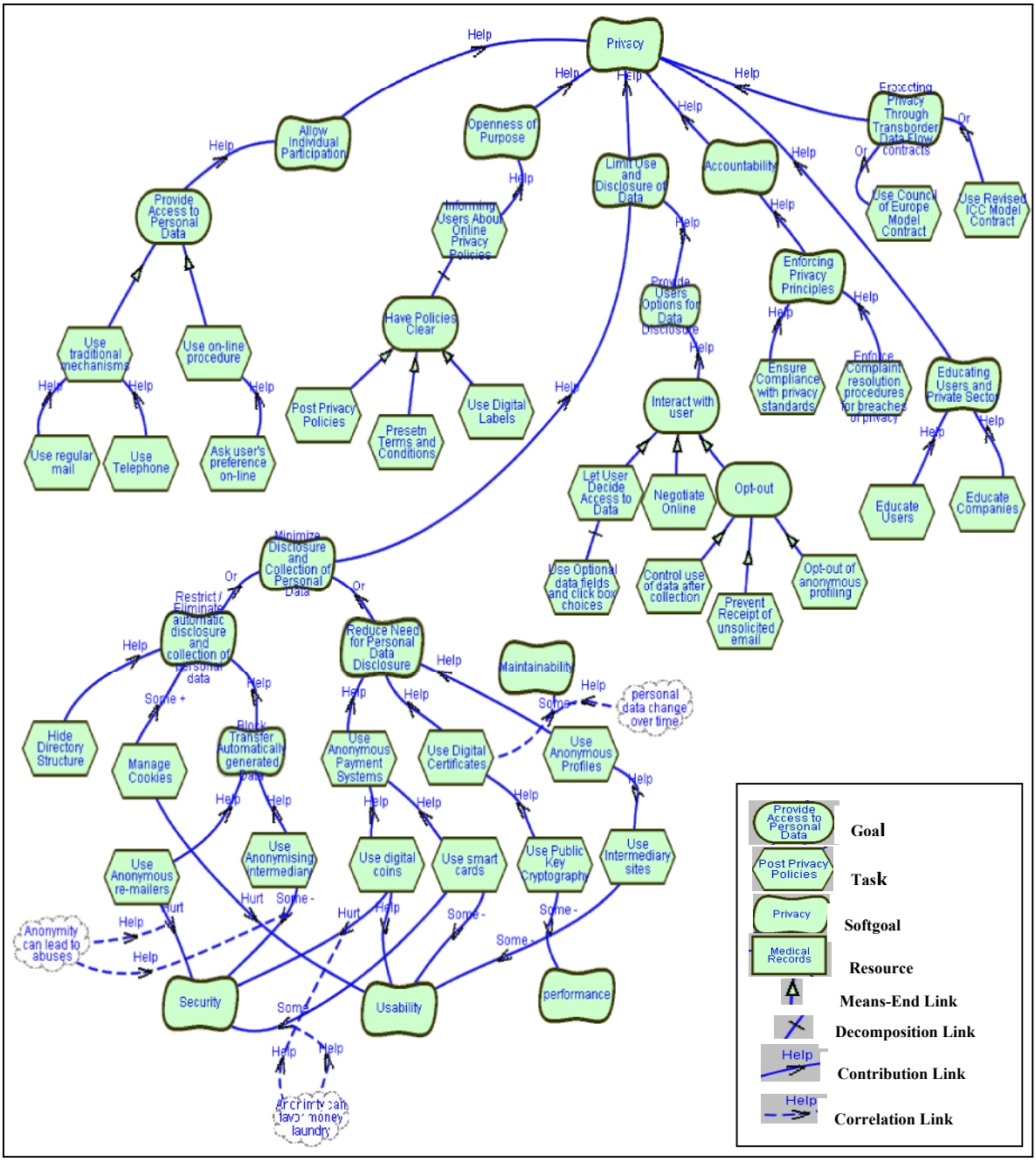

Figure 1 – Catalogue of Privacy Alternative Solutions

In Figure 1 we can see the use of i* constructs to build the catalogue. It shows a refinement process for each of the categories we used for Privacy above. For example, we see that to Minimize Disclosure and Collection of Personal Data one can Restrict Automatic Disclosure and Collection of Personal Data or Reduce the Need for Personal Data Disclosure. On choosing the first option, one might operationalize it by Hiding Directory Structure, Managing Cookies or Blocking the Transfer of Automatically Generated Data. Choosing to Manage Cookies will have negative impact on Usability since many users might have great difficult to do that (contribution link *hurt* to Usability). On the other hand, choosing the latter option would lead to two different possibilities: Use Anonymous Re-mailers like Replay or Use Anonymising Intermediary such as Anonymizer. Both options will at different levels compromise Security (*Hurt* and *Some–* contribution links)

We see this catalogue as a continuously evolving supporting tool for Privacy design. We understand that operationalization levels of this catalogue may experience differences when one tries to apply it to some domains. The catalogue is to be taken as guide to operational levels but it should still give some room for one to apply it just in some extent, extending its use in different directions. We also show in this catalogue that some of the alternative to meet privacy may conflict with other NFRs like Security and Usability. This sample catalogue does not intend to be complete. It is intended as a starting point as an example for other contributions to its development as the community enhances its understanding on privacy aspects.

3. Modelling Privacy in *i**

The ***i**** Framework can be used for both early and late phases of requirements engineering. It allows modelling relationships among actors (agents) in a strategic way. In ***i****, actors are taken to be intentional (have goals, beliefs, etc.) and strategic (concerned about vulnerabilities and seeking opportunities). Actors may be abstract (*roles* defining responsibilities), concrete (*agents* - individuals or classes with specific capabilities, machines with hardware/software functionalities), or other organizational constructs (e.g., *positions* which package a number of roles together to be assigned to a single concrete agent). Actors depend on each other forming a network of intentional dependencies. Examining this network, we can reason about opportunities and vulnerabilities. The ***i**** Framework offers two different models: the Strategic Dependency model (SD) and the Strategic Rationale model (SR); allowing different levels off abstraction to be used.

3.1 Eliciting and Modelling Agents in the Domain

One of the challenges posed by today's complex and networked domains is to understand the social relationship that underlies the domain. It is important to understand how one agent depends on another for what and to what extent. Therefore, we start to understand and model the domain by eliciting and representing the main agents involved and how they depend on each other. We start this process by drawing a Strategic Dependency (SD) model from the elicited information. The SD model depicts a process through the use of a network of dependency relationships among actors. In ***i****, a dependency is a relationship in which one actor (the *depender*) depends on another actor (the *dependee*) for something (the *dependum*) to be achieved. A dependum can be a goal, task, resource, or softgoal, reflecting the types of freedom allowed by the relationship. A goal dependency is one in which one actor depends on another to bring about a certain condition or state in the world, while the depended actor (the dependee) is free to, and is expected to, make whatever decisions are necessary to achieve the goal. Thus, it also indicates that one actor does not care how the other actor will achieve this goal. On the other hand, a task dependency means that the depender expect a certain process to be taken by the dependee. A Softgoal is similar to a goal representing a condition in the world that an actor would like to achieve. However, differently from the goal, in the softgoal the criteria for the condition being achieved are not sharply defined. Therefore, softgoals are said to be satisficed (sufficiently achieved) rather than satisfied. A resource dependency means that one actor depends on the other for the availability of an entity (physical or information).

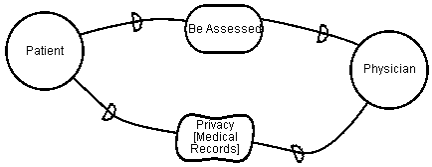

Figure 2 – Privacy between Patient and Physician

Let us take for example the health care domain. In this domain, Privacy is one of the major requirements. Individuals share a great deal of sensitive and personal information with their doctors and they expect these information to be kept away from other people especially insurance companies. Without adequate privacy protections, individuals take steps to shield themselves from what they

consider harmful and intrusive uses of their health information often at significant cost to their health [HealthPrivacy.org]. One of the most important relationships in this domain happens between the patient and the physician. Patients expect to be assessed by physicians and to have privacy regarding all the information provided to physicians along with any medical information the physician might collect or produce. Figure 2 shows the SD model representing that. There, we can see that the actor Patient depends on the actor Physician to have the goal Be Assessed achieved and also to have the softgoal of Privacy regarding medical records to be accomplished.

3.2 Understanding How Agents Achieve Their Goals

Although the SD model gives a global understanding of the domain and its relationships, we also want to have a deeper understanding on how each agent will achieve their goals. For that purpose we use SR models to refine the knowledge about the domain. The SR models describe the intentional relationships that are "internal" to actors, in terms of process elements and the rationale behind them. Goals are related to tasks through means-ends links. The tasks are the different ways in which the goal can be accomplished. Each task may consist of subgoals, subtasks, resources, and softgoals (via the task decomposition link). All elements of a task must be satisfied in order for a task to be satisfied. A goal is satisfied if any of its tasks is satisfied. Satisfied here means that it can be successfully met during the execution of the process [Yu 01].

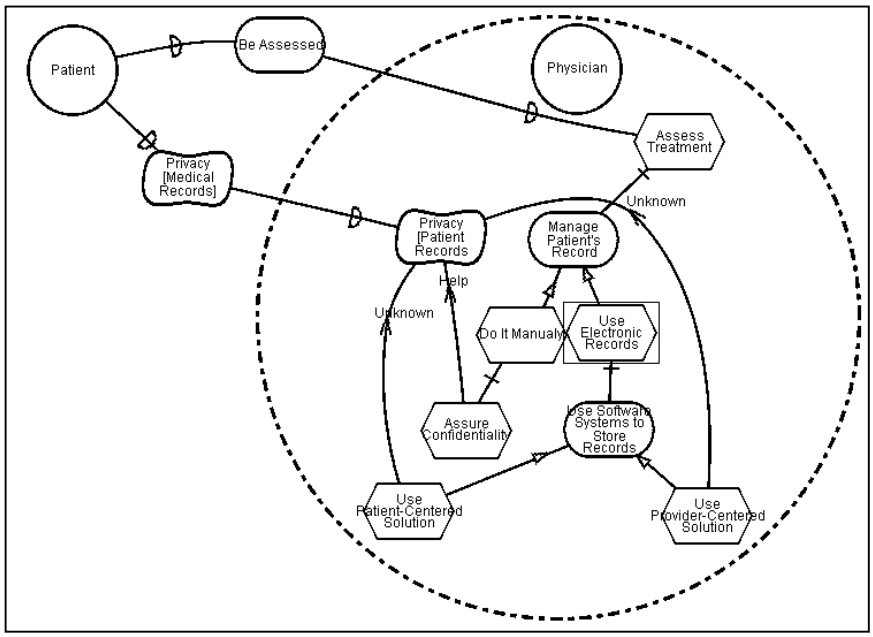

Figure 3 – Starting to Modelling Privacy

One important characteristic of SR models is that we can express different alternatives for achieving a goal. We can represent different means to get to an end, by using means-end links. For example, Managing Patient's Record is part of the Physician's job when assessing the Patient. Using an SR model we can detail how this management can occur and how the expected Privacy will affect it, i.e. which efforts the Physician might undertake to satisfice the softgoal dependency that the Patient has on him. Figure 3 illustrates the reasoning. Management of Patient's Record can be done in two different ways, either Manually or Using Electronic Records, i.e. software systems. Many physicians may decided to keep Managing patient's record manually as they do today because they do not trust in software systems to handle such a delicate subject as the patient's record. Others may be confident enough to adopt electronic records or might even be compelled to use it either by the administration or eventually by law.

When doing it manually there is a task for Assuring Confidentiality that is considered to *help* Privacy aspects. It *helps* because Patients usually trust in their Physicians so if Privacy depends only in the Physicians' efforts for keeping the records private it may be enough for the Patient.

On the other hand, when Using Software Systems we might have two different options. First, we can adopt a Patient-Centered Solution where all the patient's record will be in patient's hand. The second alternative would be to use today's solution, Provider-Centered Solution, of having it controlled by health care providers as hospitals and clinics. Initially the contribution of each of these solutions to the privacy softgoal is considered *unknown* since there are no insights about what each alternative would represent.

The contribution link types (*Help*, *Hurt*, etc) will be used later to propagate labels to qualitatively evaluate the viability of the different alternatives.

3.3 Exploring Different Alternatives

Since we have two alternatives for Managing the Patient's Record we have to model these alternatives. We may consider how the current process can be improved if new actors are introduced to the problem in order to delegate some of the responsibilities that are currently under one actor's responsibility to another actor. At first, we should represent these new actors in SD models showing the new relationships and posing the dependencies these new actors will introduce. After that, we may refine this model into SR models to understand how these new actors will manage to achieve their goals and thus how they will contribute to the process.

To guide our reasoning we use the exemplar proposed for agent-oriented software development methodologies [Yu 02] that is based on the Guardian Angel Project [Szolovits 94]. One of the goals of the Guardian Angel project is to provide automated support to assess patients with chronic diseases through the use of a set of "guardian angel" software agents integrating all health-related concerns, including medically-relevant legal and financial information, about an individual. This personal system will help track, manage, and interpret the subject's health history, and offer advice to both patient and provider. Minimally, the system will maintain comprehensive, cumulative, correct, and coherent medical records, accessible in a timely manner as the subject moves through life, work assignments, and health care providers.

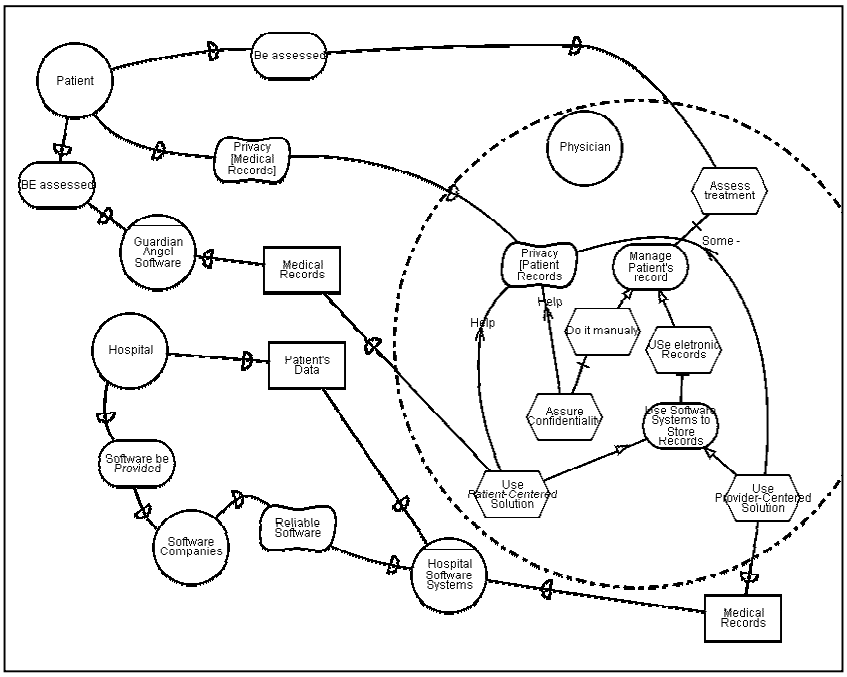

Figure 4 – Exploring Alternatives

Figure 4 shows the broader model. We arrived at this model by considering what other actors would have to be involved to address each possible solution. For the Patient-Centered Solution we introduced the Guardian Angel Software agent. For the Provider-Centered Solution we initially introduced the Hospital Software System agent. Note that we are using the term agent here in the context of conceptual modeling during requirements analysis and design. During design, decisions

might be made to map some of those conceptual agents into software agents (e.g., with intelligent or mobile capabilities).

Later, we realized that many hospitals would use software companies not only to provide the software but also to administrate it, enabling the hospital to concentrate in their area of expertise. For representing that, we use a resource dependency showing that the Physician will depend on both softwares to have the Medical Records delivered.

At this point, we may change the contributions types of each alternative to Privacy. The Patient-Centered Solution would *help* to achieve Privacy since having the data controlled by the Patient leave little room for Privacy problems, at least in a first glance. On the other hand, a Provider-Centered Solution would have a negative impact on Privacy (*some-*) since Patients may not completely trust in the Hospitals and even less in third-party companies. Thus, if at this point we label the Patient-Centered Solution as satisficed and the Provider-Centered Solution as denied we would be able to see that doing so would have the Privacy softgoal as weakly denied. If we do the opposite, propagating the contribution to the Privacy softgoal we would see that it would be weakly accepted. This allows one to evaluate the consequences of opting for one alternative instead of another. Of course we usually have more complex analysis than the above one, and that is when the propagation of contribution would help most.

3.4 Modelling Different Viewpoints on Privacy

As we are modeling from the viewpoint of each actor, they can each have different interpretations of Privacy, eventually operationalized through different mechanisms, for example, the viewpoint of web sites designers versus the viewpoint of customers. Privacy may be differently decomposed accordingly to each actor's viewpoint and thus different operationalizations for Privacy can be found representing each actor's viewpoint. Necessary tradeoffs to satisfice both viewpoints can be made targeting a solution that satisfices the stakeholders the best way possible. For each actor where privacy is evaluated we may use the catalogue presented in Section 2 to decompose privacy into possible alternative solutions and later reason how each solution will contribute to Privacy.

In our case, we must investigate how Privacy would be translated to each of the software agents involved. We will represent three different viewpoints for Privacy the Patient viewpoint, the Hospital/Software Provider viewpoint and the Physician viewpoint. Physician's viewpoint for privacy will basically be restricted to his own efforts to assure patients' Privacy. Actually, as we are adopting electronic records, satisficing Privacy will be totally dependable on satisficing Privacy within the chosen approach. We will detail the existing model into SR models as depicted in Figure 5.

We introduce the Privacy softgoal in each of the software agents. The Privacy softgoal that patient has on Physician will now also depend on the Privacy softgoals on the Guardian Angel Software and on the Hospital Software. We have then to further decompose the Privacy softgoal. The catalogue presented in Section 2 will guide us in this process. We can see in Figure 5 that in this case we chose to decompose Privacy into Providing Users Options for Personal Data Disclosure and Minimizing Disclosure and Collection of Personal Data.

In the Guardian Angel Software we can see two different alternatives for storing the Patient's Record. We may either Store it in a PDA (personal digital assistant) that will be used by the patient or Store it at a Central Location. We can see (Figure 4) that the latter would contribute negatively to Privacy (*hurt*) because it may be more vulnerable to external attacks and may depends on patient's trust on the institution providing the storage. On the other hand, Storing it in the PDA would *make* (sufficiently positive) contribution to the softgoal. Being a personal device, the PDA is most likely to be used only by the patient or by people he trusts and to whom concerns about Privacy could be ignored. It is true that when eventually remotely connected, supposing these capabilities are offered, the PDA may be vulnerable to external attacks, but in this case other considerations than Privacy would have to be made and it is out of our focus at the moment. One might suggest that storing Patient's Record at a Central Location would allow the use of more powerful hardware, leaving room for patients to be able to choose who should have access to what information. This would

contribute to Privacy and it is modeled as a *hurt* correlation link (dotted line) to the *hurt* contribution from Storing at a Central Location to Privacy.

Applying the catalogue from Section 2 we refined Minimizing Disclosure and Collection of Personal Data into the softgoal Restrict Automatic Disclosure and Collection of Personal Data which in turn is further refined into the task Hide Directory Structure since the simple fact of having a subdirectory with a name of a disease might *hurt* patient's Privacy. Imagine for example a patient navigating his records and that someone is looking. Imagine now that at the end of the directory appearing in the browser one may see "HIV". The simple fact that someone knows you have taken the HIV test can be enough to get you in trouble. To deal with this requirement, the Guardian Angel Project proposes to combine the use of XML and HL7 DTD [HL7].

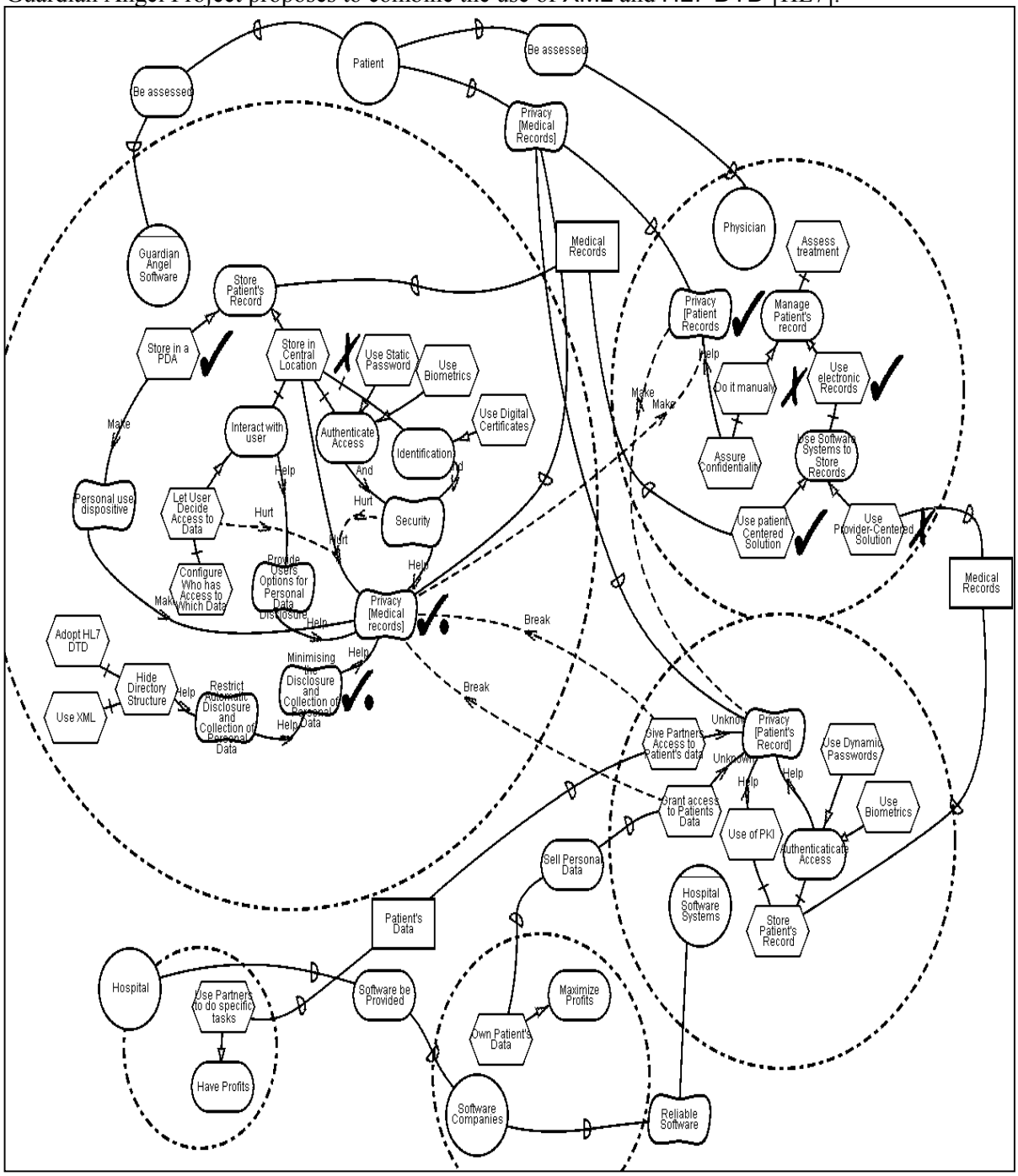

Figure 5 – Privacy Through Each Actor's viewpoint

Again applying the catalogue we first refined Providing Users Options for Personal Data Disclosure into the need Interacting with the user. Further refinement leads to a task showing that to operationalize this requirement it is necessary for the system to provide a way for patients to Configure Who has Access to Which Data.

By introducing security mechanisms, we can also diminish the vulnerability of Storing at a Central Location. The same way as we did with Privacy, we first decompose Security into Authenticate Access and Identification [Chung 00]. Refining the Authenticate Access goal, we can think about two different approaches, the use of Static Passwords or the use of Biometrics. Identification goal would be refined into the Use of Digital Certificates to ensure that the software is being accessed by the correct person. Security would contribute positively towards Privacy and it is shown with the *hurt* correlation link from Security to the *hurt* contribution from Storing at a Central Location to Privacy softgoal.

Looking through the hospital and software provider viewpoints things take a different perspective. The only mechanisms towards Privacy that Software Companies would be willing to take would be to use PKI and authentication. Furthermore, Hospitals want to use partners like clinical laboratories or image diagnosis laboratories to do part of the patient's assessment. To do that, access should be granted to Patient's Records. Access to these data would compromise Privacy in the Patient's viewpoint since it cannot be granted that hospital partners will not use patient's data in such a way that would be against patient's will. Regarding the Privacy through the Hospital Software System's viewpoint, granting access to partner would have an unknown contribution link. In the present case we use the unknown value most to express that in fact it does not matter which solution is adopted. In this case, we could have chosen not to represent any contributions at all, but we envision that this might change very quickly in a near feature and therefore we wanted to have this possible contribution clear for the sake of evolutionary concerns.

The Software Companies' viewpoint can be even worse. In order to maximize profits they may want to Own Patient's Data so they can sell them the way they want. Again, through the Hospital Software System's viewpoint it would have no clear impact on privacy. On the other hand, it would definitely compromise Privacy in Patient's viewpoint. Many cases have been recently disclosed reporting undesirable use of patient's information. For example, the Washington Post reported that CVS drug stores and Giant Food were disclosing patient prescription records to a direct mail and pharmaceutical company. The company was using the information to track customers who failed to refill prescriptions, and then sending them notices encouraging them to refill and to consider other treatments [Berman 99].

Therefore, at this point we decide to go with the Patient-Centered alternative. This is represented by the label of *denied* (*X*) next to the dependency link from Patient's Privacy softgoal to Hospital Software System's Privacy softgoal denoting that this dependency is not viable.

It remains to decide whether to use the PDA or to Store the Data at a Central Location. Although the use of Privacy mechanism can improve Privacy when Storing at a Central Location, Privacy can be more extensively granted if we Store Data in the PDA.

Up to this point, we can see that the Patient-Centered alternative Storing Data in a PDA is the alternative that presents the best contribution and therefore should be chosen to be implemented. The presence of the *denied* label next to the task denoting the Central Location Storage means that this alternative should not be adopted. On the other hand, the existence of the *satisficed* label (√) next to the Store in a PDA task denotes that this alternative of design may be adopted.

4. Privacy and Other NFRs - Reasoning among different alternatives

Of course, Privacy and Security are not the only concerns in a complex project. Many others NFRs such as availability, performance and usability can play important role in design decisions and should therefore be modeled and analyzed. Although we are presenting privacy and security modeling separately from other NFRs, we do that only for the sake of simplicity. In a real situation, privacy needs to be modeled and addressed together with other NFRs.

For example, Availability is one major concern from Physicians' viewpoint. Since NFRs are frequently difficult to elicit one may assume a proactive approach investigating what possible NFRs may be necessary for each actor. We must ask the stakeholders and ourselves what other possible NFRs such as security, availability and performance would be important to consider to each actor. One approach for this proactive investigation may be seen in [Cysneiros 01b].

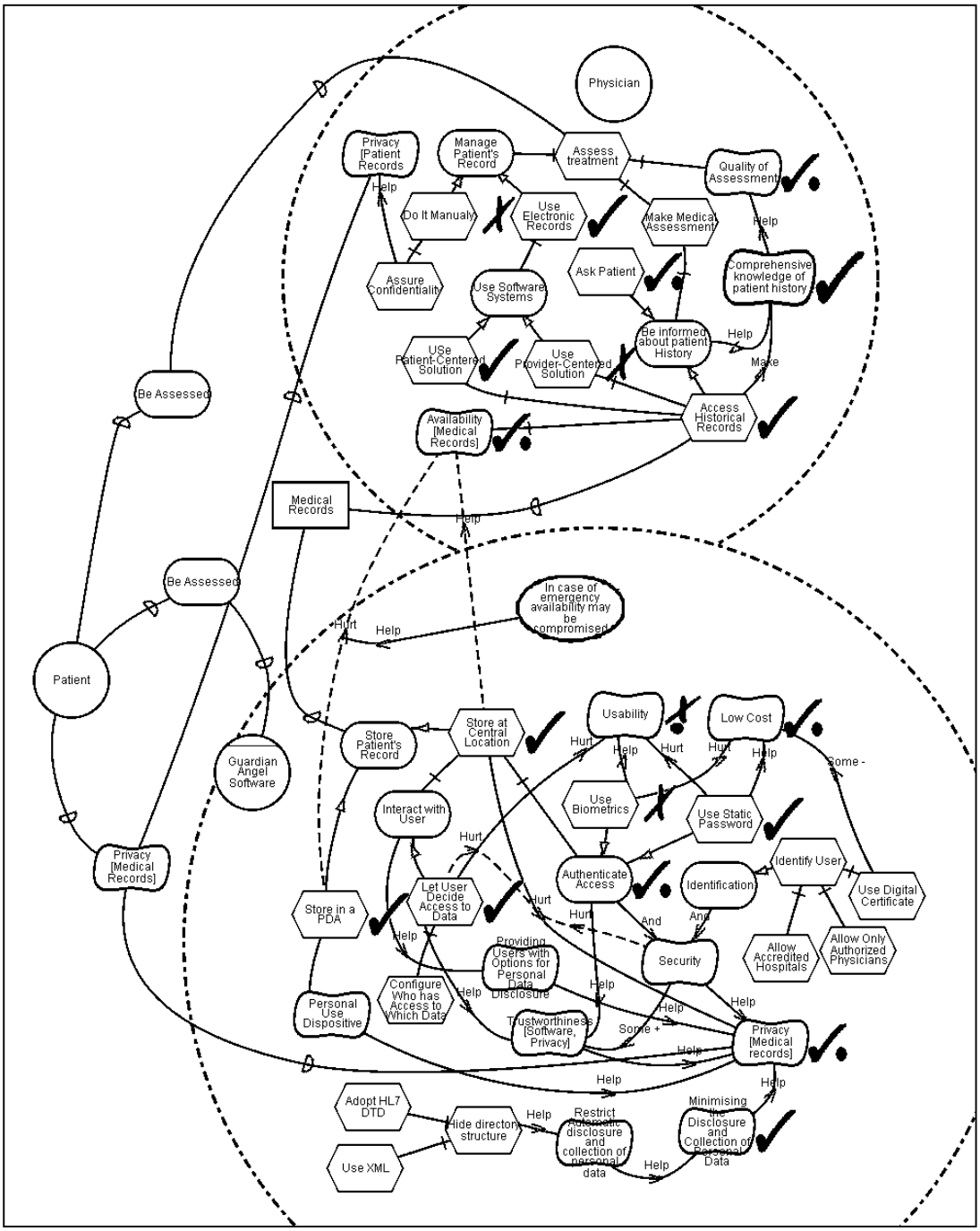

Figure 6 – SR Model After Representing and Reasoning About

Let us take the situation portrayed in Figure 6. On the one hand having patient's record Stored in the PDA up to now is the best alternative to be taken. On the other hand, if we consider the need for Patient's Record to be available we may have to carry out further investigations. Availability of Medical Records is important to allow Physicians to have Quality in their Assessment (a softgoal that decomposes the assess treatment task). In a normal situation, the Patient would be able to provide the Physician all necessary data by allowing the PDA to interface with the Physician's computer and the latter to retrieve the necessary information as long as the information solicited have

been authorized by the patient to be transferred. This alternative would still be compliant with the decision of having patient's record Stored in the PDA. However, during an emergency, the Patient may not have the physical ability to give access to his PDA or may even be unconscious. Figure 6 portrays this as a *hurt* contribution link from the task Store in a PDA within Guardian Angel Software to the Availability softgoal within Physician together with a belief stating this knowledge that *helps* in the *hurt* contribution.

On the other hand, Storing at a Central Location will contribute towards Availability because data could be accessed even without the direct intervention of the patient and virtually from anywhere. Since this alternative implies that hospitals and physicians may have access to patient's data, we further refined the task Identify User to show the need for Allowing Only Accredited Hospitals and Authorized Physicians.

Another NFR requiring tradeoffs is Usability. Reasoning about it, we realize that some tasks that *help* the Privacy softgoal would *hurt* Usability. Configuring Access may be a challenge for many Patients while to Authenticate Access depends on what alternative of authentication we decide to use. The use of Password Authentication may *hurt* usability because remembering and entering passwords long enough to be secure may be difficult to many patients. In the other hand, the use of biometrics would *help* usability because they do not call for any effort from the patient to use it. However, it may *hurt* another softgoal that represents the need to keep the Costs Low. The use of Digital Certificate was considered to impact the Low Cost softgoal but since the impact is not heavy and this task may play an important role on assuring Security we decided to use it anyway. Of course, these NFRs would not be the only ones involved in a system like this. Performance for example could be impacted if some further cryptography is used to address Security concerns if we decide to store it in a central location like a web site. However, for the sake of simplicity we will restrict the example to the NFRs modeled in Figure 6.

Figure 6 portrays not only the different alternatives but also records the design decisions taken. For example, we see that we decided to deny the alternative of Storing Patient's Data in the Hospital Software System since it *hurts* not only Privacy but also Availability. On the other hand, when choosing to satisfice the Storage of Patient's Data in the Guardian Software we decided for doing that Using a Central Location to Store it. Adopting the Authenticated Access and allowing the patient to Configure Who Would Have Access to What Data (refinement of Let User Decide Access to Data), contribute to diminish the negative impact of this decision. These two mechanisms will also contribute to Trustworthiness of patient regarding how the software can assure Privacy, which in turn will *help* to satisfice the Privacy softgoal. It is true that on doing so we will be *hurting* Usability concerns but Privacy was considered, in this case, to be more relevant than Usability. Finally, Usability is considered less important than keeping Costs Low and thus we keep the option of using password authentication instead of adopting biometrics solutions.

5. Related Work

An agent-oriented approach to requirements modelling, as exemplified by ***i****, gets at deeper issues than conventional modelling techniques. The approach therefore can potentially uncover unforeseen problems, and help stakeholders arrive at innovative solutions that address individual as well as organizational goals. Research in goal-oriented requirements engineering has provided some of the groundwork in this regard, by offering systematic techniques for discovering, refining, and addressing goals. GBRAM [Potts 94] and KAOS [VanLamsweerde 01] address goals and NFRs but do not deal with social relationships. The NFR framework [Chung 00] deals only with NFRs. How NFRs are related and impact on functional requirements is only superficially covered.

i* incorporates NFRs as softgoals. NFRs are eventually converted into functional requirements through operationalizating into tasks. Agent orientation extends goal orientation by introducing the social dimension. Agents in a social world have varying degrees of autonomy. Their behaviours are constrained by the networks of relationships that they find themselves in, even though they could potentially violate those constraints. During the early requirements stage, stakeholders are seeking to advance their strategic interests, exploring and assessing the kinds of freedoms and constraints they would face under various new systems proposals. Agent orientation therefore presents elicitation

challenges that go beyond those of goal-oriented requirements engineering. We propose here to use the ***i**** framework to support an agent-oriented approach to meeting privacy requirements during the early stages of design. As in any other approach privacy is becoming a major issue to be addressed since the very early phases of software development. The ***i**** framework offers a comprehensive support for eliciting and modelling Privacy together with other requirements since the very early stages of software development.

The ***i**** framework is complementary to other approaches addressing privacy. In [Korba 02] is shown some of the challenges of addressing privacy for agent-based e-commerce software systems together with a policy-driven approach for privacy negotiation. In [Antón 02] a taxonomy of privacy for web sites is shown with some high level categorizations together with many goals that at some level can help the designer on choosing among different alternatives for each case. In [OECD 99] is presented an inventory of instruments and mechanisms to address privacy on global networks. All the above works are important to bring to light the different approaches one might have for addressing privacy. However, having a comprehensive list of mechanisms without being able to understand their impact in the whole software design can frustrate the efforts for good quality systems. In this paper we have shown how to use ***i**** as a basis for modeling and reasoning about privacy as goals to be achieved. Works, like those mentioned above, are used to categorize privacy in such a way that it can help us on decomposing privacy into high-level sub-goals that can lead to privacy satisficing. The ***i**** framework can be used in a preliminary analysis of the domain and its inherent social relationship, later to be detailed with the many well know mechanisms to ensure privacy. The models can be used to express the different mechanisms one might consider to satisfice privacy within a domain and represent all the consequences of each alternative

A companion paper [Liu 02] elaborates on the use of actor relationship patterns in analyzing security requirements, threats and protection measures.

6. Conclusion

This work argues for the need for systematic design frameworks for modelling and reasoning about privacy, security and other NFRs. To support that reasoning we have presented a catalogue based on the guidelines from [OCDE 99]. We showed examples using the ***i**** framework to illustrate how one can model privacy as softgoals in order to assess the different alternatives to satisfice each notion of privacy and how each alternative would contribute positively or negatively for achieving privacy.

The ***i**** framework also allows one to explore different levels of abstraction by using SD and SR models, easily moving from one level of abstraction to another. Tracing the impacts of one change is also improved through the use of the ***i**** framework since we can simply represent one alternative previously satisficed as denied and vice-versa and thus evaluate the impact of these decisions on the design.

We have shown in this paper an example from the health care domain portraying how different alternatives can be modeled to satisfice privacy and how they would contribute not only to privacy but also to security and other requirements as usability, availability and cost. We have also shown that some alternatives might even contribute to privacy satisficing indirectly, e.g., by enhancing the trust the patient would have on the software. In addition, different perspectives for the same problem can be modeled as we showed here by focusing on the different viewpoints patients and hospitals might have. This is particularly important for the web domain because web providers' viewpoint may not match customers' viewpoints. Having modeled the different alternatives and their impacts one can go through a more detailed analysis of the domain and make design decisions in a less intuitive way.

The ***i**** framework has been applied in many different domains including telecommunication, smart cards and health care, including real-life case studies.

Future work includes studying more deeply the interrelationship between privacy and trust and to improve the existing prototype tool that supports the modeling and reasoning based on ***i****.

7. References

[Antón 02] Antón, A.I. and Earp., J.B. " A taxonomy for Web Site Privacy Requirements" NCSU Technical Report TR-2001-14, 18 December 2001.

[Berman 99] Privacy in the Digital Age: Work in Progress Jerry Berman & Deirdre Mulligan Nova Law Review, Volume 23, Number 2, Winter 1999. The Internet and Law

[Chung 95] Chung, L., Nixon, B. *"Dealing with Non-Functional Requirements: Three Experimental Studies of a Process-Oriented Approa*ch*"* Proc. 17th Int. Con. on Software Eng. Seatle, Washington, April pp: 24-28, 1995.

[Chung 00] Chung, L., Nixon, B., Yu, E. and Mylopoulos,J. "*Non-Functional Requirements in Software Engineering"* Kluwer Academic Publishers 2000.

[Cysneiros 01] Cysneiros,L.M., Leite, J.C.S.P. and Neto, J.S.M. *"A Framework for Integrating Non-Functional Requirements into Conceptual Models*" Requirements Engineering Journal – Vol 6 , Issue 2 Apr. 2001, pp:97-115.

[Cysneiros 01b] Cysneiros,L.M. and Leite, J.C.S.P. *"Using UML to Reflect Non-Functional Requirements"* Proceedigns of the 11th CASCON, IBM Canada, Toronto Nov 2001 pp:202-216

[Ebert97] Ebert, C. "*Dealing with Nonfunctional in Large Software System"s.* Annals of Software Engineering, 3, 1997, pp. 367-395.

[ga.org] The Guardian Angel Web Site - http://www.ga.org/ga

[HealthPrivacy] Health Privacy Project http://www.healthprivacy.org/

[HL7] HL7 SGML/XML Special Interest Group, http://www.mcis.duke.edu/standards/HL7/committees/sgml/index.html

[Korba 02] Korba, L. "Privacy in Distributed Electronic Commerce" in Proc. Of the 35th Hawaii International Conference on System Science, Hawaii, January 7-11,2002.

[Liu 02] Liu, L., Yu,E. and Mylopoulos, J. "Analyzing Security Requirements As Relationships Among Strategic Actors" Submitted to the Symposium on Requirements Engineering for Information Security (SREIS'02), Raleigh, North Carolina, Oct 15-16, 2002.

[Mylopoulos 92] Mylopoulos,J. Chung, L., Yu, E. and Nixon, B*., "Representing and Using Non-functional Requirements: A Process-Oriented Approach",* IEEE Trans. on Software Eng, 18(6), pp:483-497, June 1992

[Potts 94] Potts, C., K. Takahashi and A. I. Antón. ``Inquiry-Based Requirements Analysis"*, IEEE Software*, pp. 21-32, March 1994.

[OECD 99] "Inventory of instruments and mechanisms contributing to the implementation and enforcement of the OECD privacy guidelines on global networks" Head of Publications Services, OECD, 2 rue-André-Pascal, 75775 Paris Cedex 16, France.

[Szolovits 94] Szolovits, P., Doyle, J., Long, W.J. "Guardian Angel:Patient-Centered Health Information Systems" Technical Report MIT/LCS/TR-604, http://www.ga.org/ga/manifesto/GAtr.html

[VanLamsweerde 01] VanLamsweerde, A. "Goal-Oriented Requirements Engineering: A Guided Tour" *Proc of the 5th IEEE Int. Symp. on Requirements Engineering*, pp:249-262, 2001.

[Yu 97] Yu, E. "Towards Modelling and Reasoning Support for Early-Phase Requirements Engineering" *in Proc. Of the 3rd IEEE Int. Symp. on Requirements Engineering*, pp:226-235, 1997.

[Yu 01] Yu, E. "Agent-Oriented Modelling: Software Versus the World" Agent-Oriented Software Engineering AOSE-2001 Workshop Proceedings. LNCS 2222.

[Yu 02] Yu,E., Cysneiros.L.M., "Agent-Oriented Methodologies-Towards a Challenge Exemplar" in Proc of the 4th Intl. Bi-Conference Workshop on Agent-Oriented Information Systems (AOIS 2002) Toronto May 2002.