Addis Ababa University

College of Natural Sciences

Ontology-based Interactive Privacy Requirements Elicitation Method

Wendwesen Belay Kitaw

A Thesis Submitted to the Department of Computer Science in

Partial Fulfillment for the Degree of Master of Science in

Computer Science

Addis Ababa, Ethiopia

April 2020

Addis Ababa University

College of Natural Sciences

Wendwesen Belay Kitaw

Adviser: *Mesfin Kifle (PhD)*

This is to certify that the thesis prepared by *Wendwesen Belay Kitaw*, titled: *Ontology-based Interactive Privacy Requirements Elicitation Method* and submitted in partial fulfillment of the requirements for the Degree of Master of Science in Computer Science complies with the regulations of the University and meets the accepted standards with respect to originality and quality.

Signed by the Examining Committee:

	Name	Signature	Date
Advisor:	Mesfin Kifle (PhD)		
Examiner:	Dida Midekso (PhD)		
Examiner:	Ayalew Belay (PhD)		

Abstract

A major challenge in software engineering is to make users trust the software they use in their everyday activities. Trusting software depends on various elements, one of which is the protection of user privacy. Privacy is social, political, economic, technological and legal concepts that span across multiple discipline. Privacy requirements play a critical role in the personal information system. They should be considered since the early phases of system design. However, much of existing work on privacy requirements deals with them as security requirements, overlooking key aspects of privacy. Besides, privacy requirements are difficult to elicit, and manage due to the existence of immense challenges. Hence, there should be a systematic approach to elicit and manage privacy requirements. This thesis presents ontology-based interactive privacy requirements elicitation method that can help requirement engineers while eliciting privacy requirements in software development environment. It proposes core privacy ontology for privacy requirements engineering and an interactive privacy requirements elicitation method. We implemented the ontology, and developed an interactive environment (a tool) to facilitate the use of the ontology and to automate the method. The proposed ontology was evaluated by checking its completeness compared to other ontologies and its validity using competency questions. The results have demonstrated that the proposed ontology is complete and valid. Moreover, a controlled experiment was performed to evaluate usability of the ontology, the method and the tool. The results have demonstrated that the ontology covers the main privacy concepts, the method is efficient and easy to use, and the tool is friendly to use.

Keywords: privacy ontology, privacy requirements, requirements elicitation, requirements engineering, privacy by design (PbD).

Dedication

To: My Father

My Sister

Acknowledgements

I think no one denies that there are always certain things that go beyond our control and management let alone the teaching-learning process. It is just a matter of realization and perception. Thanks to God, St. Arsema and St. Mariam for their guidance and support to overcome life challenges and their blessing in disguise.

This thesis would not have been possible without the guidance and the help of lecturers and students of Computer Science Department of Addis Ababa University (AAU), whose insightful comments and encouragement inspired me throughout the work.

I owe my deepest gratitude to my advisor Dr. Mesfin Kifle for his time and advice that makes turning points all the way through this study. He was really inspiring to proceed at the time of difficulties and he is easily approachable. I would like to thank Dr. Ayalew Belay for his constructive comment and questions at the time of proposal defense which I rethink of them time and again until the compilation of the thesis. Besides, I would like to thank Dr. Fekade Getahun for his concern and support whenever I met him.

I would also like to recognize Dr. Amina Souag, Dr. Camille Salinesi, Assistant Professor Raúl Mazo and Dr. Mohamad Gharib for their provision of resources on their work.

I am also heartily thankful to AAU Computer Science lecturer Ato Yoseph Berhanu and his students Nathnael Minyelshowa and Tsegaab Alemayehu for their technical support on interface development.

Finally, I would like to thank my families, friends, and staffs whose silent support led me to complete my thesis. A special note of thanks goes to my friends Biniam Tibebu from Linguistic Department, and Biruk Abel from Computer Science Department of AAU for reviewing the thesis.

Table of Contents

List of Tables

List of Figures

List of Algorithms

Acronyms

EHR	Electronic Health Record
ePHRs	Electronic Personal Health Record Systems
FIPP	Fair Information Practice Principle
NFR	Non-Functional Requirements
NIST	National Institute of Standards and Technology
OWL	Web Ontology Language
PbD	Privacy by Design
PET	Privacy-Enhancing Technology
PHR	Personal Health Record
PI	Personal Information
PIA	Privacy Impact Assessment
PII	Personally-Identifiable Information
PMRM	Privacy Management Reference Model and Methodology
PRE	Privacy Requirements Engineering
RE	Requirements Engineering
SQUARE	Security Quality Requirements Engineering
SQWRL	Semantic Query-Enhanced Web Rule Language
SWRL	Semantic Web Rule Language

Chapter One: Introduction

1.1 Background

Being the main, initial and important activity of requirement engineering (RE) [1, 2, 3], requirements elicitation is the process of seeking, uncovering, acquiring, understanding and elaborating requirements for the development of computer based systems[1, 2]. It is the base for writing the software requirements specification (SRS) [1, 2]. Determining quality requirement is essential to the success of software development project [1]. Without an accurate understanding of what the stakeholders really need, projects cannot develop what the stakeholders' desire [1, 4].

In line with this, researchers have been motivated to carry out research in different areas of RE, since its origin in 1990s. Requirements elicitation is the leading empirically researched core area [5]. Even if the elicitation problem is not new and has been approached many times over 25 years, it is still considered one of the most complex and challenging issues and is far away from being considered solved and many interesting issues still need to be addressed [6]. The interest in investigating it further is still on the rise [3, 5, 6, 7, 8].

One of the primary responsibilities of requirement engineer is the elicitation of functional requirements and non-functional requirements (NFRs). NFRs are most extensively researched emerging area [5, 9]. Unfortunately, NFR concerns are normally dealt at design and implementation level and this approach results in the failure of most of the systems [9]. The chances of software success can be maximized when dealing with NFRs at the requirements level starting from the early stages of software development [9, 10, 11]. However, eliciting NFRs is not an easy task. There is a lack of elicitation mechanism for NFRs and there is a lack of proper consensus and standardization regarding NFRs elicitation techniques. A major problem of NFRs is how to measure the NFRs and how to deploy them. Privacy, security and usability are the important features and quality attributes that are normally of highest priority and the absence/deficiency/inconsistency/ambiguity of these requirements results in a system where user satisfaction remains question mark [9].

Security requirements are investigated relatively extensively followed by usability requirements [5]. Commonly, privacy requirements are investigated as part of security

requirements. However, security researches do not address privacy specifically, but think of it as part of system security overlooking aspects related to privacy such as anonymity, pseydonymity, unlinkability, and unobservability. As such they do not offer specific techniques for identifying privacy issues [12, 13]. Besides, adapting framework from other requirements engineering domain such as accessibility is not a seamless task due to the unique behavior of privacy [14]. Accessibility is a technical concept [14], while privacy is a social-technical concept [15, 16, 17]. Privacy itself is multifaceted concept [12]. There are different notions of privacy as well as data protection principles [18]. Hence, researchers need to look into the emerging NFRs such as privacy, and regulatory requirements [5].

In line with this, privacy has become a first-class citizen in the realm of NFR [14]. There is an increasing need for a systematic approach for eliciting, modeling and analyzing privacy requirements from the very early stages from the beginning and throughout the software development process, instead of patching up a built system afterwards [9, 14]. Heralded by regulators, privacy by design (PbD) holds the promise to solve the digital world's privacy problems [16]. However, its' principles and other legal privacy principles are far from technology, and moving them into technical requirements is not an easy job [14]. Hence, privacy engineering as specialized systems engineering discipline has come to existence [17, 19]. While the term privacy engineering has been around since at least 2001, attention to it as a research topic increased dramatically after 2012 [20] and only in the past few years has it comes into common usage in the privacy professionals' community [17]. As to our knowledge, even an attempt to define privacy engineering has started since 2016 [19, 20].

1.2 Motivation

According to Davey and Parker [1] and Young [4] more than 70% of software/information system have been experiencing failure or serious challenge during development process. Up to 71% of system failure being attributed to poor requirements elicitation [1]. The cost of fixing requirement elicitation problems is higher than other sources of error. It is typically 45% of projects. There is as much as a 2000:1 cost savings from finding errors in the requirements stage versus in the maintenance stage of the life-cycle [1, 4]. Each late detection of possible privacy vulnerabilities has also been proven to be extremely costly and time consuming resulting in severe delays in project delivery [21].

Privacy perspectives and concerns have shifted substantially in the post-Snowden (authorized Central Intelligence Agency (CIA) Agent) landscape [22]. Since the Snowden revelations, efforts to apply privacy-by-architecture methods and techniques in digital infrastructure design have gained in prominence, for instance, considering data minimization to protect against Transport Layer Security (TLS) client fingerprinting. Privacy protection in this scenario is dominated by those with the greatest resources and political power [20]. In 2017 Norton reported that hackers have stolen $172 billion from 978 million consumers in 20 countries. Consumers globally reported an average loss of $142 per victim and nearly 24 hours dealing with the aftermath [23].

To protect and preserve privacy, governments have been exercising different initiatives. Legislative acts such as the Health Insurance Portability and Accountability (HIPAA) 1996, UK Data Protection Act, and Australian Data Privacy Act have been formed. According to HIPAA, if an entity discloses individually identifiable health information with intent to sell, transfer, or use for commercial advantage, personal gain, or malicious harm, shall be fined not more than $250,000, imprisoned not more than 10 years, or both [24].

Hence, a reason for supporting privacy in software is legal and regulatory compliance [25]. In order to ensure holistic approaches to privacy engineering, the privacy requirements of the users of a system might extend beyond the legal demands [19, 25]. These should be considered also in software design in order to increase users' trust in the system and gain user acceptance of the system [13, 19, 25]. Therefore, the main motivation of this research is to enable requirement engineers elicit quality privacy requirement to develop trust worthy, and legal and regulatory compliant software which in turn enables consumer, citizen, service provider, private companies and public administrations minimize or avoid cybercrime, financial losses, legislative penalties, losses of trust and economic penalties[13, 23, 26].

1.3 Statement of the Problem

A major challenge in the field of software engineering is to make users trust the software that they use in their everyday activities. Trusting software depends on various elements, one of which is the protection of user privacy [12]. Privacy requirements play a critical role in the personal information system that deal with human-subjects data [9]. Personal data is the core asset at the heart of many companies' business models today [16]. A global survey

found that 88% of people are worried about who has access to their data; over 80% expect governments to regulate privacy and impose penalties on companies that do not use data responsibly [16]. However, technology constantly changes. Data is like water: it flows and ripples in ways that are difficult to predict [16] and maintain its' privacy [27]. Privacy regulation is not easy [16].

There are immense challenges [16] to elicit, reason, model and analyze privacy requirements [28, 29], and integrate them into systems [16]. The main challenges are software engineers lack of legal knowledge and expertise which in turn requires privacy specialists' collaboration (which is costly) [26], lack of reusable knowledge [14, 28], and lack of systematic methodology [13, 16, 25].

Hence, many researchers have made a significant contribution in the privacy domain. However, there are few works for privacy requirements engineering methods as well as privacy requirements elicitation methods [21, 25]. The existing methods are LINDDUN [28], PriS [12] and SQUARE for privacy [26, 27]. Still, it is unclear precisely what privacy protection means. Even law makers and judges cannot easily articulate privacy violations, privacy harms, privacy interests, and privacy problems [30]. Besides, conceptually and methodologically, privacy is often confounded with security that creates difficulty for a software engineer to easily reason about different ways to achieve privacy [16, 29]. 'Elusive', 'fuzzy', 'vague', 'disconnected from technology', or 'aspirational' are some terms employed nowadays to refer to privacy and resulted in much confusion among designers and stakeholders, and has led in turn to wrong design decisions [14] and thus it is difficult to protect privacy[16]. We need to come to terms on what it is we want to protect [16]. We need to start distinguishing security from privacy to know what to address with what means [16, 29] and to improve understanding of how to apply systems engineering and risk management processes to addressing privacy concerns [19].

Engineers often miss a solid reference, vocabulary and/or corpus detailing which specific, technical requirements and standards they must abide by and a systematic methodology to follow [14, 16, 19, 20, 25, 31]. Hence, a well-defined privacy ontology that captures all privacy related concepts and aspects along with their interrelations, would constitute a great step forward in improving the quality of privacy-aware systems [31]. However, there are

few research works on privacy ontology for requirements engineering phase with their own purpose, scope and focus. As far as our knowledge goes, the only works we can cite here are the works of Gürses [18], Webster et al. [29], Gharib et al. [31], and Hecker [32]. Besides, ontology-based requirements elicitation method that have been done so far are not meant to treat the distinct notions of privacy.

Gharib et al. [31] have proposed core privacy ontology to be used by software/security engineers while dealing with privacy requirements. However, the ontology has missed concepts and relations that exist in two prominent work. Indeed the ontology also did not structure its concepts and relations on the basis of i* modeling method. Besides, it proposed that privacy risk should be mitigated by privacy goals rather than privacy requirements which is in contrast to our claim [14, 19]. Therefore, the aim of this thesis is to propose a relatively complete core privacy ontology on the basis of i* modeling method and an interactive privacy requirements elicitation method that uses the ontology while dealing with privacy requirements. Hence, the key research questions are:

- ✓ What are the concepts and relations that need to be presented in a core privacy ontology?
- ✓ How can the ontology be used by requirements engineers while eliciting privacy requirements?

1.4 Objectives

The general objective of the thesis is to propose ontology-based interactive privacy requirements elicitation method to help requirement engineers while eliciting privacy requirements in software development environment.

The following specific objectives are identified to realize the general objective:

- ✓ to review and understand relevant concepts, theories and methodologies related to the topic under investigation,
- ✓ to assess the capacity of existing ontology-based requirements elicitation methods,
- ✓ to identify the key concepts and relations for capturing privacy requirements,
- ✓ to model and represent core privacy ontology containing concepts and relationships that guide and support privacy requirements elicitation method,

- ✓ to propose an interactive requirements elicitation method that exploits concepts, relations and axioms of the ontology,
- ✓ to implement the proposed core privacy ontology,
- ✓ to automate the proposed method through an interactive environment (a tool),
- ✓ to evaluate the completeness, validity and usability of the proposed ontology, and
- ✓ to evaluate the usability of the method and the interactive environment.

1.5 Methods

The research is concerned with solving problems and directed towards the production of method related to privacy requirement elicitation. Thus, the type of the research is applied research. In addition, as proposed solution will be tested, and experimentation and evaluation will be done in computer laboratory, the research will be done using experimental research approach. In line with this, in order to accomplish the general and specific objectives of this study, the following methodologies will be applied.

Literature Review

Related literatures from different sources will be reviewed to understand relevant concepts, theories and methodologies related to the topic under investigation and how other researches come across privacy requirements elicitation problem. Since computer science is a dynamic field, the most current and relevant researches in the area will be reviewed to assess notable characteristics of different solution methods.

Data Collection Method

To collect data about the key concepts and relations for capturing privacy requirements, Systematic Literature Review (SLR) of privacy ontologies will be conducted.

Ontology Construction Methodology

METHONTOLOGY [33] will be the methodology we will adopt to construct the privacy ontology. The methodology supports building an ontology from scratch, allows us to incorporate knowledge in the form of other ontologies and allows us to design the ontology in an implementation independent way [32]. It contains six main steps: objective specification, scope specification, knowledge acquisition, conceptualization, implementation, and evaluation [33, 34].

Development Environment and Tools

Prototype will be developed using Protégé 5.5 Editor (with its' OWL, SWRL Editor, SQWRL, and Drools inference engine) and Java programing language. Protégé will be used to develop the core privacy ontology. It will be used since it is an extensible, platform-independent environment for creating, editing, viewing, checking constraints, and extracting ontologies and knowledge bases [34, 35]. Besides, it has an intuitive and easy-to-use graphical user interface, is popular among the research community, and is also highly scalable [35]. It has an easy learning curve, while providing great user support [32]. Java will be used to develop the application and presentation component of our tool, because Protégé-OWL API, and SWRL API are java library.

Evaluation Method

- ✓ Completeness of the ontology will be evaluated using alignment table drawn up, with the concepts of our ontology on one side, and concepts of privacy ontologies found in privacy requirements engineering literature on the other side.
- ✓ Validity of the ontology will be evaluated using competency questions.
- ✓ Usability of the ontology, the method and the interactive environment will be evaluated using controlled laboratory experiment.

1.6 Scope and Limitations

This research deals only with the requirement elicitation activity of the requirement engineering process. Requirement analysis and risk-analysis activities are beyond the scope of this work, even if they are somehow intertwined and overlap with requirement elicitation. The requirements elicitation method elicits requirement only from the proposed privacy ontology. It does not support elicit requirements directly from the user.

Privacy ontologies must accommodate various legal regulations, cultural and ethical guidelines, and standards created by various industries [15, 29]. However, because we lack expertise in the legal, cultural and ethical domain; some purely legal requirements would not map to decomposition of requirements into layer, as they do not introduce technical requirements [14]; and it is not likely that a single generic privacy legislation can be created to cover all personal information in the world [14, 26], we are restricted to provide a

semantic framework for this purpose [32]. Besides, constructing core ontology remains ambitious and was found to be complex. One single team's work is not large enough [34].

To sum up, due to shortage of time, complexity of core ontology and lack of legal expertise, this research focuses on reviewing and updating the core privacy ontology proposed by Gharib et al. [31]. Besides, it focuses on proposing interactive privacy requirements elicitation method, and implementing the ontology and the method.

1.7 Application of Results

The main contribution of the research is a core privacy ontology that will be used as reference and corpus and an interactive requirements elicitation method that will be applied as a guide by requirement engineers while eliciting privacy requirements in software development environment. A consensus on requirements could even impact regulations worldwide, by homogenizing them [14]. Hence, the requirements of the ontology can be used as an input to develop homogeneous privacy regulation. Besides, the findings of the research will enrich professional literature in the area of requirements engineering and elicitation, particularly in ontology-based privacy requirement elicitation. Finally, the results of research can also be used as a stepping stone for further research.

1.8 Organization of the Thesis

The rest of the thesis is organized as follows. Chapter Two deals with literature review. It presents relevant concepts, theories and methodologies about the topic under investigation. Chapter Three explains related works that have been done in the past to solve privacy requirements elicitation problem. Chapter Four presents the system design and architecture of the proposed solution in order to build ontology-based interactive privacy requirements elicitation method. It also describes components of the architecture and privacy requirements elicitation algorithm. Chapter Five presents the implementation and evaluation of the prototype. Tools and technologies that have been used to develop the prototype are stated and description about how each component of the architecture has been implemented is clearly presented in this chapter. Chapter Five also presents online PHR Company as case study, and discusses the evaluation and the results of the proposed solution. Finally, Chapter Six presents conclusion, contribution and future works.

Chapter Two: Literature Review

This chapter presents relevant concepts, theories, and methodologies related to privacy, privacy engineering and risk management, privacy requirements, privacy requirements engineering (PRE), privacy requirements elicitation, interactive requirements engineering (RE), ontology, ontology engineering, and interactive goal model analysis and algorithms. Hence, the chapter presents:

- ✓ the definition, relation, and distinction of privacy and security to provide operational definition and context;
- ✓ the impact of privacy violation and harmful activities - privacy problems;
- ✓ the background of privacy engineering as a discipline, and its definition and components to show the hotness of our topic and the context where it resides;
- ✓ new privacy engineering objectives and privacy risk model that has been introduced in privacy engineering as distinct from security engineering;
- ✓ privacy requirements and privacy threats, and the way they can be framed/organized; how context and roles describe the relation between privacy goals and privacy requirements;
- ✓ the challenges that arise from the notion of privacy, and the three prominent PRE approaches and future direction;
- ✓ the promising PRE methodology that has been adopted as our method, and its gap not yet covered by current privacy initiatives;
- ✓ the existing privacy requirements elicitation techniques/methods and their limitations, and the commonly observed privacy requirements elicitation problems;
- ✓ the role of machines in RE and the two components of interactive RE (dialogue system and concept model);
- ✓ the definition, components, kinds, and languages of ontology;
- ✓ methontology and on-to-knowledge as relatively complete ontology engineering methodology; and
- ✓ the nature of goal-oriented requirement engineering (GORE) framework and goal model analysis algorithms.

2.1 Privacy Vs Security: Operational Definition

The common misperception is that information security equates to privacy [17]. Recently, a significant body of work treats privacy as part of security. The treatment implies that privacy shares enough characteristics with security [25]. However, while security certainly plays a vital role in enhancing privacy, there is an important distinction between them [17]. Using "privacy" as a separate term presumes that privacy has a meaning and brings with it issues distinct from security [19].

From a practical perspective, privacy is not about ***secrecy or preventing*** organizations from collecting needed personal information as part of their role in providing goods or services to their customers [36]. Privacy is about the power to control information – maintaining personal control over the collection, use, and disclosure of one's personally identifiable information (PII) and the data subject exercising that control consistent with his or her own interests and values. The data subject must be at the heart of what drives design and operational decisions concerning PII [32, 36]. Users' privacy can also be defined as the right to determine when, how, to what extent information about them is communicated to others [12]. It is also minimizing future privacy risks by protecting data after it is no longer under a user's direct control [37].

There are two essential practices that best characterize the essence of data privacy – "purpose specification" which clearly identifies why an organization needs to collect personal information and "use limitation" which refers to only using the data collected for the primary purpose specified. If the data collected will be used for other secondary purposes, then the individual must be informed and allowed to consent to the additional uses of his or her personal data [36].

When we come to information security, it is about protecting and controlling information including PII. Encryption, identity and access management, firewalls, are all about controlling the access or flow of information within the organization or between the organization and outside entities. Privacy is the other side of the coin [17]. There are security issues unrelated to privacy (e.g., confidentiality of trade secrets), just as there are privacy issues unrelated to security [19]. Security concerns focus on unauthorized activity that causes a loss of confidentiality, integrity or availability of information. While, in

privacy, the processing of PII is planned and permissible (i.e. authorized), but it creates implications for individuals' privacy. So while some privacy concerns arise from unauthorized activity, privacy concerns also can arise from authorized processing of PII [19]. Privacy control goes beyond privileged activities. A violation of privacy occurs in either way can compromise the purpose of the information and the use of the information. The subject of that information should be the only one to make changes, either directly or indirectly with his or her consent [32].

For example, a smart meter in smart grid technology can collect, record, and distribute highly granular information about household electrical use in energy sector in a secured way can have substantial privacy concerns due to such information could be used by other parties, entities or individual to learn when a house was occupied and what appliance they were using [19]. Even actions taken to protect PII (security tools such as advanced camera and sensor technology) can have privacy implications by creating concerns about the degree to which information is revealed about individuals. These privacy concerns are unintended consequences or byproduct of the system as it processes information about individual. This implies that byproduct risk model is conceptually distinct from security risk model [19]. Besides, Non-repudiation, in contrast to security, is a threat for privacy [28].

Moreover, privacy cannot be understood independently from society. It is vivid that without society there would be no need for privacy. Privacy is the relief from a range of kinds of social friction [30]. It exists - or is lost – at the boundary line between the individual and others. This boundary is in a state of flux, depending on the context within which a person operates and the degree of value derived from interactions with other people [19]. Hence, there are both socio-cultural and technical aspects to privacy [17].

Thus, it should be clear that effective privacy management will not be achieved solely on the basis of security management [19]. Security is one of several means to ensure privacy [16]. Security in the context of privacy usually refers to the concept of safeguard that is put in place to achieve privacy preservation on different levels. Safeguard is any technique that prevents unauthorized entities from gaining access to personal information [32].

2.2 Privacy Problems and Systems

Privacy problems that can result from unauthorized access to PII are generally well-recognized. They include embarrassment or other emotional distress from the unauthorized release of information, economic loss from identity theft, or physical or psychological harm from stalking. However, problems from authorized processing may be less visible or not as well understood but they also result in real consequences [19]. Let alone in software engineering, understanding and identification of privacy, privacy violations and privacy problems have been a challenge in information privacy law. Due to conceptual confusion, courts and legislatures often fail to recognize privacy problems, and thus no balancing ever takes place between privacy and counter vailing interests [30].

Privacy problems involve harm to individuals [30]. Solove D.J. [30] has identified four basic groups of harmful activities/systems (information collection, information processing, information dissemination, and invasions) and each of these groups consist of different related subgroups making a total of 16 harmful activities. The privacy problems include discrimination (stigmatization and power imbalance), frustrations, loss of trust, systemic failures in democratic institutions (such as voting), and loss of self-determination (loss of autonomy, loss of liberty, exclusion) [19]. They also include dignitary harms (reputational injury, incivility, lack of respect, or causing emotional angst) and "architectural" problems (the power that others or the government have over individuals) [30]. The consequences of these experiences can impact quality of life at both personal and societal level. It is vital that engineers understand the issue and have the conceptual tools to build systems that minimize problems for individuals when processing their information [19].

2.3 Privacy Engineering and Risk Management

Building trustworthy system that fulfills and meets specific privacy requirements demands the development of consistent, repeatable and measurable privacy engineering process, approaches, terminology and guidance that complement existing enterprise risk management and systems engineering processes for translating widely recognized, high-level privacy principles – such as the FIPPs and the 7 foundational principles of Privacy by Design (PbD) – in to effective system privacy requirements [19]. Researchers and practitioners have made groundbreaking contributions to the emerging field but few of these efforts are invested in

systematizing or generalizing their approaches. Privacy engineering work requires embracing and needs to be cognizant of the plurality (the different meaning and view point of privacy), contextuality, and contestability (the availability of multiple concepts around which formal disputes exists) of privacy as social, political, economic, and legal concepts [20]. Hence, for holistic and integrative approach [17, 20], privacy engineering is built around intradisciplinary (such as software engineering, security engineering, human-computer interaction, and machine learning) and interdisciplinary (such as law, societal norms, ethics, and technology) fields. The field also borrows knowledge and know-how from privacy research and practice [20].

Privacy engineering is defined as a specialty discipline of systems engineering focused on achieving freedom from conditions that can create problems for individuals with unacceptable consequences that arise from the system as it processes PII. This definition has two parts: privacy engineering and risk Management. Privacy engineering (inherent in the clause "achieving freedom from conditions that can create problems for individuals") provides an independent frame of reference that have been lacking in the privacy field. The definition recognizes risk acceptance, because there is no system that can be perfectly free of privacy risk [19].

2.3.1 Privacy Engineering

FIPPs have served as universal privacy values and as common general framework for translating privacy objectives to law, policy, and technology [17]. They source from data protection laws and hence they are far from technical domain [14]. PbD evolved from early efforts to express FIPPs directly in the design and operation of ICTs, resulting in PETs. PbD foundational principles serve as an overarching framework for inserting privacy and data protection early, effectively and credibly into ICTs, organizational processes, networked architectures and, indeed, entire systems of governance and oversight [17]. However, yet they are rather abstract [14]. They skewed toward privacy-by-policy approaches and barely attend to privacy-by-interaction methodologies [20]. PbD provides the "what" to do – the principles, then privacy engineering provides the "how" to do it [17] to bridge the distance between privacy principles and their effective implementation in systems [19].

In line with this, system engineering has the potential to enable privacy to be included as a key attribute of trustworthy systems. To better support engineers in this effort, privacy engineering process has been organized around five basic components and introduces new conceptual tools. Two of the components have been used in the privacy domain: ***laws, regulations and the FIPPs*** that have been used to drive privacy requirements; and ***Privacy Impact Assessment (PIA)*** that have been used to describe the system assessment process (risk identified, controls implemented and how the system meets requirements). The rest three are supplemented by components typically used in the information security: ***a risk model*** to produce a risk assessment: ***system objectives*** (e.g., confidentiality, integrity, availability) to map and evaluate system capabilities in order to provide assurance that the system meets the requirements and addresses risk appropriately; and ***a risk management framework*** to provide a process for selecting and assessing controls to manage identified risks and meet the requirements [19].

To bridge the distance between privacy principles of laws, regulations and FIPPs and their effective implementation in systems and to address concerns particular to privacy which result from authorized processing of PII and that cannot be addressed by security objectives, privacy engineering objectives and privacy risk model have been introduced to supplement, not replace the FIPPs, security objectives and security risk model [19].

Privacy engineering can provide an outcome-oriented process to resolve communication gap among different teams. System engineers could use an organizing constructs (privacy engineering objectives) to help them characterize system properties associated with privacy and to map system capabilities and controls to provide evidence of the desired level of trustworthiness. They could enable system designers or engineers to focus on the types of capabilities the system needs in order to demonstrate implementation of an agency's privacy policies and system privacy requirements. In what follows, the three privacy engineering objectives that a system should exhibit each in some degree to be considered a system that can support an agency's privacy policies and that provide a degree of precision to encourage the implementation of measurable controls for managing privacy risks are presented [19].

1. **Predictability:** is enabling reliable assumptions by individuals, owners, and operators about PII and its processing by an information system. A reliable sense of what is

occurring with PII in a system is core to building stable and trusted relationships between systems and individuals, accountability for system compliance with organizational privacy policies and system privacy requirements, and transparency for how PII is managed and to support purpose specification and use limitation of FIPPs.

2. **Manageability:** is providing the capability for granular administration of PII including alteration, deletion, and selective disclosure. It is an important system property enabling several of the FIPPs: access and amendment; accountability; minimization; quality and integrity; and individual participation. Information with insufficient granularity limited to support identifying and correcting inaccurate information, disposing obsolete information, collecting or disposing only necessary information, and implementing and maintaining individuals' privacy preferences.
3. **Disassociability:** enabling the processing of PII or events without association to individuals or devices beyond the operational requirements of the system. It captures one of the elements of privacy-preserving systems – which the system actively protects or "blinds" an individual's identity or associated activities for exposure. It is most closely associated with capabilities that could be used to implement the minimization of FIPP. Unlike confidentiality, it recognizes that privacy risks can result from exposures even within an authorized perimeter. Adapting disassociability as an objective could raise awareness of the benefits of cryptographic techniques. Furthermore, taxonomy of anonymity, de-identification, unlinkability, unobservability, pseudonymity or others could potentially support more precise control mapping and risk mitigation.

In addition to exploring relevant theories of privacy and engineering, the field calls for the development of methods, techniques, and tools – including privacy requirements elicitation and architectural techniques. It also foresees the use of them in developing information systems which refers to not only the technical artifact but the greater socio-technical system [20]. The definitions of privacy engineering methods, techniques and tools are presented as follows:

- ***privacy-engineering methods*** are approaches for systematically capturing and addressing privacy issues during information system development, management, and maintenance;

- ***privacy-engineering techniques*** are procedures, possibly with a prescribed language or notation, to accomplish privacy-engineering tasks or activities; and
- ***Privacy-engineering tools*** are (automated) means that support privacy engineers during part of a privacy engineering process [20].

2.3.2 Risk Management

Risk management is a key process that enables agencies to achieve mission goals while minimizing adverse outcomes [19]. Risk is a key aspect of systems engineering [17, 19]. Privacy engineering, therefore, requires a sufficiently robust approach to risk. This includes characterizing systems and processes, addressing identified risks through risk management approaches involving the selection and application of risk controls including mitigation and acceptance, defining and using an effective privacy risk model [17] and assessing risk [19].

2.3.2.1 Risk Model

A risk model of any kind establishes conceptual scaffolding to support structured reasoning about risks in a particular domain as represented by threats, vulnerabilities, and impacts. Privacy is one such domain and analytical frameworks based upon traditional FIPPs represent one such risk model. Fortunately, recent research has produced bases for additional privacy risk models such as Solove's taxonomy of privacy problems [30], Nissenbaum's contextual integrity heuristic and Calo's objective/subjective framing of privacy harms. These works don't constitute complete privacy risk models in and of themselves, but they do provide grounded reference points around which more complete models may be built [17].

Hence, new privacy risk model has been introduced. The model adopts terminology more suited to the nature of privacy to be able to identify privacy risk as distinct from information security risk. A more information-rich factor for a privacy risk model is to identify the operation that a system is performing on PII, which could cause an adverse effect or a problem for individuals – in short, a problematic data action which replaces threats risk factor in the information security model. In the privacy risk model, the primary risk factors are expressed as the likelihood/extent to which systems and process (working on PII) are vulnerable to or exploited by problematic data actions as well as the likelihood of a problematic data actions (with in a context), and the impact of such problematic data actions

occurrence. This risk factor implies that there are three key privacy risk characteristics that could facilitate determination of the likelihood and impact of problematic data actions [19].

1. **Data actions:** are any system operations that process PII. Thus, a privacy risk assessment should be oriented around the identification of a system's discrete data actions, and subsequent determination of which of these data actions could be problematic [19].
2. **PII:** Identifiability can be defined as the degree to which data can be directly attributed to an individual [37]. It is vivid that PII is information that can be used to distinguish or trace an individual's identity. We should consider the impact of systems on human behavior and how it should be accounted [19].
3. **Context:** are the circumstances surrounding the system's processing of PII. It is the key characteristic in determining the likelihood of a data action becoming problematic [19].

2.3.2.2 Risk Assessment

Once an appropriate risk model is identified, there are several considerations for performing the actual analysis. There should be risk analysis tools such as privacy impact assessments (PIA) to robust the analysis. Quantifying privacy is also another issue to be considered in risk analysis. Risk is a function of harm and probability. Quantifying the impact of privacy harm is problematic as it is extremely dependent on the norms of the subject population as well as the availability of data upon which to base probabilities. For example, revealing a teen's pregnancy to her father may be thoughtless and embarrassing in one culture but deadly in another. We should also consider residual risks and trade-offs [17]. Finally, in a world of limited resources, an important function of a risk assessment is to prioritize risks to enable determinations about the appropriate response and to achieve an acceptable degree of residual risk and avoid unacceptable consequences [19].

2.4 Privacy Requirements Engineering

2.4.1 Privacy Requirements

A requirement is a feature that system/product/service must have or a constraint that they must satisfy to be accepted by the client. It is an important factor that defines what different stakeholders need and how system/product/service will fulfill these needs. Requirements are

classified into functional requirements, non-functional requirements, and constraints [3, 38]. Privacy requirements are NFR [17] that capture privacy goals and its associated measures for a system under development [29]. For most systems, privacy is ancillary to the primary purpose of the system. It may be required for compliance purposes, for customer trust, for risk management, or for ethical concerns but, in theory, the base system usually functions without privacy backed in [17].

Generally, privacy threats (which are the negation of the main privacy properties see Table 3.1) [28] and privacy requirements can be framed around current IT architecture which has three distinct technical domains: the user sphere, the recipient sphere and the joint sphere [37]. These technical domains can be mapped to or considered as the three usual roles: data subject, data controller and data processor [14]. Context and roles describe the relation between privacy goals and privacy requirements [12, 25]. Privacy goals such as anonymity are expressed towards a stakeholder who has a specific role. For example, a privacy goal could be the anonymity of the name of the user shall be preserved. The refinement of this goal into a privacy requirement could be the anonymity of the name of a user shall be preserved within the database of the system-to-be from the recognition of system administrators [25]. Table 2.1 (adopted from the works of Spiekermann and Cranor [37]) summarizes the privacy spheres and resultant privacy requirements in three layer privacy requirement framework.

Moreover, it is important to consider where user's personal information will be stored or passed by, as these are the crucial elements for building in privacy [28]. It is important for engineers to understand/discover how privacy breaches can occur as a result of data transfer, storage, and processing. It is equally important for them to understand user expectations with regard to privacy-friendliness of a system [37]. Hence, the threats and the requirements can be organized by stating user privacy expectations and behavior around information systems' privacy sensitive processes/activities: data transfer, data storage and data processing [25, 28, 37]. Thus, an information flow oriented model (Table 3.1) of software-based system can be used to guide requirement analysis and to provide broader coverage. Data flow diagram (DFD) can be used to represent designated high-level system description which has a specific context. DFD is proven to be sufficiently expressive in a number of

case studies. The DFD elements are data flow, data store, process, and entity. The entity can be data subject, data controller and data processor [28].

Table 2.1 Three layer privacy requirement and engineering issues

No.	Privacy Spheres	Description	Privacy requirements	
			Engineering Issues	**User Privacy Concerns**
1	User Sphere/ Data Subject	It is any technology the user utilizes to communicate with the software [25]. It encompasses a user's device where data is stored. User devices should be fully controllable by the people who own them. Data should not flow in and out of them without their owners being able to intervene.	• What data is ***transferred*** from the client to a data recipient? • Is the user explicitly involved in the ***transfer***? (Engineers must ensure a controlled transition of data) • Is the user aware of remote and/or local application ***storing*** data on his system? For example cookies on a user's system. • Is data storage transient or persistent?	• Unauthorized collection • Unauthorized execution • Exposure • Unwanted inflow of data (i.e. "the right to be let alone" may be expanded to include the right not to be addressed by or forced to view digital services.)
2	Recipient Sphere/ Data Controller	It is company-centric sphere of data control that involves backend infrastructure and data sharing networks. Data is stored in any data recipients: servers and databases of network providers, service providers or other parties with whom data recipient shares data.	• What data is being **shared** by the data recipient with other parties? • Can the user expect or anticipate a ***transfer*** of his data by the recipient? • Is personal data adequately secured? • Is data **storage** transient or persistent • Can the ***processing*** of personal data be foreseen by the user?	• Internal unauthorized use • External unauthorized use • Improper access • Errors • Reduced judgment • Combining data

No.	Privacy Spheres	Description	Privacy requirements	
			Engineering Issues	**User Privacy Concerns**
			• Are there secondary ***uses*** of data that may not be foreseen by the user? • Is there a way to minimize ***processing***? (e.g. by delegating some pre-processing to user sphere)	
3	Joint Sphere/ Data Processor	It encompasses companies that host peoples' data and provide additional services (e.g., e-mail). Data is stored in web service provider's servers and databases	• Is the user fully aware of how his data is ***used or processed*** and can he control this? • Is there a way that third parties protect the data they receive and do not **use** it for their own purposes?	• Exposure • Reduced Judgment • Improper access • Unauthorized secondary use

In accordance with the recent software system setting, privacy requirements could also be roughly categorized in to two: bounded /priori/ privacy requirements and unbounded /posteriori/ privacy requirements. In the former, privacy requirements can be established priori in bounded system where the privacy protection of the users was wholly the responsibility of such systems and their hosting organizations. This has been true in the past for traditional and relatively closed and static systems, like health care, student records, or stock maintenance. In the latter, a range of unanticipated privacy requirements emerge a posteriori when the system comes into contact with open, hyper-connected and potentially "unbounded" environment [22]. Privacy concerns can arise at any point outside the scope of authorization where PII is processed [19]. Hence, effective treatment of privacy requirements requires development of privacy requirements engineering/elicitation approaches, methods, techniques, and tools [20, 22] that fit with the inherent openness and fluidity of the environment through which user data and information flows [22].

2.4.2 Privacy Requirements Engineering (PRE) Problems

PRE process has the objective of finding a complete and consistent set of privacy requirements for the system-to-be, given domain assumptions and facts about the environment, such that if they were all satisfied then the privacy problem would be solved. However, these claims to completeness, consistency and requirements satisfaction cannot hold with respect to privacy requirements. This problem is not specific to privacy, but nevertheless is exacerbated due to the nature of privacy notions [18]. There are immense challenges to define privacy requirements [18, 29] and to integrate privacy into systems [16]. The challenges that arise due to the existence of various contextual, relational and non-absolute/vague notions of privacy are [18]:

1. **Legal Compliance:** PRE must be done with wide and deep understanding of legislative requirements, standards, and policies. Software engineers often do not have that knowledge and must therefore collaborate with a privacy specialist which is costly. Besides, it is always difficult to bridge the gap between legal language and computer language [26]. Stakeholders are also likely not to share the vocabulary of the experts [18]. Exactly how legislation applies to technology is not a trivial matter [14, 18, 29]. Hence, any method for engineering privacy requirements has to include procedures to capture legal constraints [18].
2. **Technology Advancement/Contrivability/ [16, 22, 27]:** This may not have any legal or technical counter-parts. This may require to contrive new definitions and categories of privacy concerns [18]. Ensuring user privacy in such settings is non-trivial [22].
3. **Lack of Universality or Standard:** There are no universally accepted notions of privacy that can be used to elicit privacy requirements devoid of social context [18]. There is lack of agreement on privacy basics within the community and organizations due to different meanings and viewpoints attached to privacy around the world [14, 29]. Even there is a difference in selection, definition, conceptualization, solutions, notions and terminology of privacy among existing privacy RE approaches [18, 20, 25] (this is a topic of substantial interest in the field [20]). This leads to:
    a. **Lack of Reusable Knowledge:** lack of common structured requirements corpus that can be used as a reference by IT practitioners [14],

b. **Lack of standardized privacy principle** [14], and
c. **Lack of agreed-upon methodology** [13, 16, 25]: There is also lack of methodology that extract privacy requirements from ontology [13]. Besides, system development life cycles and organizational engineering processes rarely leave room for privacy considerations [16].

4. **Lack of satisfiablity:** abstract privacy properties such as data integrity and confidentiality that act as shadows of privacy notions may not be satisfiable [18].
5. **Subjectivity:** privacy notions and the "satisfaction" of abstract privacy properties may be interpreted differently by those who use the system or those whose information is stored in the system [18].
6. **Environmental Factors:** the conditions of the environment in which a system exists my override the abstract privacy properties secured by the system [18].
7. **Counter-factuality**: privacy requirements may rely on counter-factual arguments about privacy or privacy breaches which require a deeper analysis of assumptions, facts, and even other counter-factuals [18].

The existing requirements engineering methods has limitations to address these challenges. Hence, we need a way to define privacy requirements without ignoring these aspects and/or at least recognizing them as problems when addressing privacy concerns and interests [18]. Software engineers would be helped by a systematic way of developing privacy requirements that is suitable for use in all software development environments [26].

2.4.3 Existing Privacy Requirements Engineering Approaches and Future Direction

Existing state-of-the-art in PRE and the wider privacy literature have stated that there are four classes of privacy RE approaches based on their treatment of privacy requirements pertaining to: Compliance – Privacy by Policy; Access control – Privacy by Architecture; Usability – Privacy by Interaction; and Verification – Technical Point Control [17, 22]. Most efforts that have been made over the past several decades have followed three prominent approaches: Privacy-by-Architecture /Privacy as confidentiality/, Privacy-by-Policy /Privacy as Control/ and Privacy-by-Interaction /Privacy as Practice/ [18, 20]. Privacy-by-Policy involves a "trust us" mentality to do the right thing for users, while privacy by architecture involves "trust the system" [17] and privacy is technically enforced

[37]. Privacy by Interaction focuses on sociotechnical designs that would improve privacy in social settings. This approach's techniques can help design teams create interactions respectful of social and ethical norms [20]. There is also Technical-Point-Controls approach that shall not be overlooked and which comes to existence to mitigate privacy risks that cannot be handled by Privacy-by-Architecture [17]. Privacy-by-Policy is a pragmatic approach and focuses on the implementation of the notice and choice principles of FIPPs [37]. It is the most common approach taken by organizations today [17, 37].

All the three prominent approaches fundamentally differ in what they consider to be privacy problems and solutions due to the interdisciplinary nature of privacy. This might be seen as a productive plurality in research. Gaps, vulnerabilities, and limitation of each approaches comes from their focus on certain discipline such as law, societal norms, ethics, and technology and their applications as if they are independent solution [20]. For example, only focusing on privacy requirements derived from data protection legislation - Privacy by Policy - are insufficient due to the shortcomings of data protection legislation within the countries where they apply, the impossibility of such universally applicable regulation to cover user and community concerns, failurity of legislation to follow technological changes, and difficulty to apply legislations to new technology [18]. Moreover, none of the existing proposals for PRE offer solutions to address the lack of universality and subjectivity with respect to notions of privacy [18]. Hence, RE methods that make use of theoretical frameworks provided by legal scholars such as a taxonomy of privacy [30] to elicit privacy requirements and model privacy expectations and implications are especially interesting and more appropriate for addressing (subjective) privacy requirements and RE concepts beyond legal and security concerns than other models [18].

Even if all the three approaches are applied in concert, some privacy concerns might fall out of scope [20]. The three fundamental weaknesses of the approaches are system centric, syntactical and attribute-driven. ***System-Centric*** refers to the issue that the treatment of privacy remains focused around one system and does not effectively tackle the complexity of privacy requirements in highly open and interconnected settings. ***Syntactical*** refers to the problem of "first hop" in personal data that do not address protection and/or use of the once shared data across subsequent communications or further sharing with third parties – "multi-hop". ***Attribute-Driven*** refers to limiting the disclosure of a subset of data attributes such as

PII and not information that can be inferred from other shared information. Therefore, the software engineering research in general and RE in particular shall progress sufficiently in addressing privacy requirements in potentially unbounded settings. The following are the three fundamental shift in perspectives [22].

1. **System-Centric to Cross-Domain Privacy Requirements:** Personal data no longer belongs to any single domain, application, administrative, organizational, geographical, or contextual boundary. They regularly cross a range of boundaries. And what is required is a shift in perspective - from that of system-centric and data-origin-centric view of privacy to addressing posteriori privacy requirements arising from the cross-domain nature of contemporary settings. There is a necessity for creating new or adapt existing privacy requirement models to facilitate the notion of "accountability" and "transparency" and to support the integration of privacy requirements that may emerge a posteriori [22]. Agencies may need to take a programmatically expansive view of the boundary of a system in order to fully assess privacy risk [19].
2. **Privacy Management to User Empowerment:** In cross-domain information flow, privacy management is quite impossible. Individual users (and not only organizations) are the potential managers, beneficiaries, and victims in terms of gains and losses from their privacy disclosure and protection. Privacy is highly contextual. What is required is a shift in perspective - from that of privacy management to empowering users with regards to their data and information [22]. The focus must be on the empowerment of the data subjects as it is in their vital interest to keep personal information disclosure to a limit [32]. This is important particularly in "newer" system, i.e. Online Social Networks (OSNs) where the end user is the major active player. Therefore, RE approaches should incorporate adaptive decision making techniques, preserve history of the data exchange, and review the utility and cost of each data sharing transaction [22].
3. **Shallow Privacy to Deep Privacy Requirements:** the existing approaches focus on an attribute-driven view of privacy rather than derived attribute. What is required is a shift in perspective - from that of a shallow view of privacy to a deeper view of privacy requirements that goes beyond basic attributes and considers derived attributes and synthetic data within the scope of privacy requirements. Therefore, there is a need for significant advancement in the existing RE methods to, at least, start incorporating

simple derived attributes, or that of probabilistically derived attributes though history of sharing and interaction. Addressing such issues can act as a stepping stone towards tackling the hard challenge of deep privacy [22]. There is a need for significant advancement in existing RE methods and privacy requirement ontology to address a "***deep***" view of privacy [18, 22].

2.4.4 Privacy Requirements Engineering Methodology

There has been lack of a systematic methodology to translate privacy principles into more prescriptive requirements, specifications, standards, best practices, and operational-performance criteria. This problem has already been tackled in accessibility which show structural similarities to privacy. Privacy and accessibility share many commonalities: they are categories of NFR; more specifically, they are quality-in-use attributes whose determination depends on the specific user and context of use. Besides, they involve complex and interdisciplinary issues: both impact users' rights and are thus contemplated by legislation [14].

However, accessibility is just a matter of human-computer interaction, while privacy is a socio-technical systems issue [14, 15]. Even if there is a major philosophical difference exists between privacy and accessibility, some of the current status of practice in accessibility requirements can be adopted to PRE, analyzing as well the gaps not yet covered by current privacy initiatives [14]. In what follows are the activities that should be included in PRE methodology and their respective gap [14]. All of the activities are also implied and presented in PriS methodology [12, 25].

1. **Standardizing Privacy Principles:** There are already several initiatives which state the principles that must be met by privacy-compliant systems. These initiatives are indeed closer to the technical domain for example data minimization, use limitation, consent and choice. However, they are still abstract. Even there is not yet a consensus on a closed list of principles. OASIS privacy management reference model and methodology (PMRM) is in good position to assume standards accepted by stakeholders [14].
2. **Identifying Roles for Privacy:** a role is any stakeholders that can affect, be affected by, or perceive themselves to be affected by a decision or activity related to data

processing. The roles that usually take part are a data subject (who is the natural person whom the data is about), a data controller (which determines the purposes and means for processing the data), and a data processor (who processes data on behalf of the data controller). In addition, the roles may assume data protection authority, public authority, and agency [14]. Thus, privacy requirements can be assigned to and organized around one or more of these roles [12, 14, 25].

3. **Structuring Privacy Requirements into Layers:** 'high-level' or 'strategic' goals are further decomposed and specified to sub-goals and operational goals until 'low level' technical privacy requirements can be discovered and operationalized with in a given context [12, 14]. We must refine privacy goals into sub-goals until all the necessary actions and information are represented at the leaves levels of the graphs. These actions and information are called operationalizations [29]. Each layers refine the previous one into more detailed requirements: ***principles*** which define the foundations necessary for a system to be privacy-respectful, ***guidelines*** which provide the specific goals that authors should work toward in order to meet a principle, ***success criteria*** which define observable, testable, and measurable items (checkpoints) for each guideline, and ***techniques*** which define a reliable, implementable way to meet several success criteria. These days, there have been different attempts to refine privacy principles into more detailed points, which demonstrate i* framework fits within the current state of the technique, where gaps need to be nonetheless covered. All the existing techniques leave gaps in the decomposition process. However, a comprehensive set of requirements could be derived from all these existing techniques, and then be structured according to i* framework. I* operationalizes privacy requirements, moving from abstract requirements (soft goals), to concrete requirements (goals) and design solutions which operationalize these requirements (tasks). This approach also allows conjugating different definitions of privacy, from the perspectives of different agents playing different roles, so it seems quite promising to use the i* framework [14].
4. **Providing Technique Sources/Catalogue:** After privacy goals are clearly defined, we must identify how to reach them [16]. Providing an evolving catalogue of privacy patterns or techniques which instruct/guide engineers on the specific ways to meet

privacy requirements is important. PET groups disparate solutions, which are described at different levels of abstraction and focus on meeting different privacy principles. They do not help to refine the requirements but to meet them once they are stated. The PriS methodology [12] has tried to resolve this limitation of PET [14]. On a different perspective, the OASIs PMRM, ISO/IEC 29101, and the NIST privacy controls define a set of privacy services that every system should include as part of their privacy architecture. However, there is neither guidance on the requirements to be met, nor how the services must be used to fulfill them. The basic challenge of this topic is identifying, defining, and refining the privacy requirements that are used to select appropriate PETs or privacy services. Thus, there should be a well-developed common requirement corpus [14] or core ontology [18] to serve this purpose. We should reconcile the different privacy requirement approaches to come up with a corpus [14].

5. **Defining Level of Conformance:** here, we have to define levels of relevance of privacy requirements at the success criteria layer, so that different levels of compliance with the standardized body of requirements can be required, targeted, claimed, and certified. The concept of different levels of privacy protection is already adopted by different legislations, which usually attach it to different degrees of sensitivity of personal information: different types of personal information require different measures. These different measures would be mapped to different subsets of success criteria, each assigned to a respective level of conformance [14].

2.5 Privacy Requirements Elicitation

The enduring confidence of individuals, businesses, and regulators in organizations' data handling practices is a function of their ability to express core privacy commitments and requirements, which also promote efficiencies, innovation, and competitive advantages [17]. Privacy must be on engineers' requirements radar from the start of a new IT project. It needs to enter the system development life cycle at such an early point that architectural decisions around data processing, transfer, and storage can still be made [16]. However, there are many challenges in their identification [18, 29] as discussed in Section 2.4.2. Privacy requirements may be difficult to quantify and specify [29] and it is not mathematically

precise [32]. The state of the art lacks systematic methodologies and reusable knowledge or classification of measurable aspects of privacy to reason, model and analyze privacy threats, elicit privacy requirements, and instantiate privacy enhancing countermeasures, accordingly [28, 29]. Even, the existing approaches differ in notions and terminology. Hence, aligning software to meet privacy requirements is a challenging task [25].

The problem of eliciting privacy concerns is comparable to the problem of eliciting requirements [18]. The requirements engineering team determines what structured requirements elicitation techniques, approaches, and tools to use for requirements elicitation. The techniques provide detailed guidance on how to produce requirements for developers and stakeholders. Different literature stated the following techniques/methods [26, 27].

- Misuse cases (MUC)
- Soft Systems Methodology (SSM)
- Quality Function Deployment (QFD)
- Controlled Requirements Expression (CORE)
- Issue-based information systems (IBIS)
- Joint Application Development (JAD)
- Feature-oriented domain analysis (FODA)
- Critical discourse analysis (CDA)
- Accelerated Requirements Method (ARM)

Some have been developed specifically with security in mind, whereas others have been used for traditional requirements engineering and could potentially be extended to security and privacy requirements. There is no elicitation technique which covers and handles everything effectively in all cases. Moreover, the commonly observed problems of privacy requirements elicitation are as follows [26]:

- It is a time- and labor-intensive human activity.
- It needs privacy specialists/experts (which is costly).

Thus, we need a way or a systematic approach, technique and tool that should reduce the amount of time needed to develop privacy requirements and prevent requirements leaks even though privacy specialists are not involved in the elicitation [26]. Instead of dealing

with privacy specialist and stakeholder, requirement engineers shall dialogue with a machine to elicit privacy requirements.

2.6 Interactive Requirements Engineering

In interactive RE, machines can be responsible and can play the role in directing users to specify their requirements. Automatic and interactive RE has two components: appropriate user interface such as a dialogue system that is a supporting tool for human-machine interaction which is used to conduct the communication during requirement elicitation and a concept model such as ontology to represent knowledge about a domain or a task such as RE and human- machine interaction [39]

Dialogue is a conversation between two or more agents, be they human or machine: human-human dialogue and human-computer dialogue. The latter is involved in a dialogue system, a computer program that communicates with a human user in a natural way [40]. Dialogue systems are designed to communicate with human beings, extracting and analyzing information from their dialogue-based expressions, so as to accomplish certain tasks (e.g. exchanging information and providing services) in relatively natural manners. Most human communications in history are based on dialogues. Thus, a dialogue system provides a more natural, comfortable and convenient way for human-machine interaction [39].

Typically, a dialogue system consists of six components: input, fusion, dialogue manager, knowledge base, fission and output [39, 40]. The core components of a dialogue system are dialogue manager and knowledge base [39, 40]. A dialogue managers is the program which coordinates the activity of several subcomponents [40]. It is responsible for deciding what to do given the user's input and context [41]. The four main approaches to dialogue management are finite-state and frame-based, information state and the probabilistic based, plan based, and collaborative agent based. Finite state-based dialogue managers are the simplest dialogue managers. The dialogue structure is represented in the form of state transition network, and the dialogue control is system-driven and all the system's utterances are predetermined [39, 40]. As an extension of finite state, frame-based models simulate the approach of form filling, which allows some degree of flexibility. Therefore, a frame-based dialogue system shall be adopted to control the flow of dialogue [39].

Based on the kind of knowledge they contain, the different knowledge sources/bases and models commonly used in dialogue systems are dialogue model, dialogue history, domain model, task model, and user model [41]. Domain model and task model are the important one. First, domain model holds the domain knowledge that will be referred to in the conversations. It usually contains the structure of the domain and comprises a subset of the general world knowledge. Its simplification is ***conceptual model*** [39, 41]. Often, conceptual model alone is enough for the domain knowledge of the dialogue system [39]. Second, task model represents two kinds of tasks: user's task and system's task. A user task is non-linguistic and takes place in the real world and the model involves the user's goals and how they can be achieved. Model of system task describes how the system's communicative and other tasks should be carried out. It can consist of a hierarchical representation of subtasks which capture the task structure. The decision of dialogue manager is highly dependent on the status of the user task or subtask [41]. Therefore, the knowledge base of a dialogue system shall contains domain knowledge of requirement elicitation and the task knowledge for guiding user to elicit requirements [39].

In addition to the structure, by considering the source of information (in other word the direction of information flow) which determines the interaction, tasks of dialogue systems can be classified into four categories: slot-filling task, database search task, explanation task, and more complex tasks [39, 42]. In slot-filling task, the user has his goal and has the information about accomplishing the task. Slot-filling task shall be used because question will be proposed by the machine, and users will respond with their decisions. This implies that requirements engineers know what they are going to do, how it is going to be done, and why they are doing it. Therefore, a slot-based dialogue system shall be adopted, and requirements elicitation tasks shall be modeled as a group of slot-filling tasks. These tasks shall be performed specifically according to the knowledge related to RE, which is built in knowledge-base of the dialogue system [39].

2.7 Ontology and Ontology Engineering

Unlike traditional process-driven RE, knowledge-driven RE, as a novel RE paradigm, is conducted under the guidance of domain knowledge. Hence, information hidden in the domain can be retrieved without much help from domain experts. Knowledge engineering

methods, like propositional logic, predicate logic and other rule-based methods focus on how to solve the problem (such as RE problems) rather than the knowledge itself. So the resulting knowledge is often implicit and difficult to be maintained, shared or reused. On the contrary, the main concern of ontology is the contents of knowledge and approaches to accumulate it. It builds the foundation for common knowledge. There are many important benefits in applying ontology [39]. Such tools could potentially support more precise semantic processing on requirements descriptions [43]. Ontologies are known for being a good way to formalize knowledge for guiding requirements elicitation and hence to produce requirements specifications with better quality [43, 44]. RE based on predefined ontologies can make the job of RE much easier and faster [34]. So, currently, ontology is one of the most popular and powerful knowledge engineering methods widely applied in different applications [39].

The Artificial-intelligence literature contains many definitions of ontology; many of these contradict one another. For the purposes of this thesis an ontology is a formal explicit description of concepts in a domain of discourse (classes (sometimes called concepts)), properties of each concept describing various features and attributes of the concept (slots (sometimes called roles or properties)), and restrictions on slots (facets (sometimes called role restrictions)). Ontology together with a set of individual instances of classes constitutes a knowledge base. In practical terms, developing an ontology includes: defining classes in the ontology, arranging the classes in a taxonomic (subclass–superclass) hierarchy, defining slots and describing allowed values for these slots, and filling in the values for slots for the instances [45].

The main goal of the following sections is to discuss components of ontology, kinds of ontologies and language used, and ontology engineering methodology.

2.7.1 Ontology Components

Depending on the expressivity of an ontology [46], different ontology components can be defined such as classes/concepts, attributes[34]/slots/roles/properties, facets/role restrictions, instances, axioms, relations and terms [45, 46].

- **Classes/concepts:** Classes describe concepts in the domain. A class can have subclasses that represent concepts that are more specific than the superclass. For example, the class

of all wines can be divided into red, white, and rosé wines [45]. Classes are the main components of ontology and can be defined in different and (complementary) ways [46]:

- By their textual definitions: for example, the concept "person" is defined by the sentence "an individual human being",
- By a set of **properties**: for example, the concept "person" has the property "name", "birth date" and "address" ; note that a property can be reused for several concepts.
- By a logical definition composed of several logical formulae/axiom/: **axiom** is a combination of concepts and semantic relations. For example the concept "person" is defined by the formula "LivingEntity ∩ MovingEntity" [46]. In practice ontologies are usually composed of simple axioms [47]. Axioms are a set of assertions specifying what is true in the domain [48]. (In OWL) they are statements about classes or individuals [47]. They are used to connect classes and properties with some logical information about them [48]. They are categorized into different types such as symmetric and transitive and specified as complex objects that refer to concepts and relations [49].
- A concept can also be defined by the set of **instances** that belong to it. For example, "Martin Luther King" is an instance of the concept "person".

❖ **Terms:** classes, instances and properties are referenced by one or more symbols. Symbols are terms that humans can rapidly understand roughly by reading them.

❖ **Relations:** finally all these ontology components are connected through relations. ***Semantic relations*** link only concepts together: for example, the location relationship indicates that city concept is localized in a country concept. ***Instance relations*** connect only instances and instance relations are often instances of semantic relations, although it is not always the case. Some relations between instances can be contextual and cannot be generalized to all instances of their concept. An example of instance relation is that the city instance named Paris is localized in the country instance named France. All cities are localized in a country. ***A contextual instance relation*** can be that the person instance named "John Travolta" is localized in the city instance named "Paris" at the point in time 31 January 2010. The ***terminological relations*** express the relationships that terms can have: for example, the term "person" is synonym to the term "human being".

2.7.2 Kind of Ontology and Language Used

There are several types of ontologies. The word "ontology" can designate different computer science objects depending on the context. For example, an ontology can be: a ***thesaurus*** in the field of information retrieval or a ***model*** represented in OWL in the field of linked-data or a ***XML schema*** in the context of databases. It is important to distinguish the different forms of ontologies to clarify their content, their use and their goal. Several classifications of ontologies have been presented in the literature. This section focuses on two of these classifications [46].

First, according to the expressivity/the usage of the above ontology components and formality of the languages used (natural language, formal language), there are four kinds of ontologies [46]:

1. **Information Ontologies:** focus on concepts, instances and their relationships. Their goal is to propose an overview of a current project in order to express the state of the project. They are normally described by means of visual language such as Mind Map.
2. **Linguistic/Terminological Ontologies:** can be glossaries, dictionaries, controlled vocabularies, taxonomies, folksonomies, thesauri, or lexical databases. This type of ontology mainly focuses on terms and their relationships. The roles of linguistic ontologies are twofold: to present and define the vocabulary used, and to avoid ambiguity. Simple knowledge organization system (SKOS) and resource description framework (RDF) can be used to describe this type of ontologies.
3. **Software Ontologies:** provide conceptual schemata whose main focus is normally on data storage and data manipulation, and are used for software development activities, with the goal of guaranteeing data consistency. They mainly focus on properties, concepts, relations and terms. They are normally defined with conceptual modeling languages used in software and database engineering. The most well-known one is called unified modeling language (UML).
4. **Formal Ontologies:** require a clear semantics for the language used to define the concept, clear motivations for the adopted distinctions between concepts as well as strict rules about how to define concepts and relationships. This is obtained by using

formal logic (usually first order logic or Description logic) where the meaning of the concept is guaranteed by formal semantics. This ontology type is the only one that contains logical definition. The logical definition of a concept is composed of one or more logical formulae (or axiom). The purpose is not simply retrieval and storage of data but reasoning. Formal ontology does not focus on term and textual definition even if they could be defined in the ontology. Terms are only used as symbol in order to help user during the manipulation of logical formula. The ontologies are defined with different formal language such as description logics (DL), conceptual graphs (CG), first order logic (FOL), and web ontology language (OWL). Ontology development tools or ontology editors such as OntoEdit, Ontolingua, and Protégé (that provide a variety of features and use different languages and formalisms [35]) can be used to define the formal ontologies. Thanks to these formal definitions and rules, the inference engine can enters into a dialog with a user.

Second, ontologies can be classified based upon the scope of the objects described by the ontology or on the domain granularity. Here, we have six kinds of ontologies [46]:

1. **Local/Application Ontologies:** are specializations of domain ontologies where there could be no consensus or knowledge sharing. This type of ontology represents the particular model of a domain according to a single viewpoint of a user or a developer.
2. **Task ontologies:** contains knowledge to achieve a task, on the other hand the domain ontology describes the knowledge where the task is applied.
3. **Domain Ontologies:** is only applicable to a domain with a specific view point. That is to say that this viewpoint defines how a group of users conceptualizes and visualizes some specific phenomenon. This domain ontology could be linked to a specific application.
4. **Core Reference Ontologies:** is a standard used by different group of users. This type of ontology is linked to a domain (such as privacy) but it integrates different viewpoints related to specific group of users. The ontology is the result of the integration of several domain ontologies (such as health, bank, and education). It is often built to catch the central concepts and relations of the domain. It is appropriate

to develop the ontology using middle-out approaches where core concepts are identified and then generalized and specialized to complete the ontology.

5. **General Ontologies:** are not dedicated to a specific domain or fields. They contain general knowledge of a huge area.
6. **Foundational/Top Level/Upper Level Ontologies:** are generic ontologies applicable to various domains. They define basic notions like objects, relations, events, processes and so on. All consistent ontology has a foundational ontology. Domain or core reference ontologies based on the same foundational ontology can be more easily integrated.

2.7.3 Ontology Engineering Methodology

Several methodologies for ontology engineering are proposed to design ontologies. The most complete ones are Methontology [33] and On-to-knowledge. Nevertheless, this research area is still in development. All these methodologies are composed of several activities. The development process is not a linear process but a refinement one where each activity can be repeated several times. Among all the activities the most important are: Ontology specification (purpose, scope, and objective/goal specification), Knowledge Acquisition, Conceptualization, Formalization, Implementation, Evaluation, Maintenance, and Documentation [46].

2.8 Interactive Goal Model Analysis and Algorithm

In goal-oriented requirement engineering (GORE) framework, goal models are unique among models used to capture a system domain and requirements in that their structure naturally leads to an analysis of the achievement of objectives as well other important domain properties such as privacy, security, risk, or trust. We can gain further benefits (such as domain understanding, communication, scoping, model improvement, and requirement elicitation) from goal models by applying systematic analysis such as (forward/backward) satisfaction analysis [53]. Generally, GORE frameworks allow representation of one or more stakeholder needs (goals), which may be assigned to an agent, and which may have relationships to other goals, often describing how a goal can be achieved.

In line with this, there are several approaches that use the concept of goal such as KAOS, GBRAM, AGORA, NFR, i*, Tropos, and GRL [53]. Among these, i* modeling method is a widespread framework in the software engineering field that supports goal-oriented and agent-oriented modeling and reasoning method of socio-technical systems and organizations [52]. The i* builds upon the NFR framework [53, 54]. It has been used as a basis for agent-goal modeling in the GRL and Tropos frameworks. As such, it includes many existing goal model language concepts [54]. Besides, the i* is quite promising to structure privacy requirements into layers and to conjugate different definitions of privacy by different stakeholders [14, 29]. Thus, It is reasonable to adopt the latest and the consolidated version of i* modeling method (iStar 2.0) proposed by Franch et al. [52].

Privacy goals are interpreted by refining it into lower level goals and eventually linking them to implementable mechanisms and privacy requirements. Various goals and mechanisms may contribute to privacy in varying degrees. Each stakeholders' interpretation of privacy may lead to different goal refinements and mechanisms. The various interpretations of privacy can be collected and organized into a catalogue or an ontology for reference during requirements elicitation, analysis and design. The catalogue or the ontology allows representing different ways of achieving a goal. This facilitates choosing the one best suited to the problem being analyzed. Besides the operationalizations for privacy, possible correlations to other, maybe conflicting requirements are presented. This allows showing that one specific solution might achieve privacy and contribute positively or negatively to other requirements, for example security [29].

When we come to goal model analysis, we should focus on techniques which analyze a model after its creation, as opposed to techniques which direct the creation of models. A great variety of techniques/procedures for analyzing goal models in RE have been proposed in recent years [53]. Appropriate techniques can be selected based on the selection guidelines proposed by Horkoff and Yu [53] for problems at hand. Among the available techniques, i* (forward or backward) satisfaction analysis procedures/algorithms are recommended for domain containing a high degree of social interaction, or having many stakeholders with differing goals [53], and for early and high level RE [54]. Thus, we have adopted backward satisfaction analysis algorithm proposed by Horkoff and Yu [54] that can be used to find a set of acceptable options, given desired goal satisfaction levels [53] and

that can let us start from the desired top-level goals and work ‘‘backwards’’ along contribution paths to determine what combinations of choices will satisfy desired sets of objectives [54]. This algorithm can only find alternatives already in the model [53]. The identified option(s) should be considered as privacy requirement(s) that satisfy privacy goal and/or that mitigate threat.

2.9 Rule and Integrity Constraint

For two basic reasons, we should set rules to represent the invalid relationships between concepts in an ontology. First, the invalid relationships are used to denote the invalidity in the instantiated ontology. Second, some of them cannot be reasoned with available reasoners. Even if reasoners can deal with them, the invalid relationships will not be explicitly pointed out by the reasoners. Table 2.2 illustrates some of the group of SWRL rules that can be applied in an ontology [39].

Table 2.2 Rules and descriptions of an ontology

No.	**Rules**	**Description**
1	Decompose (x, x) ⇒ invalid (x, x)	*Decompose* relationship is non-reflective.
2	Rely (x, y) ∧ rely (y, z) ⇒ rely (x, z)	*Rely* relationship is transitive.
3	Decompose (x, y) ∧ Decompose (y, x) ⇒ invalid (x, y)	*Decompose* relationship is asymmetric.
4	Decompose (x, y) ∧ Decompose (y, z) ⇒ Decompose (x, z)	*Decompose* relationship is transitive.
5	Decompose (x, y) ∧ Decompose (z, y) ⇒ invalid (x, y) ∧ invalid (z, y)	*Decompose* relationship is inverse-functional.
6	Associate (x, y) ∧ Associate (z, y) ⇒ invalid (x, y) ∧ invalid (z, y)	*Associate* relationship is inverse-functional.

Besides, restrictions on the use of the constructs of the adopted i* model are stated textually as integrity constraints in Table 2.3 [52]. Restriction is a special kind of class description with which all individuals in that class will satisfy the restriction [48]. Concerning the integrity constraints we should remark [52]:

- Softgoals cannot be decomposed; instead, contributions are used to identify which elements influence the satisfaction of softgoals.
- An intentional element cannot be both a depender and being decomposed.
- When a dependency's depender is an intentional element, its type needs to be concordant with that of the dependum; the same happens for the dependum and the dependee when it is an intentional element.

Table 2.3 Integrity constraints over the i* language

	Actor links (ActorRelationship)
IC1	The ActorRelationship must connect actors of the same type
IC2	Cycles are not allowed regardless of the ActorRelationship
	Intentional Element Links (IELinks)
IC3	MeansEnd can only have tasks as from and goals as to
IC4	When a Decomposition has a goal as to, it can only have goals as from
IC5	When a Decomposition has a task as to, it can only have tasks or resources as from
IC6	It is not allowed Decomposition with a resource or a softgoal as a to
IC7	Contribution can only have softgoals as to
	Dependencies
IC8	The depender IE cannot be a to in any IELink
IC9	When the depender IE is a goal, the dependum can be a goal or a task
IC10	When the depender IE is a softgoal, the dependum can only be a softgoal
IC11	When the depender IE is a task, the dependum can be a task or a resource
IC12	When the depender IE is a resource, the dependum can only be a resource
IC13	When a dependum is a goal, the dependee IE can be a goal or a task
IC14	When a dependum is a softgoal, the dependee IE can be only a softgoal
IC15	When a dependum is a task, the dependee IE element can be a task or a resource
IC16	When a dependum is a resource, the dependee IE element can be only a resource

Chapter Three: Related Work

This chapter presents related works that have been done to solve privacy requirements elicitation problems, and discusses in detail how our solution advances the state of the art. This chapter is organized around the two important dimensions of requirements engineering: methodology and knowledge [28]. The first section presents the three privacy requirements elicitation methodologies that exist in the literature [21, 25] that have been done to address lack of methodology to support privacy requirements at early stage of software development. The second section presents works that have been done towards an ontology-based privacy requirements elicitation method to address the lack of reusable knowledge which can be used as a reference by IT practitioners [14].

3.1 Privacy Requirements Elicitation Method

3.1.1 PriS Methodology

It is a goal-oriented approach in order to elicit and integrate privacy requirements into the system design phase [12, 21, 25]. It provides a holistic approach from 'high level' goals to 'privacy-compliant' IT systems. The method focuses on bridging the gap between the design and the implementation phase. Privacy requirements such as unlinkability and unobservability are modeled as organizational privacy goals. Further privacy process patterns are used to identify system architectures, which support the privacy requirements. The method starts with a conceptual model which considers stakeholders, processes, goals, privacy requirements, and process patterns [25]. Enterprise knowledge development (EKD) framework is the base for conceptual model. Process patterns describe generic process models using activities and flows. Seven patters are defined which relate to the privacy goals [21]. The method prescribes four activities: eliciting privacy-related goals that are specific to the organization, analyzing the impact of privacy goals on organizational processes to refine goals and processes, modeling affected processes using privacy-process patterns, and identifying the technique(s) from PETs [21]. However, the method does not consider assets and risk assessment [21].

The PriS Method is demonstrated through an e-voting case study and career office system. The results indicate that the method can be used to effectively link organizational privacy

needs to alternative system implementations (PETs) that satisfy these needs and can guide designers to make informed decisions regarding the choice of the most suitable technological solution (PETs). PriS's application also showed that there are a great number of repetitive tasks. Besides, the PriS way-of working is formally defined to enable the development of automated tools for assisting its application. Thus, there is a need for development of tool that will automatically identify the impact of privacy goal in the goal-process structure [12].

3.1.2 SQUARE for Privacy

Bijwe and Mead [27] adapted Security Quality Requirements Engineering (SQUARE) methodology (which was their previous work) in order to support the elicitation of privacy requirements at the early stages of software development process [21]. The extended framework includes the same steps as the original SQUARE method. Actually, the whole nine steps of SQUARE methodology is modified to accommodate privacy, the Computer-Aided Privacy Requirements Elicitation Technique (PRET) tool has been developed independently as one kind of technique to be selected in step five of the methodology [26, 27]. First, the tool questions developer and stakeholder; second, developer and stakeholder discuss; third, they answer the questions; forth, the tool searches for the appropriate privacy requirements and displays them; finally, developer and stakeholder adapt them to the particular requirements [26]. The technique uses a database of privacy requirements based on privacy laws and regulations. The methodology is risk driven [21].

Although the steps may appear to be similar on the surface, different techniques come into play in the automated support provided by the SQUARE tool. The reliability of the PRET tool itself needs to be improved. The PRET database must be enhanced to cover other laws. Besides, the PRET is a generic tool, the requirements engineer needs to verify and tailor its output to specific needs through other techniques [27].

3.1.3 LINDDUN Methodology

The methodology is introduced as a privacy threat analysis framework in order to support the elicitation and fulfillment of privacy requirements [21, 28]. It is a threat driven approach. The primary contribution is the systematic methodology to model privacy specific threats. To achieve this, first of all, application Data Flow Diagram (DFD) is created based on the

high-level system description of the use case scenarios. Second, privacy threat types are identified by negating eight privacy properties (such as unlinkability, anonymity, pseudonymity, plausible deniability, undetectability and unobservability, confidentiality, content awareness, and policy and content compliance) [21, 28]. Third, the threats are mapped to DFD elements type (Data Flow (DF), Data Store (DS), Process (P) and Entity (E)) using Table 3.1 as a guide to determine the corresponding threats they are subject to (marked with x). Forth, an extensive catalogue of privacy-specific threat tree patterns are generated to detail privacy threats instances in a designated system. Fifth, the identified privacy threats are documented as misuse cases that presents a collection of threat scenarios in the system. Sixth, the method has a room for risk assessment to evaluate and prioritize the threats. Seventh, the privacy requirements of the system are elicited from the misuse case. Finally, appropriate PETs are selected [28]. However, it focus only on threat. Even if it has a room for risk assessment, privacy-specific risk assessment technique is out of its scope. It also does not consider assets [21].

There are insights concerning the methodology. Some privacy threats, in contrast to security, affect DFD elements pairwise (or sometimes group-wise). For instance, unlinkability implies the relation of two or more items of interest. Besides, privacy emphasizes relationships between instances of DFD elements or relationships between a DFD element and an entity, while security focuses on each individual DFD component in a more local way. The process element in the DFD is less important for privacy because privacy cares more about the relationships between entities and data. It is quite the opposite in the case of security [28].

Table 3.1 Privacy concerns with corresponding to LINDDUN components (privacy threats) and DFD Element type (E, DF, DS, P) susceptible to threats

No.	Privacy Properties	Privacy Threats	E	DF	DS	P
1	Unlinkability	**L**inkability	X	X	X	X
2	Anonymity & Pseudonymity	**I**dentifiability	X	X	X	X
3	Plausible deniability	**N**on-repudiation		X	X	X
4	Undetectability & Unobservability	**D**etectabiliity		X	X	X
5	Confidentiality	Information **D**isclosure		X	X	X
6	Content Awareness	Content **U**nawareness	X			
7	Policy & Content compliance	Policy & Content **N**oncompliance		X	X	X

3.2 Towards an Ontology-based Privacy Requirements Elicitation Method

There have been studies [2, 39, 43, 50, 51] undertaken that have proposed ontology-based reasoning methods to guide requirements elicitation. However, their interest was in functional requirements, was not in non-functional requirements and privacy requirements. Since this research has been started and undertaken, we are not aware of another work that have been done related to ontology-based privacy requirements elicitation methods (which is based on core ontology not limited to particular domain [34]) that has been proposed to solve problems particular to security and privacy requirements elicitation other than the works of Souag et al. [34], and Gharib et al. [31].

Souag et al. [34] in 2015 have proposed security ontology for security requirements elicitation. Their objectives are to provide a generic platform containing knowledge about the core concepts related to security and to develop interactive requirements elicitation environment to facilitate the use and reuse of the ontology. Their main contribution has been a core security ontology. The ontology was extracted from security requirements ontologies and security ontologies used for security RE using systematic and syntactic analysis. The authors have also developed the environment. The method is evaluated in maritime domain. The controlled experiment demonstrated that the ontology helps requirements engineers in eliciting security requirements by allowing them to exploit security-structured knowledge. This was made possible via the interactive environment that dynamically generates the necessary queries. Despite all this effort, the goal of constructing this kind of security ontologies remains ambitious and was found to be more complex than expected. One single team's work is not enough. A truly complete security ontology remains a utopian goal [34]. However, the focus of the study was security, rather than privacy [31]. The ontology was organized around three dimensions (Organization, Risk, and Treatment) which are considered to be modules [34] missing privacy dimension.

Gharib et al. [31] in 2017 have conducted a research on "Towards an Ontology for Privacy Requirements via a Systematic Literature Review" which superseded the old work by Webster et al. [29], Hecker [32] and Gürses [18]. For background information, we present key points about these works in the next paragraph.

Webster et al. [29] in 2005 presented a reusable knowledge base expressed in the form of a catalogue with strategies to satisfy privacy requirements. However, the catalogue is restricted to health care domain and two acts (Ontario and Canada). Besides, the structure used to store and retrieve information from this catalogue can pose some challenges [29]. Hecker [32] in 2009 created a generic privacy ontology as well as extensions to it (created by application domain expert) and the ability to extend the generic one to cater for other application domains. However, the ontology is too generic to apply it to a specific domain [46]. Besides, the author did not articulate any technique/method used to acquire privacy concepts. Gürses [18] in 2010 proposed privacy requirements engineering (RE) ontology. The ontology represents adopted and new RE concepts to overcome the limitations of the old RE model. However, the ontology did not address the requirements elicitation problem only it supports elicitation process [18]. Hence, the author used user surveys, interviews, ethnographic studies and content analysis as a source of privacy requirements elicitation. Besides, the author did not mention how the review of existing definitions of privacy notions was conducted. Therefore, these works have been superseded by the more recent work of Gharib et al. [31] in 2017.

The main purpose and focus of this work has been to identify the key concepts and relations in order to propose a novel ontology. Hence, 34 studies were selected and analyzed out of 240 relevant papers. The selected studies were investigated and 55 concepts and relations (27 organizational, 10 risk, 8 treatment, and 9 privacy) were identified. Furthermore, based on its importance for capturing privacy requirements and the frequency of its appearance in the selected studies, 38 key concepts and relations (17 organizational, 9 risk, 5 treatment, and 7 privacy) were selected that can be used for capturing privacy requirements in their social and organizational context. The ontology has been organized around four dimensions (organizational, risk, treatment and privacy dimensions).

The objectives of the research were considered to be achieved since the research questions posed have been answered. Moreover, the identified concepts/relations were used for proposing a privacy ontology to be used by software engineers while dealing with privacy requirements [31]. However, the research has three gaps and limitations. First, it is vivid that there is no complete system in the field of computer science, the proposed ontology have missed concepts and relations that exist in the two prominent works: the works of Brooks et

al. [19] and Drgon et al. [57]. Second, the ontology did not structure its concepts and relations on the basis of i* modeling method which is a widespread modeling and reasoning method of socio-technical systems and organizations [52], and which is quite promising to structure and to conjugate different definitions of privacy by different stakeholders [14, 29]. Thus, we argue that the ontology should be developed on the basis of i* method. Third, the ontology proposed that privacy risk should be mitigated by privacy goal rather than privacy requirements. As we have discussed in Section 2.3 and 2.4.4, the purpose of privacy engineering and privacy requirements engineering have been translating widely recognized, high-level privacy principles or goals into effective and prescriptive privacy requirements [14, 19]. In reality, privacy risks are mitigated by low level and operational privacy requirements rather than high level and abstract privacy goals [14] and privacy principles [19]. Thus, we claim that privacy requirements should be used to mitigate privacy risk.

3.3 Summary

As we have already discussed, PriS [12] does not consider assets and risk assessment, LINDDUN [28] does not consider assets, and the SQUARE for privacy [27] requires developer and stakeholder discussion which is time consuming and costly. Thus, our method will consider assets and risk assessment, and also minimize the cost and time involved in engineers and stakeholders discussion as we assume our ontology will represent the privacy experts/specialists knowledge that often can't be possessed by both software engineers and stakeholder. Even if both SQUARE and LINDDUN recognize the importance of reusable knowledge in addition to the lack of methodology, they have used database and catalogue to represent privacy requirements and threats respectively. Therefore, our method will use ontology which is the most popular and powerful knowledge engineering methods widely applied in different applications [39]. Besides, PriS and the LINDDUN are manual even most of SQUARE's steps are manual except step five. Thus, our method will be automated.

When we come to ontology for privacy requirements elicitation, recognizing the focus of Souag et al. [34] has been on security, Gharib et al. [31] have conducted a research which superseded the old work by Gürses [18], Webster et al. [29], and Hecker [32] to address the distinct notion of privacy. However, the research has three gaps and limitations that should be filled by this thesis.

Chapter Four: Proposed Solution

This chapter discusses the overall design of Ontology-based Interactive Privacy Requirements Elicitation Method proposed in this thesis. First, the chapter define design goals. Then, it illustrates the general overview of the system architecture of a tool that automates the method. Finally, it describes the two main components of the method that are the main components of the tool as well: core privacy ontology and its' rules and integrity constraints; and an interactive privacy requirements elicitation method, the inputs, the expected outputs and the algorithm implemented to generate the output.

4.1 Design Goal

The fundamental goal of the method is to produce a complete and valid privacy requirements ontology (that represent privacy expert knowledge) that can be easily used by requirement engineers while dealing with privacy requirement elicitation. Thus, the design goals of the solution are:

- **Completeness:** the core privacy ontology shall cover and integrate key privacy concepts and relations that exist in the literature [34, 35].
- **Validity:** the ontology must give reliable output (answers) to questions [34, 35]. It must work or behave well in the way we expect.
- **Usability**: It refers to the extent to which a system, product or service can be used by specified users to achieve specified goals with effectiveness, efficiency and satisfaction in a specified context of use[1]. Thus, the ontology and the method shall be used for privacy requirement elicitation by requirement engineers [34, 35].

4.2 System Architecture

The method proposes to use a core ontology that embeds privacy specific knowledge. It relies on a collection of rules and algorithm that automatically extract relevant privacy requirements from the core privacy ontology. First, the requirement engineers dialogue with a tool through slot filling forms which are logically presented to the engineers. Second, the engineer's response will be passed onto the application. The application will convert the

[1] https://www.iso.org/obp/ui/#iso:std:iso:9241:-11:ed-2:v1:en

input into format that can be processed by machines and passed onto an inference engine. The engine then knows user's decision and it will consult the ontology knowledge base with the decision. After that, an output will be generated by the engine according to the decision and sent to the application. Finally, the application produces a textual privacy requirements specification as an output. Figure 4.1 shows the system architecture of the tool that automates the proposed method.

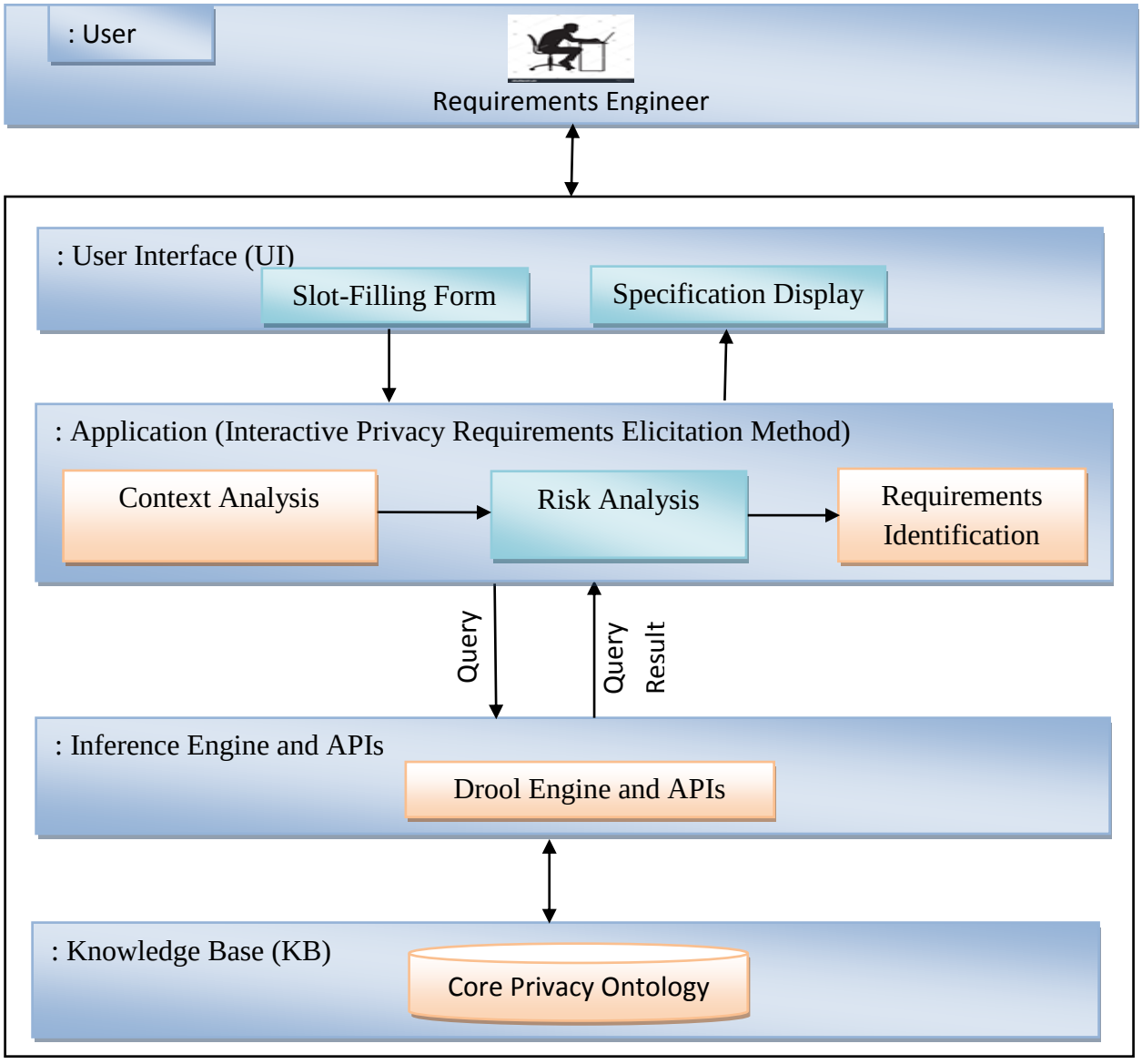


Figure 4.1 System architecture of the tool

We have adapted our architecture from the architecture proposed by Souag et al. [34, 35], and enhanced or changed four elements of the architecture. First, As privacy is socio-

technical concept and is highly dependent on context, we have enhanced the valuable asset identification activity and have come up with context analysis activity (which deals with dependencies and link among actors and their intentional element expressed in structured form and their analysis). Second, we have enhanced the requirments identification activity in a way that it can deal with three types of privacy requirments: privacy requirements that satisfy only privacy goals, privacy requirements that mitigate only threats, and privacy requirements that satisfy privacy goal and mitigate threats. Third, Jess rule engine has been changed with Drool rule engine implementation which allows rules to operate directly on java objects than Jess [60]. Forth, we have proposed core ontology specific to privacy.

The major components of the architecture are:

1. **Requirements Engineers:** are users who are responsible to define privacy requirements.
2. **User Interface (UI):** is the front end (the outer component) of the solution. It is used to accept input from the user and feed it into the system. After the user information is processed, the final result is displayed on the user interface. Textual privacy requirements specification is produced as an output.
3. **Application:** It is the component where part of the interactive privacy requirements elicitation method activities are automated and utilized: Context Analysis, Risk Analysis, and Requirements Identification. These analysises are guided by the core privacy ontology. Because, it automate the method, the application component is elaborated as an interactive privacy requirements elicitation method in Section 4.4.
4. **Inference Engine (IE) and API:** Inference Engine is a reasoning component. It is responsible for deciding what to do given the user's input and context. It searches the possible combination of rules from the ontology for providing feedback to the users. To be able to manipulate the ontology through the application and the engine, the layer of APIs (Application Programming Interfaces) is introduced.
5. **Knowledge Base (KB):** Here, core privacy ontology is the main component of the proposed solution that guides the elicitation process. It models key privacy concepts and relations, axioms and mechanisms. So that, it can support the development of PRE methods and tools, and it can guide requirement engineer while dealing with privacy requirements.

In the next sections, the two main components (the knowledge base and the interactive privacy requirements elicitation method) of the proposed solution are presented in detail.

4.3 Knowledge Base: Core Privacy Ontology

To address the first research question of the thesis, our method proposes a core privacy ontology that considers the descriptions of the most important concepts related to privacy requirements and the relationships among them. "Core" refers to the union of knowledge (high-level concepts, relationships, attributes) present in the literature [35]. The method also proposes that the ontology should be developed on the basis of i* modeling method to realize the benefits of goal-oriented requirement engineering (GORE), and to facilitate guidance and further goal analysis algorithm. Thus, we have adopted and transformed the consolidated version of i* modeling method proposed by Franch et al. [52] into our ontology using the transformation approach applied by Najera et al. [55] and Najera [56].

The ontology shall bind to the following design/development principles:

- It should make it possible to consolidate and capitalize the knowledge of the research community, by creating an entry point or a link for the various existing ontologies in the literature [35].
- It should make it possible to create a generic platform of different privacy concepts (threats, risks, requirements, etc.) to harmonize the semantics of existing privacy concepts in the literature [35].
- It should be represented and structured in a way that facilitates its use as guidelines for the requirements elicitation process [29].

In line with this, six main steps are used to develop the ontology which have been adopted from METHONTOLOGY methodology proposed by Fernández-López et al. [33] and applied by Souag et al. [34]. First, the objective behind the ontology construction must be defined in the beginning, including its intended uses, scenarios of use, end-users, etc. Second, the scope that stipulates the field covered by the ontology must be defined. Third, the knowledge acquisition step aims at gathering from different sources the knowledge needed for the ontology construction. Fourth, the knowledge is structured in a conceptual model that contains concepts and relationships between them. Fifth, implement the ontology using a software environment such as Protégé; this includes codifying the ontology in a

formal language (RDF or OWL/XML). Finally, the validation step guarantees that the resulting ontology corresponds to what it is supposed to represent. The details about how the first four steps have been applied are presented in this section and the last two steps are detailed in Section 5.3 and 5.5.2.

Step 1. Objective Specification:

The main objective of the target ontology is to provide a generic platform containing knowledge about the core concepts related to privacy (goals, threats, vulnerabilities, countermeasures, requirements, etc.). This ontology will be a support for the elicitation of privacy requirements and the development of PRE methods and tools; it will be in particular used in the context of our proposed method. The ontology will be a meta-view for the different privacy ontologies in the literature. It should harmonize the privacy terminology spread in these ontologies and help requirements engineers communicating together [34].

Step 2. Scope Specification:

The ontology covers the privacy domain in its high level aspects (threats and treatments) as well as its organizational ones (assets and actors) [34] and its privacy ones (privacy goals and privacy principles). See Step 3 below for all privacy concepts covered by the ontology.

Step 3. Knowledge Acquisition:

We have used the Systematic Literature Review (SLR) methodology applied by Gharib et al. [31] to acquire key concepts and relations that are necessary to develop core privacy ontology (even to undertake our thesis). Five electronic database resources (such as IEEE Xplore, ACM Digital Library, Springer, Google Scholar, and Citseerx) have been used to primarily extract data. The core privacy ontology proposed by Gharib et al. [31] includes core concept and relations that exist in most paper, thus (any variant or dialect of) any domain ontology can be linked to and used with the ontology, as long as they embed some expected knowledge. This implies that we can adopt and incorporate this ontology to our proposed ontology. Hence, we have used 35 key concepts and relations (16 organizational, 9 risk, 5 treatment, and 5 privacy) from the ontology. We have reviewed and updated it with two prominent work: with OASIS Privacy Management Reference Model and Methodology (PMRM) [57]; and with "An Introduction to Privacy Engineering and Risk Management in

Federal Systems" [19] that introduces privacy engineering as discipline and that come up with new privacy concepts.

Martín et al. [14] suggested that OASIS PMRM is in good position to assume standard privacy principles accepted by stakeholders. Hence, we acquired 14 Privacy Principles from the PMRM [57]. Besides, Brooks et al. [19] in 2017 presented new privacy engineering and risk management concepts for federal systems. These concepts establish the basis for a common vocabulary to facilitate better understanding and communication of privacy risk within federal systems, and the effective implementation of privacy principles. The work introduces two key components to support the application of privacy engineering and risk management: privacy engineering objectives and a privacy risk model for detail see Section 2.3.1 and 2.3.2 respectively. In order to consider and give special emphasis to privacy aspects and notions that cannot be treated by security engineering and risk model, we reused privacy objectives (such as predictability, manageability, disassociability) and security criterions (such as confidentiality, integrity and availability) relevant to privacy with the name privacy criterions. We also reused problematic data actions as type of threat in our ontology.

In line with this, Privacy goals [14, 29] and privacy principles [19, 57] are privacy requirements in their high level abstraction, it is not appropriate to use them directly to mitigate threats or operationalize privacy. Thus, we reorganized mainly the treatment and the privacy dimension proposed by Gharib et al. [31] to come up with new ontology structure. Table A.1 in the annex A presents the three studies and their corresponding privacy concepts and relations. The knowledge acquisition step and part of the conceptualization phase were performed manually relying essentially on tables to align the concepts and relations of the different sources with the help of document reference [34].

Step 4. Conceptualization:

Based on the outcomes of the knowledge acquisition step, concepts and relationships have been defined, organized and structured in a glossary, and then have been put together and represented in a conceptual model (Figure 4.2 which is adapted from the work of Souag et al. [34, 35]) in accordance with i* modeling method and its integrity constraints proposed by Franch et al. [52]. The model is easy to understand, independently of any implementation

language. The names of the concepts and the relationships have been chosen based on their importance for capturing privacy requirements and the frequency of their appearance in the selected studies (Table A.1 in the Annex A). In accordance with the work of Gharib et al. [31], our concepts and relations have been organized around four dimensions: Organization Dimension, Risk Dimension, Treatment Dimension, and Privacy Dimension. We come up with 20 organizational, 11 risk, 6 treatment, and 15 privacy concepts and relations making a total of 52 concepts and relationships.

Notice: Due to space and time we do not include goal trust and permission trust dependencies, and monitor concept in the organization dimension, all instances of processing, all privacy goals, and we only demonstrated the refinement of transparency goal into operational requirements in Figure 4.2.

1) Concepts of the Privacy Ontology

a) Organizational Dimension

As privacy is socio-technical problem, this dimension includes the organizational concepts of the system where the context and the circumstance surrounding privacy can be analyzed in terms of its agentive entities, their objectives and informational entities, their social dependencies and expectations concerning such dependencies [31].

Agentive Entities: represent the active entities of the system, three concepts along with two relations were selected: ***Actor*** represents an autonomous entity that aims at achieving their intentionality and goals by exercising their know-how, in collaboration with other actors [31, 52]. It cover two entities: A ***role*** represents an abstract characterization of an actor in terms of a set of behaviors and functionalities, and roles can be a specialization of one another [31] (Is-a: links actors of the same type for example a programmer role may be specialized into junior and senior programmer roles [52]); and an ***agent*** is an actor with concrete, physical manifestation [52], and it can ***play*** a role or more [31]. ***Plays*** links an agent to a role for example a particular person may play the role of project management [52].

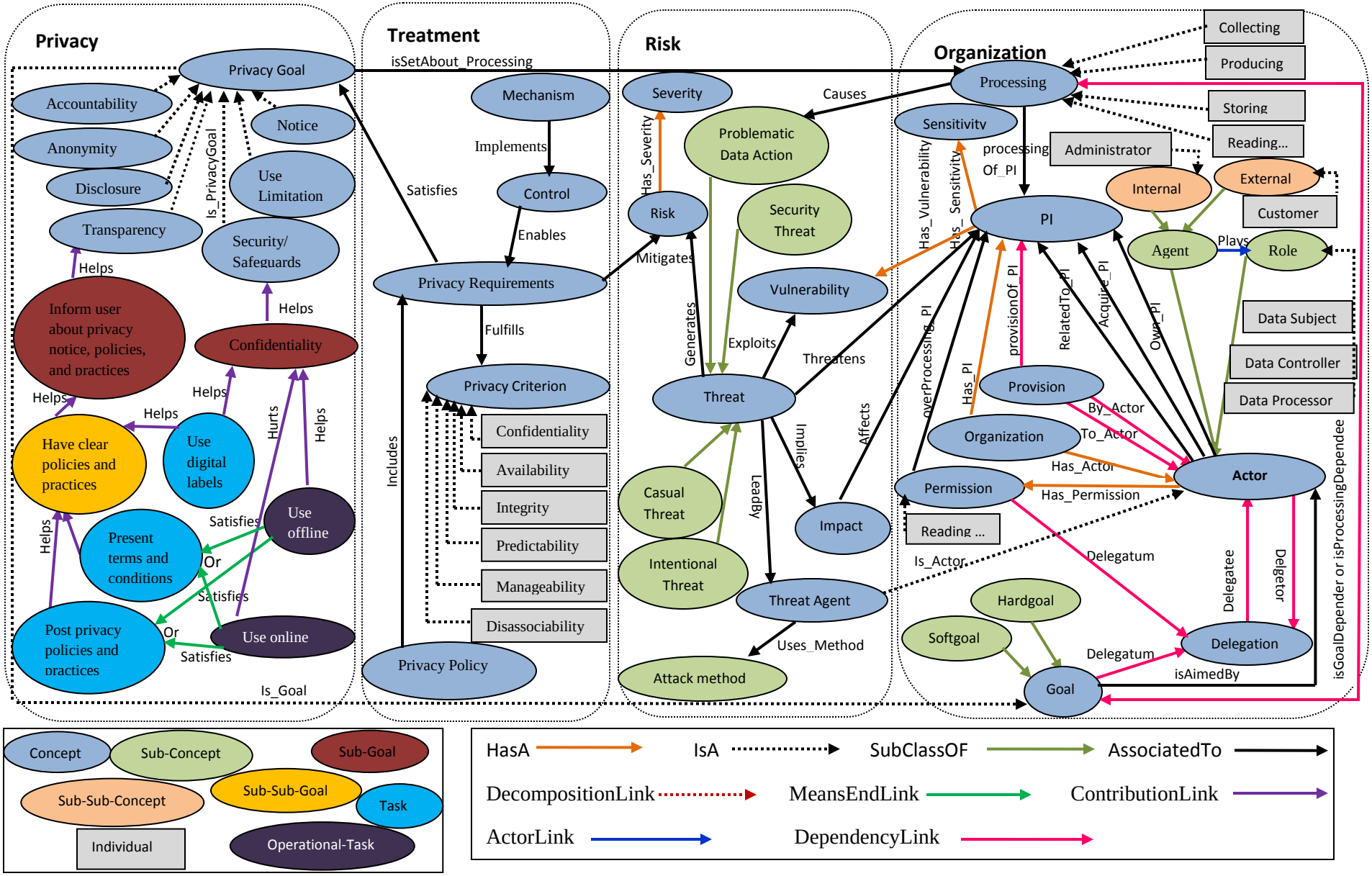


Figure 4.2 The core privacy ontology

Intentional Entities [31]/Elements [52]: Intentionality of actors is made explicit by identifying their intentional elements inside their boundary. The boundary delineates accurately what is under the actor's control. Inside the boundaries, four types of intentional elements can be declared: goals, softgoals, tasks and assets (resources) [52]. The behavior of actors is determined by the objectives they aim to achieve. Therefore, the goal concept is adopted to represent such objectives. A ***goal*** is a state of affairs that an actor intends to achieve, and it can be refined through and/or-decompositions of a root goal into finer sub-goals [31]. There are two kinds of goals: hardgoals and softgoals. For example "provide health care" is an organization's hardgoal. NFRs such as privacy are modeled as softgoals to be satisfied from the viewpoint of the various stakeholders [29]. Softgoals express a goal whose fulfilment is not clear-cut; instead, its satisfaction condition is subject of interpretation – subjectivity. Thus, we focus on softgoals. A ***task*** represents an activity whose execution is prescribed according to some established procedure. An ***asset*** stands for a physical or intentional entity that is produced or provided by the actor [52].

Intentional elements in actors are connected using several types of intentional element links. This way actors are able to express complex intentionality in a structure form, facilitating later analysis [52]. The interrelationships between intentions inside an actor are depicted via three types of links: Means-Ends links, Decomposition links, and Contribution links. Means-Ends links show the alternative tasks which can accomplish a goal. Decomposition links show the intentions which are necessary in order to accomplish a task. Contribution links show the effects of softgoals, goals, and tasks on softgoals [54]. These intentional elements that are linked with softgoal can help in (or avoid to) attain a softgoal [52]. Decomposition links allow decomposing complex elements into simpler ones of the same type with the only exception of resources which are allowed to appear in task decompositions. While means-end links can be viewed as a connection between the problem space (the end) and the solution space (the means), decomposition links do not change the space. Decompositions maybe AND-, OR- or XOR-decompositions [52]. ***And-decomposition*** implies that the achievement of a root requires the achievement of all its sub. While in ***Or-decomposition*** the achievement of any of its sub is enough [31].

Contribution can be positive (supporting) or negative (damaging), and can be an implication or just a connection, yielding four types of contributions links (make, help, break, and hurt)

[52]. ***Make/Brake*** indicates positive/negative contribution strong enough to satisfy/deny the parental softgoal, and ***Help/Hurt*** indicates positive/negative contribution not sufficient by itself to satisfy/deny the softgoal, respectively [29, 52, 54]. Although these distinctions are coarse grained, they are enough to help us decide whether we need further refinement and search for more specific softgoals and operationalizations or not. Contribution links enable NFRs decompositions up to a point where the operationalizations for a 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. Operationalizations are typically specified as tasks (privacy requirements in our case), each indicating a particular way of doing something [29].

Informational Entities: There are two main types of information: personal information (PI) and public information [31]. ***Personal Information (PI)*** (as an asset) is our concern. It is any data that describes some attribute of, or that is uniquely associated with, a natural person [57]. It can be composed of several parts, and the ***part-of*** concept is adopted to capture the relation between PI and its sub-parts [31]. We use the term PI as a proxy for other terminology, such a PII, personal data, non-public personal financial information, protected health information, sensitive personal information [57]. ***Sensitivity*** is the level of PI, (e.g. high, medium or low) which determines the different levels of privacy protection adopted by different legislations [14, 32] in different country or territory [32]. For example, PI could be much more strongly protected (by law) in countries within the European Union (EU), than in Middle-East (ME) countries. Therefore, the level of privacy the data subject receives would be high in the EU and lower in ME countries [32].

Information Type of Process: actors may process PI for achieving their goals [31]. ***Processing*** is operation or set of operations performed upon PI that can include, but is not limited to, the collection, retention/storage, logging, analysis, generation, transformation or merging, use, disclosure, transfer, and disposal of PI [19]. The ontology adopts ten types of processing relationships between goals and PI: Collecting, Producing, Storing, logging, Reading, Modifying, Analyzing, Disclosing, Transferring, and Disposing that indicate a goal (Goal_Depender) achievement depends on ***Processing PI*** [19, 31] (Processing_Dependee and implies resource dependency). As depender intentional element cannot be a to in any

intentional element link [52], we considered the relationship between privacy goals and processing as privacy goals are established or set about processing of PI.

Information Ownership & Permissions: ***own*** concept relates personal information to its legitimate owner, who has full control over its usage, which can be controlled depending on permissions. A ***permission*** is consent of a particular processing of a particular object in a system. The ontology considers ten different types of permissions (such as Collecting, Producing, Storing, logging, Reading, Modifying, Analyzing, Disclosing, Transferring, and/or Disposing permission over PI) [31].

Entity Interactions: actors may depend on each other for achieving their goals [31, 52], executing tasks, and accessing resources [52]. Whatever needs that are not inside the actors' boundary, need to be fulfilled in collaboration with other actors through dependencies. Not just actor links (such as is-a and plays), but also dependencies do connect actors. A dependency is a relationship between two actors: one of them, named depender, depends for the accomplishment of some internal intention (such as goals, softgoals, tasks, and resources (PI)) on a second actor, named dependee. The dependency is characterized by an intentional element (dependum) which represents the reason of dependency. There are four types of dependency or interaction (relationships): goal dependency: the dependee shall satisfy the goal; softgoal dependency: the dependee shall sufficiently satisfy the softgoal; task dependency: the depender requires a dependee to execute a task in a prescribed way; and resource dependency: the dependee has to make a resource available to the depender [52].

Therefore, the ontology adopts the four types of interactions. ***PI provision*** (implies resource dependency) indicates that an actor has the capability to deliver PI (resource) to another one. PI provision has one attribute that describes the provisioning type, which can be either confidential or non-confidential, where the first guarantee the confidentiality of the transmitted information while the last does not [31]. ***Delegation*** differentiates between actors that delegate to other actors permission and actors that delegate execution (and/or processing) on goals, tasks, and resources (if you do not do the job, I hurt). Delegation between two actors happens when delegater delegates to delegatee permission or execution on the delegatum [64]. ***Goal delegation*** (a goal dependency) indicates that one actor delegates the responsibility to achieve a goal to other actors. ***Permission delegation*** (a task

dependency) indicates that an actor delegates the permissions over a specific information to another actor [31].

Entity Social Trust: the notion of trust and distrust are adopted to capture the actors' expectations of one another concerning their delegations. ***Trust*** indicates the expectation of trustor that the trustee will behave as expected considering the trustum; while ***distrust*** indicates the expectation of trustor that the trustee will not behave as expected [31]. We can consider actor as a depender or a dependee; goal, permission, and PI as dependum; and goal delegation or goal trust, permission delegation or permission trust, and processing PI or PI provision as goal dependency, task dependency, and resource dependency respectively.

Monitor: To observe the operation of processes and to indicate when exception conditions occur [57], we rely on monitoring to compensate the lack of trust and distrust in a trustee concerning the trustum [31].

b) **Risk Dimension**

The risk dimension gathers concepts related to threats, vulnerabilities, attacks, and threat agents. ***Risk*** is a measure of the extent to which an entity or individual is *threatened* by a potential circumstance or event, and typically is a function of: (i) the adverse impact that would arise if the circumstance or event occurs; and (ii) the likelihood of occurrence [19]. ***Severity*** is the level of risk, e.g. high, medium or low [34]. A ***vulnerability*** is a weakness related to PI that can be exploited by a threat [31, 34] (e.g. weak password [34]). ***Threat*** is a potential incident that threaten PI by exploiting a vulnerability [31]. It is a violation of privacy criterion [34]. We classified threat into two: security threat and problematic data action. ***Security threat*** are a kind of threat that threaten security/safeguard implemented to protect privacy. ***Problematic Data Action*** is a data action that causes an adverse effect, or problem, for individuals. It is a type of threat specific to privacy. Data actions are system operations that process PI such as data collection, data retention, data use, and data transfer. Privacy risks can arise at any stage of PI processing from collection through disposal inside or outside the authorization boundary of the agency [19]. ***Impact*** is the magnitude of cost or harm from the threat [19]. Each threat implies an impact [34]. A threat can be either natural, accidental, or intentional [31, 34]. Therefore, the ontology differentiates between two types of threat, ***casual threat*** (natural or accidental) and ***Intentional threat*** [31]. ***Threat Agent*** is

the person (or program) who carries out the threat. The name 'threat agent' was chosen to cover both types of threat, either intentional (carried out by an attacker) or unintentional or casual (carried out by any person or program, not necessarily an attacker). Finally, ***Attack method*** refers to the different methods used by threat agent to accomplish their attacks, such as sniffing, spoofing, and social engineering [34].

c) **Treatment Dimension**

This dimension introduces countermeasure concepts to mitigate risks [31]. ***Privacy requirements*** define the protection capabilities provided by the system, the performance and behavioral characteristics exhibited by the system, and the evidence used to determine that the system privacy requirements have been satisfied. Each privacy requirement is expressed in a manner that makes verification possible via analysis, observation, test, inspection, measurement, or other defined and achievable means [19]. A requirement is some quality or performance demanded of an entity in accordance with certain fixed regulations, policies, controls or specified services, functions, mechanisms or architecture [57]. Privacy requirements should enable or fulfill privacy criterions [19], controls and mechanisms [57].

Privacy Criterion defines privacy properties such as predictability, manageability, disassociability, confidentiality, integrity and availability [19]. It can also be considered as a constraint on assets [34]. ***Privacy Mechanisms*** are the packaging and implementation of services and functions into manual or automated solutions [57]. They realize controls [34]. ***Privacy Controls*** are administrative, technical and physical safeguard employed within an organization or domain in order to protect and manage PI. They express how privacy policies must be satisfied in an operational setting [57] e.g., alarm or password [34]. ***Privacy policy*** are laws, regulations, contractual terms and conditions, or operational rules or guidance associated with the collection, use, transmission, trans-border flows, storage, retention or destruction of PI [57]. A privacy policy is a privacy statement that defines the permitted and/or forbidden actions to be carried out by actors of the system toward information [31]. It can be used as requirements [34]. Its implementation should satisfy/demonstrate privacy criterions [19].

d) Privacy Dimension

The central theme of this dimension is privacy goals and its associated privacy principles and their refinements. Privacy goals can be refined to the level of privacy requirements that enable or operationalize its achievement [14, 29]. ***Privacy goal*** defines an aim to counter threats and prevents harm to PI by satisfying privacy criterion concerning such information [31] or set about processing of PI. We can consider privacy principles as high level privacy goals [14, 29]. ***Privacy principles*** are foundational terms which represent expectations, or high level requirements, for protecting PI and privacy, and which are organized and defined in multiple laws and regulations, and in publications by audit and advocacy organizations, and in the work of standards organizations [57]. So, we derived the following 14 operational privacy principles stated by Drgon M. et al. [57] and Brooks S. et al. [19] as our privacy goals. The Operational Privacy Principles are composite definitions, intended to illustrate the operational and technical implications of commonly accepted Privacy Principles [57].

1. ***Accountability:*** ensure and demonstrate compliance with privacy policies and privacy principles to the various domain owners, stakeholders, regulators and data subjects by the privacy program, business processes and technical systems (e.g. produce audit report) [57]. Agencies should also clearly define the roles and responsibilities with respect to PI for all employees and contractors, and should provide appropriate training to all employees and contractors who have access to PI [19].
2. ***Notice:*** provide information, in the context of a specified use and in an open and transparent manner, regarding policies and practices exercised within a domain [57].
3. ***Consent and Choice:*** enable data subjects to agree to the collection and/or specific uses of some or all of their PI either through an opt-in affirmative process, opt-out, or implied [57].
4. ***Collection Limitation and Information Minimization:*** (exercised by the information processor) limit the personal information collected, processed, communicated and stored to the minimum necessary to achieve a stated purpose [57] or legally authorized purpose [19] and, when required, demonstrably collected by fair and lawful means [57].

5. ***Use Limitation:*** (exercised by the information processor), ensure that PI will not be used for purposes other than those specified and accepted by the data subject or provided by law, and not maintained longer than necessary for the stated purposes [57].
6. ***Disclosure:*** enable the transfer, provision of access to, use for new purposes, or release in any manner, of PI managed within a domain in accordance with notice and consent permissions and/or applicable laws, and make known the information processor's policies to external parties receiving the information [57].
7. ***Access and Amendment [19]:*** allow an adequately identified data subject to discover, correct or delete, PI managed within a privacy domain; provide notice of denial of access; options for challenging denial when specified; and "right to be forgotten" implementation [57].
8. ***Security/Safeguards:*** ensure the confidentiality, availability and integrity of PI collected, used, communicated, maintained, and stored; ensure specified PI will be de-identified and/or destroyed as required [57].
9. ***Information Quality:*** ensure information collected and used is adequate for purpose, relevant for purpose, accurate at time of use, and, where specified, kept up to date, corrected or destroyed [57].
10. ***Enforcement:*** ensure compliance with privacy policies, agreements and legal requirements and give data subjects a means of filing complaints of compliance violations and having them addressed, including recourse for violations of law, agreements and policies, with optional linkages to redress and sanctions. Such Functionality includes alerts, audits and security breach management [57].
11. ***Transparency [19]:*** (available to data subjects) allow access to an information processor's notice, (policies [19]) and practices relating to the management of their PI, and establish the existence, nature, and purpose of use of PI held about the data subject [57].
12. ***Anonymity:*** prevent data being collected or used in a manner that can identify a specific natural person [57].
13. ***Information Flow:*** enable the communication of PI across geo-political jurisdictions by private or public entities involved in governmental, economic, social or other

activities in accordance with privacy policies, agreements and legal requirements [57].

14. ***Sensitivity:*** provide special handling, processing, security treatment or other treatment of specified information, as defined by law, regulation or policy [57].

2) Relationships of the Privacy Ontology

Building relationships involves clarifying the relationships among the classes and defining the hierarchy. It is the process of adding axioms and restrictions to the ontology [48]. High-level relationships between those concepts were defined. They were categorized into two major kinds. i* relationships that are used to relate i* language core concepts and general relationships that are used to relate other concepts in the ontology. The i* relationship is further classified into three kinds (see the organizational dimension in the previous topic for their description). First, actor links such as Plays. Second, intentional element link which in turn is categorized into three kinds: means-end link such as satisfies, decomposition links such as And and Or, and contribution links such as Make, Help, Break, and Hurt. Third, dependency link such as Delegater, Trustor, Monitor, Delegatee, Trustee, and Monitoree. When we come to general relationships, they are IsA, HasA, SubClassOf and AssociatedTo (such as threatens, affects). The relationships between the concepts of the core privacy ontology can be organized and described around the four dimensions (see Figure 4.2).

Organizational Dimension: An organization has PI (Has_PI). An organization also has actors that it deals with (Has_Actor). The Actors can be Agent or Role (SubClassOf). An Agent in turn can be internal or external (SubClassOf). It plays one or more role (Plays_Role). An actor can relate to, own, and/or acquire PI (Relate_to_PI, Own_PI, and/or Acquire_PI). PI has sensitivity (Has_Sensitivity). A goal can be hardgoal or softgoal (SubClassOf). An actor (delegater) can delegate a goal (goal delegation) to another actor (delegatee) to get delegator's goal satisfied or achieved. An actor has permission (Has_Permission). Then, an actor can also delegate a permission (permission delegation) to another actors so that the delegatee can perform a certain processing tasks (such as collecting, producing, reading and/or modifying) over the PI on behalf of delegator. An actor can provide PI (ProvisionOf_PI= PI provision) *to* another actor, and an actor can be provided PI (ProvisionOf_PI) *by* an actor. Goal achievements depends on processing of PI. Thus we have goal depender and processing dependee relationships or links.

Risk Dimension: A PI is threatened by one or many threats (Threatens). Threat can be security threat or problematic data actions (SubClassOf). These threats exploit vulnerabilities in the PI (Exploits). The threat agent is an actor (Is_Actor) who leads an attack (LeadBy) and uses attack methods (Use_Method) to achieve an attack. A threat implies an impact (Implies), for example: "A denial of service attack implies a server downtime". The impact affects one or more PI (Affect). A threat can be intentional, or casual (SubClassOf). A threat generates a risk (Generate) with a certain level of severity (Has_Severity).

Treatment Dimension: Privacy requirements mitigate a risk (Mitigate) and satisfy (Satisfy) privacy goal expressed by an actor (ExpressedBy). Privacy requirements fulfill (Fulfills) one or more privacy criterion. For instance, the requirement "The application shall ensure that each user will be able to execute data actions for which he/she has permission at any time/every week" satisfies the security criteria Confidentiality and Availability. Control enables a privacy requirement (Enable). For example, the control "password" enables the requirement "The application shall ensure that each user will be able to execute data actions for which he/she has permission". Privacy mechanism implements privacy control (Implements). Privacy policy incorporates (Includes) privacy requirements.

Privacy Dimension: Privacy goal is a goal (Is_Goal) that is set about processing of PI (is_Set_About_Processing). Notice, anonymity, transparency, accountability, Consent and Choice, Collection Limitation and Information Minimization, Use Limitation, Disclosure, Access and Amendment, Security/Safeguards, Information Quality, Enforcement, Information Flow, or Sensitivity is a privacy goal (Is_PrivacyGoal). The softgoal, inform user about privacy notice, policies, and practices helps transparency (Help). The goal have clear policies and practices helps inform user about privacy notice, policies, and practices goal (Help). The goals post privacy policies and practices, present terms and conditions, use digital labels help to satisfy the goal inform user about privacy notice, policies, and practices. Confidentiality and security softgoals can help satisfy Use Digital Labels goal (Help). Post Privacy Policies and Practices, and Present Terms and Conditions can be operationalized or satisfied by the tasks use online or use offline (Satisfies). The two tasks (the operationalization) which can be considered as privacy requirements in turn cause direct impacts on the softgoals Confidentiality and Security. Use online or (Or) use offline tasks

can help or hurt Confidentiality (Help or Hurt). Thus, we can analyze these impacts. Some users might be confident enough to post privacy policies online. Others will prefer to use offline methods. When posting privacy policies and practices online or offline one must consider the security aspect of the information available [29].

3) Attributes and Axioms of the Core Privacy Ontology

In addition to concepts and relationships, an ontology contains axioms and attributes. Attributes describe each concept of the ontology. For instance, the concept agent has a name (its type is String) (see Table 4.1). The meta-information (term) such as agent and threat can be used to define axioms, constraints and rules that help to maintain the consistency of the proposed privacy requirement ontology [48]. Table 4.2 illustrates some axioms with their descriptions and the related concepts.

Table 4.1 Part of the table of attributes

Concept	**Attribute**	**Value type**
Agent	Name	String
Password	Minimum length	String

Table 4.2 Part of the table of axioms

Description	**Expression**	**Concepts**
A threat can be either intentional or Casual	∀ x: threat ⇒ Intentional Threat (x) V Casual Threat (x)	Threats
A agent can be either Internal Agent or External Agent	∀ x: agent ⇒ Internal Agent (x) V External Agent (x)	Agent

Figure 4.2 presents the core privacy ontology proposed in this thesis. It includes the four dimensions, with their corresponding concepts and relationships.

4.4 Interactive Privacy Requirements Elicitation Method

The proposed interactive (machine guided) privacy requirements elicitation method is inspired by and adopted from PRE methodology [14] which has been discussed in Section 2.4.4. The major enhancement is that every activity of the method is guided and/or

supported by core privacy ontology. This is the distinctive nature of our method that makes it different from another privacy requirements elicitation methods that exist in the literature. Besides, the originality of our method lies in the development of core privacy ontology (on the basis of i* modeling method) that can be used with any domain ontologies, as long as they embed some expected knowledge; and the defined rules and presented algorithm allows the method to automatically exhibit an appropriate ontological semantics to the requirements engineer (privacy goals, agents, objects, threats, privacy requirements…) [35] so that the engineer can interactively and iteratively analyzes and decides the inputs to the method and generate textual privacy requirement specification. Figure 4.3 shows an activity diagram that explains and illustrates how privacy requirements are generated using the proposed privacy requirements elicitation method.

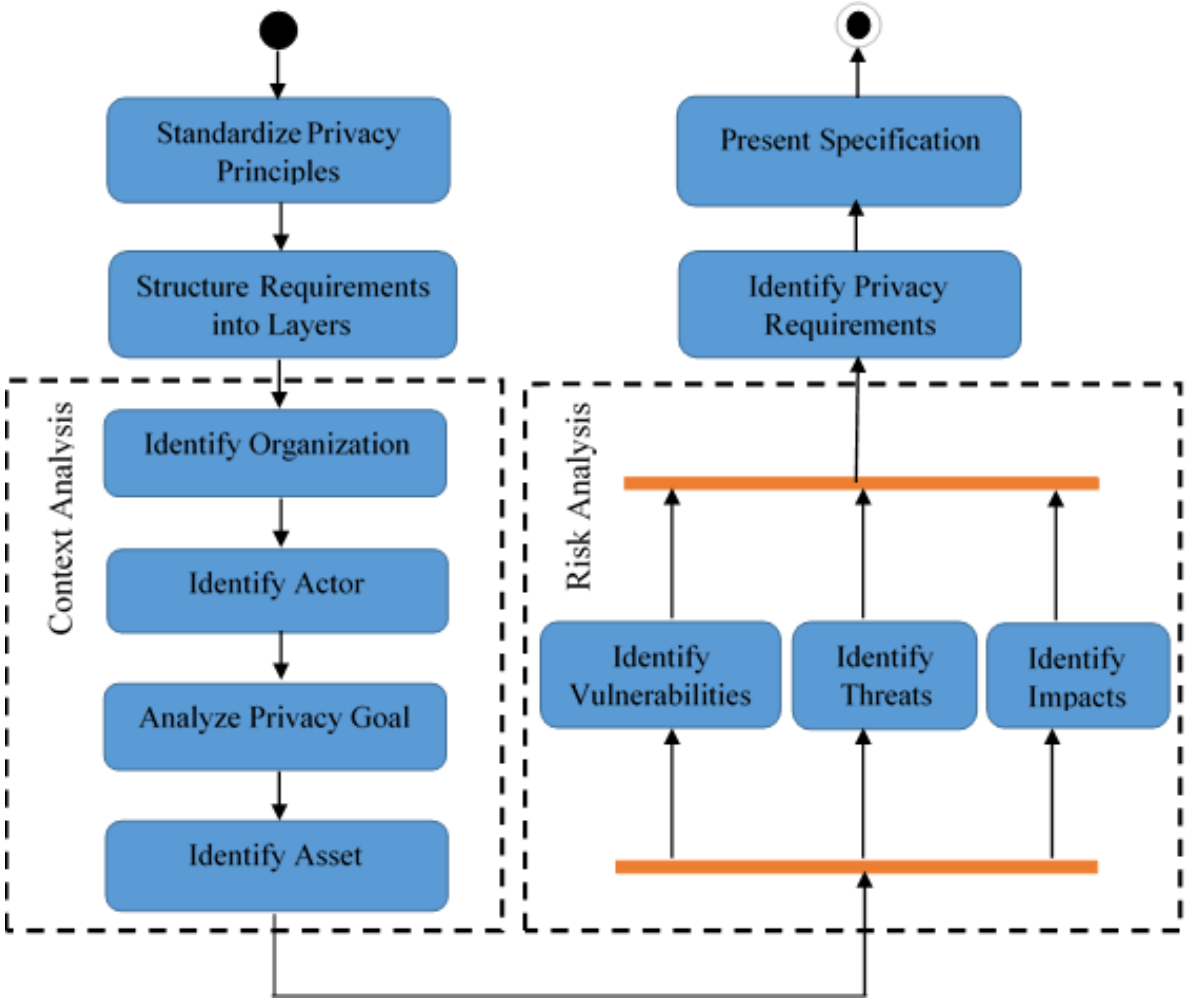


Figure 4.3 an activity diagram for privacy requirements elicitation method

The methods activities are:

1. **Standardize Privacy Principles:** standard principles accepted by stakeholders shall be captured from standardization effort. In our case, we adopted standard in our core privacy ontology from OASIS PMRM as suggested by Martín et al. [14].
2. **Structure Privacy Requirements into Layers:** the main objective of this activity is decomposing and refining high level privacy requirements such as privacy goals and privacy principles into operational or implementable (low level) privacy requirements. The structuring and refinement has already been done at the ontology level in accordance with i* modeling method as suggested by Martín et al. [14] and Webster et al. [29].
3. **Context Analysis:** analyze the circumstances surrounding the system's processing of PI such as the organization where the system resides, its agents and their roles, its assets and its different goals. To come up with the context, requirements engineers have to perform the following sub-activity:
    a. **Identify Organization:** organization has their own privacy notion, interpretation, and concerns. These can be specifically represented in domain ontology. Thus, in this activity, we should identify the organization for which we want to elicit privacy requirements.
    b. **Identify Actor:** The different agents in the organization play different kinds of role so that the organization serves its purpose and achieves its goals. Agent and their roles can represent different privacy responsibility and concerns. Agents and their usual roles such as data subject, data controller and data processor should be identified in this activity. Hence, privacy requirements can be assigned to and organized around one or more agents and their roles.
    c. **Analyze Privacy Goal**: The main objective of this sub-activity is analyzing and identifying the different stakeholders' interpretation of privacy and privacy goal refinements and mechanisms, and other privacy goal(s) that are influenced by the operationalization of a certain privacy goal(s). These interpretations have already been represented at the ontology level using the i* modeling method. Here, we have focused on analyzing goal model rather than creating a model. Algorithm 4.1 presents the adopted backward goal model evaluation/reasoning algorithm

which has been discussed in Section 2.8. The application of the algorithm and the i* modeling method is simplified through the benefits of ontology, rule, and reasoner. Actually, we have adopted two satisfaction values/labels/ such as satisfied and denied in terms of chosen and dropped. We can summarize the algorithm steps as follows. ***Initiation:*** the user decides on an analysis question and applies corresponding initial evaluation labels to the model. ***Propagation:*** the evaluation labels are propagated through the model. When an alternative is chosen, it is consider to be satisfied and its associated intentional elements will be selected for further judgement and choice until the option(s) (operational tasks) that satisfy the top level goal are identified. In the other way round, when an alternative is not chosen, it is considered to be denied and will be dropped together with its associated intentional elements. ***Softgoal resolution:*** because at each level of the goal model the user decides on the displayed alternative, there will not be contradicting values or labels [54].

Input:
Top Level Privacy Goal (TLPG)
Output:
Operational Task (OT). //The operational task(s) in this case are privacy requirements that are specified to satisfy privacy goals and that may not necessarily meant for mitigating threat. OT is requirements array. //
Begin
    Get the TLPG that the user chooses
    GetRelated goals that help the achievement of the TLPG and display them
    Get sub-goal that refine the upper goal and that the user chooses
    GetRelated sub-sub-goals that help the achievement of the sub-goal and display them
    Get sub-sub-goal that refine the sub-goal and that the user chooses
    GetRelated tasks that can satisfy the chosen goal and display them
    Get the task that the user chooses
    GetRelated operational tasks that operationalize the upper task and display them
End

Algorithm 4.1 Backward Privacy Goal Analysis Algorithm

d. **Identify Asset:** An actor has goal(s) to protect or attack an asset. An asset in our case is Personal Information (PI). So, we have to identify the PIs that shall be protected from the proposed ontology.

4. **Risk Analysis**: identify the impact of threat/problematic data action/ on PI. We have to identify threats, vulnerabilities, and impacts in accordance with the three key privacy risk characteristics such as data action, PI (such as degrees of sensitivity [14, 32]) , and context which has been discussed in Section 2.3.2.1.
5. **Privacy Requirements Identification:** Identify privacy requirements that mitigate the identified threats and/or that satisfy privacy goals. Besides, we should identify privacy goal(s) or other privacy requirements that are influenced by the elicited privacy requirements. Algorithm 4.2 presents privacy requirements classifier algorithm that classify privacy requirements as privacy requirements (PR) that satisfy privacy goals (GoalPR), as privacy requirements that mitigate threats (ThreatPR), and as privacy requirements that satisfy privacy goal and mitigate threats (BothPR). GoalPR, ThreatPR, and BothPR are requirements array.
6. **Specification Presentation:** finally, produce and present textual privacy requirements specification.

```
Input:
PR and OT. //PR and OT are requirements array. //
Output:
GoalPR, ThreatPR, and BothPR
Begin
   Initialization: GoalPR = 0, ThreatPR = 0, BothPR = 0
   Initialization: k  = 0,  j = 0
   Repeat until k <= OT. Length
            Repeat until j <= PR. Length
               If  OT[k] = PR[j]  then
                     BothPR.Add (PR[j])
               Else
                     GoalPR.Add (OT[k])
                     ThreatPR.Add (PR[j])
               End if
            End For
   End For
   Display GoalPR, ThreatPR, and BothPR
End
```

Algorithm 4.2 Privacy Requirements Classifier Algorithm

4.5 Privacy Requirements Elicitation Algorithm

Given organization, actors and their role, privacy goals, and assets, the engineer wishes to know what are the potential threats (and/or problematic data action)? The vulnerabilities? The attackers? The attack methods? The privacy requirements to mitigate the threats and/or to satisfy privacy goals? The engineer also wants to produce textual privacy requirements specification to have a better view and analysis of his PI, vulnerabilities and privacy requirements [35]. To overcome these issues, we have developed an algorithm which runs over the proposed ontology. The algorithm is inspired by security pattern search engine proposed by Guan et. al [48]. The main purpose of the algorithm is to simplify the exploration of privacy concepts, relations and requirements in the ontology in accordance with context. The basic nature of the algorithm is it takes users choice as an input and query the ontology for related instances, and returns results in accordance with each choice of user narrowing down query results to user choice at each stage of dialogue progressively, interactively, and iteratively. Algorithm B.1 in the Annex B presents the overall privacy requirements elicitation algorithm that contains Algorithm 4.1 and 4.2.

4.6 Summary

This Chapter has presented the system design and architecture of the proposed solution. It has described the enhancements or changes that have been made on the architecture. It has also described components of the architecture. It has mainly described the knowledge base and the interactive privacy requirements elicitation method components of the proposed solution and the algorithms necessary to elicit privacy requirements.

The development process of the proposed ontology has followed three design principles. It has also adopted the six main steps of METHONTOLOGY methodology. The acquired concepts and relations have been defined, organized and structured in a table, in a glossary, and in a conceptual model. These concepts and relations have been organized around four dimensions: Organization Dimension, Risk Dimension, Treatment Dimension, and Privacy Dimension. This chapter has also described six activities of the proposed method and their respective algorithms where necessary, and the privacy requirements elicitation algorithm as a whole.

Chapter Five: Implementation and Evaluation

To optimize usability of the ontology, our method was automated via an interactive environment (a tool) that the requirements engineer can use during privacy elicitation. The tool was implemented on Java. It offers two main functionalities to requirements engineers. First, it allows to perform privacy requirements elicitation which contains three sub-functions: context analysis, risk analysis and requirement identification. Second, the tool allows to produce natural language specification document. In this chapter, we explain tools and technologies used to implement our method and why we choose them, present Online PHR Company as a case study, implement and demonstrate the ontology, the method and the tool, present the evaluation, and discus the evaluation results.

5.1 Development Tool and Technology

We organize and present the tools and technologies used in accordance with the component of the proposed system architecture presented in Section 4.2.

A. Knowledge Base

The core privacy ontology was implemented using the Protégé editor, OWL (Web Ontology Language) and SWRL (Semantic Web Rule Language) Editor.

Ontologies via Protégé can be developed in a variety of formats including OWL, RDF(S), and XML Schema [35]. Different ontology languages provide different facilities. The most recent development in standard ontology languages is OWL from the World Wide Web Consortium (W3C) [58]. OWL is a computational logic-based language such that knowledge expressed in OWL can be exploited by computer programs[2]. OWL bases upon RDF, but it has a far greater expressiveness [32] and facilitates greater machine interpretability than that supported by XML, RDF, and RDF Schema (RDF-S) by providing additional vocabulary along with a formal semantics [35]. Thus, we used OWL 2 for the development of our ontology.

In addition to OWL-Protégé editing, SWRL Editor[3] was used to edit the axioms (rules) of the ontologies. SWRL extends the set of OWL axioms to include Horn-like rules [59]. It

[2] https://www.w3.org/OWL/
[3] https://github.com/protegeproject/swrlapi/wiki/SWRLEditor

allows the definition of rules to enrich the semantic of OWL ontologies. The defined rules allow to deduct and to add new relations between the created individuals in an ontology [35]. SWRL is an expressive OWL-based rule language that can be used to increase the amount of knowledge encoded in OWL ontologies. Semantically a SWRL rule can be considered as an additional type of OWL axiom [61].

B. Inference Engine and APIs

Protégé-OWL API, and SWRL API was used to be able to manipulate the ontologies through inference rule engine and then Java application.

The Protégé-OWL API is an open-source Java library for manipulating ontologies in OWL format. The API provides classes and methods to load and save OWL files, to query and manipulate OWL data models, and to perform reasoning based on SWRL rules. Furthermore, the API is optimized for the implementation of graphical user interfaces[4]. In our context, our tool invokes this API to manipulate the core privacy ontology [35].

Authoring and management of SWRL rule bases requires specialized tools that are not typically present in standard OWL development environments. A tool—called the SWRLAPI provides a rich development environment for working with SWRL rules. The SWRLAPI is built on the widely-used Protégé-OWL ontology toolkit [61]. In general, the SWRLAPI provides a collection of software components and Java-based APIs. These components and APIs can be used by developers who wish to work with OWL-based SWRL rules and SQWRL (Semantic Query-Enhanced Web Rule Language) queries in their applications[5]. To execute SWRL rules or SQWRL queries, you need a SWRLAPI-based rule engine implementation. Currently, a Drools-based SWRL rule engine implementation is provided. The SWRLAPI uses the OWLAPI to manage OWL ontologies. The SWRLAPI can be used to create SWRL rule engine and/or SWRL query engine using an ontology created by the OWLAPI, to create and execute a SWRL rule and/or SQWRL query using this engine, and then to process the results[6].

[4] https://protegewiki.stanford.edu/wiki/ProtegeOWL_API_Programmers_Guide
[5] https://github.com/protegeproject/swrlapi/wiki
[6] https://github.com/protegeproject/swrlapi

The Protege-based SWRLTab Plugin and the standalone SWRLTab automatically include Drools inference engine so it does not need to be installed separately[7]. In our context, this SWRLAPI-based rule engine implementation was used to create and execute SWRL rules and SQWRL queries on core privacy ontology during privacy requirements elicitation (context analysis, risk analysis and requirements identification) and to build rule-driven java application [60, 61].

To test and extract relevant knowledge from the core privacy ontology, SQWRL was used. SQWRL is a SWRL-based query language that provides SQL (Structured Query Language)-like operators for extracting information from OWL ontologies. Two mechanisms are provided by the SWRLAPI to execute SQWRL queries: (1) a Java API that provides a JDBC-like interface, called the SQWRL Query API, which can be used to execute queries and retrieve query results in Java applications, and (2) a graphical user interface called the SQWRL Query Tab that supports interactive querying and results display[8]. SQWRL define a selection operator allowing the retrieval of instances described in an ontology. This operator is noted sqwrl:select. It also offers aggregation functions (such as max, min, avg, count), set operators and grouping functions [35, 62]. For example, the following query replies to the question "What are the vulnerabilities (Vul_x) exploited by the threat Th_z?" [35]. More questions and SQWRL queries that are used to test and evaluate our ontology are presented in Annex C Table C.1.

$$Threat\_exploits\_Vulnerability\ (Th_z, ?Vul_x) \text{ -> } sqwrl:\ select\ (?Vul_x)$$

C. Application and Presentation

The application and presentation component of our tool has been developed using Java. IntelliJ IDEA IDE (Integrated Development Environment) was used to facilitate coding and debugging. IntelliJ IDEA is a cross-platform IDE that provides consistent experience on officially released 64-bit versions of Windows, Mac, and Linux operating systems. It requires a minimum of 2 GB of free RAM, 2.5 GB and another 1 GB for caches, and 1024x768 monitor resolution[9]. Thus, our tool should work with different current platform especially with Windows 8 and above. This component interacts with the end user during

[7] https://github.com/protegeproject/swrlapi-drools-engine/wiki/SWRLAPI-Drools-Engine
[8] https://github.com/protegeproject/swrlapi/wiki/SQWRL
[9] https://www.jetbrains.com/help/idea/installation-guide.html#

the elicitation process. Depending on end user choices, it displays the results of queries based on the other components. It also produce the specification (a word document) that contain the privacy requirements.

5.2 Case Study: Online PHR Company

There is a widespread agreement on the advantages of electronic personal health record systems (ePHRs). Several countries have set the implementation of national or regional integrated ePHRs as a goal to enhance the safety, quality, and delivery of healthcare. Integrated ePHRs is a transformative health technology [63]. One of transform happens because PHRs empower patients to participate in their own care in ways that were not able to do so far [64]. ePHRs are providing health consumers with greater access and control to their health records by shifting these records from being a health provider-centered Electronic Health Record (EHR), to a patient-centered electronic personal health record (ePHR) [63]. This is a shift in perspective from privacy management to user empowerment [22] which is one of the required progress in software engineering discussed in Section 2.4.3. As is the case with any technological innovation, ePHRs' privacy have been a key concern for healthcare consumers [63]. Figure 5.1 (adopted from the work of Roehrs et al. [65]) shows one of the three types of ePHRs architecture [63]. The model's purpose is to allow a unified view of health records which are distributed in several health organizations [65].

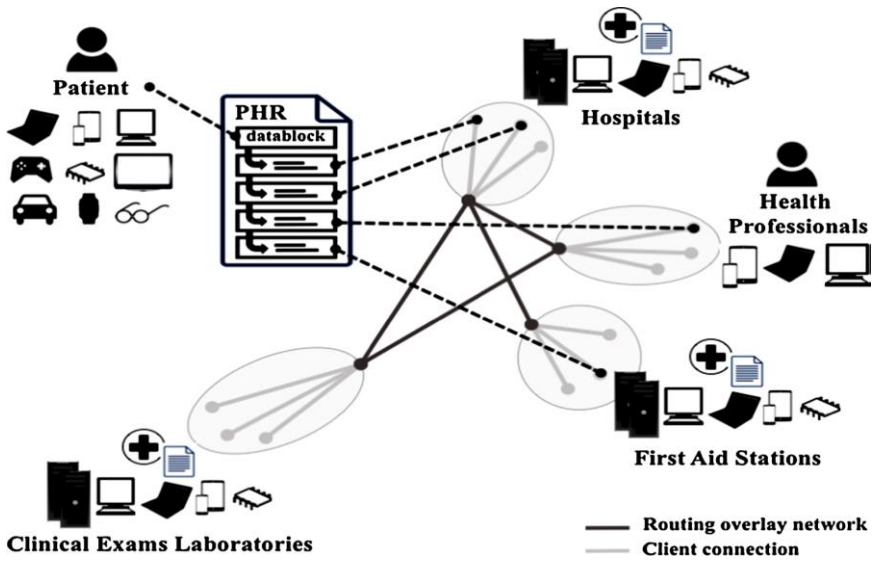
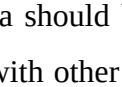
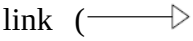


Figure 5.1 Integrated or unified ePHRs model

Technologies of ePHRs concerns about their connectivity to EHRs and patient privacy and security. Therefore, ePHRs and its data, affect not only the patient, but other stakeholders such as payers, providers, and clinicians. Implementation of ePHRs involves multiple stakeholders with different interpretations and expectations; more importantly it involves changes in the custody of data, patient privacy, and consent management. These stakeholders are related to each other in a social setting, where at the same time each stakeholder is autonomous with its own goals, motivations and intentions. Hence, ePHRs are complex and socio-technical systems. There are many sensitive issues in the health process with regards to PHRs that need to be investigated. In PHR analysis, we need to answer questions such as: Who is the provider of PHR? Who has access to the patient data and why? And how the system can empower the patient? And how can the patient privacy be managed? Therefore, thorough analysis of stakeholders' preferences, objectives, intentions, their dependency and trust relationship, and the policies (such as HIPAA [24]) that are involved are crucial for the success of ePHRs. This necessitates the use of advanced conceptual modelling and early RE methodologies. Without a systematic elicitation of PHR requirements and stakeholders expectations, it is hard to determine how any of the currently proposed technologies meet the needs of PHRs. Therefore, we expect that implementation of our method will help us to identify the organizational context of an online PHR Company along with the goals of actors and their social relationships and clarify the privacy requirements of the system to-be [64].

As an example, we assume that Alice wants to visit a neurologist for the pains she feels in her wrists. While she likes to share with her doctor all her health record, she prefers to avoid sharing attention deficit problem she had experienced during her pre-teenage. However, she doesn't like to entirely delete this part of data from her health record, because she has decided to counsel with a psychiatrist later this month to discuss her poor performance in last semester at the university. Her psychiatrist decides to seek counsel for Alice's case from one of his colleagues in a research institute (RI) specialized in ADHD (attention-deficit hyperactivity disorder). Alice agrees to share her Health data if only the relevant part will be transmitted, her data only be used for her treatment purpose. And the data will be removed from the RI repository as soon as her case is closed [64].

In this case, Alice, the neurologist, and the psychiatrist are the agents. In an abstract level they can be represented by actor such as Patient and Clinician. Usually, actors have specific goals that they like to fulfill. If they are not capable to fulfill them individually, they depend on other actors in the system to fulfill these goals. Alice visits two Clinicians, a Neurologist (C1) which is Dereje and a Psychiatrist (C2) which is Daniel. Figure 5.2 (adopted from the work of Samavi and Topaloglou [64]) represents this with two actor instantiation relationships of the Clinician role. Daniel not only depends on patient for her personal data, but he depends on Dereje for up-to-date personal health data of Alice. Alice stores all her health data on an online PHR company and she personally keeps it up-to-date. So, when she meets the psychiatrist, she has already entered previous data from her visit with Dereje. It seems that a stand-alone PHR can potentially play the role of an interoperable EHR [64]. The i* notation are: Actor (), Role (), Agent (), Depend () on, Goal (), Task (), Resource (), Softgoal(), Means-ends link (), Decomposition link (), and Contribution link (). The qualitative i* labels are satisficed (), denied (), partially satisficed (), partially denied (), unknown (), and conflict () [52, 64].

In line with this, next we will show ownership, trust, delegation, and monitor concepts of privacy. In the case of PHRs, a patient as the owner of his/her health data should be in a position to enforce his/her privacy in any instance of sharing information with other parties involved in the healthcare process. All privacy policies (such as HIPAA [24]), acknowledge the patient's rights to delegate, monitor or specify purpose when his/her data are shared with others. These policies need to be operationalized by the specific requirements in the system-to-be [64].

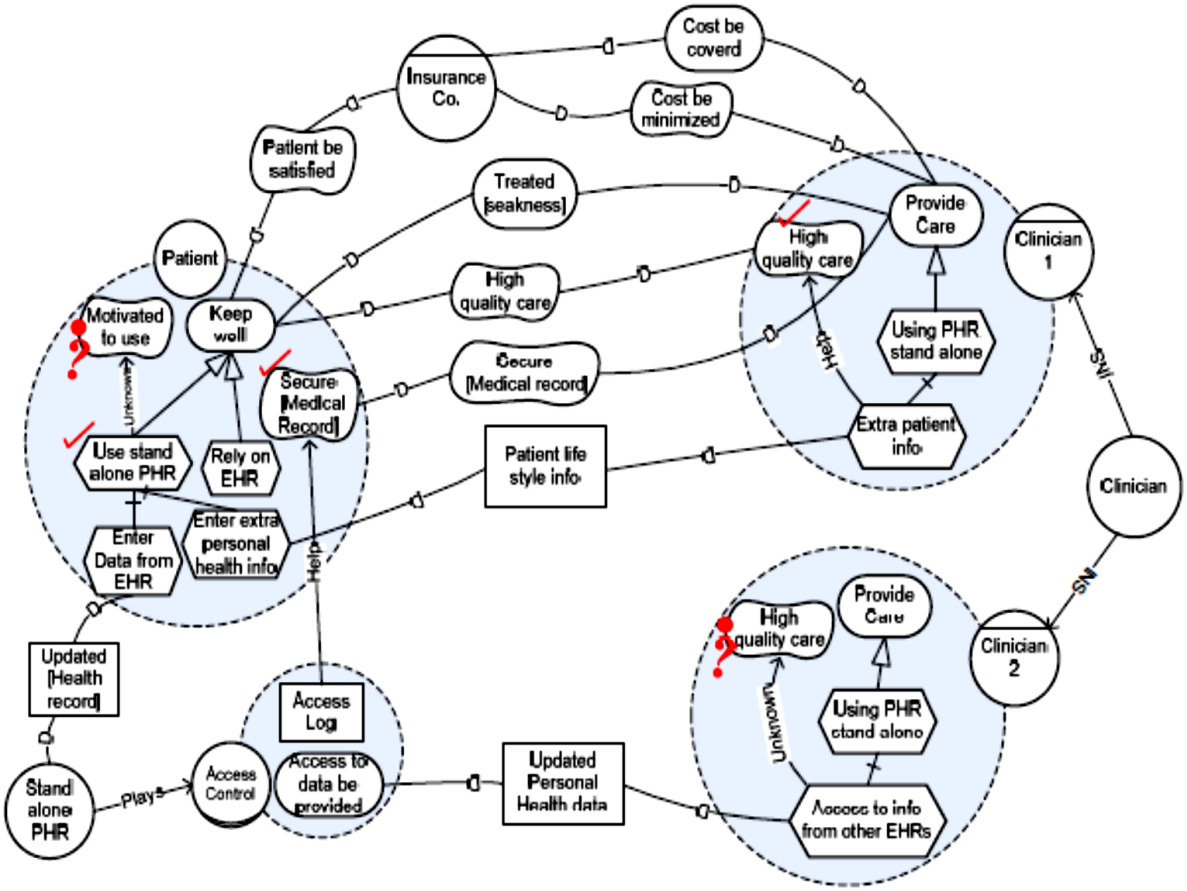


Figure 5.2 i* Rationale Model when patient uses a standalone PHR system

Figure 5.3 (adopted from the work of Samavi and Topaloglou [64]) depicts the trust, monitor and delegation relationship between Alice, her Psychiatrist (Daniel) and the RI. Alice owns the resource PHR (such as Updated personal health information, and personal life style information), and Daniel owns the treatment goal. The treatment goal depends on processing (such as reading and modifying) of PHR. The owner (O) of a goal or resource has full authority of the fulfillment of a goal or use of a resource. Alice trusts (T) her doctor (Daniel). Interestingly, Daniel and the Research Institute (RI) trust each other on a resource that neither of them owns. The model explicitly shows that there is not a trust relationship between Alice and RI. Alice delegates usage on her PHR to her doctor (Daniel). This delegation can be a delegation for execution of a goal or task or resource and labeled by **De**. Daniel in order to fulfill his goal (Alice's Treatment) wants to delegate patient information to the RI. Since Daniel is not the owner of the resource, a functional requirement must be introduced: Daniel needs Alice's consent for delegation. The type of delegation here is

delegation of permission and labeled by **Dp** (e.g. reading permission over Alice personal health information only for the purpose of her treatment). If RI uses the resource, it should be for the purpose mentioned in the permission only [64].

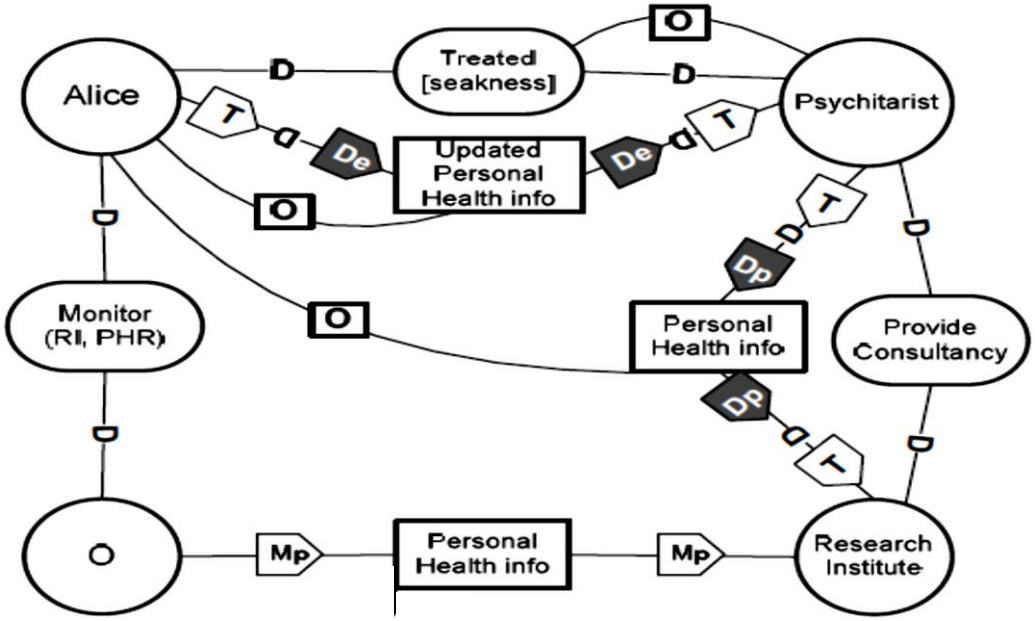


Figure 5.3 Partial trust, delegation, monitor, ownership and functional dependency model

Figure 5.3 also depicts Alice distrusts the RI and would like to enforce her privacy. She hires an ombudsman (O) to monitor the RI records to ensure the usage purpose and the deletion of record after the specified retention period. The functional dependency between Alice and the ombudsman is captured by the Monitor (RI, PHR) goal. The edge between Ombudsman and RI represents the monitoring on permission which is labeled by **Mp** [64].

We use the case study throughout the rest of this chapter to demonstrate how our method can help requirements engineers to elicit privacy requirements for online PHR Company. The engineer wants to i) capture the stakeholders' goals in a quick manner, and ii) produce the corresponding well-formed textual privacy requirements [35, 64].

5.3 Implementation of the Ontology

On the basis of i* method stated in Section 4.3, we implemented the core privacy ontology conceptual model (presented in Figure 4.2) using protégé classes, properties, and individuals. In the implementation process, we modified some existing classes and

relationships and created new ones to maintain completeness, consistency and expressivity of the ontology. Seven steps were followed to implement the ontology [34, 56, 58]:

1. Creating classes and class hierarchy,
2. Creating object and datatype properties,
3. Describing and defining classes,
4. Invoking Reasoner,
5. Creating instances of the classes,
6. Creating and executing SWRL, and
7. Creating and executing SQWRL.

In the first step, we arranged the classes in a taxonomic (subclass–superclass) hierarchy. For example, Figure 5.4 shows the dependency class hierarchy which is the subclass of OWL: Thing. We made sibling classes disjoint (an individual (or object) cannot be an instance of more than one class [58]) where appropriate. We made the sub-classes of each GoalDependum, SoftgoalDependum, TaskDependum, and PIDependum classes not disjoint, because the reason for the delegation, trust, provision, and monitor dependency relationships can be a goal, a softgoal, a task or a PI. In other words, a delegated goal, softgoal, task or PI can also be monitored, and/or trusted, and a delegated, monitored and/or trusted PI can also be provided. Besides, we made Delegation, Trust, and Monitor super-classes not disjoint, because the instances (such as PermissionOver, ExecutionOn, and/or ProcessingOn) that describe the Delegation, Trust, and Monitor dependency relationships can be the same for all the classes. We added new concepts and relationships in the ontology to properly and expressively represent the proposed conceptual ontology. Here, we created new classes such as permission type, provision type, processing type, contribution type, decomposition type, trust type, and intentional element type to describe the type that members of classes can have.

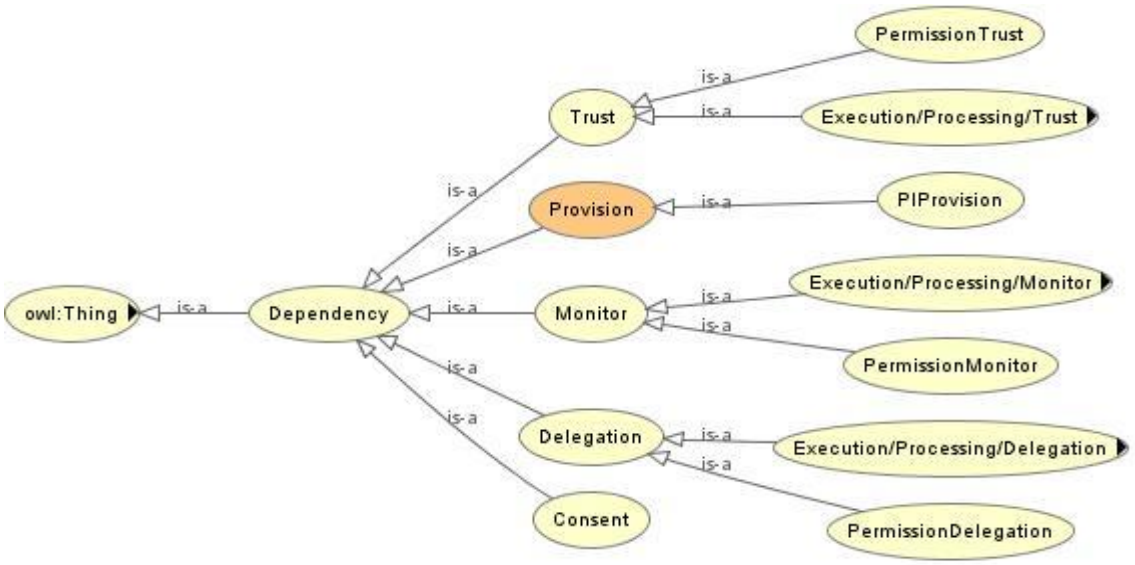


Figure 5.4 Dependency class hierarchy

In the second step, we created two main types of properties (object properties and datatype properties) in our ontology to represent relationships. Object properties are relationships between two individuals, while datatype properties describe relationships between an individual and data values. Properties link individuals from the domain to individuals from the range. Properties may have a domain and a range specified. However, it is generally advised against doing this [58]. Thus, we have created object properties presided by domain and followed by range separated by underscore (_) to clarify the relationships except for most iStar relationships and some other relationships. For example the object property Actor_owns_PI has a domain Actor, a range PI and the actual object property is owns.

In the third step, we described and defined classes (where appropriate) in terms of existential restriction, and universal restriction. We used these restrictions to define the necessary conditions of a primitive class and convert the necessary conditions into necessary and sufficient conditions and come up with defined classes. Using universal restrictions, we created closure axioms to explicitly restrict the open world reasoning in OWL. Besides, we defined classes such as privacy criterion, privacy goal, severity level, and sensitivity level as enumerated classes by precisely listing the individuals that are the members of the class [58]. Figure 5.5 shows the class description and definition of the class GoalNode.

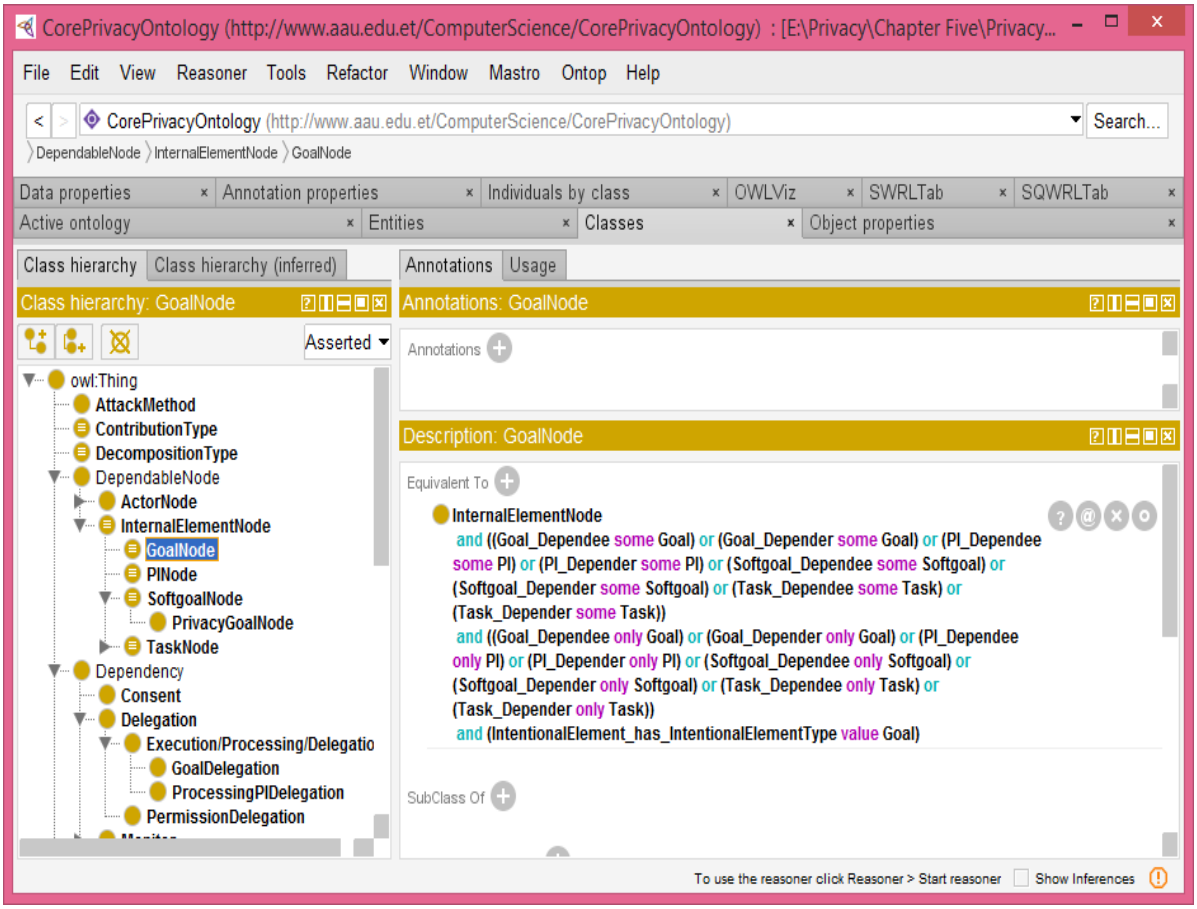


Figure 5.5 GoalNode class definition and description

In the fourth step, we used HermiT 1.4.3.456, Pellet, and FaCT++ 1.6.5 reasonor to come up with inferred axiom and to verify the consistency and completeness of the ontology, and corrected the pitfalls. After we started using SWRL Tab and SQWRL Tab, we only used Pellet, and FaCT++ 1.6.5 reasonor to verify our ontology due to SWRL built-in atoms are not supported yet in HermiT 1.4.3.456 reasonor.

In fifth step, we created individuals for online PHR Company. Then, we created an object property assertion for each individuals to link them with each other so that privacy knowledge can be easily explored, queried, and extracted. Practically, we found that the concept PI provision in the proposed ontology should be represented as sub-class as well as as instance of the class Provision dependency so that we can represent the dependency, provider, receiver and the dependum expressively and clearly.

Finally, in the sixth and seventh step, to work with OWL-based SWRL rules and SQWRL queries in our interactive environment application, we created and executed SWRL Rules and SQWRL queries using SWRL Tab and SQWRL Tab of protégé. Formal questions to the ontology is presented in Annex C Table C.1.

5.4 Implementation of the Method

Implementing ontology with OWL and Protégé is not enough. Thus, an interactive environment (a tool) was developed in Java to make the ontology usable for end users not familiarized with Protégé and SQWRL and to automate the method proposed in Section 4.4. The tool facilitates the exploration of the ontology. It automatically and dynamically generates the necessary SQWRL queries and rules for obtaining the information related to organization, agents, role, privacy goals, tasks, assets, threats, vulnerabilities, and privacy requirements. It makes it possible to generate a specification (a word document) that summarizes the result of the analysis. The Protege-based SWRLTab Plugin automatically include Drools engine that plays the role of the engine[10]; it is intended to wait for SQWRL queries. Once a query is received, the engine processes it and then sends the result to the tool. Thus, the requirements engineers do not interact directly with it [34].

A screenshot of the user interface is presented in Figure 5.6. Privacy requirements elicitation process was performed, with three main windows: Context Analysis (on the left side), Risk analysis, and Privacy Requirements Identification (on the right side). First, the tool allows the user to load the core privacy ontology. Then, in the Context Analysis window, it allows the user to choose organization, its agent and their role, the different stakeholder privacy goals and their associated sub-goals, sub-sub-goals, tasks and operational tasks, and the valuable PIs (Assets) that needs to be protected. The rest are displayed on the left. For each PI, the tool displays the corresponding threats. For each chosen threat, the tool displays the corresponding vulnerabilities. And finally, for each chosen vulnerabilities, the resulting list of privacy requirements to mitigate them is presented. The “Export to Word File” button leads to the generation of the specification document that summarizes the analysis and the relevant privacy requirements [34].

[10] https://github.com/protegeproject/swrlapi-drools-engine/wiki/SWRLAPI-Drools-Engine

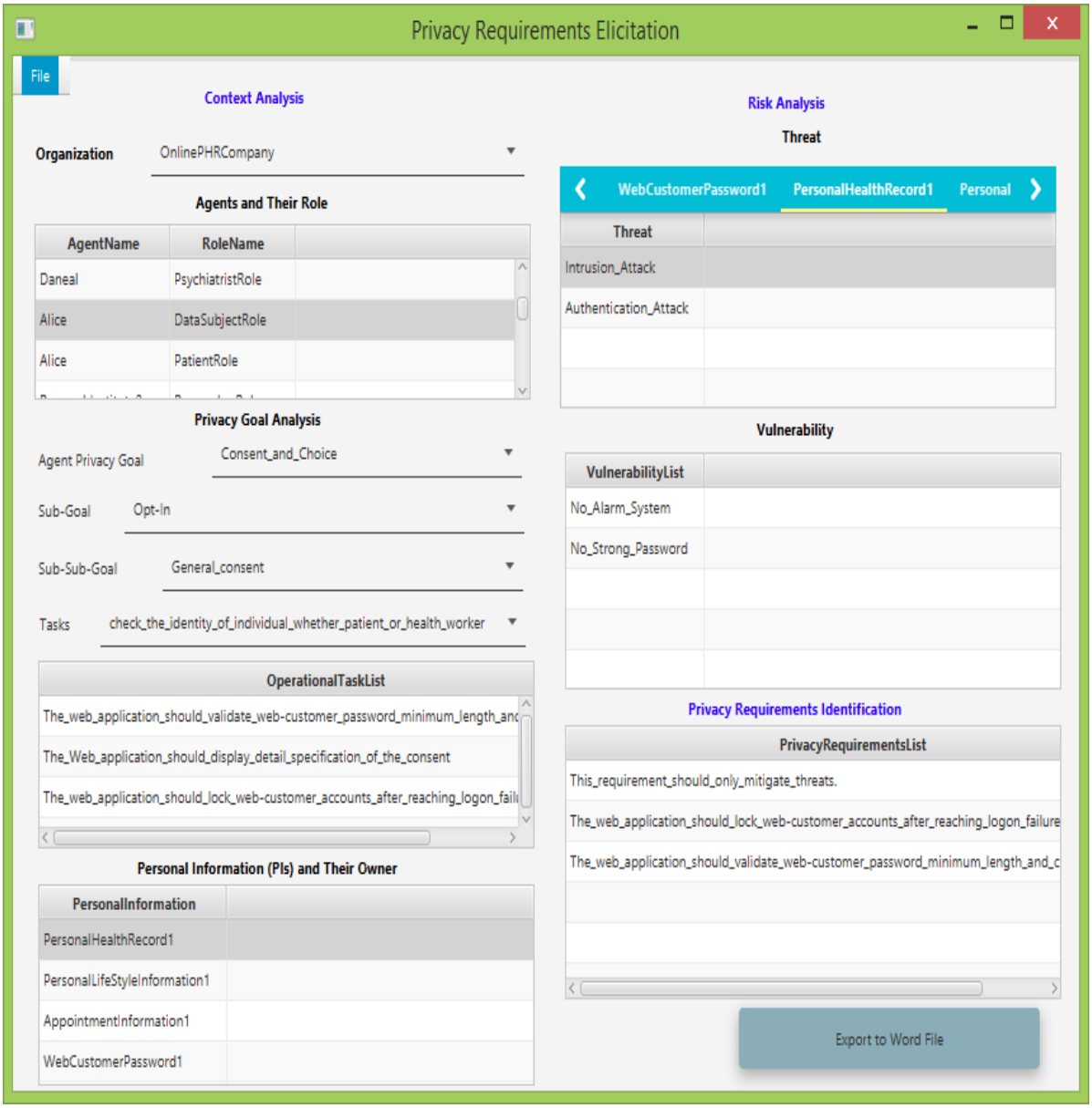


Figure 5.6 A screenshot of the interactive environment

5.5 Evaluation and Discussion

As stated in Section 4.1 the fundamental goal of the thesis is to provide complete, valid and usable platform containing knowledge about the core concepts and relations related to privacy. Thus, the evaluation focuses on assessing the completeness, validity and usability of the core privacy ontology. Besides, it focuses on assessing the usability of the method and the tool. We have used alignment table, competency questions, controlled laboratory experiment, and questionnaire methods to evaluate the solution.

5.5.1. Completeness

The completeness criterion verifies that our ontology integrates the knowledge that exists in other ontologies. By completeness, we want to prove that the proposed ontology is 'more' complete than the ones covered in literature. An alignment table was drawn up, with the concepts of our ontology on one side, and concepts of other ontologies on the other side (these ontologies were chosen, and not others, because they have been used in requirements engineering contexts) [34, 35]. Table 5.1 presents the result of the alignment.

Table 5.1 The alignment table of ontologies used for privacy requirements elicitation

Concepts of the ontology	**Other Ontologies**		
	Gharib et al. [31]	**Velasco et al. [66]**	**Dritsas et al. [67]**
PI (Asset)	PI (Asset)	Asset	Asset
Organization	Organization	-	-
Actor	Actor	-	-
Agent	Agent	-	-
Role	Role	-	-
Privacy Goal	Privacy Goal		Objective
Threat	Threat	Threat	Threat/Deliberate attack
Problematic Data Action	-	-	-
Vulnerability	Vulnerability	-	Vulnerability
Risk	-	Risk	-
Severity	Severity	Valuation Criteria	-
Impact	Impact	-	-
Threat Agent	Threat Actor	-	Attacker
Attack Method	Attack Method	-	-
Privacy Criterion	Privacy Constraint	-	-
Privacy Requirement	Privacy Requirement	-	Privacy Requirement
Privacy Control	-	Safeguard	Countermeasure

Our ontology and the ontology of Gharib et al. [31] contains PI as the only Asset, while the rest two ontology contains PI as one kind of Asset. Besides, as there are socio-cultural aspects to privacy [17] and privacy is contextual among community and organizations [14, 29], our ontology and the ontology of Gharib et al. [31] contain agentive entities, their intentional elements and social dependencies [31]. Thus, these ontologies contain Organization, Actor, Agent, Role and Privacy goal, but the rest two do not contain these concepts. All of the ontology considered Threat, two of them (the ontology of Gharib et al. [31] and Dritsas et al. [67]) neglect the concept Risk generated by a threat, and its Severity. Only our ontology contains Problematic Data Action which is more information-rich factor for a privacy risk model [19] (as discussed in Section 2.3.2.1). Except the ontology of Velasco et al. [66], all the ontology contains Vulnerability. Our ontology and the ontology of Gharib et al. [31] contains Impact, and Attack Method, but the rest two does not contain them. Except the ontology of Velasco et al. [66], all the ontology uses Threat Agent with different name. Our ontology and the ontology of Gharib et al. [31] contains Privacy Criterion with different name, but the rest two does not contain Privacy Criterion. Our ontology, the ontology of Gharib et al. [31], and the ontology of Dritsas et al. [67] contains Privacy Requirement. The ontology of Velasco et al. [66] does not contains Privacy Requirement at all. All, except the ontology of Gharib et al. [31] contains Privacy Control with different name. Therefore, the results demonstrate that the proposed core privacy ontology is relatively more complete than the other ontologies used.

Similar to the works of Souag et al. [34], we have experienced that the goal of constructing complete core privacy ontology remains ambitious and more complex than expected. Even this experience is aggravated due to the different notions and conceptualizations of privacy.

5.5.2. Validity

We used competency questions [48] to validate the proposed ontology. The ontology must be able to give reliable answers to these questions using its terminology. Table C.1 in Annex C (adopted from the works of Souag et al. [34]) present informal (expressed in natural language) and formal (expressed in SQWRL) questions to the ontology that a requirements engineer is likely to encounter during requirements elicitation. These questions guide the requirements engineer during the requirements elicitation process. They should be regarded

as indicative of what the ontology can deal with and reason about [34]. For example, Figure 5.7 demonstrates delegation dependency question and result for:

- The informal questions: Who delegates for whom? What? for what reason? On what consent? And what types of permission?
- The formal question: delegates (?DelegaterList, ?DelegateeList) ^ PermissionDelegation (?PermissionDelegationList) ^ Permission_has_PermissionType (?PermissionDelegationList, ?PermissionTypeList) ^ has_Consent(?PermissionDelegationList, ?ConsentList) ^ PermissionOverPIDelegatum (?PermissionOverPIDelegatumList) -> sqwrl:select (?DelegaterList, ?DelegateeList, ? PermissionDelegationList, ?PermissionTypeList, ? PermissionOverPIDelegatumList, ?ConsentList).

Those questions and their results have demonstrated that the ontology could be used in privacy requirements elicitation phase. Thus, it can be concluded that the ontology is valid.

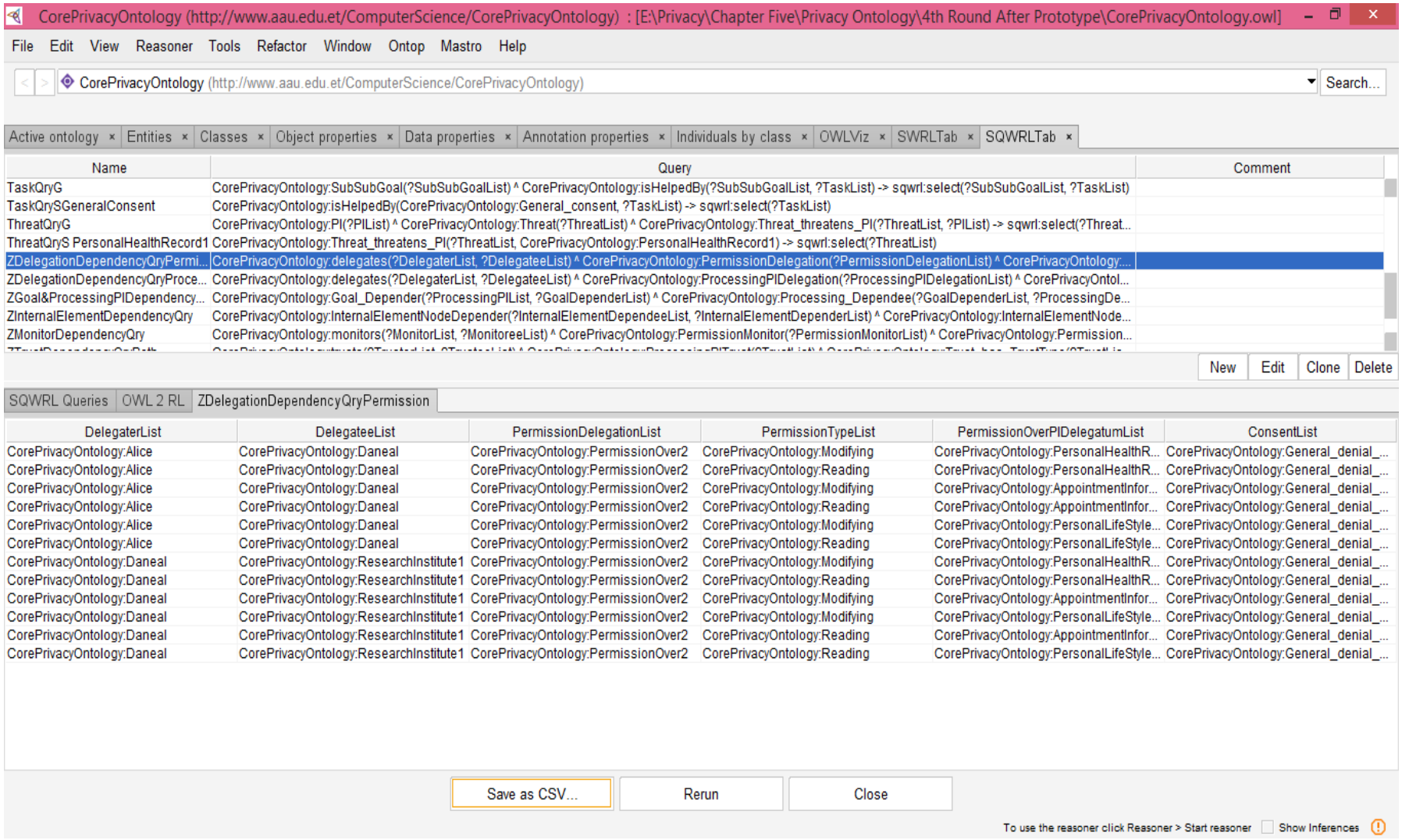


Figure 5.7 Delegation dependency query and result

5.5.3. Usability

Any proposed method remains incomplete until its usability and benefits are evaluated by its end-users. Controlled experiments are one way to evaluate methods. During experiments, evaluators need to follow a thorough process. The process explicates the objectives of the evaluation, the different hypotheses to test, the subjects that perform the experiments, and the different variables to measure [35].

We have used method evaluation model proposed by Moody [68] as evaluation method. By combining two dimensions of method (pragmatic) success: actual effectiveness and adoption in practice, the author proposes a theoretical model and associated measurement instrument for evaluating Information System (IS) design methods as well as methods used in other domains. The evaluation procedure carried out has been adapted from Pfleeger [69]. The procedure steps are conception (define the goals or objectives of the experiment), design (define hypotheses, select experimental subjects, and define variables), preparation (readying the subjects and experimental environment), execution, analysis, and reporting results. Throughout the evaluation process, the guidelines for empirical research in software engineering proposed by Kitchenham et al. [70] have been taken into consideration [35].

Based on the evaluation method, procedures, and guidelines, we have come up with the following experimental design. The main goals of our experiment has been to measure: the coverage of the core privacy ontology and its usability, the usability of the method for producing privacy requirements, and the usability of the tool (the interactive environment). We have adopted perceived usefulness and perceived ease of use [68] (and their associated measurement variables and questions [34, 35]) which are fundamental determinants of user acceptance and system usage [71]. We have measured usability based on these two variables. The Hypotheses have been: (1) the privacy ontology provides the main concepts for privacy requirements elicitation process, (2) the method is easy to use, (3) the method is efficient, (4) Users find the graphical interface to access the privacy ontology easy to use, and (5) Users find that the tool is easy to use overall [35]. For the time being the experiment has been conducted on final year computer science postgraduate students who were easily accessible and who took software engineering, and computer security and privacy courses. We contacted 11 regular and 20 extension students by mail and 9 regular students (who we

mate them at computer laboratory and took their phone number) by phone making a total of 40 students (20 regular and 20 extension). The experiment has been conducted on 7 subjects (who were available at the time of evaluation). 1 of them were female, 6 male. The experiment included orientation, a presentation of the core privacy ontology, presentation of the method, demonstration of the interactive environment, and a session of manipulation by the participants. At each phase, participants were asked to fill in a questionnaire (adopted from the works of Souag et al. [34, 35]) presented in Annex D. The results extracted from the questionnaire are summarized as follows:

A. The privacy ontology and its usability

Questions 1-2 in Annex D allowed us to evaluate the usability of the ontology with subjects. Their results are summarized in Figure 5.8 and 5.9.

Figure 5.8 shows results of question 1 (Q1) Do you find that the privacy ontology has the main concepts for privacy requirements elicitation? The Figure show that most participants agree that the privacy ontology includes the main concepts. Figure 5.9 represents results of Q2 (Does the privacy ontology help in finding new elements (privacy requirements, threats, vulnerabilities …)?). It shows that most participant agree that the ontology helps in discovering new elements since it is not easy to bear in mind hundreds of threats, vulnerabilities, and their corresponding privacy requirements [34, 35]. Figure 5.8 and 5.9 show a quite high level of satisfaction, which is encouraging. Thus, Hypothesis (1) is mostly validated, which means that the proposed privacy ontology provides the most important concepts for privacy requirements elicitation.

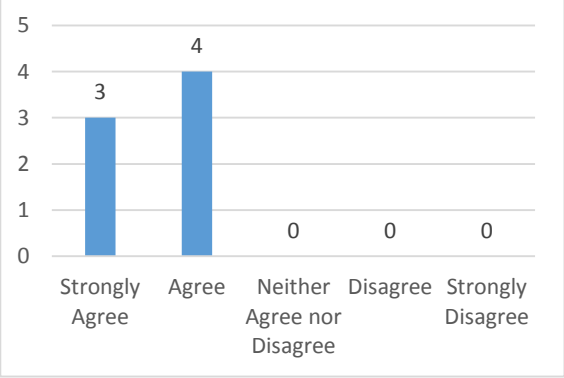


Figure 5.8 Results of Question 1 (Q1)

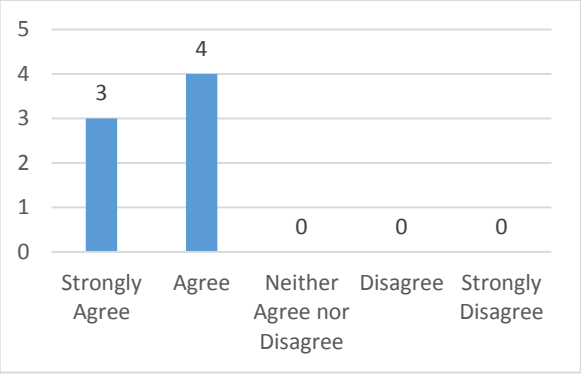


Figure 5.9 Results of Q2

Besides, the core security ontology of Souag et al. [34] was organized around three dimensions, while our ontology like the core privacy ontology of Gharib et al. [31] has been organized around four dimensions (adding privacy dimension). Let alone the ontology of Souag et al. [34], we have added new concepts and relations and/or modified the existing ones even on each dimensions of the ontology of Gharib et al. [31]. For example, in the organizational dimension, new concepts such as task have been added. Instead of the concept use, we have used processing. We have specified information provision into PI provision. In the risk dimension, new concepts such as problematic data action as a kind of threat have been added. In the treatment dimension, instead of the concept privacy constraints, we have used privacy criterion. New concepts such as predictability, manageability, and disassociability as privacy criterion, and privacy control have been added. In the privacy dimension, consent and choice, collection limitation and information minimization, use limitation, disclosure, access and amendment, security/safeguards, information quality, enforcement, information flow, and sensitivity as privacy goals/principles have been added as new concepts. Thus, our ontology contains new important concepts and relations than the core privacy ontology of Gharib et al. [31].

B. Usability of the Method

Questions 3-4 in Annex D allowed us to evaluate the usability of the method with subjects. Their results are summarized in Figure 5.10 and 5.11.

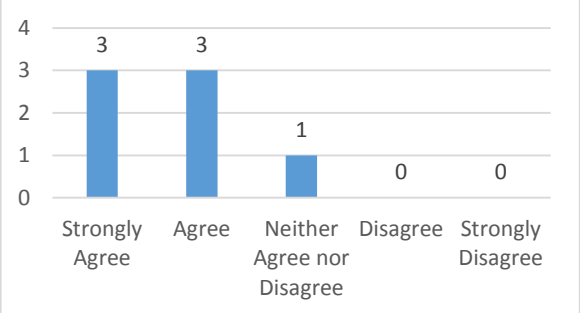


Figure 5.10 Results of Q3

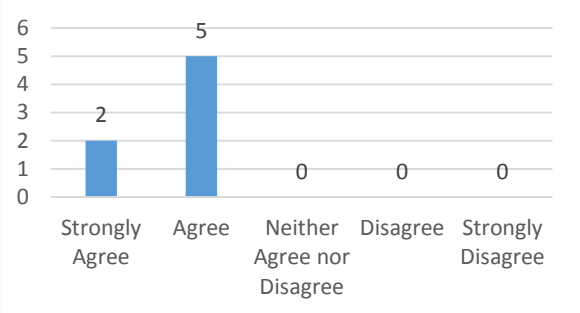


Figure 5.11 Results of Q4

Q3 was related to effort required for the elicitation of privacy requirements using the method and Q4 was about the ease of use of the method. Most participants expressed high level of agreement. Thus, these results validate hypotheses (2) and (3). This implies that the method is easy to use and efficient (time saving) with a high level of confidence.

SQUARE for Privacy [27] uses databases of privacy requirements based on privacy laws and regulations that involves developer and stakeholder discussion to adopt them to particular requirements which is time consuming, while our method refines, structures, and represents high level privacy goals, laws, and/or principles into low level operational privacy requirements in accordance with different stakeholders' interpretation of privacy on the basis of i* method. Our method does not need developer and stakeholder discussion to adopt privacy laws to a particular requirements as our ontology contains deferent stakeholders' definitions of privacy. For example, in our case study, Alice privacy needs and interpretations are already structured and represented in our ontology and can be queried at any time without her involvement. Thus, our method needs less effort than SQUARE for Privacy [27] and is easy to use to perform privacy requirements elicitation.

C. Usability of the prototype implementing the ontology and the method

Questions 5-6 in Annex D allowed us to evaluate the usability of the tool implementing the ontology and the method as a whole with subjects. Their results are summarized in Figure 5.12 and 5.13.

Figure 5.12 shows that almost all participants appreciated the interactive environment that simplifies and facilitates access to the ontology, and revealed that is nice to use such kind of interface than directly using the ontology.

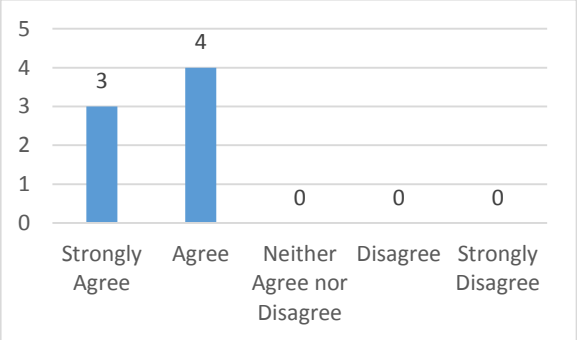


Figure 5.12 Results of Q5

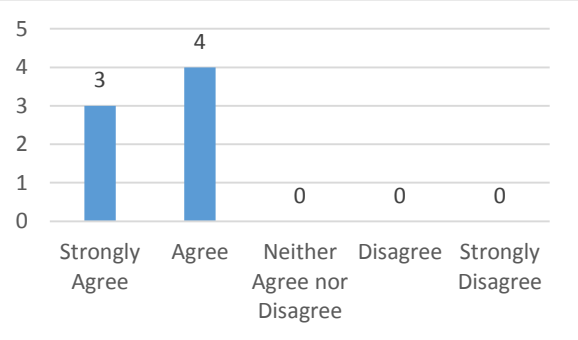


Figure 5.13 Results of Q6

Figure 5.13 shows that most participants agreed that the tool implementing the method is friendly and easy to use. These results validates hypotheses (4) and (5) which means that users find the graphical interface to access the privacy ontology easy to use.

As we have used the latest Protégé and Protégé-based SWRLTab Plugin that automatically include Drools engine (which is a SWRL API-based SWRL rule engine implementation), creating and executing SWRL rules and SQWRL queries are relatively simpler than the implementation of Souag et al. [34]. This environment also simplifies the development of the interface which in turns makes the interface clear and easy to use.

5.6 Summary

This chapter has presented the implementation of the ontology and the methods proposed in chapter four and their evaluation. It has explained the development tools and technologies that have been used and why they were chosen. It has also described Online PHR Company as case study, and the importance of Electronic Personal Health Record systems (ePHRs) as transformative health technology in relation to privacy.

Completeness of the ontology has been evaluated using alignment table drawn up, with the concepts of our ontology on one side, and concepts of other privacy ontologies on the other side. The result demonstrates that the proposed privacy ontology is relatively complete than other ontologies. Besides, validity of the ontology has been evaluated using competency questions. The questions and their results have demonstrated that the proposed ontology is valid.

Besides, usability of the ontology, the method and the interactive environment has been evaluated using controlled laboratory experiment, and collecting the participants' feedback through questionnaires. All the results have shown a quite high level of satisfaction, which is encouraging. Thus, we can conclude that all the hypotheses are validated with high level of confidence.

Chapter Six: Conclusion and Future Works

6.1 Conclusion

The thesis presents ontology-based interactive privacy requirements elicitation method that can help requirements engineers while eliciting privacy requirements in software development environment (a task known to be difficult, costly, and time- and labor-intensive human activity due to lack of reusable knowledge, lack of systematic methodology, and engineers' lack of legal knowledge and expertise which in turn requires privacy specialists' collaboration). To address what are the concepts and relations that need to be presented in a core privacy ontology (the first research question)? We have acquired key concepts and relations through a systematic literature review method. The concepts and relations have been defined, structured and represented in a table, glossary and conceptual model, and implemented in Protégé OWL. Thus, the first research question has been considered to be achieved. Then, to systematically exploit the privacy knowledge represented in the ontology, interactive (machine guided) privacy requirements elicitation method has been proposed and implemented in an interactive environment (a tool). Hence, the second research question (How can the ontology be used by requirements engineers while discovering privacy requirements?) has been considered to be achieved. Therefore, the objectives of the research have been considered to be achieved since the research questions have been answered.

More importantly, we have evaluated the completeness and validity of the ontology as well as the usability of the ontology, the method, and the tool. The completeness of the ontology has been evaluated with regards to existing ontologies used in privacy requirements engineering methods using alignment table. The result has shown that our ontology is relatively complete than other ontologies. In addition, competency questions showed that the ontology answers what it is asked and it behaves as we expect it. Thus, it is valid. Besides, the controlled experiment demonstrated that the ontology covers the main privacy concepts and helps requirements engineers in eliciting privacy requirements by allowing them to exploit privacy-structured knowledge via the tool that dynamically generates the necessary queries. The experiment also revealed that the method is efficient and easy to use, and the tool is friendly to use.

Besides, we have found that the representation of the ontology in accordance with i* modeling method is expressive and logical. In our case study, we have experienced that actors (such as Alice, psychiatrist (Daniel), Neurologist (Dereje) and the Research Institute), their intentional elements, their interaction, and their delegation, trust, monitor and provision dependency relationships have been represented clearly and logically as Depender, Dependee, Dependum and they are easy to read and query. We have also observed that as far as the core privacy ontology has links to privacy concepts and relations to other domain specific privacy ontology, we can instantiate the core privacy ontology with this ontology. As there can be a number of domain specific privacy ontology that can be linked with the core privacy ontology, we should have an automated way of linking domain specific ontologies with the core privacy ontology.

Despite all these efforts, we have experienced that let alone privacy, specific privacy concept such as consent management requires a thorough knowledge, experience, and expert support to represent them in the core privacy ontology. Besides, developing core privacy ontology (specifically speaking acquiring, conceptualizing and implementing privacy concepts and relations) has been time consuming and tedious. Thus, this research should involve many teams with expertise from different discipline.

6.2 Contribution

The contributions of the thesis are:

- core privacy ontology developed on the basis of i* modeling method that realize the benefits of goal oriented modeling and analysis in early and high level requirement engineering,
- interactive privacy requirements elicitation method that can be applied with the core privacy ontology in an integrated way (which came up with a systematic method called ontology-based interactive privacy requirements elicitation method), and
- interactive environment (a tool) that automates the method to simplify exploration of the ontology by the users (whether they have knowledge about Protégé, SWRL rule, and SQWRL Query or not).

6.3 Future Works

In the future,

- Our method will be enhanced to incorporate requirements prioritization activity, and PETs in the ontology so that we can link privacy requirements to technical solution.
- Being the ontology is developed based on the i* modeling method, we can and will integrate and use the ontology and its reasoning features with existing goal oriented approaches, such as Secure Tropos, KASO and GBRAM, so that it can help requirements engineers who use these approaches.
- We will enhance our method to analyze impacts of a selected privacy requirements on other requirements.
- We will enhance our method to analyze the degree of PI sensitivity.
- Automated way of linking domain specific ontologies with the core privacy ontology will be developed.
- The method will be implemented in multiple case studies from different domain to better validate it.
- The controlled experiment will be performed with a larger number of participants to improve the validity of the results.
- To generalize the results, we will conduct a field experiment (practitioner acceptance testing) to evaluate the likelihood of the adoption of the method in the real world.

References

[1] B. Davey, and K. R. Parker, "Requirements Elicitation Problems: A Literature Analysis," *Issues in Informing Science and Information Technology,* Vol. 12, 2015, pp. 71-82, retrieved from http://iisit.org/Vol12/IISITv12p071 082Davey1929.pdf, Latest accessed on March 27, 2018.

[2] D. V. Dzung and A. Ohnishi, "Ontology-based Reasoning in Requirements Elicitation," 2009 *Seventh IEEE International Conference on Software Engineering and Formal Methods*, 2009, pp. 263- 272.

[3] U. Sajjad and M. Q. Hanif, "Issues and Challenges of Requirement Elicitation in Large Web Projects," Unpublished Master Thesis, School of Computing, Blekinge Institute of Technology, 2010.

[4] R. R. Young, *Effective Requirements Practices*, Addison-Wesley Publications, Boston, 2001.

[5] T. Ambreen, N. Ikram, M. Usman and M. Niazi. "Empirical research in requirements engineering: trends and opportunities," *Requirements Engineering*, Springer-Verlag, London, 2016.

[6] P. Spoletiniand and A. Ferrari, "Requirements Elicitation: A Look at the Future through the Lenses of the Past," *2017 IEEE 25th International Requirements Engineering Conference*, 2017, pp. 476-477.

[7] I. A. Al-Fataftah and A. A. Issa, "A Systematic Review for the Latest Development in Requirement Engineering," *World Academy of Science, Engineering and Technology International Journal of Humanities and Social Sciences*, Vol.6, No.4, 2012, pp. 583-590.

[8] H. C. Cheng and J. M. Atlee, "Research Directions in Requirements Engineering," *Future of Software Engineering (FOSE'07) 2007 IEEE*, 2007.

[9] S. Ullah, M. Iqbal and A. M. Khan, "A Survey on Issues in Non-Functional Requirements Elicitation," 2011 IEEE, 2011, pp. 333-340.

[10] M. Rahman and S. Ripon, "Elicitation and Modeling Non-Functional Requirements: A POS Case Study," 2012.

[11] H. Kaur and A. Sharma, "Non-Functional Requirements Research: Survey," *International Journal of Science and Engineering Applications*, Vol. 3, No. 6, 2014, pp. 172-182.

[12] C. Kalloniatis, E. Kavakli and S. Gritzalis, "Addressing privacy requirements in system design: the PriS method," *Requirements Engineering*, Springer-Verlag, London, Vol. 13, 2008, pp. 241-255.

[13] M. Gharib, M. Salnitri, E. Paja, P. Giorgini, H. Mouratidis, M. Pavlidis, J.F. Ruiz, S. Fernandez, and A. D. Siria, "Privacy Requirements: Findings and Lessons Learned in Developing a Privacy Platform," *IEEE 24th International Requirements Engineering Conferenc*e, 2016, pp. 256–265.

[14] Y.-S. Martín, J. M. Alamo, and J. C. Yelmo, "Engineering Privacy Requirements: Valuable Lessons from Another Realm," 2014.

[15] D. N. Jutla and P. Bodorik, "Sociotechnical architecture for online privacy," *IEEE Security and Privacy*, Vol. 3, 2005, pp. 29-39.

[16] S. Spiekermann, "The Challenges of Privacy by Design," *Communications of the ACM July 2012*, Vol. 55, No. 7, 2012, 38-40.

[17] A. Cavoukian, S. Shapiro, and R. J. Cronk, *Privacy Engineering: Proactively Embedding Privacy, by Design*, Ontario, Canada, 2014.

[18] F. S. Gürses, "Multilateral Privacy Requirements Analysis in Online Social Network Services," Unpublished Dissertation, Engineering, Arenberg School, 2010.

[19] S. Brooks, M. Garcia, N. Lefkovitz, S. Lightman, and E. Nadeau, "An Introduction to Privacy Engineering and Risk Management in Federal Systems," *National Institute of Standards and Technology Internal Report 8062*, 2017.

[20] S. Gürses and J. M. del Alamo, "Privacy Engineering: Shaping an Emerging Field of Research and Practice," *IEEE Symposium on Security and Privacy*, 2016, pp. 40-46.

[21] A. Pattakou, C. Kalloniatis, and S. Gritzalis, "Security and Privacy Requirements Engineering Methods for Traditional and Cloud-Based Systems: A Review," *CLOUD COMPUTING 2017: The Eighth International Conference on Cloud Computing, GRIDs, and Virtualization*, 2017, pp. 145-151.

[22] P. Anthonysamy, A. Rashid, and R. Chitchyan, "Privacy Requirements: Present & Future," 2017.

[23] Norton, "2017 Norton Cyber Security Insights Report," 2017, retrieved from https://us.norton.com/cyber-security-insights-2017, last accessed on February 1, 2020.

[24] The Senate and House of Representatives of the United States of America in Congress, "Health Insurance Portability and Accountability (HIPAA) Act of 1996," PUBLIC LAW 104–191, 1996 .

[25] K. Beckers, "Comparing Privacy Requirements Engineering Approaches," *2012 Seventh International Conference on Availability, Reliability and Security*, IEEE, 2012, pp. 574-581.

[26] S. Miyazaki, N. Mead, and J. Zhan, "Computer-Aided Privacy Requirements Elicitation Technique," *2008 IEEE Asia-Pacific Services Computing Conference*, 2008, pp. 367-372.

[27] A. Bijwe and N. R. Mead, "Adapting the SQUARE Process for Privacy Requirements Engineering," *Technical Note CMU/SEI-2010-TN-022*, Software Engineering Institute, Carnegie Mellon University, 2010.

[28] M. Deng, K. Wuyts, R. Scandariato, B. Preneel, and W. Joosen, "A privacy threat analysis framework: supporting the elicitation and fulfillment of privacy requirements," *Requirements Engineering*, Vol. 16, 2011, pp. 3-32.

[29] I. Webster, V. Ivanova, and L. M. Cysneiros, "Reusable Knowledge for Achieving Privacy: A Canadian Health Information Technologies Perspective," *in Proceeding of VIII Workshop in Requirements Engineering*, Porto, Portugal, 2005, pp. 112-122.

[30] D. J. Solove, "A Taxonomy of Privacy," *University of Pennsylvania Law Review*, Vol. 154, No. 3, 2006, pp. 447-560.

[31] M. Gharib, P. Giorgini, and J. Mylopoulos, "Towards an ontology for Privacy Requirements via a Systematic Literature Review," *Conference Paper*, 2017.

[32] M. Hecker, "A Generic Privacy Ontology and its Applications to Different Domains," Unpublished Dissertation, Digital Ecosystems and Business Intelligence Institute, Curtin University of Technology, 2009.

[33] M. Fernández-López, A. Gómez-Pérez, and N. Juristo, "METHONTOLOGY: From Ontological Art Towards Ontological Engineering," *Proceedings of the Ontological Engineering AAAI-97 Spring Symposium Series,* Stanford University, 1997, pp. 33-40.

[34] A. Souag, C. Salinesi, R. Mazo, and I. Comyn-Wattiau, "Security Ontology for Security Requirements Elicitation," *Springer International Publishing Switzerland*, 2015, pp. 157–177.

[35] A. Souag, "AMAN-DA: A knowledge reuse based approach for domain specific security requirements engineering," Unpublished Dissertation, Université Paris 1 Panthéon-Sorbonne, 2015.

[36] A. Cavoukian, *Operationalizing Privacy by Design: A Guide to Implementing Strong Privacy Practices*, Toronto, Ontario (Canada), 2012.

[37] S. Spiekermann and L. F. Cranor, "Engineering Privacy," *IEEE Transactions on Software Engineering,* Vol. 35, No. 1, 2009, pp. 67-82.

[38] D. Zowghi and C. Coulin, "Requirements Elicitation: A Survey of Techniques, Approaches, and Tools," *Engineering and Managing Software Requirement*s, Springer, Berlin Heidelberg, 2005, pp. 19-46.

[39] X. Zhang, "An Interactive Approach of Ontology-based Requirement Elicitation for Software Customization," Unpublished Master Thesis, Department of Computer Science, University of Windsor, 2011.

[40] T. H. Bui, "Multimodal Dialogue Management: State of the Art," Centre for Telematics and Information Technology, University of Twente, Enschede, Netherlands, Technical Report TR-CTIT-06-01, 2006.

[41] A. Flycht-Eriksson, "A Survey of Knowledge Sources in Dialogue Systems," 1999, pp.41-48.

[42] M. Araki, K. Komatani, T. Hirata, and S. Doshita, "A Dialogue Library for Task-Oriented Spoken Dialogue Systems," *Linkoping Electronic Articles in Computer and Information Science,* Vol. 4, No.21, 1999, pp. 1-7.

[43] H. Kaiya and M. Saeki, "Using Domain Ontology as Domain Knowledge for Requirements Elicitation," *14th IEEE International Requirements Engineering Conference (RE'06),* 2006.

[44] D. Dermeval, J. Vilela, I. Bittencourt, J. Castro, S. Isotani, P. Brito and A. Silva, "Applications of ontologies in requirements engineering: a systematic review of the literature," *Requirements Engineering*, Springer-Verlag, London, Vol. 21, 2015, pp. 405-437.

[45] N. F. Noy and D. L. McGuinness, "Ontology Development 101: A Guide to Creating Your First Ontology," *Stanford Knowledge Systems Laboratory, Technical Report KSL-01-05*, Stanford University, Stanford, 2001, pp. 1-25.

[46] C. Roussey, F. Pinet, M. A. Kang, and O. Corcho, "An Introduction to Ontologies and Ontology Engineering," *Ontologies in Urban Development Projects*, Springer-Verlag, London, 2011, pp. 9-38.

[47] S. Williams and R. Power, "Grouping axioms for more coherent ontology descriptions," *6th International Natural Language Generation Conference (INLG 2010),* Dublin, Ireland, 2010.

[48] H. Guan, H. Yang, and J. Wang, "An Ontology-based Approach to Security Pattern Selection," *International Journal of Automation and Computing,* Vol.13 No. 2, 2016, pp. 168-182.

[49] S. Staab and A. Maedche, "Axioms are Objects, too – Ontology Engineering beyond the Modeling of Concepts and Relations," *Workshop on Applications of Ontologies and Problem-Solving Methods, ECAI 2000,* Berlin, 2000.

[50] C. Salinesi, E. Ivankina and W. Angole, "Using the RITA Threats Ontology to Guide Requirements Elicitation: an Empirical Experiment in the Banking Sector," *2008 First International Workshop on Managing Requirements Knowledge (MARK'08),* 2008, pp.11-15.

[51] S. Farfeleder, T. Moser, A. Krall, T. Stalhane, I. Omoronyia, and H. Zojer, "Ontology-Driven Guidance for Requirements Elicitation," *ESWC 2011, Part II, LNCS 6644,* Springer-Verlag, Berlin Heidelberg, 2011, pp. 212–226.

[52] X. Franch, L. López, C. Cares, and D. Colomer, "The i* Framework for Goal-Oriented Modeling," *Domain-Specific Conceptual Modeling,* Springer, Cham 2016, pp. 485-506.

[53] J. Horkoff and E. Yu, “Analyzing Goal Models – Different Approaches and How to Choose Among Them,” *Conference: Proceedings of the 2011 ACM Symposium on Applied Computing (SAC)*, TaiChung, Taiwan, 2011.

[54] J. Horkoff and E. Yu, “Interactive goal model analysis for early requirements engineering,” *Journal of Requirements Engineering*, Springer-Verlag, London, Vol. 21, No. 1, 2014, pp. 29-61.

[55] K. Najera, A. Perini, A. Martinez, and H. Estrada, “Supporting i* model integration through an ontology-based approach,” *CEUR Proceedings of the 5th International i* Workshop (iStar 2011)*, 2011, pp. 43-48.

[56] K. Najera, “An Ontology-Based Approach for Integrating i* Variants,” Unpublished Master Thesis, Mexico, Cuernavaca, Morelos, 2011.

[57] M. Drgon, G. Magnuson, and J. Sabo, “Privacy Management Reference Model and Methodology (PMRM) Version 1.0.,” *OASIS Committee Specification 02*, 2016.

[58] M. Horridge, *A Practical Guide to Building OWL Ontologies Using Protégé 4 and CO-ODE Tools Edition 1.3*, University of Manchester, 2011.

[59] I. Horrocks, P.F. Patel-Schneider, H. Boley, S. Tabet, B. Grosof, and M. Dean, “SWRL: A semantic web rule language combining OWL and RuleML,” *W3C Member submission*, Vol. 21, No. 79, 2004.

[60] M. J. O'Connor and A. Das, “A Pair of OWL 2 RL Reasoners. OWL: Experiences and Directions (OWLED),” *9th International Workshop*, Heraklion, Greece, 2012.

[61] M. J. O'Connor, R. D. Shankar, C. Nyulas, A. K. Das, and M. A. Musen, “The SWRLAPI: A Development Environment for Working with SWRL Rules," *OWL: Experiences and Directions (OWLED), 4th International Workshop*, Washington, D.C., U.S.A, 2008.

[62] M. J. O'Connor and A. K. Das, “SQWRL: a Query Language for OWL. OWL: Experiences and Directions (OWLED),” *6th International Workshop*, Chantilly, VA, 2009.

[63] A. Yaser, A. Alsahafia, and B. V. Gay, "An Overview of Electronic Personal Health Records," *In Health Policy and Technology*, Vol. 7, No. 4, 2018, pp. 427-432.

[64] R. Samavi, and T. Topaloglou, "Designing Privacy-Aware Personal Health Record Systems," *Advances in Conceptual Modeling – Challenges and Opportunities*, *Lecture Notes in Computer Science*, Springer, Berlin, Heidelberg, 2008, pp. 12-21.

[65] A. Roehrs, C. A. da Costa, and R. Righi, "OmniPHR: A Distributed Architecture Model to Integrate Personal Health Records," *Journal of Biomedical Informatics*, Vol. 71, 2017, pp. 70-81.

[66] J. L.Velasco, R. Valencia-Garc´ıa, J. T. Fern´andez-Breis, and A. Toval, "Modelling reusable security requirements based on an ontology framework," *Journal of Research and Practice in Information Technology,* Vol. 41, No. 2, 2009, 119-133.

[67] S. Dritsas, L. Gymnopoulos, M. Karyda, T. Balopoulos, S. Kokolakis, C. Lambrinoudakis, and S. Katsikas, "A knowledge-based approach to security requirements for e-health applications," *Journal for E-Commerce Tools and Applications,* 2006.

[68] D. L. Moody, "The Method Evaluation Model: A Theoretical Model for Validating Information Systems Design Methods," *European Conference on Information Systems (ECIS) 2003 Proceedings,* Vol. 79, 2003.

[69] S.L. Pfleeger, "Experimental design and analysis in software engineering," *Annals of Software Engineering,* Vol. 1, No. 1, 1995, pp. 219-253.

[70] B. A. Kitchenham, S. L. Pfleeger, L. M. Pickard, P. W. Jones, D. C. Hoaglin, K. E. Emam, and J. Rosenberg, "Preliminary guidelines for empirical research in software engineering," *IEEE Transactions Software Engineering*, Vol. 28, No. 8, 2002, pp. 721-734.

[71] F. D. Davis, "Perceived Usefulness, Perceived Ease of Use, and User Acceptance of Information Technology," *MIS Quarterly,* 1989, pp. 319-340.

Annexes

Annex A: Concept Definition of the Core Privacy Ontology

Table A.1 has been built up for ontology concepts definition. It includes the ontologies used as an entrance point.

Table A.1 Ontology concepts definition using privacy ontologies and models from literature

Dimension		**Concepts of Our Ontology**	**Gharib et al.**	**Martín et al.**	**Brooks et al.**
Organizational Dimension	Agentive	Actor	Actor	Actor	Actor
		Role	Role	Role	Role
		Agent	Agent	Agent	Agency, Employees and Contractors
		Is-A	Is-A	Is-A	-
		Plays	Plays	Is Played By	-
	Intentional	Goal	Goal	Goal	Goal
		Task	-	Task	-
		Refinement	Refinement	Refinement	-
	Informational	PI (Asset)	PI (Asset)	PI	PII (strategic asset)
		Part-Of	Part-Of	-	-
		Own	Own	Own	-
		Processing	Use	Processing	Processing
		Processing PI	Using Information	Processing PI	Processing PII
	Interaction	Goal delegation	Objective Delegation	Delegate	-
		Permission	Permission	Permission	-
		Permission Delegation	Permission Delegation	-	-
		PI provision	Information Provision	-	-
		Monitor	Monitor	Monitor	Monitor
		Goal Trust	Goal Trust	-	Trustworthiness
		Permission Trust	Permission Trust	Trustworthy	Trustworthiness

Dimension	Concepts of Our Ontology	Gharib et al.	Martín et al.	Brooks et al.
			Processing	
Risk Dimension	Risk	Risk	Risk	Risk
	Threat	Threat	-	Problematic Data Action
	Problematic Data Action	-	-	Problematic Data Action
	Intentional Threat	Intentional Threat	-	-
	Casual Threat	Casual Threat	-	-
	Vulnerability	Vulnerability	-	Vulnerability
	Attacker	Attacker	-	-
	Attack Method	Attack Method	-	-
	Impact	Impact	Impact	Impact
	Threaten	Threaten	-	-
	Exploit	Exploit	-	Exploit
Treatment Dimension	Privacy Requirements	Privacy Requirements	Requirements	Privacy Requirements
	Mitigate	Mitigate	Mitigate	Mitigate
	Privacy Criterion	Privacy Constraints	-	Privacy Objectives
	Privacy Mechanisms	Security/Privacy Mechanism	Mechanisms	-
	Privacy Control	-	Privacy Control	Privacy Control
	Privacy Policy	Security/Privacy Policy	Privacy Policy	Privacy Policy
Privacy Dimension	Privacy Goal	Security/Privacy Goal	Privacy Principles	Privacy Principles
	Notice	Notice	Notice	-
	Anonymity	Anonymity	Anonymity	-
	Transparency	Transparency	Openness	Transparency

Dimension	Concepts of Our Ontology	Gharib et al.	Martín et al.	Brooks et al.
	Accountability	Accountability	Accountability	Accountability
	Consent and Choice	-	Consent and Choice	-
	Collection Limitation and Information Minimization	-	Collection Limitation and Information Minimization	Minimization
	Use Limitation	-	Use Limitation	Purpose Specification and Use Limitation
	Disclosure	-	Disclosure	-
	Access and Amendment	-	Access, Correction and Deletion	Access and Amendment
	Security/Safeguards	-	Security/Safeguards	Security
	Information Quality	-	Information Quality	Quality and Integrity
	Enforcement	-	Enforcement	-
	Information Flow	-	Information Flow	-
	Sensitivity	-	Sensitivity	-

Annex B: Privacy Requirements Elicitation Algorithm

Input:

Organization, agents and their role, top level privacy goals (TLPG) and their tasks, and asset (PI)

Output:

Privacy requirements (PR) that satisfy top level privacy goals (GoalPR), privacy requirements that mitigate threats (ThreatPR), and privacy requirements that satisfy privacy goal and that mitigate threats (BothPR). GoalPR, ThreatPR, and BothPR are requirements array.

Begin

Load or instantiate the core privacy ontology

Get the organization that the user chooses

GetRelated agents with their role and display them

Repeat until the user needs no further requirements elicitation

Get agents with their role that the user chooses

GetRelated TLPG of the agent, and display them for further analysis

Repeat Until the user inputs another agent with role

Get the TLPG that the user chooses

GetRelated goals that help the achievement of the TLPG and display them

Get sub-goal that refine the upper goal and that the user chooses

GetRelated sub-sub-goals that help the achievement of the sub-goal and display them

Get sub-sub-goal that refine the sub-goal and that the user chooses

GetRelated tasks that can satisfy the chosen goal and display them

Get the task that the user chooses

GetRelated operational tasks that operationalize the upper task and display them

GetRelated Assets (PI) that the agent wants to protect and display them.

Get the asset that the user chooses to protect

GetRelated Threats that threaten the asset and display them.

Get the threat that the user chooses to mitigate

GetRelated Vulnerabilities that are exploited by the threat and display them.

Get the vulnerability that the user chooses to mitigate

GetRelated privacy requirements (PR) that mitigate the vulnerability and display them

Call RequirementsClassifier with OT and PR

Display list of Name of organization, Agent and their role, Assets, Threats and Privacy requirements (shuch as GoalPR, ThreatPR, and BothPR)

End For

End For

End

RequirementsClassifier with OT and PR

Begin

Initialization: GoalPR = 0, ThreatPR = 0, BothPR = 0

Initialization: k = 0, j = 0

Repeat until k <= OT. Length

Repeat until j <= PR. Length

If OT[k] = PR[j] then

BothPR.Add (PR[j])

Else

GoalPR.Add (OT[k])

ThreatPR.Add (PR[j])

End if

End For

End For

Return GoalPR, ThreatPR, and BothPR

End

Notice:

The operational task(s) in this case are privacy requirements that are specified to satisfy privacy goals and that may not necessarily meant for mitigating threat.

Algorithm B.1 Privacy Requirements Elicitation Algorithm

Annex C: Informal and Formal Questions to the Ontology

Table C.1 Informal and formal questions to the ontology

Privacy Requirements Elicitation Activities		**Informal and formal questions**
Context Analysis	Identify Organization	**What are the organizations in the scope of the project?**
		Organization(?OrganizationName) -> sqwrl:select(?OrganizationName)
	Identify Actor	**Who are the Agents and their Roles in the chosen organization?**
		Organization_has_Actor (OnlinePHRCompany, ?AgentName) ^ Agent_plays_Role(?AgentName, ?RoleName) -> sqwrl:select(?AgentName, ?RoleName)
	Analyze Privacy Goal	**What are the chosen Agents' Privacy Goals? (For example "Alice's Goals"?)**
		PrivacyGoal (?PrivacyGoalList) ^ Agent_aims_Goal (?AgentName, ?PrivacyGoalList) -> sqwrl:select(?AgentName, ?PrivacyGoalList)
		PrivacyGoal (?PrivacyGoalList) ^ Agent_aims_Goal (Alice, ?PrivacyGoalList) -> sqwrl:select(?PrivacyGoalList)
		PrivacyGoal (?PrivacyGoalList) ^ Agent_aims_Goal (Pawlos, ?PrivacyGoalList) -> sqwrl:select(?PrivacyGoalList)
		What are the sub-goals that help the achievement of the chosen goals? (For example "ConsentandChoice"?)
		PrivacyGoal (?PrivacyGoalList) ^ isHelpedBy (?PrivacyGoalList, ?SubGoalList) -> sqwrl:select(?PrivacyGoalList, ?SubGoalList)
		isHelpedBy(Consent_and_Choice, ?SubGoalList) -> sqwrl:select(?SubGoalList)
		What are sub-sub-goals that help the achievement of the chosen sub-goals? (For example "Opt-In"?)
		SubGoal (?SubGoalList) ^ isHelpedBy (?SubGoalList, ?SubSubGoalList) -> sqwrl:select(?SubGoalList, ?SubSubGoalList)

Privacy Requirements Elicitation Activities		Informal and formal questions
		isHelpedBy(Opt-In, ?SubSubGoalList) -> sqwrl:select(?SubSubGoalList)
		What are the tasks (means to achieve the goals)? (For example "General consent"?)
		SubSubGoal(?SubSubGoalList) ^ isHelpedBy(?SubSubGoalList, ?TaskList) -> sqwrl:select (?SubSubGoalList, ?TaskList)
		isHelpedBy(General_consent, ?TaskList) -> sqwrl:select (?TaskList)
		What are the sub-tasks that operationalize the chosen tasks? (For example "Check the identity of individual whether patient or health worker"?)
		Task (?TaskList) ^ Task_isSatisfiedBy_OperationalTask (?TaskList, ?OperationalTaskList) -> sqwrl:select (?TaskList, ?OperationalTaskList)
		Task_isSatisfiedBy_OperationalTask (check_the_identity_of_individual_whether_patient_or_health_worker, ?OperationalTaskList) -> sqwrl:select (?OperationalTaskList)
	Identify Asset	**What are the Personal Information (PI) to be protected in the chosen organization? What is description of each PI?**
		Organization_has_PI (OnlinePHRCompany, ?PersonalInformation) ^ Agent (?OwnerAndProvider) ^ Actor_owns_PI (?OwnerAndProvider, ?PersonalInformation) -> sqwrl:select(?PersonalInformation, ?OwnerAndProvider)
		Organization_has_PI (OnlinePHRCompany, ?PersonalInformation) ^ Actor_owns_PI (**Alice**, ?PersonalInformation) -> sqwrl:select(?PersonalInformation)
Risk Analysis		**What are threats that threaten the chosen PI? (For example "Alice's PersonalHealthRecord1"?)**
		PI (?PIList)^ Threat (?ThreatList) ^ Threat_threatens_PI

Privacy Requirements Elicitation Activities	Informal and formal questions
	(?ThreatList, ?PIList) -> sqwrl:select(?ThreatList, ?PIList)
	Threat_threatens_PI(?ThreatList, PersonalHealthRecord1) -> sqwrl:select(?ThreatList)
Privacy Requirements Identification	**What are the privacy requirements to consider to mitigate the risk?**
	Threat (?ThreatList) ^ Threat_exploits_Vulnerability(?ThreatList, ?VulnerabilityList) ^ Vulnerability_isMitigatedBy_PrivacyRequirements (?VulnerabilityList, ?PrivacyRequirementsList) -> sqwrl:select(?VulnerabilityList, ?ThreatList, ?PrivacyRequirementsList)
	Threat_exploits_Vulnerability(Authentication_Attack, ?VulnerabilityList) ^ Vulnerability_isMitigatedBy_PrivacyRequirements (?VulnerabilityList, ?PrivacyRequirementsList) -> sqwrl:select(?VulnerabilityList, ?PrivacyRequirementsList)
Internal Element Node Dependency Analysis	**What are the internal element node such as Goal that depends on internal element node such as PI?**
	InternalElementNodeDepender(?InternalElementDependeeList, ?InternalElementDependerList) ^ InternalElementNodeDependee (?InternalElementDependerList, ?InternalElementDependeeList) ^ Dependum(?DependumList) -> sqwrl:select(?InternalElementDependerList, ?InternalElementDependeeList, ?DependumList)
	What are the Goal that depends on processing of PI and their type of processing?
	Goal_Depender (?ProcessingPIList, ?GoalDependerList) ^ Processing_Dependee (?GoalDependerList,

Privacy Requirements Elicitation Activities	**Informal and formal questions**
	?ProcessingDependeeList) ^ Processing_has_ProcessingType (?ProcessingPIList, ?ProcessingPITypeList) ^ ProcessingPIDependum (?ProcessingPIDependumList) -> sqwrl:select(?GoalDependerList, ?ProcessingDependeeList, ?ProcessingPITypeList, ?ProcessingPIDependumList)
Trust Dependency Analysis	**Who trusts or distrusts who? On what? and for what reason?**
	trusts (?TrustorList, ?TrusteeList) ^ ProcessingPITrust (?TrustList) ^ Trust_has_TrustType (?TrustList, ?TrustTypeList) ^ ProcessingPITrustum (?ProcessingPITrustumList) -> sqwrl:select(?TrustorList, ?TrustTypeList, ?TrusteeList, ?TrustList, ?ProcessingPITrustumList)
	Who trusts who? On what? and for what reason?
	trusts (?TrustorList, ?TrusteeList) ^ ProcessingPITrust (?TrustList) ^ Trust_has_TrustType (?TrustList, Trust) ^ ProcessingPITrustum (?ProcessingPITrustumList) -> sqwrl:select(?TrustorList, ?TrusteeList, ?TrustList, ?ProcessingPITrustumList)
	Who distrusts who? On what? and for what reason?
	distrusts (?TrustorList, ?TrusteeList) ^ ProcessingPITrust (?TrustList) ^ Trust_has_TrustType (?TrustList, Distrust) ^ ProcessingPITrustum (?ProcessingPITrustumList) -> sqwrl:select(?TrustorList, ?TrusteeList, ?TrustList, ?ProcessingPITrustumList)
Delegation Dependency Analysis	**Who delegates for whom? What? for what reason? On what consent? And what the types of processing or permission?**
	delegates (?DelegaterList, ?DelegateeList) ^ ProcessingPIDelegation (?ProcessingPIDelegationList) ^ Processing_has_ProcessingType (?ProcessingPIDelegationList, ?ProcessingTypeList) ^ has_Consent(?ProcessingPIDelegationList, ?ConsentList) ^

Privacy Requirements Elicitation Activities	**Informal and formal questions**
	ProcessingPIDelegatum (?ProcessingPIDelegatumList) -> sqwrl:select(?DelegaterList, ?DelegateeList, ?ProcessingPIDelegationList, ?ProcessingTypeList, ?ProcessingPIDelegatumList, ?ConsentList)
	delegates (?DelegaterList, ?DelegateeList) ^ PermissionDelegation (?PermissionDelegationList) ^ Permission_has_PermissionType (?PermissionDelegationList, ?PermissionTypeList) ^ has_Consent(?PermissionDelegationList, ?ConsentList) ^ PermissionOverPIDelegatum (?PermissionOverPIDelegatumList) -> sqwrl:select(?DelegaterList, ?DelegateeList, ? PermissionDelegationList, ?PermissionTypeList, ? PermissionOverPIDelegatumList, ?ConsentList)
Monitor Dependency Analysis	**To solve the distrusts, Who monitors who? What? for what reason? In accordance with what consent?**
	monitors (?MonitorList, ? MonitoreeList) ^ PermissionMonitor (?PermissionMonitorList) ^ Permission_has_PermissionType (?PermissionMonitorList,?PermissionTypeList) ^ has_Consent(?PermissionMonitorList, ?ConsentList) ^ PermissionOverPIMonitorum (?PermissionOverPIMonitorumList) -> sqwrl:select(?MonitorList, ?MonitoreeList, ?PermissionMonitorList, ?PermissionTypeList, ?PermissionOverPIMonitorumList, ?ConsentList)

Annex D: Evaluation Form

(Controlled Experiment)

Phase 1: Orientation

Aim of the experiment

Evaluate the proposed method and its tool-implementation against a controlled experiment with a case study and a questionnaire.

Participants

The experiment is mainly intended for final year computer science postgraduate students who took Software Engineering and Computer Security and Privacy Courses.

Timing

You will be able to complete the experiment within 30 minutes.

Consent of participation and confidentiality

By filling in the questionnaire you consent to your voluntary participation in this experiment.

The data collected through the experiment will be kept confidential and will be stored securely, and will be deleted after completion of the experiment related activities. This questionnaire is anonym as well as the results obtained from the experiment.

We appreciate your time and effort! Thank you very much,

Wendwesen Belay

Background Information

Age: _____

Sex: _____

Position such as student (MSc or PhD) and/or practitioner (security and/or privacy experts, or requirements engineer):_____

Sector of activity: _____

If you are a student, Program (Regular/ Extension): _____

If you are a practitioner, years of experience: _____

Phase 2: The privacy ontology and its usability

1. Do you find that the privacy ontology has the main concepts for privacy requirements elicitation?
    a. Strongly agree ☐
    b. Agree ☐
    c. Neither agree nor disagree ☐
    d. Disagree ☐
    e. Strongly disagree ☐
    Comments: _____
    _____
2. Does the privacy ontology help in finding new elements (privacy requirements, threats, vulnerabilities …)?
    a. Strongly agree ☐
    b. Agree ☐
    c. Neither agree nor disagree ☐
    d. Disagree ☐
    e. Strongly disagree ☐
    Comments: _____
    _____

Phase 3: Usability of the method

3. Do you think that the method reduces the effort required for the elicitation of privacy requirements?
    a. Strongly agree ☐
    b. Agree ☐
    c. Neither agree nor disagree ☐
    d. Disagree ☐
    e. Strongly disagree ☐
    Comments: _____
    _____

4. Do you think the method is easy to use overall?
    a. Strongly agree ☐
    b. Agree ☐
    c. Neither agree nor disagree ☐
    d. Disagree ☐
    e. Strongly disagree ☐
    Comments: _____
    _____

Phase 4: Usability of the prototype implementing the presented ontology and method

5. Do you find the interface to access to the core privacy ontology easy to use?
    a. Strongly agree ☐
    b. Agree ☐
    c. Neither agree nor disagree ☐
    d. Disagree ☐
    e. Strongly disagree ☐
    Comments: _____
    _____
6. Is this prototype friendly (clear and easy) to use?
    a. Strongly agree ☐
    b. Agree ☐
    c. Neither agree nor disagree ☐
    d. Disagree ☐
    e. Strongly disagree ☐
    Comments: _____
    _____

DECLARATION

I, the undersigned, declare that this thesis is my original work and has not been presented for a degree in any other university, and that all source of materials used for the thesis have been duly acknowledged.

Declared by:

Name: _Wendwesen Belay_____

Signature: _____

Date: _____

Confirmed by advisor:

Name: _Mesfin Kifle (PhD)_____

Signature: _____

Date: _____