UNIVERSITAT AUTÒNOMA DE BARCELONA

DEPARTAMENT D'ENGINYERIA DE LA
INFORMACIÓ I DE LES COMUNICACIONS

A GENERAL-PURPOSE SECURITY FRAMEWORK

SUBMITTED TO UNIVERSITAT AUTÒNOMA DE BARCELONA
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE
DEGREE OF DOCTOR OF PHILOSOPHY IN COMPUTER SCIENCE

PROGRAMA DE DOCTORAT EN INFORMÀTICA

by Miquel Colobran Huguet
BELLATERRA, SEPTEMBER 2015

Supervisor :
JOSEP M. BASART MUÑOZ

I CERTIFY THAT I HAVE READ THIS THESIS AND THAT IN MY OPINION IT IS FULLY ADEQUATE, IN SCOPE AND IN QUALITY, AS A DISSERTATION FOR THE DEGREE OF DOCTOR OF PHILOSOPHY

Bellaterra, September 2015

Dr. Josep M. Basart Muñoz
(Supervisor)

A totes les persones *que m'han ajudat i animat en aquesta aventura, especialment na Yolanda Cardiel, en Josep Maria Arqués i el meu director en Josep Maria Basart.*

There is no way to security;
security is the way [a]***.***

[a] Rewritten from Mahatma Gandhi's quote : "There is no way to peace; peace is the way."

PREFACE

In general, one has the feeling that a work of this kind is completely systematic. One thinks that there is a starting point. Once established, one supposes, it will be a lot of complex work and it requires months or years. This view assumes that the job is clearly defined and it *just* has to document "*as a diary*" the path which has been made.

Reality is nothing to do with that. Those works are, each and every one of them, a new product and that means they are, in a high percentage, an art. There is a small percentage of inspiration that tells how you could orientate work. From there, as any artist, one should do sketches, many sketches of parts of the work. One has to study how the elements relate each other in order to obtain an harmonious work. One has to test, search out items to bring to the work and study how to get a coherent final work.

A study of this dimensions requires many years of preparation, a lot of effort and many people taking part in order to help in choosing colors, position, shape and therefore, somehow they are involved in the outcome of this piece.

Finally a day comes when the artist exhibited his work in public. Like all complex and polychromatic work, some people evaluate technical aspects, others evaluated some details or the whole composition. Maybe, some elements are not seen at first glance. As any work it is subject to a subjective component and the peace don't have to get an unanimous judgment.

Anyway, this is the art piece. All together with its colors, tones and shades.

ACKNOWLEDGMENTS

The history of this work, as all of these kind, is long. Allow me to digress a bit. Thesis creation may seem a solitary activity. This is half true. I have been very lucky and I have had the constant support of many people.

I must thank specially the unconditional support I received from my thesis advisor Dr. Josep M. Basart. Without him, this work would not exist. Also I stake my life on that he is the best advisor a student could wish. He knew from the beginning that my work would not only encompass technical aspects but others newer and unexplored aspects of computing. I found out much later.

Yolanda Cardiel and Josep Maria Arques always supported me and endured my long talks full of doubt. Without you I would give up several times.

Dra. Esther Morón taught me that computer science is no longer a purely technical subject and it needs from other disciplines in order to succeed in the twenty-first century society. She always have given me valuable advices.

Dr. Arcadi Oliveres gave the view that historically security had been integrated, and currently is forming a knowledge field. So, through my eyes, I sow clearly that computer security was more than a pure technical area. It has to be one more of the securities in the existing world too. Instead of being considered a different security, it had to become part of the general idea of. I owe him, therefore, that he has expanded my concept on what security is.

Dr. Miguel Angel Lopez and Dr Josep Pujolar had the patience to resolve doubts as well as specific issues that have arisen over the course of the thesis. I was bothering them occasionally with “complex” questions for me (not for them) and I ‘stole‘ ” ten minutes or a lunch time even. Their patience was superb.

I really appreciated the support and assistance received from The Consorci de Serveis Universitaris de Catalunya (CSUC) that allowed me to apply my research to the CSUC institution. Particularly to Dr. Miquel Huguet Vilella and Mr. Jordi Guijarro. Both have made me this part of the research quite easy.

I am very thankful for suggestions from many people such as Dra. Mercè Villanueva, Dr. Miguel Angel Garcia, Dr. Vicent Alcantara, or all the chats with Dr. Vittorio Galleto throughout those years. All suggestions have been successful and valuable.

I am specially thankful to Dr. Stephen Cheskiewicz and his family. They taught me that friendship has no boundaries and this research has made me this priceless gift. The support and courage received has been outstanding.

There are many other people who have helped one way or another in the work in order to reach its end. So, Dr. Rafael Boix, Dr. Joan Pasqual or the SLIPI of the Faculty of Economics and Business. The latter have made their personal efforts to provide me support and resources (a quiet place) whenever I needed it.

To my family, I thank the support I received. All of them have stood silently and patiently during this years.

Finally, who put the seed of this dream many, many years ago, Maria Rosa Calvo and Montse Ros into Toni Coarasa and me. Maria Rosa and Montse taught us the value of work and study when we was about 10 or 12 years old. Toni defended his thesis in 1999[1] and I do it later.

And thanks all the people (there are many) who I have not mentioned (unintentionally). Those people consciously or unconsciously made this adventure real. If I forgot some person is the result of an oversight and there is no bad-intention, but without the persons mentioned, most likely this work would not have begun even.

This work represents a first attempt to make computer information security an area that is part of the knowledge area of security.

In your hands you have, therefore, a document which proposes that computer security is in relation to the rest of securities, very alike. A means of protecting what is considered valuable from potential threats that may damage.

Thank you everyone.

[1] Quantum Effects on some low and high energy processes beyond the standard model. Directed by Dr. Joan Solà

ABSTRACT

Computer Science has undergone major transformations throughout its short history. It started with great machines and very restricted and specialized environments and It has become in small devices that are part of society and daily life of every person. Security has been one of the areas most affected by those changes and has undergone major changes in technology also. For this reason, we think that the "traditional" definition of computer security is narrow, especially if we consider the new securities that have appeared in other areas of knowledge. Current definition comes from the 70s and security, in the twenty-first century, is conceptually, theoretically and practically something different.

Therefore, the main objective of this thesis is review the concept of computer security itself in order to propose a definition together with a framework model capable to be implemented. In order to achieve it, an analysis method is proposed. The analysis method is based on conceptual methods of obtaining knowledge (knowledge acquisition) used in knowledge engineering. The conceptual model is performed using the Class Diagram (UML) as a graphical representation language. After that, apply the proposed method to a set of selected sources, in order to obtain the model. The conceptual model of the concept of security is expressed as a set of concepts and relationships among concepts

Based on the proposed model, an algebraic expression of the concept of security is drawn, and finally the model is implemented by means of a knowledge-based system using an ontology.

Consequently, the study's principal contributions are the development of a methodology of conceptual analysis and a definition of security along with its framework.

The framework is expressed in algebraic manner also and is capable to be implemented using technologies such as Java, providing security metrics.

The structure of the thesis is as following: In part 1, a theoretic approach to the study of security, paying attention to other disciplines not related to engineering. An historical approach of the study of the concept of security is made, having special attention to those concepts or models proposed by scholars in the field of security (not exclusively in the field of computer security). Part 2 explains the tools used to build the model. Modeling tools are used both conceptual and knowledge based ones. A method of analysis is constructed and used in the model design. In part 3 a generic model of security is proposed. The aim is to propose an integrative model that includes many of the existing securities. Additionally an algebraic formulation of the security model is made. Finally, part 4 is dedicated to apply the proposed model to a real scenario. This demonstrates that the model is operative and capable to measure the level of security.

keywords: security, knowledge modeling, framework, computer security, metric, ontology

CONTENTS

LIST OF FIGURES

LIST OF TABLES

Chapter 8 **127**

Chapter 10 **161**

Appendix B **203**

Appendix C

Appendix E **231**

Appendix G **255**

Introduction

CHAPTER

1

Introduction

"Growth demands a temporary surrender of security."

— Gail Sheehy

Contents

This chapter describes the reasons for writing this work and the objectives to achieve. Thesis structure is also shown.

Security has been a major concern for humankind. It is a concept present somehow in all aspects of human life. The word “securitas” appears in Cicero (first century BC) as a philosophical term, evolved to a social concept with Thomas Hobbes (1588-1679) and later a government matter until the end of cold war. Just a few decades ago, Buzan [Buz83] highlighted a lack of conceptual work. Therefore, the concept of security has been highly reviewed in International Relations (I.R.). Nowadays security can be seen from many perspectives and thus analyzed from any of them such as the societal, psychological, economic, technological, geopolitical, philosophical, human or even an environmental point of view.

Despite the fact that security is a transverse concept to many knowledge fields, it has been modeled, applied and developed in a different manner depending on the area in which it has been used. Areas such as Philosophy, Social Science and lately Computer Science have focused on the notion of security.

Computer Science was born in 1940 with von Newman work [vN93]. The first security model appeared in 1976 [BL76]. From this original work, many other models have been created (for example Chinesse Wall [CW87]) until the current model [SFK00]. Many of these changes or refinements of the initial model have been produced as a consequence of technological changes (the advent of Internet, the emergence of the personal computer or the massive use of digital technologies through small digital devices such as PDAs, cell phones or GPS trackers).

Nowadays the widespread use of digital systems has made computer science become intertwined at all levels of society, constituting an inherent element. The software has evolved to become tools that are simple to use without any computer skills like email. There are forums, blogs, wikis or social networks like Facebook® or Whatsapp®. Therefore, this new situation makes computer technology have technical, social and human dimensions at the same time. Bearing in mind that almost everyone uses some of those applications we are facing a social phenomena.

One of the fundamental features of social facts, as sociologist Durkheim[2] maintains, is that they are “something more” than the sum of individuals, just the same as a person is not the simple sum of its cells or the activity of a computer is not only the exchange of electricity between its transistors. All form a higher-order entity and should be treated and studied using this paradigm. Computing is now

[2] The term *fact* is used in a broad sense. We are not interested now on starting any discussion about whether it is a fact or phenomenon. It has been widely studied by Durkheim.

intertwined in the social fabric and thus it carries its own set of new problems.

Therefore, despite computer science is social fact, their security models, with no more than 50 years of antiquity such as [CW87] [BL76] [BN89], are barely connected with works on security from the International Relations field [Wol52] [Buz83] [Rot95], and the underlying notion.

As a result, nowadays there are several "securities" and a number of models from different fields. Despite the importance of the issue, surprisingly, there is no common vocabulary, procedures, definition or model to share knowledge about security. The accepted definition of security in the International Relations field was made by Wolfers [Wol52] and there is no one accepted in Computer Science field. For example, the definition of computer security made by ISO/IEC 17799:2005 [ISO05] has no relation with the one used in I.R. field.

Several authors [FW06] [And03] [vSvS05] claim the necessity to review the security concept in order to integrate those models and concepts in a more general framework. Having a common body of knowledge (CBK) [TG07] has advantages such as shared vocabulary, knowledge, development or metrics

1.1 Preliminary Note

This brief note is written for the main purpose to emphasize differences between *security* and *safety*. The reason comes from the necessity to avoid using confusing terms.

Security and *safety* are words that seem clear and precise at first glance, but they may have very different meanings depending on the context. For sure it's easy for native speakers to manage two terms into their languages. But there are real linguistic traps for the others, as Ludovic Piètre–Cambacédès and Claude Chaudet highlight in its paper *The SEMA referential framework: Avoiding ambiguities in the terms security and safety* [PCC10]. It is really hard for languages which manage one word for the two concepts because ambiguity is always present. Thus, "Linguistics and translation are responsible for some of the ambiguity regarding the terms safety and security" [PCC10]. Languages as Catalan (seguretat), Danish (sikkerhed), Portuguese (segurança), Spanish (seguridad) and Swedish (säkerhet) use just one word to define security and safety. English (security and safety) and French (sûreté and sécurité) use two.

European Union provides as much as 23 official languages, English included.

Thus, is very difficult to manage terms such as.

As this dissertation comes from a person whose mother tongue (Catalan) uses one word for both terms, the author really tried to make the correct use of the two English terms, but the author apologizes in advance for any mistake introduced into the work.

1.2 Motivation

In computing, although it appears otherwise, security is an underdeveloped concept from a conceptual viewpoint, since it is restricted to technical security and protecting the system along with its information. Classical works as Bell-Lapadula [BL76] or Clark Wilson [CW87] are focused on protecting access to information. Reality about that is with the emergence of networks and Internet in the nineties and personal computers in recent years, the concept of security has become much broader and even more technical. Thus we find areas such as network security, security systems, security and response to incidents or computer forensics. Looking beyond the field of computer security, it shows that safety is a topic widely discussed and debated for many years (several hundred indeed) to areas of knowledge such as Philosophy, Social Science or International Relations. Thus, we are convinced that reviewing the notion of "security" in these areas can provide us a wider vision and generic models than any of the models that come from engineering. Therefore, the object of study of this thesis is the exploration of new alternatives based on other concepts of security.

1.3 Thesis goals and contributions

The work is mainly focused on the security concept and its modeling. The goals the author hopes to achieve in this research are:

- Review the concept of security in order to probe that computer security is not "one of a kind" but "a kind of".
- Integration of many securities under one framework and definition. This research aims at presenting a flexible security framework that could be common to various disciplines and in turn allowing the use of a common language.

- The framework has to be suitable to be implemented. It's important also, but not indispensable, to get some security metrics.

The main objectives of the thesis have been split into smaller objectives.

- Use or create a conceptual methodology to highlight the elements, concepts and relations of a concept. The methodology is later applied to the security notion in order to obtain a model. That objective is achieved by means of Knowledge Engineering.
- Make a definition of security based on the previous findings. Current concept of security, far from outdated, is narrow. The work should lead us towards a broader concept of security capable to include security the way it is currently understood along with many other existing “securities” in our society.
- Formalize the model of security. A formal expression of security allows making further research in the theoretic field.
- Use a methodology to model a concept under a framework suitable to be implemented. The use of knowledge modeling techniques allows the creation of very flexible and rigorous knowledge models that can be implemented in information systems. It's important also, but not indispensable, to get some security metrics. That objective is achieved by means of Java and Ontologies.

Contributions of the thesis are in several aspects.

- Theoretical and fundamental research contributions:
  - Define and construct a generic security model. Analyze whether this proposed security model can be applied to different areas, and specifically in the field of computer security.
  - Specify if computer security could be included into a broader safety concept.
  - A definition of security.
  - A formal expression of security.
- Methodological contributions:

  - A generic conceptual analysis methodology for building models that could be implemented is proposed.
  - Time dependency appears as an essential element and the framework reveals time dependency as a key element that affects the level of security. A model of security based on time is proposed.

- Applied contributions:

  - Prototype model made in Java.
  - Security Metrics.
  - A knowledge base ontology.

A unified security framework is useful in several areas because it uniforms the vocabulary, the way of handling security and also provides a common referent. As mentioned, the researcher considers extremely important for the study of security to survey other fields of knowledge such as International Relations and Philosophy. Especially when they have studied the issue for centuries.

Broadly speaking, security research has a few works on the security concept and a cornucopia of operational concepts on the International Relations as well as on Computer Security fields. Indeed most research about computer security relies on Information Security and thus by itself, it implies an operational concept. In this sense, most studies are constrained by this primary assumption. Computer security is based mainly on protecting confidentiality, integrity and availability of information (CIA triad) and so are their formal models.

Therefore computer security, as a concept, has been subordinate to the protection of information. Lately the concept has been expanded slightly and currently security is not only about preserving information. Current technologies comprise protecting information from interception, modification and destruction by means of techniques such as backup systems, authentication, firewalls, IDS or honeypots.

In short, there is a need to redefine computer security in more global terms in order to obtain more generic and flexible models. It has to be defined not just in technological terms but also in social, legal or human terms.

1.4 Thesis Outline

The document is divided into 4 parts, 11 chapters and 7 appendices structured as follows (Figure 1.1):

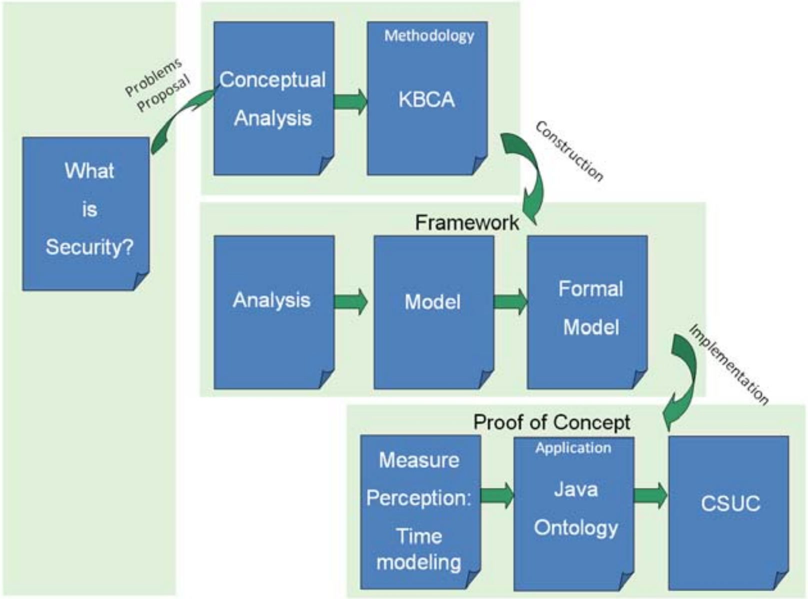


Figure 1.1 – Graphical outline of the research.

First part reviews the fundamentals of security. **Chapter 2** makes a review of the concept of security throughout history. This research aims to identify how the concept evolved, and the current situation.

Second part is focused on establishing a generic conceptual modeling methodology for security (**Chapter 3**). To achieve it, we have to create a methodology that allows us to obtain the model. **Chapter 4** reviews the set of tools necessary to be able to generate a model from informal text descriptions. The methodology proposed is general and thus it can be applied to any field.

Third part, starting from the methodology proposed in the previous part, builds the model (**Chapter 5**). In **Chapter 6** an analysis of the obtained framework is carried out. The chapter also explains how this model, which is intended to be integrative, fits with existing securities. A formal description, in algebraic form, of the model obtained is made in **Chapter 7**.

Last part exhibit the validation of the proposed model. The model is applied to a real case. In order to achieve it, a lightweight model of perceived security and its time dependency is proposed **Chapter 8**, a java implementation of the model is drawn in **Chapter 9** and applied to a real scenario (**Chapter 10**).

Finally (**Chapter 11**), we summarize main conclusions of the work presented and a number of future research lines that raise from this research.

As part of the memory, several appendices are included in order to clarify and complete some of the contributions of this thesis. In **Appendix A** information related to the extraction of knowledge to create the model is detailed. **Appendix B** and **Appendix C** contain two examples of the methodology for creating the security framework. In **Appendix D** the article on the methodology of extracting knowledge from text [CB13] is found. **Appendix E** is composed by the diagrams used in the case study. **Appendix F** contains data relevant to this research from the survey carried out with Dr. Stephen Cheskiewicz. Finally, **Appendix G** contains an example of the security level lightweight model.

Bibliography

[And03] James M. Anderson. Why we need a new definition of information security. *Computers & Security*, 22(4):308 – 313, 2003.

[BL76] David E. Bell and Leonard J. LaPadula. Secure Computer System: Unified Exposition and Multics Interpretation. Technical Report MTR-2997, The MITRE Corporation, Bedford, MA, USA, March 1976.

[BN89] D. F. C. Brewer and M. J. Nash. The chinese wall security policy. In *IEEE Symposium on Security and Privacy*, pages 206–214. IEEE Computer Society, 1989.

[Buz83] Barry Buzan. *People, States and Fear*. Harvester- Wheatsheaf, Brighton, 1983.

[CB13] Miquel Colobran and Josep M. Basart. Knowledge based concept analysis method using concept maps and uml: Security notion case. *World Academy of Science, Engineering and Technology*, 7(2):437 – 444, 2013.

[CW87] David D. Clark and David R. Wilson. A Comparison of Commercial and Military Computer Security Policies. *IEEE Symposium on Security and Privacy*, page 184, 1987.

[FW06] Stefan Fenz and Edgar Weippl. Ontology based it-security planning. In *PRDC*, pages 389–390. IEEE Computer Society, 2006.

[ISO05] Switzerland ISO, Geneva. Code of practice for information security management. Norm ISO 17799:2005, International Organization for Standardization, 2005.

[PCC10] Ludovic Piètre-Cambacédès and Claude Chaudet. The SEMA referential framework: avoiding ambiguities when dealing with security and safety issues. In *4th Annual IFIP WG 11.10 International Conference on Critical Infrastructure Protection (CIP 2010)*, Washington, DC, USA, Mar 2010. Not included in the conference proceedings (selected for publication in IJCIP).

[Rot95] Emma Rothschild. What is security? the quest for world order. *Daedalus*, 124(3):53–99, June 1995.

[SFK00] Ravi S. Sandhu, David F. Ferraiolo, and D. Richard Kuhn. The nist model for role-based access control: towards a unified standard. In *ACM Workshop on Role-Based Access Control*, pages 47–63, 2000.

[TG07] M. Theoharidou and D. Gritzalis. Common body of knowledge for information security. *IEEE, Security & Privacy*, 5(2):64–67, 2007.

[vN93] John von Neumann. First draft of a report on the edvac. *IEEE Ann. Hist. Comput.*, 15(4):27–75, October 1993.

[vSvS05] Basie von Solms and Rossouw von Solms. From information security to...business security? *Computers & Security*, 24(4):271 – 273, 2005.

[Wol52] Arnold Wolfers. National security as an ambiguous symbol. *Political Science Quarterly*, 67(4):481–502, 1952.

Part I

Security Essentials

CHAPTER

2 Security Background

"Distrust and caution are the parents of security."

— Benjamin Franklin

Contents

This chapter reviews the concept of security, which is one of the most important and influencial concepts. Security has been a key concept in the development of society.

The term security is ambiguous in contents and form, making it difficult to define and therefore perform modeling and subsequent implementation. Besides, security is not an absolute term. There is no completely safe or unsafe system.

The work begins by analyzing the key definition, which is the basis. The author refers to the concept of security. It's hard to define security. The word security, probably in its broader sense, is referred to the absence of risks [NG03]. Although one has to bear in mind that the term has multiple uses and can take different meanings depending on the area or field to which it refers. It's very different to talk about national security (the security of a state), road safety (the safety of pedestrian or cars), safety associated with a nuclear plant or safety from government to citizens in front of a natural disaster like an earthquake.

The result of defining security greatly differs depending on the applied field. For example, reviewing definitions of security in the field of International Relations [Wol52] [HA10] and definitions of security in the Computer Science field [ISO05] [Lan01] reveals that there is barely anything in common. The former defines security in terms of threats, fear and values. The later in terms of preservation of information, focusing on confidentiality, integrity and availability known as the CIA triad in Computer Science.

Therefore, there are many security definitions. In this sense, security is considered a multidimensional concept. A concept is multidimensional when "...its concepts are categorized according to different characteristics, and thus showing their different dimensions" [Kag97].

2.1 Security Concept, a blurred notion

The concept comes from the Latin securitas and refers to the quality of safe, in other words, that is free from harm or risk. Something sure is somewhat true, firm and indubitable. Security, therefore, is a certainty.

The dictionary, in its effort to define terms, provides a definition of security that is inaccurate. Thus, the Oxford English Dictionary includes "security" as a quality of "safe". As an adjective, "safe" is defined as free from hurt or damage and clear of any danger, harm or risk, sure, indubitable and somehow infallible. Security is, in this usage, a quality of human beings and things that have their freedom unrestricted.

Unfortunately, the definition introduces more confusion. "On one hand the concept of security itself, ... ambiguous and indeterminate and, on the other, a huge burden of subjectivity" [Gse98].

Dictionaries emphasize the quality of “safety” of the security notion within the meaning of free and clear of danger, threat, harm or attack. A closer look at these terms reveals that in fact they are not synonymous but worse, introducing other senses to the concept of security. This could be thought to induce more and better theoretical discussion and therefore tends to improve the concept but it introduces “noise” that impedes the clarification and does not incorporate new elements. As a result, security is compared with concepts such as justice, freedom, peace or power. All of them go to be considered within the field of security and the debate becomes complex, with multiple elements and security left as being a central concept and will become a concept often entailed to other elements.

Much of this debate has its historical roots. Traditionally they have tended to identify security with military capabilities and/or to a lesser extent, economic capacity (key to building military capacity, as a nation’s economic resources are the basis of their ability to wage war). Consequently, the area of study that is centered in security is the field of International Relations.

Recently, security has been seen from a broader perspective. So, in the words of Richard Ullman, “defining national security merely (or even primarily) in military terms conveys a profoundly false image of reality” [Ull83].

Reality about the concept of security is much more complex. As safety is part of many aspects of our lives, this causes the term to become full of nuances. One could speak of “multiple securities” or multi-dimensional concepts of security. The different dimensions of security are intimately linked to each other forming a whole complex and deep.

2.2 Historical review

There is agreement, at present, that the term “security” has a positive value. Now, which has been the relationship between the label “security” and the notion that has been associated in different historical periods? In fact, the term “security” has been combined with many different concepts, not always in a good way.

Different notions of the term have sought to adapt needs over time, to the extent that its development has gone hand by hand with the development of social organizations. Thus the concept of security has undergone deep changes, theoretical and practical.

The word “securitas” appears in Cicero (first century BC) as a philosophical term meaning “absence from grief / tranquility of mind”. Cicero associates “securitas” with “joy for life”, that is, the traditional Graeco-Roman philosophical ideal of eudaimonia (usually translated as happiness). By Seneca, two generations later, “securitas” is thought to be “good characteristic of wise men” (almost identical to eudaimonia). Considered as the goal of philosophers to overcome the fear of death, and in this respect equal to God. With the Emperor Augustus the concept becomes political, so “securitas” is linked to “Pax Romana”. Because of that the meaning is associated with stability or tranquility.

For Christians the concept “security”, that comes from Seneca, means “certitudo” (certainty), and therefore highly ambiguous. Christians thought that only God had awareness and certainty of salvation. For humans, as Weaver described [Wae09], this kind of “security” would be presumptuous.

The Middle Ages, period from 476 AD until the fall of Constantinople held by the Turks in 1453, were characterized by an organization of social life that took as basis a universal structure of religious domination and local political power structure. From these two structures are derived the precarious sense of human security that prevailed in medieval Europe; the feudal state.

The development of the feudal structure was made possible through mercantilism and trade that took place in the cities that make up with a common monarch. Therefore, the state arises, with three key elements: territory, population and government. It is the need to provide security for these items under the responsibility of the governor of state. Thus, security became a political term and a goal to achieve.

The nobles had the need to maintain security and control of their kingdoms, some of them located far-reaching, so they were forced to delegate power to local control. The peasants demanded security because of barbarians of the surrounding land and the presence of thieves. It was also necessary to provide security against invading armies. With this scene, the development of the feudal system and structure was an almost inevitable fact. However, all this came at the great expense of the common man. During this period the concept of security was associated with protection. On a practical level, security was based on principles of intimidation and deterrence (not on principles of partnership and cooperation as will later) as a means to achieve peace and security.

With the dismantling of the feudal system, new social structures arise. A new structure of order and power became necessary and centralized power made its public appearance. This kind of power is able to regulate the new and the more complex social relations generated by the expansion of trade and new technologies. The historical response to this need were the absolute monarchies, which spurred the development of social regulatory capacity of the State.

The work of the philosopher Thomas Hobbes (1588-1679) is considered as the break line with the Middle Ages. Hobbes made "security" to the central notion of the modern state. In his writing considers the security as the peace derived from the "social contract" that citizen hand over the "power" to the State (Leviathan), to the detriment of their freedom to ensure their safety. The State, thus, becomes, as the guarantor of security (seen as peace). Security is referred in particular to goods or property.

Contemporary philosophers and subsequents, follow a similar thought than Hobbes. Thus, Leibniz (1646, 1716) argues that the State must assure to citizens the fair and peaceful coexistence of the human community itself. This means that the State is a provider of security. Rousseau (1712, 1778) is characterized by being the first really democratic thinker. Rouseeau believes the state was born of a free agreement among men who have joined together to designate the governor. As the president has been elected by the people, at any time, when the people want, it can be replaced. At the same time, the mission of the rulers is always doing the will of the people. So that, security remains a national affair. Although the State, in this case, is appointed by the people. No variation on the concept of security is made by Montesquieu (1689-1755). His argument on the State and individual freedom is that it can only be maintained if the government's powers are divided.

The French Revolution of 1789 swung the concept of security from a matter of the State to a personal matter. Thus, The Declaration of Rights of Man and Citizen approved by the National Assembly of France (August 26, 1789), in its Article 2 lists the natural and inalienable rights of man, which predate established powers and are considered to be applicable anywhere and any time:

> "
>
> *Article II*
>
> *The aim of all political association is the preservation of the natural and imprescriptible rights of man. These rights are liberty, property, security, and resistance to oppression. "*

Security, finally, finds its specific identity and is no longer an issue exclusively for the state to be concerned with the public good.

In the twentieth century, the concepts of state security and the role of the nation are strengthened and evolve tightly accordingly to the political and geopolitical scenarios. The great wars between states national and / or alliances of them, allow a sustained evolution of security concepts.

The notion evolves into broader concepts that integrate various elements of the national state and contribute quantitatively and qualitatively. This leaves safety a matter exclusively of state. Its concept is no longer associated only to power, politics and military. Other elements come into play such as economics, politics or technology. These elements came to be directly related to safety and they contribute to state security, national security, human security, political security, ecological security, among many others that are no longer the responsibility of the armed forces or the State.

This evolution can be seen especially from the end of the Cold War, where the center of the security notion is transferred from national interest within a bipolar world to the disappearance of one of the blocks that requires another adaptation to the design of international security to new and constantly changing realities. The study of security is greatly enhanced, and the work of Buzan [Buz83] People, States and Fear shows that the concept of security, nevertheless, was relatively underdeveloped. In his remarkable survey he points out that most of the work on security came from the field of empirical strategic studies for which "security" is the core concept. Thus, in general, the core concept is not developed at all.

Emma Rothschild, meanwhile, in his remarkable article "What is security?" [Rot95] describes the directions that security have been extended since the early nineties.

Vertically security has changed in two ways:

- From the security of nations to the security of groups and individuals.
- From the security of nations to the security of international system.

According to Rothschild, this concept has also extend horizontally:

- From military to political, economic, social , environmental or "human security"

- From the government to press, local government and all kind of abstract forces

The actors responsible for security have multiplied. Today, the State must share its central role in this field with a number of new actors such as international institutions, governments local and regional NGOs, opinion public and even market forces.

So, in last decades, the safety study as a concept has been carried out from multiple points of view and in turn approached as a multidimensional notion. Also great changes have generated an intense debate and plenty of work around the concept of security as an analytical concept and model. Nowadays security has to be considered as global, universal and indivisible.

2.3 Philosophical background

The interpretation of safety has generated two main lines of thought in the world. The idealistic and realistic. The first, legislative, wants to achieve security through a set of rules that allow peace[3]. The realistic line of thought raises the situation "as is" and looking for solutions to obtain security[4].

The concerning or principle is Power by realists and Peace by Idealists. Security is conceived by idealists as a result of permanent peace and by realistics as the result of the exercise of power.

The currents of thought spread, and a systematization is proposed by M. Wight (later on fulfilled by his disciple H.Bull) in three main currents. In this sense it is better to speak of ideal types of classification or dominant lines of thought.

2.3.1 Realism

Realism (in international relations theory) is one school of thinking within is prioritized national interest and security over ideology, moral concerns and social reconstructions. This term is often synonymous with power politics.

Realism was born with Thomas Hobbes, the first author to include security in their philosophical problems. In his book "Leviathan" Hobbes attributed the task

[3] Some greatest exponents of this line of thinking are A. ZIMMERN (1936) *The League of Nations and the Rule of law*, M.ANGELL (1910) *The Great Illusion* and B.RUSSELL (1936) *Which Way to Peace.*

[4] Some greatest exponents of this line of thinking are E.H. CARR (1981) *The twenty years Crisis* and MORGENTHAU (1960) *Politics Among Nations*.

of preserving the integrity of the citizens and to free the individual from the uncertainties of the anarchic nature of the world. In other words, to provide security. As mentioned by Gabriel Orozco:

> *“ For Hobbes the security concept is not restricted only to the security of physical existence, but goes further and extends also to social stability that allows to enjoy a life free of threats ”* [Oro06]

The conceptual breakthrough that makes Hobbes on security is not identify protection with security as had been so far. Throughout the Middle Ages, the territories were forced to fend off invasions of the barbarians and therefore the castles were defensive structures designed to keep out unwanted intruders or invaders. Hobbes puts safety as a key factor for the establishment of the modern state and the satisfaction of the general welfare.

The line of thought initiated by Hobbes, has formed the realist school, with some distinctive features. Realists believe that the state, as an entity, is not benevolent to others, but rather selfish and competitive. The states are inherently aggressive and obsessed with security.

Thus, for example, a characteristic feature is that the pursuit of national security, states strive to reach as many resources as possible, becoming predators of global resources.

This aggressive accumulation, however, leads to a security dilemma. Increasing security can lead greater instability. The opponents build their own weapons in response and create an unstable situation and greater tension. Therefore, security can become a zero-sum game where only relative gains make sense. There are no universal principles, instead, a state should always be aware of the actions of the states around it and must use a pragmatic approach to solving problems that arise.

Morghentau, one of the greatest exponents of realism, argued that international politics is a struggle for power (among those who wield it and those on which it is exercised). His vision of security (seen as peace) is that this is due to the forces inherent in human nature, which leads to the existence of conflicting interests and conflicting moral principles that will never be fully realized, but it can be closer to them through the balance of interests and the reconciliation of conflicts.

Starting with the second edition of Politics Among Nations, Morgenthau included a section in the opening chapter called “Six Principles of Political Realism.”

- Political realism believes that politics, like society in general, is governed by objective laws that have their roots in human nature.
- The main signpost of political realism is the concept of interest defined in terms of power, which infuses rational order into the subject matter of politics, and thus makes the theoretical understanding of politics possible.
- Realism does not give ’interest defined as power’ a meaning that is fixed once and for all, but recognizes that the determining kind of interest varies depending on the political and cultural context in which foreign policy is made.
- Political realism is aware of the moral significance of political action. It is also aware of the tension between the moral command and the requirements of successful political action. Realism maintains that universal moral principles cannot be applied to the actions of states in their abstract universal formulation, but that they must be filtered through the concrete circumstances of time and place.
- Political realism refuses to identify the moral aspirations of a particular nation with the moral laws that govern the universe.
- The political realist maintains the autonomy of the political sphere; he asks “How does this policy affect the power of the nation?” Political realism is based on a pluralistic conception of human nature. A man who was nothing but “political man” would be a beast, for he would be completely lacking in moral restraints. But, in order to develop an autonomous theory of political behavior, “political man” must be abstracted from other aspects of human nature.

In short, some features of the realist school of thought are based on the principle of accumulating power, so that security comes as a result. Thus, one could consider power as a principle of safety.

2.3.2 Universalism

The Universalism or idealism aims to:

> *“ transform the international system in a scenario where they could establish the necessary conditions for lasting peace in international society as a whole, or as suggested by Immanuel Kant, a “perpetual peace” ”* [HC06]

Universalism is based on the ideas of Immanuel Kant. Kant, in his “Essay for Perpetual Peace (1795)”, postulated that the creation of a “Confederation of States” linked by many rules of morality would avoid war. That “perpetual peace” only “... achieved materialize once it were possible ensure security at all members of the international community” [HC06].

Therefore, the individual, in this line of thought is much more important than the state. It starts from the idea that the interests of all men are on human community are identical and the relations between states are entirely cooperative.

Security, thus, according to Kant is jurisdiction of the state, which is the guarantor of the inalienable rights of its citizens. This idea is quite similar with Hobbes theories, but Kant goes further than Hobbes when interpreting the problem of security from the relationship between the states according to moral standards. As mentioned by Gabriel Orozco:

> *“ Kant realizes that the only way to achieve security is to create an international legal system similar to that in the interior states. Therefore, it considers central to international institutions to legislate and to coact or suppress the violent actions of the states, so that liberate humanity from the scourge of war. ”* [Oro06]

International Relations does not fulfill the whole Kantian philosophical system. Hedley Bull, the representative best known of the English School of International Relations, introduced what he named “Kantian tradition”.It is based onto the belief in an international community based on a permanent cooperative and the idea that international behavior is governed by the moral. States, voluntarily, lose relevance in benefit of a bigger transnational society.

For Universalists, the principle of security is emancipation. It can only be acquired (in Theory of International Relations) by states and later on by groups and individuals. In this context, emancipation is the freeing of constraints.

> *"...Emancipation is the disappearance of any legal, social, economic, moral, political and physical constraints. Freedom must be tempered, in turn, to the knowledge of the rights of others. The basis of emancipation is the idea of reciprocity of rights that should exist in the universal community. The reciprocity of rights (" My freedom depends on your freedom") pushes the process of emancipation. The Kantian approach considers safety (emancipation) of individuals is the ultimate goal of universal community. "*
> [Gse98]

2.3.3 Racionalism

Rationalism or Grotian tradition, is situated between the two previous schools of thinking, finding the in-betweens. On one hand denies the anarchy of the first (the lack of respect for international law). On the other hand also denies the desire for emancipation, to free.

Conceptually, this line of thinking "does not accept the widespread conflict, neither think those interests should always be similar between people" [Vie05]. Its vision is a society of states with defined rules and institutions, which eventually may be conflicts, but where agreed regulations tend rather to facilitate relationships and limit conflict.

Shares with the Hobbesian tradition are regulated by rules and institutions that limit the system. The relationships between states are in terms of coexistence and cooperation. International operations are the economic and social relations and as a result an exponent obtain Trade.

Security, in this context, is a distributive game. In words of Gabriel Orozco:

> *"Grotian conception of international politics interprets that states are arranged as a series of rules and behaviors consistent with the kind of societies every state creates. In this sense, international politics ,understood by Grotius, is neither just about the conflict between states nor is based on an absolute identity of interests. it remembers a game which is partly distributional and partly also productive. "*
> [Oro06]

So, instead of talking about "the pursuit of security", one would be talking about spaces of stability, as the conflict remains a real possibility, tempered by a set of rules, which must be maintained by the actors.

Finally, for rationalists the principle of security is order. This feature appears depending on the existing order. The order, as Sainz stated, “...is explained as a situation in which the basic aims and objectives of the actors are achieved and maintained through sharing common norms and standards” [Gse98, page 30].

2.4 Computer Science security concept

Computer security is usually known as a branch of computer technology applied to computers and networks. Thus, its objective includes protection of information and property from theft, corruption or natural disaster. But also computer security has to allow that the information and property to remain accessible to its expected users. The term involves all processes and mechanisms by which sensitive and valuable information and services are protected from intended or unintended attacks.

Defining “computer security” is not trivial. The definition has to be broad enough to be valid for any system but specific enough in order to describe what security is. Thus, in the context of computer science, security is the protection against access, destruction or alteration to information and regardless it be intended or unintended. In this sense, computer security could be defined as “... the ability of a system to protect information and system resources with respect to confidentiality and integrity” [JFR07]. That definition includes information and surrounding components, i.e. hardware and software.

Computer Security is usually associated with three core areas, the well known acronym “CIA”. Confidentiality ensures information is accessed only by authorized persons. Integrity takes care that authorized persons only could modify information, and Availability is responsible for having the information available to authorized persons. “CIA”, therefore, is focused only on information.

Despite all of these efforts, computer security definitions reveals some drawbacks.

- Current definitions and therefore the concept involve mainly the information. Just in some of them, the surrounding elements are included as a necessity to protect information.

- There is no common accepted definition.

- Computer security, from the point of view of the security concept development, is in its early stages. Security notion is about two thousand years and computer security is only around fifty years.
- Because of the existing definitions are based on "information security" the outcome are operational definitions and there is a lack of theoretical definitions.
- There is barely no relation with security definitions existing in other fields, despite the notion is the same.
- Computer security has moved by its own path, and therefore its definitions and models are not tied to the traditional concept of security, despite having many elements in common.
- Reducing computer security to only protecting information is rather simplistic. Security definition in real-time and critical systems such as air traffic control or nuclear plants have to include key elements such as life or environment either as an active or passive manner. A failure (intentional or accidental) of computer system leads to severe risk not just information but other much more important values.

Therefore the current definition needs to be enlarged because Computer Science is a key element in societies and a societal phenomenon. Thus, the notion of computer security just protecting the information is inadequate.

2.4.1 Computer Science historical review

Development of computer security started from the notion of national security in Internationals Relations field. Therefore has a military origin, which seeks only to protect the information. Computer security has its origins in the 1960s, when multi-user systems emerged needing mechanisms for protecting the system.

At this early stage protection was from system to its users and among them. Systems were in very controlled environments and used by very specific people. Besides, systems were not public available. Therefore protection mechanisms were simpler than now. The RAND report by Willis Ware [WoCSoDC79] was the starting point.

Mainframes promoted the development of initial formal security models to regulate access to classified or sensitive data, such as Bell-Lapadula [BL76]. Cryptography, as an academic discipline related to computer science, started because the necessity to protect backup media and communication among systems. The most important contribution was the concept of public-key by Diffie and Hellman [DH76].

Advent of personal computers changed considerably the scene. A computer could be purchased in small units by companies and organizations, even without involving IT departments. Users managed the PC by its own, storing data locally. The developed security models were utterly unnecessary. Other formal models to reflect the new situation were necessary [BL76] [CW87] [BN89].

Digital communications came into scene with networks. At this very beginning, communication security was considered equal as data storing and thus cryptography was the main mechanism. The first steps of the Internet made information transport and computer control the two main issues.

The widespread use of networks and its availability to be used by the society, made a twist on the situation. The web showed up and emphasized the easiness to transmit information everywhere to everyone; the power to get, put and move information and also the weakness of information security and its models. Issues such as privacy, use of web by children, pornography, international data moving or cyber terrorism made computer security a social need to acquire in political proportions. Thus computer security is no longer a technical issue but a societal one. The new perspective implies awareness, education of society and industry have to involve all employees, customers and entities that deal with the organization.

Computer security is extremely new into the concept of security (Figure 2.1). The big conceptual change that occurs between the concept of security and computer security relays on the latter that it looks for "security automation" instead of considering it a "cottage industry". In this sense formal models of security initiated by Bell-Lapadulla [BL76] seek protection of information without human intervention.

Because of those new set of phenomena; considering computer security and other securities in global terms is a must.

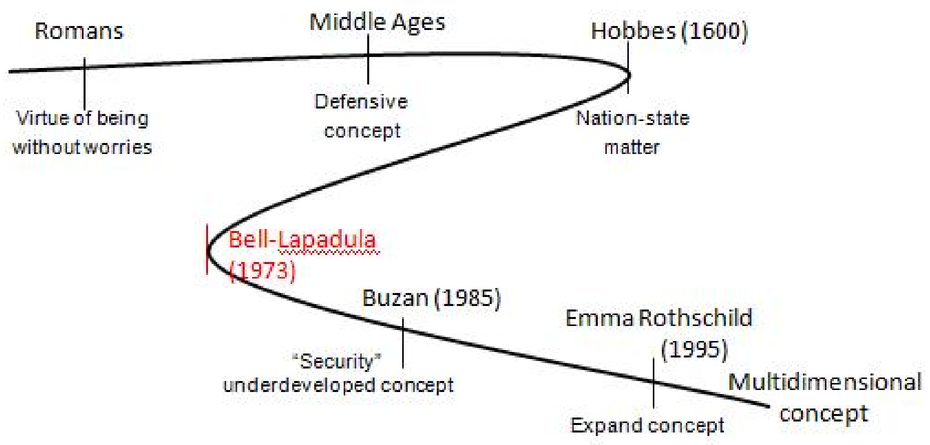


Figure 2.1 – Security concept time line.

2.4.2 Computer security models

Security in computer science field is mainly focused on information control access and thus the milestone is to provide a reliable system capable of guarantee the protection of information from inappropriate or unauthorized access. Their development, therefore, has been a set of formal models. Those models, essentially, define subjects and objects. Objects are constructs such as files, programs, directories or ports and subjects are entities such as users, process or threads that perform some sort of operation. Both have a set of security attributes. When a subject tries to access an object, the operating system examines the security attributes in order to decide if the access is allowed or denied.

Broadly speaking, access control models, could be classified as:

- Mandatory Access Control (MAC) [BL76] [Bib77] [BN89]. The main feature of those models is the set of rules (the policy) is centrally controlled. Thus all security relies on the security policy administrator. Users cannot modify the permissions.
- Discretionary Access Control (DAC) [Den76] [LS77]. Those models delegate to users the ability to make policy decisions and/or assign security attributes. Users has control over the objects it owns and thus they have the capability to determine the permissions other users have over those objects.
- Role Based Access Control (RBAC) [CW87] [SFK00]. In addition of subjects and objects, there are roles. A role is and abstract entity which defines

certain operations over certain objects. After that, the role is assigned to subjects in order to obtain the current permissions. RBAC overcomes MAC and DAC models because of it is capable of implement them.

However, computer science has enlarged its security concept somehow. Data protection alone is inadequate and security, hence, is seen as a process in which the probability of an incident that adversely affect the system and its availability is assessed. This security notion includes the attacks coming from external or internal sources. External attacks have grown in recent years with the connection of Internet to any system and was necessary to include it in some manner.

2.5 Current models crisis

Computer security mainly relies on CIA triad. This approach carries several lacks.

- The CIA Triad is completely focused on information. It promotes a limited view of security that tends to ignore several factors. For example, Availability takes care to ensure that access to resources when needed. In terms of information security, availability in itself does not guarantee that someone else is not making unauthorized use of hardware resources.

- Some authors [And03] [vSvS05] highlight computer security definition needs to be reviewed.

- There is a necessity to share security knowledge [FW06] in order to improve security on working systems. Current security approach makes it hard. Knowledge engineering such as Ontology approach [FPM09] tries to mitigate the problem.

- Currently, “Information security is thus not just about technology issue, is also about people and process also” [AVC10]. Thus security definitions have to include this approach, because the social scene needs moving the definition to a social and technical inclusive definition.

All of that moves security to a different stage. Security is no longer a technical problem related to data access control or system access control. It is related to social aspects such as law or human behavior. Owing to that, security is currently tackled in a more global approach. One of the technologies that fits in this scenario is Knowledge Engineering. There are several works working with security ontologies. The need for a security ontology is a *"fait accompli"* by the scholar community [BEK09] [KLK05] [HSD07] [RHTN01] [FPM09].

2.6 Conclusions

The notion of security was initially developed in the area of philosophy. Afterwards, security was a concept used widely within the field of International Relations. The concept had a slow but persistent development. Especially because of the world wars and the Cold War, where there has been an expansion of the concept of security.

Some decades ago, the emergence of computer security models and its highly technological expansion inside the social fabric has resulted in a crisis.

Security, as a concept, is ambiguous, subjective and undeveloped. For this reason there are many definitions, which confirms the subjectivity and ambiguity of the term. Due to the large burden of subjectivity of the concept, security is not an unambiguous concept.

Bibliography

[And03] James M. Anderson. Why we need a new definition of information security. *Computers & Security*, 22(4):308 – 313, 2003.

[AVC10] Rathnakar Acharya, Dr. V. Vityanathan, and Dr. Pethur Raj Chellaih. Article: Secured information access based on bell lapadula model a case of novel publishing company. *International Journal of Computer Applications*, 11(8):37–45, December 2010. Published By Foundation of Computer Science.

[BEK09] S. Beji and N. El Kadhi. A knowledge based process proposal for mobile security. In *Developments in eSystems Engineering (DESE), 2009 Second International Conference on*, pages 166–172, 2009.

[Bib77] J. K. Biba. Integrity considerations for secure computer systems. Technical report, Bedford, MA, 1977.

[BL76] David E. Bell and Leonard J. LaPadula. Secure Computer System: Unified Exposition and Multics Interpretation. Technical Report

MTR-2997, The MITRE Corporation, Bedford, MA, USA, March 1976.

[BN89] D. F. C. Brewer and M. J. Nash. The chinese wall security policy. In *IEEE Symposium on Security and Privacy*, pages 206–214. IEEE Computer Society, 1989.

[Buz83] Barry Buzan. *People, States and Fear*. Harvester- Wheatsheaf, Brighton, 1983.

[CW87] David D. Clark and David R. Wilson. A Comparison of Commercial and Military Computer Security Policies. *IEEE Symposium on Security and Privacy*, page 184, 1987.

[Den76] Dorothy E. Denning. A lattice model of secure information flow. *Commun. ACM*, 19(5):236–243, 1976.

[DH76] Whitfield Diffie and Martin Hellman. New Directions in Cryptography. *IEEE Transactions on Information Theory*, 22(6):644–652, 1976.

[FPM09] Stefan Fenz, Thomas Pruckner, and Arman Manutscheri. Ontological mapping of information security best-practice guidelines. In *Business Information Systems, 12th International Conference on Business Information Systems, BIS 2009*. Springer Berlin Heidelberg, 4 2009.

[FW06] Stefan Fenz and Edgar Weippl. Ontology based it-security planning. In *PRDC*, pages 389–390. IEEE Computer Society, 2006.

[Gse98] Nora Sainz Gsell. *La OSCE en la Europa post-bipolar: Un estudio sobre la gestión de conflictos en el espacio ex-soviético*. PhD thesis, Universitat Autònoma de Barcelona, 1998.

[HA10] Patricio Haro Ayerve. *La ley de seguridad nacional, útil herramienta política : desde el retorno a la democracia, hasta la publicación de las políticas de defensa 2003*. PhD thesis, FLACSO sede Ecuador, enero 2010.

[HC06] Ricardo Hormazábal and Eduardo Carreño. Introducción a la teoría de las relaciones internacionales. Technical Report 14, Departamento de Gobierno y Gestión Pública del Instituto de Asuntos Públicos de la Universidad de Chile., December 2006.

[HSD07] Almut Herzog, Nahid Shahmehri, and Claudiu Duma. An ontology of information security. *International Journal of Information Security and Privacy*, 1(4):1–23, 2007.

[ISO05] Switzerland ISO, Geneva. Code of practice for information security management. Norm ISO 17799:2005, International Organization for Standardization, 2005.

[JFR07] Anil K. Jain, Patrick Flynn, and Arun A. Ross. *Handbook of Biometrics*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2007.

[Kag97] K Kageura. *Multifaceted/Multidimensional Concept Systems. Handbook of Terminology Management. Volume 1: Basic Aspects of Terminology Management*. Amsterdam/Philadelphia: John Benjamins, 1997.

[KLK05] Anya Kim, Jim Luo, and Myong Kang. Security ontology for annotating resources. In *Proceedings of the 2005 OTM Confederated international conference on On the Move to Meaningful Internet Systems: CoopIS, COA, and ODBASE - Volume Part II*, OTM'05, pages 1483–1499, Berlin, Heidelberg, 2005. Springer-Verlag.

[Lan01] Carl E. Landwehr. Computer security. *International Journal of Information Security*, 1:3–13, 2001.

[LS77] Richard J. Lipton and Lawrence Snyder. A linear time algorithm for deciding subject security. *J. ACM*, 24(3):455–464, 1977.

[NG03] Melissa Nobile González. *México y la agenda contemporánea de seguridad internacional : un estudio sobre los alcances del uso del concepto de seguridad humana*. PhD thesis, Universidad de las Américas Puebla, May 2003.

[Oro06] Gabriel Orozco. El concepto de la seguridad en la teoría de las relaciones internacionales. *Revista CIDOB d'Afers Internacionals*, (72):161–180, 2006.

[RHTN01] Victor Raskin, Christian F. Hempelmann, Katrina E. Triezenberg, and Sergei Nirenburg. Ontology in information security: a useful theoretical foundation and methodological tool. In *Proceedings of the 2001 workshop on New security paradigms*, NSPW '01, pages 53–59, New York, NY, USA, 2001. ACM.

[Rot95] Emma Rothschild. What is security? the quest for world order. *Daedalus*, 124(3):53–99, June 1995.

[SFK00] Ravi S. Sandhu, David F. Ferraiolo, and D. Richard Kuhn. The nist model for role-based access control: towards a unified standard. In *ACM Workshop on Role-Based Access Control*, pages 47–63, 2000.

[Ull83] Richard H. Ullman. Redefining security. *International Security*, 8(1):pp. 129–153, 1983.

[Vie05] Edgar Vieira. Evolución de las teorías sobre integración en el contexto de las teorías de las relaciones internacionales. *Papel Político*, (18):235–290, December 2005.

[vSvS05] Basie von Solms and Rossouw von Solms. From information security to...business security? *Computers & Security*, 24(4):271 – 273, 2005.

[Wae09] Ole Waever. *Reconceptualizar la seguridad en el siglo XXI*. UNAM, 2009.

[WoCSoDC79] W.H. Ware, United States. Defense Science Board. Task Force on Computer Security, United States. Dept. of Defense, and Rand Corporation. *Security Controls for Computer Systems: Report of Defense Science Board, Task Force on Computer Security*. RAND Corporation, 1979.

[Wol52] Arnold Wolfers. National security as an ambiguous symbol. *Political Science Quarterly*, 67(4):481–502, 1952.

Part II

Concept and Knowledge Modeling

CHAPTER

3

Concept Representation

"I hope that readers will enjoy the trip through some perhaps exotic seeming countries that lie on the borders between the sciences and the humanities, and return to their home disciplines with useful insights, such as a sense of the limitation of disciplinary boundaries, as well as with some new formal tools."

— Joseph Goguen

Contents

In this chapter, knowledge modeling and knowledge representation techniques such as conceptual maps, conceptual modeling or object orientation are reviewed. All of them are capable to analyze and graphically represent a concept.

3.1 Introduction

A concept is done by means of terms, relationships and operations. Therefore any concept or idea is not isolated, but forms systems. Concepts have the properties of the understanding or intention (a set of defining properties of the object) and extension (set of objects that fall under the concept.). Consequently, a concept is a cognitive unit of meaning, sometimes defined as a "unit of knowledge" (a concept describes an abstract idea).

Concepts are comprised of characteristics and therefore, the abstraction of "mental image" of a concept is classified and categorized according to such characteristics. That allows the classification of other concepts in the same category or class or even in subclasses of these classes. The grouping process is done by relating the aspects and qualities common to many objects. The set of all concepts gives us a representation of the world.

3.1.1 Properties of a concept

Defining a concept is related to the essential properties (which are the main characteristics for the understanding) and the description to accidental properties (which are the ones that could be removed). Desirable properties or characteristics of a concept are described in [ISO99] [Bal97] [fHI04]

- A concept "depicts or correspond to a set of objects" and "are organized into concept systems".
- The objects are "perceived or conceived" and the objects "are abstracted or conceptualized into concepts".
- The intension of a concept is "the set of characteristics that come together as a unit to form the concept".
- The extension is all the objects to which that concept is concerned.
- Concepts are comprised of characteristics.

  The abstraction of a concept is classified and categorized according to such characteristics. That allows the classification of other concepts in the same category or class or even in subclasses of these classes.

- Essential characteristics are the characteristics “indispensable for the understanding of the concept in a particular field of knowledge”. If any of those characteristics is absent, then the concept changes.

- Non-essential characteristics are those that if the characteristic were removed, the essential “concept would not be altered”.

- Concepts are language-independent.

  Words describing a concept may differ due to different languages or even in a given language there are a variety of possibilities.

- Concepts are mental or logical representations of reality.

  All concepts are abstract, by this point of view, and exist purely mentally. However concepts prepare a way for the human mind to classify and to understand the minds perceptions.

- Concepts are negotiated within a knowledge community.

  The concept needs an agreement about its features and characteristics.

- Concepts are related to other concepts.

- Concepts do not need symbols but use them for means of communication.

  A concept does not need any kind of symbol (like a word) to exist. Thus, the symbol becomes a way to communicate.

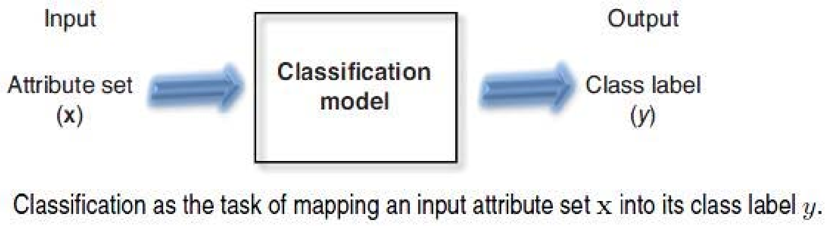
- Concepts should be operational in the broadest sense, although this should not be interpreted as requiring quantification.

- Concepts that establish definitional connections with other terms are to be preferred.

- Concepts should remain reasonably close to ordinary language.

3.1.2 Relations among concepts

There is no sense in a concept to stand alone. Somehow there must exist relations to other concepts. Therefore, when a new concept is included into a conceptual system, the operation involves a task of classification. According to [TSK06, p: 145] Classification “is the task of assigning objects to one of several predefined categories” (figure 3.1).



Classification as the task of mapping an input attribute set $\mathbf{x}$ into its class label y.

Figure 3.1 – Classification task. (Introduction to Data Mining [TSK06])

Concepts are organized systematically and characterized according to the relationships established with other concepts within a conceptual system. These relations are:

- Generic / specific

  That is a hierarchical relationship. Concepts are identified by their category membership. A generic concept could be considered superordinate to other more specific concepts. Once made, subordinate concepts share all the characteristics of the generic. But they also have some peculiarities that differentiate, making them more specific.

- Part / Whole

  The relationship among concepts that one concept is composed by one or more concepts/subconcepts which are themselves instances of another concept. Usually a concept/subconcept can only be “assigned” to one whole at a time. The set of concepts/subconcepts make up the whole.

- Polyvalent

  There is the possibility that a concept could be in different places in the same conceptual system.

3.1.3 Representing concepts

Concept representation involves the characteristics of the concept, the related concepts and the relationships established with other concepts within a conceptual system. This process, as it involves classification, could be achieved using automated tools. Visual classification tools make it easy to classify objects, organize concepts and represent concepts within the conceptual system. Currently, different categories of visual tools can be found [DM08] [Epp06]:

- **Mind maps.** From a central point, a mind map is a diagram to represent words, ideas, tasks, or other items around the idea.
- **Conceptual diagrams.** A conceptual diagram employs a graphic conceptual framework to visually structure information or learning content with the help of pre-defined categories. The categories are usually derived from a (domain-specific) theory or model.
- **Visual metaphors.** Visual metaphors are graphic structures that use the shape and elements of a familiar natural or man-made artifact or of an easily recognizable activity or story in order to use the typical associations to convey additional meaning about the content.
- **Tree Maps.** Classifies ideas into categories or groups. This type of map organizes information into levels according to importance, size or attributes.
- **Flow Maps.** Flow maps graphically depict a sequence of events in order. They can be used to represent complex processes. Multiflow Maps appears when there are multiple outcomes.
- **Compare and Contrast Maps.** Mainly used to compare, in summarized way, information about differences and similarities aspects of two issues or topics.

3.1.4 Analyzing concepts

According to Oxford English Dictionary, analysis is “A detailed examination or study of something so as to determine its nature, structure, or essential features”. That process involves breaking it into smaller parts in order to gain a better understanding of it. An analysis could be applied almost to anything, even abstract ideas such as concepts. Concept analysis could be thought as a kind of definition, because it clarifies the concept definition and the boundaries, which are formed.

The analysis of concepts or conceptual analysis is used to establish a system between the concept and those with whom it relates. It is a kind of method developed from the analytic philosophy. As a result, conceptual analysis classifies objects and their qualities based on their common features obtaining sets of classes or categories. This is currently used in philosophy of science.

As describing a concept or modeling a conceptual system is a classification problem, it is an inevitable search for methods to handling objects, concepts,or both simultaneously. The majority of methods such as Formal Concept Analysis [GW97], Conceptual Maps [NC06] or Object Oriented techniques [O'D05, Pre05, CY91] begin at their first stages with some sort of analysis.

3.2 Formal Concept Analysis

In Philosophy, a formal concept is defined as those concepts that have no substantive content at all. The concept, thus, is a form applicable to a multitude of things. The formal concepts are objects to be determined, even indeterminate. In ordinary language we use some words as formal concepts such as "entity", "thing" or "organization". They are formal if not determining content.

The term "conceptual model" is somehow ambiguous. It could be understood as a model of a concept or a model that is conceptual. Mainly models are concepts, usually of real world. When modeling a concept, it is not essentially the truth or falsity of the concept that is being modeled. This is why conceptual models are used in fields such as software developing or artificial intelligence for building expert systems and knowledge-based systems.

The scope of conceptual models is extensive. From concrete like a physical object, going through formal as a mathematical model, a concept or a category (like fruit) to a large domain (as might be the universe.)

As defined by Uta Priss in *Formal Concept Analysis in Information Science*,

> *" Formal Concept Analysis (FCA) is a method for data analysis, knowledge representation and information Management "* [Pri06]

The purpose of Formal Concept Analysis or FCA is to automatically find groups of objects (or entities) that share in common a group of attributes. FCA works on a set of objects and their properties (attributes), which comprises a group of objects that share a subset of attributes and a group of properties that has all the attributes shared by these objects. In general, describing a concept or modeling a conceptual system is a classification problem.

It is a mathematical technique that allows us to show underlying abstractions extracting conceptual structures of a data set in a data table, formally a context, by building a concept lattice, also known as Galois lattice. It is based on the philosophical idea that a "concept" consists of two parts: its extension, formed by all objects belonging to that concept, and its intention, which comprises all attributes shared by those objects. The FCA has been used in realms like representation of knowledge, psychology, linguistics, sociology, mathematics and computer sciences.

3.2.1 Origin

Formal concept analysis was introduced by Rudolf Wille [Wil82], using both the lattice theory as the theory of order, built on Garret Birkoff's 1940 work. The mathematical foundation of FCA is described by Ganter & Wille [GW97]. Recent works, close related to computer science, are the ones by Josep Goguen [Gog05].

The technique is capable of extracting conceptual structures of a data set. The issue is important enough, that the International Conference on Conceptual Structures (ICCS)[5] give conferences since 1993, and the International Conference on Formal Concept Analysis (ICFCA) started in 2003.

3.2.2 Fundamentals

The mathematical foundations are mainly extracted from [VML02, ABH+02].

Definition A *formal context* $\mathcal{K}$ is a triple $\mathcal{K} = (O, A, I)$ where O and A are sets and I is a relation between O and A. The elements of O are called the objects and the elements of A are called the attributes of the context. Formally it can be regarded as a subset of the Cartesian product (incidence relation), i.e. $I \subseteq O \times A$.

In order to express that an object d is in a relation I with an attribute a, we write dIa or $(d, a) \in I$ and read it as "the object d has the attribute a".

Definition Let X be a set of objects in a context $\mathcal{K} = (O, A, I)$. The *intension* of X, noted X' is the set of attributes common to all objects in X:

$$X' = \{a \in A : dIa, \ \forall \ \ d \in X\}$$

Definition Let Y be a set of objects in a context $\mathcal{K} = (O, A, I)$. The *extension* of Y, noted Y' is the set of objects common to all attributes in Y:

$$Y' = \{d \in O : dIa, \ \forall \ \ a \in Y\}$$

[5] http://conceptualstructures.org

Definition The *derive* of intension an extension are:

$$X'' = (X')' \quad and \quad Y'' = (Y')'$$

Definition A *formal concept* in a formal context $\mathcal{K} = (O, A, I)$ is a pair (X, Y) where X is a set of objects of $\mathcal{K}$, and Y is a set of attributes of $\mathcal{K}$, such that $X' = Y$ and $Y' = X$.

We say that X and Y are the extension and intension, respectively, of concept (X, Y).

Example Consider the set of objects $O = \{car, bicycle, motorbike, van, ski, taxi\}$ with properties $A = \{wheels, fuel, individual, snow, engine\}$. The relation is given in Table 3.1:

I	wheels	fuel	individual	snow	engine
car	✓	✓			✓
$bicycle$	✓		✓		
$motorbike$	✓	✓	✓		✓
van	✓	✓			✓
ski			✓	✓	
$taxi$	✓	✓			✓

Table 3.1 – Cross-table of relation I.

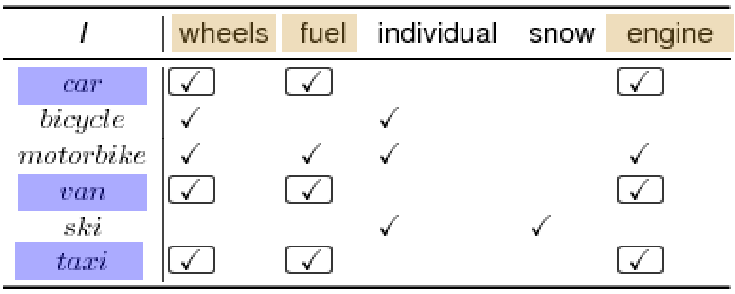
To create a formal context (Figure 3.2):

1. Pick a set of objects e.g $B = \{car\}$.
2. Derive the attributes $B' = \{wheels, fuel, engine\}$.
3. Obtain $(B')' = B'' = \{wheels, fuel, engine\}' = \{car, van, taxi\}$.
4. $(B'', B') = (\{car, van, taxi\}, \{wheels, fuel, engine\})$ is a formal concept. A dual approach can be taken starting with an attribute.

The concepts of a context can be naturally partially ordered: a concept $C1$ is “less” than another $C2$ when all the objects in $C1$ are also in $C2$.

Definition Let (X_1, Y_1) and (X_2, Y_2) concepts of formal context $\mathcal{K} = (O, A, I)$. The concept (X_1, Y_1) is *subconcept* of (X_2, Y_2), and is represented by

$$(X_1, Y_1) \leq (X_2, Y_2), if\ X_1 \subseteq X_2$$



I	wheels	fuel	individual	snow	engine
car	☑	☑			☑
bicycle	✓		✓		
motorbike	✓	✓	✓		✓
van	☑	☑			☑
ski			✓	✓	
taxi	☑	☑			☑

Figure 3.2 – Formal Concept.

Lemma Let (X_1, Y_1) and (X_2, Y_2) concepts of formal context $\mathcal{K} = (O, A, I)$. Then

$$(x_1, y_1) \leq (x_2, y_2) \Longleftrightarrow Y_2 \subseteq Y_1$$

Lemma The relation $\leq$ is a partial order on the set of concepts from one context $\mathcal{K} = (O, A, I)$.

Lemma Let $\mathcal{F}$ a family of sets of objects and $\mathcal{G}$ a family of sets of attributes in a formal context $\mathcal{K}$.

$$\left(\bigcup \mathcal{F}\right)' = \bigcap \{X' : X \in \mathcal{F}\}$$

$$\left(\bigcup \mathcal{G}\right)' = \bigcap \{Y' : Y \in \mathcal{G}\}$$

Definition $(\mathcal{R} = (O, A, I), \leq)$ is a lattice. For any concepts set $\{(X_k, Y_k) : k \in \mathcal{K}\}$, the *supremum* and the *infimum* are given by

$$sup\left(\{(X_k, Y_k) : k \in \mathcal{K}\}\right) = \left(\left(\bigcup_{k \in \mathcal{K}} X_k\right)'', \bigcap_{k \in \mathcal{K}} Y_k\right)$$

$$inf\left(\{(X_k, Y_k) : k \in \mathcal{K}\}\right) = \left(\bigcap_{k \in \mathcal{K}} X_k, \left(\bigcup_{k \in \mathcal{K}} Y_k\right)''\right)$$

Properties

Let $\mathcal{K} = (O, A, I)$ a formal context. Given $X, X_1, X_2 \subseteq O$, and $Y, Y_1, Y_2 \subseteq A$, is verified:

$$(1)\;\; X_1 \subseteq X_2 \Longrightarrow \;\; X_2' \subseteq X_1'$$

$$(2)\;\; Y_1 \subseteq Y_2 \Longrightarrow \;\; Y_2' \subseteq Y_1'$$

$$(3)\ \ X \subseteq X''$$

$$(4)\ \ Y \subseteq Y''$$

$$(5)\ \ X = X'''$$

$$(6)\ \ Y = Y'''$$

$$(7)\ \ X \subseteq Y' \Longleftrightarrow Y \subseteq X' \Longleftrightarrow X \times Y \subseteq I$$

3.2.3 Graphical Representation

The sets of formal objects and formal attributes together with their relation to each other can be represented by a $n \times m$ cross table (incidence matrix). The elements on the left side are the entities (formal objects). The elements at the top are formal attributes and the relation between them is represented with a Boolean value (graphically a checkmark or a cross) in cell (d,a) whenever object d has attribute a. This table is called “formal context”.

From the table, algorithmically, a Galois reticulum is constructed, represented by its corresponding Hasse diagram that contains all the original information, but organized in a way that shows the data structure.

As Formal Concept Analysis (FCA) is a discipline that studies the hierarchical structures induced by a binary relation between a pair of sets, a Hasse diagram (also called a line diagram) is a type of mathematical diagram, in order theory, used to represent a finite partially ordered set. There are many ways to construct the diagram, in the work of Kuznetsov & Obiedkov [KO01] many algorithms for constructing concept lattices are reviewed.

Briefly, each edge of the Hasse diagram of the concept lattice connects some concept C to the concept formed by the join of C with a single object. Thus, one can build up the concept lattice by finding the neighbors in the Hasse diagram of known concepts, starting from the concept with an empty set of objects. It is difficult to draw “good” Hasse diagrams due to there are number of possible ways to make the diagram for a given context, as shown in figures 3.3, 3.4 (page 49).

Example Consider the set of objects a: ant, b: beetle, f: fly, s: spider on which have been observed following properties 6l: 6 legs, 8l: 8 legs, f: fly, s:sting/bite. The incident table of the relationship is shown in Table 3.2 (–page 49–).

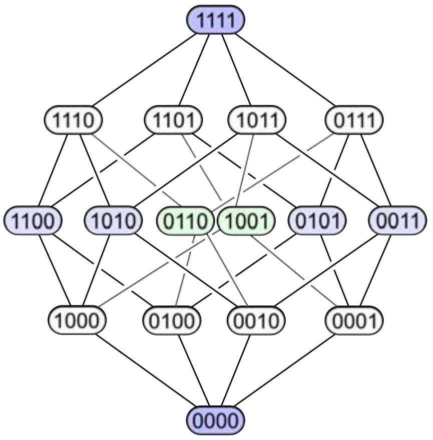


Figure 3.3 – Hasse diagram. (Wikipedia)

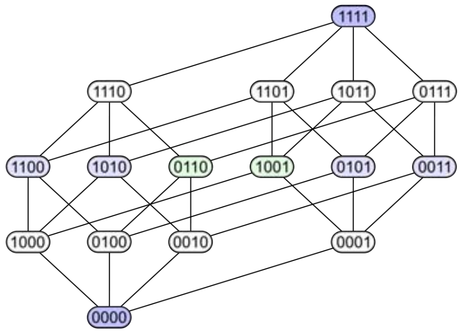


Figure 3.4 – Hasse diagram of same lattice. (Wikipedia)

	6 legs	8 legs	Fly	Sting/bite
ant	X			
scorpion		X		X
fly	X		X	
spider		X		X
wasp	X		X	X

Table 3.2 – Formal Context

3.3 Conceptual Maps

Basically, a concept map is a way to visualize the mental "map" of concepts and their relationships, as well as the structure and hierarchy of these relationships. One important aspect of concept maps is their ability to show large amounts of information in a compact format.

Concept maps achieve its goal of represent concepts and their relationships in a graphical way, which is one of the most important features. In this context, concept is defined as "a perceived regularity in events or objects, or records of events or objects, designated by a label." [NC06] and become a kind of "graphical tools for organizing and representing knowledge." [NC06].

A concept map represents "a body of knowledge along with their interrelationships in the form of a directed graph." [Hub07] and concepts are usually enclosed in circles or boxes of some type and relationships between concepts are indicated by a connecting line linking two concepts. Main features of Conceptual Maps are:

- Simplicity.

  Concept Maps should be simple and clearly show the relationships between concepts.

- From generic to specific.

  More general ideas are displayed at the top of the structure. More specific ones at the bottom.

- Uniqueness.

  Concepts are unique (never are repeated).

- Summary

  A concept map has to be seen as a short form of representing information.

3.3.1 Origin

The technique of concept mapping was developed by Joseph D. Novak and his research team at Cornell University in the 1970s as a means of representing the emerging science knowledge of students. Novak's work is based on the cognitive theories of David Ausubel (assimilation theory). Concept maps have their origin in the learning movement called constructivism. In particular, constructivists hold that learners actively construct knowledge.

Concept Maps have gone much further, and have not been restricted to the field of education. It's possible to find it everywhere such as cooperative environments, sciences ,business or government. Apart from those fields even in software engineering. By comparison, the work by Thomas Hubbard [Hub07] does a type of mapping between the concept mapping and object-oriented design and Lee A. Freeman proposes using concept maps on requirement elicitation stage [Fre04].

Mind Mapping is a popular related technique by Tony Buzan. He describes mind maps formed by a central word or concept and "around the central word you draw the 5 to 10 main ideas that relate to that word. You then take each of those child words and again draw the 5 to 10 main ideas that relate to each of those words" [BB95].

There is a huge difference between concept maps and mind maps. While mind map has only one main concept, concept map may have several. Besides, a mind map can be represented as a tree while a concept map may need a network representation.

3.3.2 Fundamentals

The main elements for Concept Maps are:

- Concept

  A concept is an event or a regular object which is called with a name or label (Novak & Gowin, 1988). There are concepts that define specific elements such as “home” and others that define abstract notions. They are untouchable despite exist in reality (security, freedom).

- Proposition

  Two or more concepts linked by link words to a semantic unit.

- Linking words

  Link the concepts to establish the type of relationship. Mainly are prepositions, conjunctions, adverbs and general all non-concept words. The linking words are used to join two or more concepts to form propositions.

3.3.3 Graphical Representation

The conceptual map is represented as a lattice of lines that meet at various points, mainly using two graphic elements, boxes and arrows. Concepts are represented as boxes and are connected with labeled arrows in a downward-branching hierarchical structure. The relationship between concepts can be articulated in linking phrases such as “is made of”, “help to”. Concepts are placed inside the box and the words are written next to the line connecting the concepts. To make a concept map, there are some steps to follow:

- Make a list with the main ideas or concepts.

- Select the concepts that derive from each other, even the ones witch have a cross relationship

- Use lines to connect the concepts. Write on each line its linking word.

- Build the diagram. Concepts must be represented from the more general to more specific in descending order. By convention, the concepts are written in capital letters and linking words in lowercase. The linking words might be verbs, prepositions, conjunctions or any other conceptual link.

There are some graphics tools to create Conceptual Maps in a graphical way such as Compendium®[6] or FreeMind®[7]. Probably the most known is CmapTools®[8] from IHMC. CmapTools supports the construction of "knowledge models" about a topic. Due to its origin is maintained and some works on it exist [NC04].

Example Suppose we want to describe a television. Its parts and for whom is used. You would get a diagram like the one shown in Fig. 3.5 (p.52)

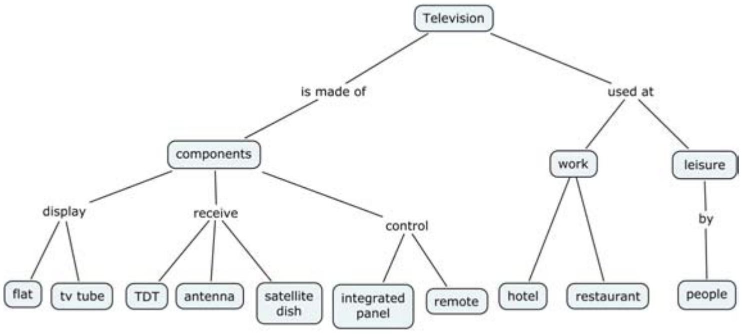


Figure 3.5 – Conceptual Map example. (Using CmapTools)

3.4 Object Oriented Approach

Object-oriented analysis is a method based on defining "all classes" (categories into this context), "and the relationships and behavior associated with them that are relevant to the problem to be solved" [Pre05, p. 217]. Hence, object-orientation is a way to model the world according to some systematic methodology.

Object orientation can be found, as an idea or philosophy, in Plato. According to Plato, the real world is mere instances of class objects in the world of ideas. Aristotle also inadvertently advanced object orientation by expressing things as matter and form. Software objects also have characteristics (properties or attributes) and behaviors. All objects are members of a larger class and, in terms of programming,

[6] http://compendium.open.ac.uk
[7] http://freemind.sourceforge.net/wiki/index.php/Main_Page
[8] http://cmap.ihmc.us

inherit private data structure and operations defined for that class. A software object maintains its characteristics in one or more “variables” and implements its behavior with “methods”. A method is a function or subroutine associated with an object.

Object-orientation is widely used far beyond software development. For example it is used in electronics assembly [LO02], automation engineering [MFC99] or one of the most successful fields, database design [KST92] [Kho90] [Hin98] [GSC91]. Besides Object approach has become a way to model the world by non-computer experts through a high level language such as Modelica®[9] [SZ09].

3.4.1 Origin

Object-oriented programming arose in the early 70’s [Cap03] and the object-oriented paradigm, from the point of view of generation programming, is up to now the latest. The programming languages started from the machine code going through assembly language, structured high-level languages to end with object oriented languages.

3.4.2 Object Oriented Paradigm

The object-oriented paradigm is based on the way people see the world, that is, “objects”. All these objects are distinguished by the characteristics (attributes) and behaviors (methods) they present. Therefore, the object oriented modeling includes two basic aspects, which are the structural dimension and the dynamic behavior of objects. The structural dimension focuses on the passive or static aspect. It is related to the static structure of objects that are part of the system. The dynamic behavior is related with the active or dynamic aspect. This describes the behavior and the interrelation of the objects that make the system.

The interest in object-orientation is that it provides concepts and tools which allow users to model and represent the real world as closely as possible. These concepts and object-oriented tools are technologies that allow real-world problems are expressed in a more easy and natural way than other paradigms such as procedural. Object-oriented paradigm contains some fundamental elements:

1. **Object.** In the “real” world objects are the entities of which the world is comprised. However, objects are not isolated entities. Everything that happens

[9] https://www.modelica.org/

in the world is related somehow by the interactions between the objects. Therefore, from a structural point of view, an object can be defined as an entity with a set of attributes or properties, the behavioral and the capacity to react to events.

In computer science an object is seen as a unit. The properties or attributes become *data*, the behavioral or actions *methods* and the events *messages*. The actions are all activities that the object is able to perform and the properties are all the features that distinguish the object. In addition, an object is an instance of a class (or category). Therefore, an instance of a class is a synonymous of the word object. Object is a more general term, but objects and instances are both representative of a class. The structure of an object is composed by:

(a) Attributes / Properties

Are the observable characteristics of an object. They describe an aspect of the object. In technical terms are the data (variables) related to the state of an object. Usually an attribute can take a value defined by an enumerated domain (set of specific values).

(b) Methods

Is the set of actions (called operations) that an object can make and therefore characterize his behavior. The methods are commonly used to modify properties of the object. In more technical terms, is the procedure or function that is invoked to act on an object.

(c) Events / messages

The events are the “stimuli” that an object receives and sends to other objects. The system handles the event by sending the right message to the relevant object. Once again, talking a bit more technical, a message is an invocation for an object to execute one of their methods with some parameters. All the messages an object can answer is called protocol.

In object-orientation, often the system is thought in terms of objects, operations, methods and messages that are transferred between such objects. The interactions among objects can be graphically represented as shown in Fig. 3.6 (p.55).

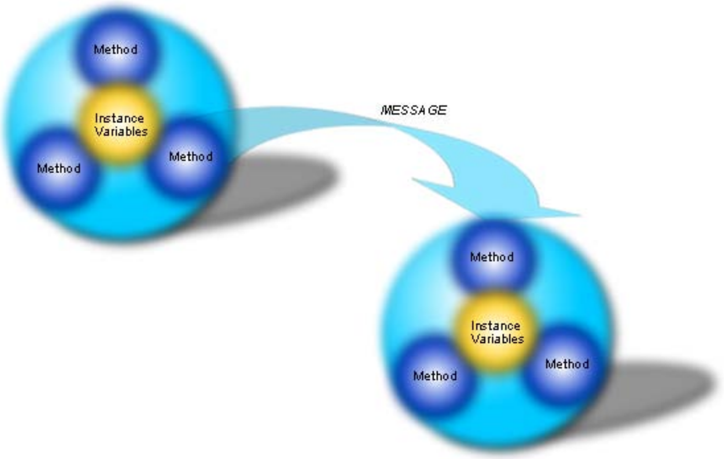


Figure 3.6 – Message sending/Method invocation between objects.

2. Classes. A class is a collection of objects of similar type. In this sense, the class can be seen as a model or prototype that defines the variables and methods common to all objects of that class. Once a class is defined, any number of objects can be created which belong to that class. The creation of an object from the class is known as instantiation. The Object oriented paradigm and its methodologies must meet some principles:

(a) Abstraction. Refers to the fact of representing essential features and behavior of an object without including the background details. Thus, the object acts as a model that can perform tasks, change its status and communicate to other objects in the system.

(b) Modularity. The property of broke an application in smaller parts (called modules) that must be independent of the other parties. Each module (also known as class) has two parts. The interface, which shows only its external view and the implementation that contains the mechanisms to perform the appropriate behavior. Classes, therefore, will be perceived as black boxes so that one only knows the behavior but not the internal details.

(c) Polymorphism. The ability of an operation to exhibit different behaviors in different instances. An operation can have the same name in different classes and each class operation run differently. for example, the object “animal” must be able to perform the breathing function. An insect, a person or a fish perform the same function, albeit in different ways.

(d) Inheritance. Inheritance is the process by which objects can acquire the properties and operations of the objects of another class. Bearing in mind that classes relate to each other, which is usually done by grouping objects into classes and these into trees that reflect the common behavior, the result is a classification hierarchy.

3.4.3 Graphical representation

Visual modeling is the key question for Object Oriented approach. Different methodologies for modeling have existed, however, widespread use of and acceptance of the Unified Modeling Language (UML) closed the discussion. Aim of UML was to represent the design by means of a graphic model. The lack of standardization that existed in the graphic representation prevented the designs could be easily shared between different designers.

UML is the modeling language for software systems most known and used today and is a de facto industry standard approved by the OMG (Object Management Group). It is a set of specifications for object-oriented notation, which are composed of different diagrams that represent different stages of developing a software project.

The language combines techniques from data modeling, object modeling and component modeling. It can be used with all processes, along the Software Development Life Cycle (SDLC). UML has synthesized the notations of the Booch method, the Object-modeling technique (OMT) and Object-oriented software engineering (OOSE) by fusing them into a single, common and widely usable modeling language.

It is a graphical language for visualizing, specifying, constructing and documenting a system. It has the tools to describe a schema of the system (the model), including conceptual issues such as system functions, expressions of programming languages, database schemes or reusable components. Besides, UML has several types of diagrams, which show different aspects of the entities represented.

The aim of UML is to model any kind of systems (not just software) using the concepts of object orientation. Its history [Ora02] started with Booch and Rumbaugh looking for a unified modeling language (UML) in 1994 under the auspices of Rational® Inc. After several revisions, in 2005, UML was approved by ISO as ISO/IEC 19501:2005 Information technology - Open Distributed Processing - Unified Modeling Language (UML) Version 1.4.2. Fig. 3.7 (p.57)

An UML model consists of three classes of construction blocks:

- Elements: The elements are abstractions of real or fictitious things such as objects or actions.
- Relationships: the elements relate to each other.
- Diagrams: These are collections of elements along with their relationships. The class diagram shows a set of classes, interfaces and relationships. This is the most common diagram in describing the design of object-oriented systems.

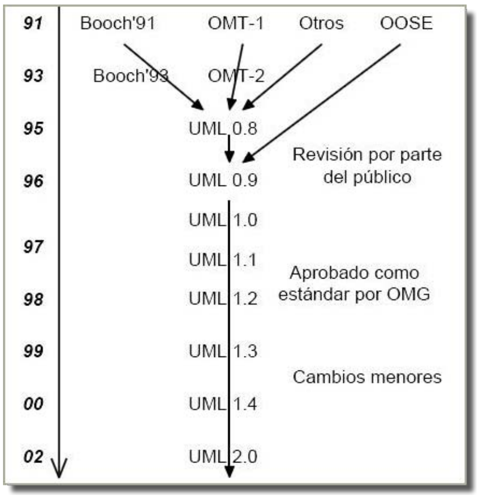


Figure 3.7 – UML timeline. (Unified Modeling Language (UML) [Ora02])

A diagram is a graphical representation of a set of elements along with their relationships. In order to properly represent a system, UML offers a wide variety of diagrams to visualize the system from several perspectives. UML 2.0 includes 13 types of diagrams. To understand it is useful to categorize them hierarchically, as shown in Fig. 3.8 (p.58).

3.5 Conclusions

Despite a concept as a mental construction, several techniques have been developed in order to categorize or classify them. The useful techniques are the ones with a graphical representation capability.

The aim of UML is to model any type of systems (not just software) using the object orientation concepts.

This research uses concepts maps and UML class diagram in next chapter in order to create a methodology of conceptual analysis. Besides, a variation of formal concept analysis to model the relation between two concepts in chapter 7 (Formal security model) is applied.

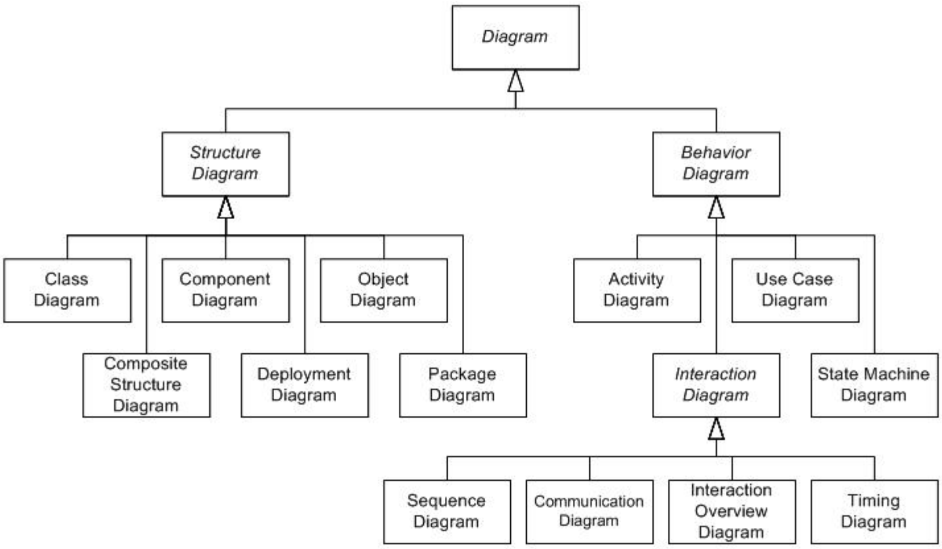


Figure 3.8 – UML Diagrams. (Adapted from Wikipedia)

Bibliography

[ABH+02] J.A. Alonso, J. Borrego, M.J. Hidalgo, FJ. Martín, and J.L. Ruiz. Una introducción al análisis formal de conceptos en pvs. *I Taller Iberoamericano sobre Deduccion Automatica e Inteligencia Artificial,IBERAMIA*, pages 33–46, 2002.

[Bal97] David A. Baldwin. The concept of security. *Review of International Studies*, 23:5–26, 1997.

[BB95] Tony Buzan and Barry Buzan. *The Mind Map Book*. BBC Books, London, 2 edition, 1995.

[Cap03] Luiz Fernando Capretz. A brief history of the object-oriented approach. *ACM SIGSOFT Software Engineering Notes*, 28(2):6, 2003.

[CY91] Peter Coad and Edward Yourdon. *Object-Oriented Analysis*. Prentice-Hall, London, 1991.

[DM08] L. Dillard and B. Myers. Visual teaching tools: Concept maps. Technical report, University of Florida, May 2008.

[Epp06] Martin J Eppler. A comparison between concept maps, mind maps, conceptual diagrams, and visual metaphors as complementary tools for knowledge construction and sharing. *Information Visualization*, 5(3):202–210, 2006.

[fHI04] Canadian Institute for Health Information. North american collaborating center. 10th annual conference on international classification of functioning, disability and health (icf). 2004.

[Fre04] Lee A. Freeman. The effects of concept maps on requirements elicitation and system models during information systems development. In *Proc. of the First Int. Conference on Concept Mapping*, 2004.

[Gog05] Joseph Goguen. What is a concept? *Lecture Notes in Computer Science : Conceptual Structures: Common Semantics for Sharing Knowledge*, pages 52–77, 2005.

[GSC91] S Goutas, P Soupos, and D Christodoulakis. Formalization of object-oriented database model with rules. *Information and Software Technology*, 33(10):741 – 757, 1991.

[GW97] Bernhard Ganter and Rudolf Wille. *Formal Concept Analysis: Mathematical Foundations*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 1st edition, 1997.

[Hin98] M. L. Hines. Conceptual object-oriented database: A theoretical model. *Information Sciences*, 105(1-4):31 – 68, 1998.

[Hub07] Thomas Hubbard. Concept map based software engineering. Master's thesis, Tufts University, 2007.

[ISO99] Switzerland ISO, Geneva. Iso/fdis 704 terminology work - principles and methods, 1999.

[Kho90] S Khoshafian. Insight into object-oriented databases. *Information and Software Technology*, 32(4):274 – 289, 1990.

[KO01] Sergei O. Kuznetsov and Sergei A. Obiedkov. Algorithms for the construction of concept lattices and their diagram graphs. In Luc De Raedt and Arno Siebes, editors, *PKDD*, volume 2168 of *Lecture Notes in Computer Science*, pages 289–300. Springer, 2001.

[KST92] Won Kim, Mark Scheevel, and Chris Tomlinson. Object-oriented databases for new applications. *Future Generation Computer Systems*, 7(2-3):317 – 327, 1992.

[LO02] Wen-Yau Liang and Peter O'Grady. An object-oriented approach to the concurrent engineering of electronics assemblies. *Computers in Industry*, 47(2):239 – 254, 2002.

[MFC99] Claudio Maffezzoni, Luca Ferrarini, and Emanuele Carpanzano. Object-oriented models for advanced automation engineering. *Control Engineering Practice*, 7(8):957 – 968, 1999.

[NC04] Joseph D. Novak and Alberto J. Cañas. Building on new constructivist ideas and cmaptools to create a new model for education. Technical report, Technical Report IHMC CmapTools, 2004.

[NC06] Joseph D. Novak and Alberto J. Cañas. The theory underlying concept maps and how to construct them. Technical Report 2006-01, Technical Report IHMC CmapTools, 2006.

[O'D05] Mike O'Docherty. *Object-oriented analysis and design : understanding system development with UML 2.0*. John Wiley & Sons, 2005.

[Ora02] Enrique Hernández Orallo. El lenguaje unificado de modelado (uml). *Manuales Formativos ACTA, nş 26*, October 2002.

[Pre05] Roger S. Pressman. *Software engineeering: a practioner's approach*. McGraw-Hill, Boston, EUA, 6 edition, 2005.

[Pri06] Uta Priss. Formal concept analysis in information science. *Annual Review of Information Science and Technology*, 40:521–543, 2006.

[SZ09] Anton Sodja and Borut Zupancic. Modelling thermal processes in buildings using an object-oriented approach and modelica. *Simulation Modelling Practice and Theory*, 17(6):1143 – 1159, 2009.

[TSK06] Pang-Ning Tan, Michael Steinbach, and Vipin Kumar. *Introduction to Data Mining*. Pearson Education, 2006.

[VML02] Petko Valtchev, Rokia Missaoui, and Pierre Lebrun. A partition-based approach towards constructing galois (concept) lattices. *Discrete Mathematics*, 256(3):801–829, 2002.

[Wil82] R. Wille. Restructuring lattice theory: an approach based on hierarchies of concepts. In *Rival, I. (ed.): Ordered Sets*, pages 445–470. Boston, 1982. seminal publication on formal concept analysis.

CHAPTER 4 Knowledge Modeling

"Those who would give up essential Liberty, to purchase a little temporary Safety, deserve neither Liberty nor Safety."[a]

— Benjamin Franklin

[a] A false dichotomy. There is no security without privacy.

Contents

This chapter proposes a methodology that can be applied to any knowledge area to make a model based on non-formal descriptions (text, polls, surveys ...), resulting in a graphic diagram; a class diagram.

The methodology relies on knowledge engineering.

In order to obtain a security concept model, knowledge engineering and concept analysis techniques are used. Concept Analysis, a branch of analytical philosophy, aims at decomposing the elements, relations and meanings that compose a concept [BR09][Nuo10]. There are several methods such as the Wilson's method [Wil63], the Rodgers evolutionary method [Rod89] or the Walker and Avant model [Wal95]. Knowledge Engineering was defined in 1983 by Edward Feigenbaum and Pamela McCorduck [FM83] as "an engineering discipline that involves integrating knowledge into computer systems in order to solve complex problems normally requiring a high level of human expertise".

Our model has to be constructed from existing security studies, models and concepts related to security. The desired characteristics for the model are:

- As simple and intuitive as possible.
- As generic as possible.
- Easiness to extend the model with new knowledge and the ability to share it.
- Could be implemented in computer systems and thus automated.
- Capable of store knowledge and deduce new and useful knowledge.
- Provide some kind of security measures

In order to analyze and design the model a systematic approach is required. Thus the schematic steps to achieve the goal are:

- Review several sources related to security.
- Extract knowledge from sources.
- Integrate all these knowledge in a single model.

4.1 Knowledge Based Concept Analysis (KBCA)

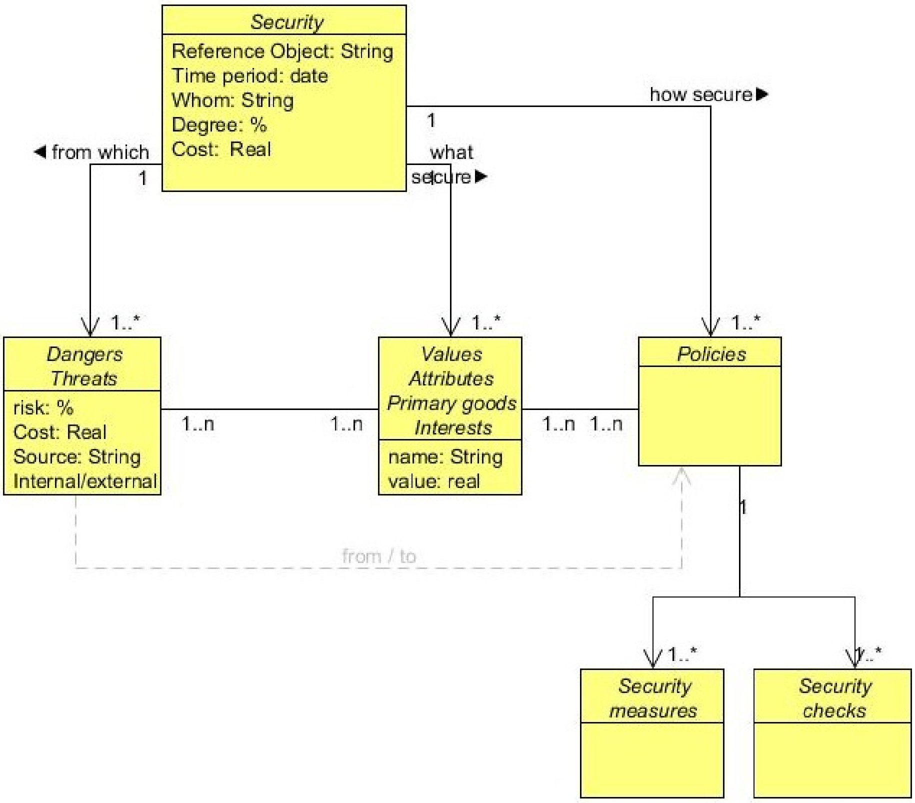
KBCA was proposed by Colobran and Basart [CB13]. It's a methodology of knowledge extraction and representation from a source such as a report, an article, a book an interview or any other source. The method obtains the elements from a text and using knowledge elicitation, concept maps and UML, creates a graphical representation of a concept, a class diagram. This methodology is a seven stepwise way and as stated by the authors, the methodology could be extended in order to allow obtaining a class diagram from several sources. In table 4.1 (KBCA Flow diagram) the steps are shown.

	Stage	**Description**	
Step 1	Choose	Choose knowledge source.	Knowledge elicitation
Step 2	Extract	Select key text elements.	
Step 3	Collect	Insert into database and number.	
Step 4	Categorize	Create list of categories.	
Step 5	Assign	Assign into category.	
Step 6	Map elements	Create concept map.	Concept map
Step 7	Class diagram	Construct class diagram.	UML

Table 4.1 – KBCA flow diagram.

Relevant features of KBCA are:

- It is incremental.
- It extracts the relevant features of the used source.
- It is possible extend in order to use it in several sources.
- It is possible to use sources from different fields.
- The outcome is a graphical model that can be implemented using object oriented technologies as well as knowledge engineering (Figure 4.1).



-- Threat --

overpopulation
environment
...

-- whom --

Individual
Institution
State
System

-- value --

freedom form physical violence	(if whom = individual)
Absence fo threats to one's property	(if whom = individual)
protect personal belongings	(if whom = individual)
economic wherewithal	(if whom = individual)
functions	(if whom = institution)
purpose	(if whom = institution)
territorial integrity	(if whom = state)
freedom of action	(if whom = state)
healthy economy	(if whom = state)
preservation of "way of life"	(if whom = state)

-- Reference object --
-- Object of security --

International security

Economic security
Environmental security
Social security
Health security
Entitlement

Personal security

Figure 4.1 – Class diagram of a security concept.

4.1.1 Method in detail

A detailed explanation of the methodology is provided in the whole article that is in **Appendix D**.

4.2 Extended KBCA (E-KBCA)

To obtain a class diagram from several sources, the KBCA methodology needs to be extended. Extending KBCA basically lies in obtaining the final class diagram incrementally using concept maps and class diagrams obtained from each source. Broadly speaking, the tasks involved in the process are shown in Figure 4.2.

- **Task 1.** It involves the creation of a research question and a systematic review in order to get the sources related to what we want to model. The sources have to focus on the concept under study, either from an operational, theoretical or descriptive point of view.

  A systematic review ensures the sources are relevant to the research question. There are several systematic review methodologies such as [Kit04] [BMN05].

- **Task 2.** Once the systematic research is finished, the knowledge needs to be extracted. Thus for every related source, the seven step KBCA methodology is applied. The result is a class diagram for every document.

- **Task 3.** As all sources are related to the research question, their class diagram will be similar. All those schemes are unified in one.

4.2.1 Method in detail

Source selection. It is done by means of any methodology of systematic review such as [Kit04] [BMN05]. These tasks involve creating a research question, choosing literature sources and sieving sources in order to retrieve only the relevant sources.

Knowledge extraction and representation. KBCA is applied to every selected source. The features are extracted. Conceptual map and class diagram are created.

Model creation. Fusion of the conceptual maps and the class diagrams to a general conceptual map and class diagram. The outcome is the model.

As the methodology is incremental by its nature, after obtaining every class diagram it could be integrated in the final model. Used this way, the model creation task is made at every cycle. After adding the last document concept map, the final model is done. In Figure 4.3 the stages are detailed.

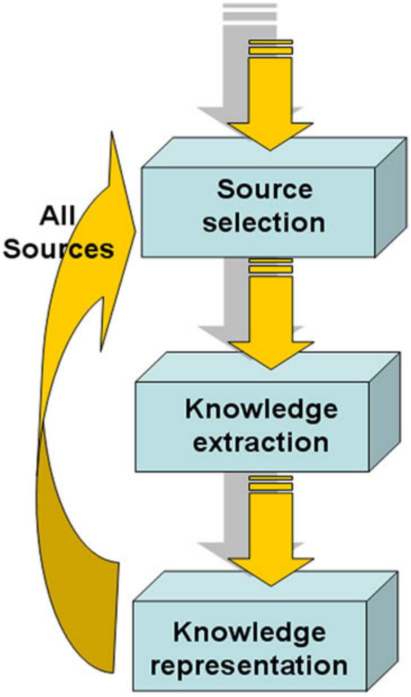


Figure 4.2 – Step schema.

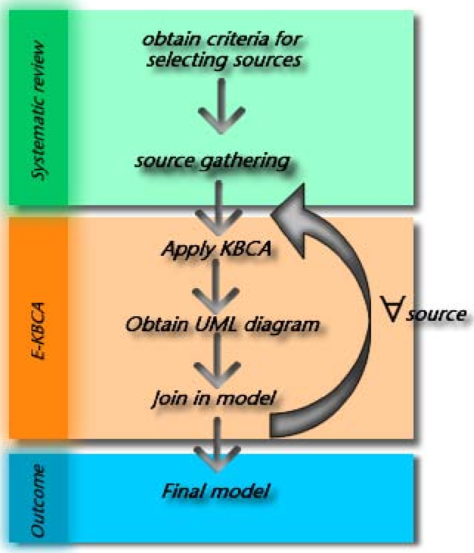


Figure 4.3 – Stages.

4.3 Conclusions

A methodology for exploring the underlying elements in a concept and the relationships between them is proposed. The result is an abstract concept, which requires specific elements to produce “the definition”. This definition is extremely flexible and can be adapted to any field.

Bibliography

[BMN05] Jorge Biolchini, Paula Gomes Mian, and Ana Candida Cruz Natali. Systematic review in software engineering. Technical Report RT-ES 679/05, COPPE/UFRJ, Rio de Janeiro, RJ, Brasil, May 2005.

[BR09] Moyra A. Baldwin and Pat Rose. Concept analysis as a dissertation methodology. *Nurse Education Today*, 29(7):780 – 783, 2009.

[CB13] Miquel Colobran and Josep M. Basart. Knowledge based concept analysis method using concept maps and uml: Security notion case. *World Academy of Science, Engineering and Technology*, 7(2):437 – 444, 2013.

[FM83] E.A. Feigenbaum and P. McCorduck. *The Fifth Generation*. Addison-Wesley, 1983.

[Kit04] B. Kitchenham. Procedures for performing systematic reviews. Technical report, Keele University and NICTA, 2004.

[Nuo10] Anita Nuopponen. Methods of concept analysis-a comparative study. *LSP Journal-Language for special purposes, professional communication, knowledge management and cognition*, 1(1):4 – 12, 2010.

[Rod89] B. L. Rodgers. Concepts, Analysis, and the Development of Nursing Knowledge: The Evolutionary Cycle. *Journal of Advanced Nursing*, 14:330–335, 1989.

[Wal95] Kay Coalson. Walker, L.O. Avant. *Strategies for theory construction in nursing*. Appleton & Lange, Norwalk, CT, 3rd ed. edition, 1995.

[Wil63] John Wilson. *Thinking with concepts*. Cambridge University Press, 1963.

Part III

Security Concept Model

CHAPTER

5

Value Based Security Framework

"There is no such thing as perfect security, only varying levels of insecurity."

— Salman Rushdie

Contents

Other knowledge areas have formulated models for the concept of security. In this chapter a review is made. After that, the methodology developed in the previous chapter is applied. Security is a concept, and therefore the proposed methodology can be applied. The chapter attempts to identify the common elements underlaying the notion of security. To do so, first of all, relevant documents in the field of security are selected and the methodology is applied in order to obtain a security framework.

In the last chapter a methodology called KBCA has been introduced in order to be applied to the concept of security. Therefore, in order to obtain the security framework, a literature review of existing security concepts is made. The research is based on a set of sources related to the concept of security from a conceptual or operational perspective.

Security is a concept present in some way in all aspects of human life. Hence it can be seen from many perspectives and thus analyzed from any of them like social, psychological, economic, technological, human or even environmental. Several scholars have completed studies on the security concept such as Hobbes [Hob99], Kant [Kan02], Ullman [Ull83], Buzan [Buz83], Baldwin [Bal97], Roschild [Rot95] among many many others. The approaches could be classified in the following way:

Analytical Define security in terms of describing the concept and its features is an analytical description. A study of this kind, conducted by [Mes08], highlights properties like threat in the core concept (Figure 5.1).

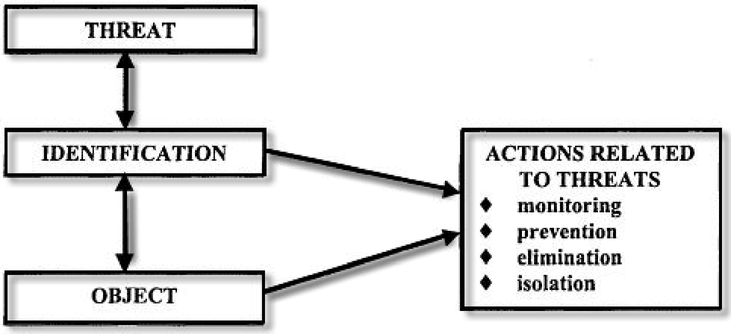


Figure 5.1 – The core of the concept of security. (How Complex Systems Studies could help in Identification of Threats of Terrorism? (Mesjasz 2008))

Relational The relational approach tries to define security in terms of elements and relations in a context-neutral way. The relational approach is more interested in describing and modeling the behavior than the essence of security.

Peter Digeser [Dig94] highlights that security does not have any "fixed content" because its meaning in any context will depend upon what is to be secured (called the "referent object"). By this author, most conceptions of security presuppose content and "it attempts to import a fixed content into a term that necessarily permits variability".

Rhonda Powell [Pow08] stated that any security needs to specify "(1) security for whom (an agent or patient), (2) security of what (an interest or value), (3) security from what (a threat or risk) and (4) who or what will provide protection".

Therefore, the relational approach leads to a concept with no meaning until it is specified. Besides, the concept does not depend on the level and context it is applied.

Operational An operational analysis involves some kind of premise. Thus, applied to a concept, the operational analysis includes relations with other concepts, the context that is analyzed and therefore how it could be defined. There are a plethora of works on this area. Wolfers [Wol52] in his remarkable work, introduced several elements. According to Wolfers, security has a wide range of goals; is a degree of protection to values previously acquired; implies a time range; protect and preserve core values. Ullman [Ull83], mainly concerned on national security, introduced new elements; several classes of dangers; several classes of measures. Baldwin [Bal97] in his 1997 paper on security highlights "..Most such efforts, however, are more concerned with redefining the policy agendas of nation-states than with the concept of security itself.". His conceptual analysis stated several specifications in the concept of security such as, How much security?; From what threats?; By what means?; At what cost? and In what time period?

Formal A formal analysis in concepts implies elements, relations and a methodology to get both and how they are connected one each other. Formal concept analysis (FCA) methodology was introduced by Uta Priss [Pri06]. A description is made in **section 3.2**

5.1 Model creation

5.1.1 Source selection criteria

In order to create the security framework, the E-KBCA methodology is used. To perform this work, information retrieval and survey methodologies relied on [Kit04] [BMN05] are used to obtain a set of primary sources to elaborate the framework.

The main criteria used in this study to select works are:

- Articles focused on the security concept.

- Articles related to the security definition. Definitions should not be operational, i.e. not using the notion of security to make an ad hoc security.
- The articles have to be descriptive about the security model proposed.
- Articles from any knowledge area related to safety such as International Relations or Computer Security.

The systematic review needs a research question, used in all selected sources. The research question is :

Security AND (model OR modeling OR definition OR redefinition OR analysis OR concept analysis OR formal)

The sources selected (Table 5.1) has been chosen according to the following criteria:

- Reliable sources.
- The Sources are cornerstone in the field.
- They posses quality criteria in their content.
- Include technical sources and social sources in order to obtain a broader and more generic view of the security concept. For example, IEEE only stores technical articles.

Sources
Association for Computing Machinery (ACM)
ScienceDirect
Google Scholar
Institute of Electrical and Electronics Engineers (IEEE)
JSTOR
Springer Verlag

Table 5.1 – Selected sources.

	Research Question	Inclusion Criteria	Exclusion Criteria	Relevant Articles
(ACM) Association for Computing Machinery	122	94	46	14
Google Scholar	366	127	37	8
(IEEE) Institute of Electrical and Electronics Engineers	122	49	25	4
JSTOR	124	21	7	2
ScienceDirect	116	54	26	2
Springer Verlag	65	36	15	2

Table 5.2 – Source gathering.

5.1.2 Source gathering

The author started by gathering, as far as possible, any publication related to security and its various aspects. The search was conducted inside the relevant and known sources of literature shown in Table 5.2. As the search is wide, there are a lot of articles proposed. Initially 915 articles where selected by the search criteria, search string and literacy sources. From those, only 166 articles have been selected according with the inclusion and exclusion criteria. A first read was performed in order to get a general idea and discard initially selected articles. A second read was carried out for deeper understanding and analysis of models, concepts and relations among them. As a result, 32 articles where considered to contain some relevant information for the research. Finally, in the process of selecting relevant articles, a quality analysis of the concepts used in those articles lead us to identify the most used (Figure 5.2).

The list of relevant works are contrasted with experts of security in the field of International Relations Dr. Arcadi Oliveres and Dr. Rafael Grasa. As a result, some works were incorporated in the final list [Wol52] [Ull83] [Buz83] [Dig94] [Rot95] [Bal97] [Mes08] [Pow08]. The results that highlight from this analysis are:

- The field of International Relation have tackled in great detail the concept and its elements.

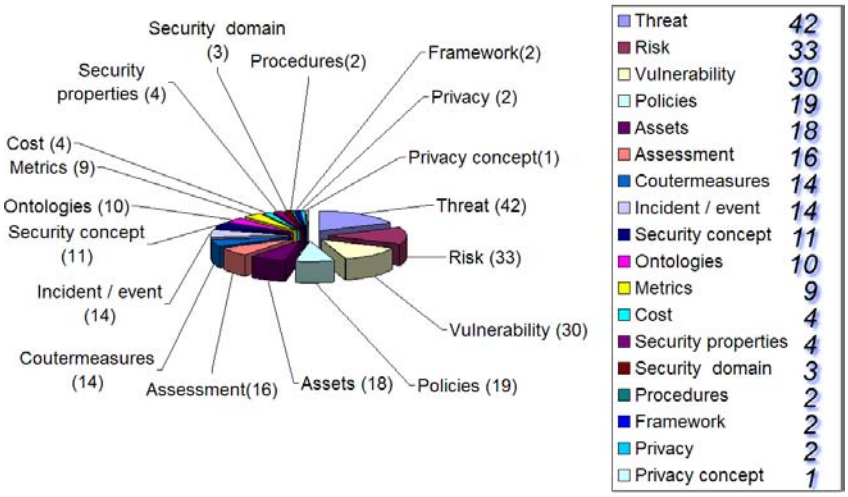


Figure 5.2 – Concepts from articles.

- A lack of conceptual studies of the security concept in the field of computer science.
- Computer science when referring to security is mainly on *Information Security*.
- The studies in computer science that best describe the security concept are close related to ontologies.
- Studies related to Knowledge Engineering and Ontologies are the ones that better operate security from a conceptual perspective.

5.1.3 Knowledge extraction

Once all relevant articles are gathered, next stage is to apply E-KBCA. The methodology is applied and several CMmaps of the security concept representing the underlying notion of security in several fields are obtained (figure 5.3). The CMaps could be reviewed in **Appendix A**.

5.2 Framework Description: Security Hexagon Model (SHM)

After applying the methodology, the final class diagram is obtained (Figure 5.4). The concept of security is expressed in UML[10] notation. The concept, ordered in a hierarchical way is shown in Figure 5.5

[10] A description of UML (Unified Modeling Language) can be found in chapter 3, section 3.4.3. Graphical representation.

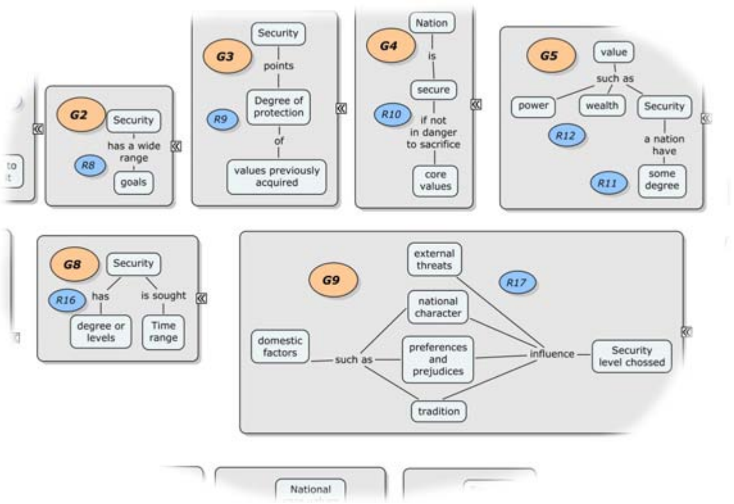


Figure 5.3 – Security concept cmap from an article.

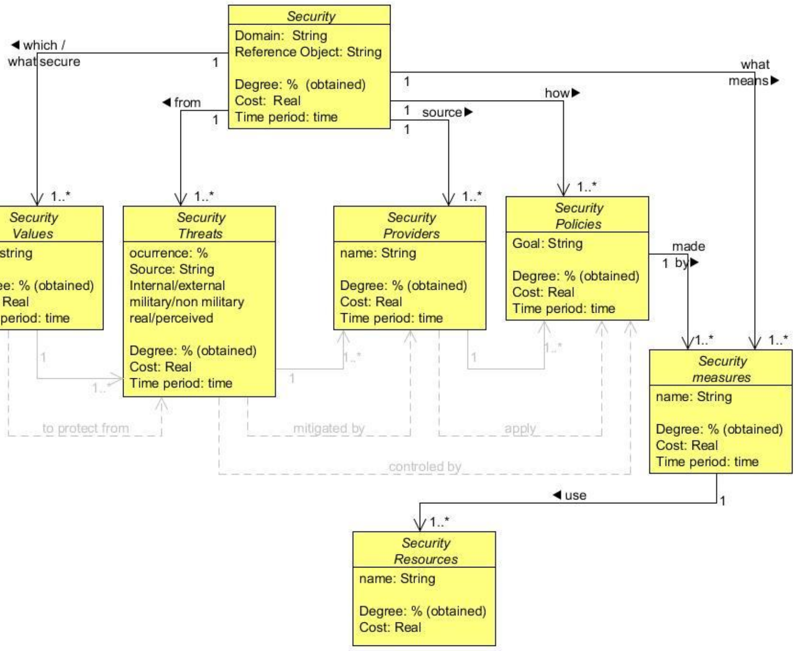


Figure 5.4 – UML representation of the obtained CMap.

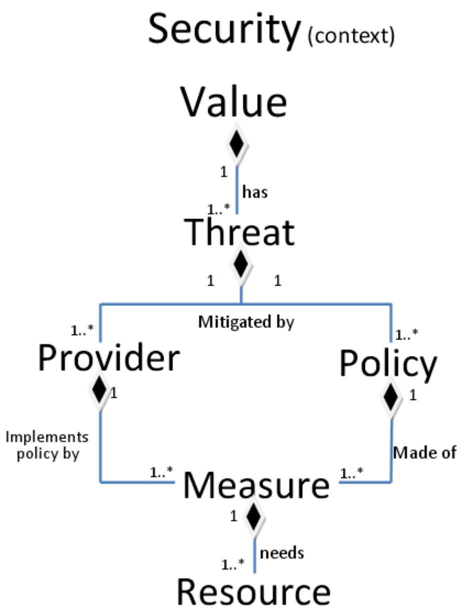


Figure 5.5 – Security concept hierarchically ordered.

In order to simplify working with the model, it could be represented by the Security Hexagon Model (SHM). This representation places concepts in its vertex and relations in its edges as shown if Figure 5.6. Thus, the concepts involved on the security framework are:

- **Context or Referent.** Provides the frame or reference to apply the security concept, i.e. national security and personal security are both securities, but very different in relation on what and how it is applied.

- **Values.** The elements one is interested in protecting. Let's bear in mind that "values" are completely subjective. Therefore, there are just a few values in any security. For example, homeland security has only three core values, identity, independence and territorial integrity [Wol52].

- **Threats.** Objects that "supply" uncertainty (lack of safety). The threats perception gives us the amount of perceived insecurity. The more we have identified and the higher the perceived probability that happens, the less we will feel secure. There are several threat definitions [SGF02, page 8] [Gro09, page 3]. Any security scenario has a number of threats. For example in the computer science field, the document SP 800-30 "Risk Management Guide for Information Technology Systems" presents a short list of common threat sources [SGF02, page 13].

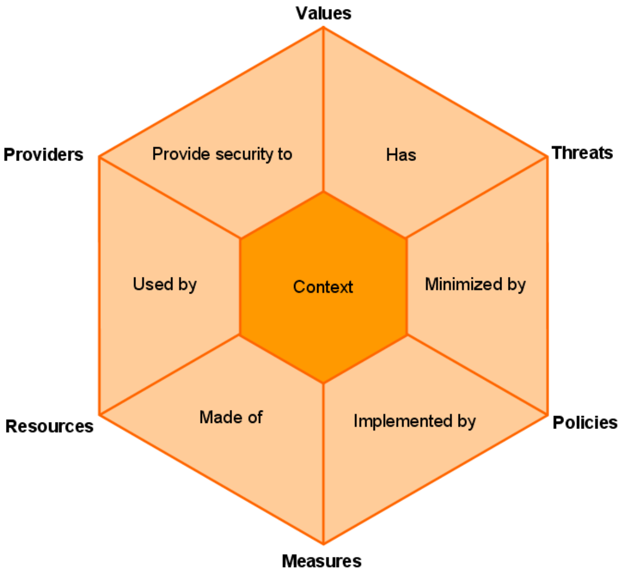


Figure 5.6 – Security Hexagon Model (SHM).

- **Providers or Agents**. Elements that provide security. If nothing or no one provides security to the values we want to protect, then we have no security at all. As Wolfers stated, "security is a matter of degree" [Wol52] and agents, that provide security, are the indicators that supply the degree of perceived security.

- **Policy**. A set of actions in order to mitigate the influence of a threat. A high level definition of policy is "acceptable behavior, expected practices, and responsibilities for an organization" [McG02].

- **Measure**. There are many definitions for the concept. Indeed the essay by Hecker [Hec08] highlights several definitions and even emphasize that the word metric and measure are used with the same sense. Only a high level definition for measure is needed "Procedure or mechanism that reduces security risk" [Min06].

- **Resource**. Resources are all that is needed in order to achieve the measure goals. The definition provided by the English Oxford dictionary fits the proposal "a stock or supply of money, materials, staff, and other assets that can be drawn on by a person or organization in order to function effectively"

Ordered in a general-specific manner, the resulting conceptual ordering is hierarchic (shown in Figure 5.7). At the top is context, the most general. That is the security we are defining. At the bottom, the most “simple” or concrete element that the security model could be decomposed. Besides, every security concept answers a question as shown in Table 5.3.

	Concept	Question
C	Context	Named security
V	Value	What / Whom (protect)?
T	Threat	Of what (protect)?
		Against what (protect)?
I	Policy	How (protect)
P	Provider	By who (protect)?
		Whom (protect)?
M	Measure	By means of
S	Resource	Using what

Table 5.3 – Concepts.

5.3 Graphical Framework representation

The “Hexagon security graphic” (Figure 5.6) is useful in working on security. It intuitively shows the concepts present and how they relate. Figure 5.7 shows, by means of a concept map, the hierarchical relation of the concepts and how they influence each other. If any element is missing, indeed, there is no security at all.

5.4 Conclusions

This chapter has analyzed several approaches to the concept of security. All of them rely in several elements and relations, but just a few tackle security in a context-free way. From the methodology developed, a new framework of security is obtained and it is represented as a Hexagon. This representation places the concepts (Table 5.3) in its vertex and its relations in the edges (Figure 5.6). The next chapter will analyze deeply the elements and features of the framework.

The intention is not to exhaust the concept of security, because most probably it has no meaning because the concept itself is dynamic, but to have more tools to explore the concept in a systematic way.

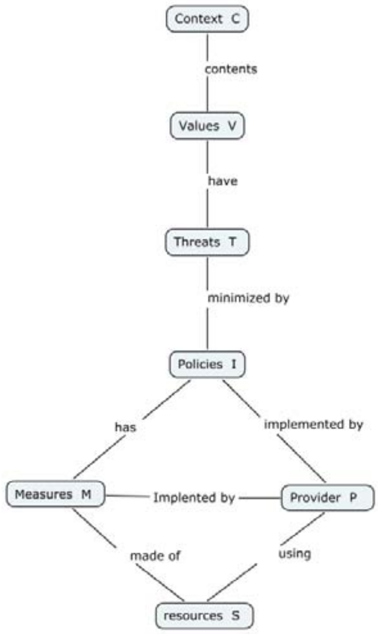


Figure 5.7 – Conceptual map of the security notion.

Bibliography

[Bal97] David A. Baldwin. The concept of security. *Review of International Studies*, 23:5–26, 1997.

[BMN05] Jorge Biolchini, Paula Gomes Mian, and Ana Candida Cruz Natali. Systematic review in software engineering. Technical Report RT-ES 679/05, COPPE/UFRJ, Rio de Janeiro, RJ, Brasil, May 2005.

[Buz83] Barry Buzan. *People, States and Fear*. Harvester- Wheatsheaf, Brighton, 1983.

[Dig94] Peter Digeser. The concept of security. Presented at the Annual Meeting of the American Political Science Association 14 September 1994. (Unpublished) Obtained from author., 1994.

[Gro09] The Open Group. Risk taxonomy. Technical report, The Open Group, January 2009.

[Hec08] A. Hecker. On system security metrics and the definition approaches. In *Emerging Security Information, Systems and Technologies, 2008. SE-*

CURWARE '08. Second International Conference on, pages 412–419, 2008.

[Hob99] Thomas Hobbes. *The Leviathan*. Alianza, 1999.

[Kan02] I. Kant. *Sobre la paz perpetua*. Alianza Editorial, 2002.

[Kit04] B. Kitchenham. Procedures for performing systematic reviews. Technical report, Keele University and NICTA, 2004.

[McG02] M. McGovern. Opening eyes: building company-wide it security awareness. *IT Professional*, 4(3):52–54, 2002.

[Mes08] Czeslaw Mesjasz. How complex systems studies could help in identification of threats of terrorism? *Unifying Themes in Complex Systems IV*, pages 379 – 389, 2008.

[Min06] Ministerio de Administraciones Públicas. *Magerit-versión 2. Metodología de Análisis y Gestión de riesgos de los sistemas de información.* Ministerio de Administraciones Públicas. España., June 2006.

[Pow08] Rhonda L Powell. *Security and the right to security of person*. PhD thesis, Oxford University, UK, 2008.

[Pri06] Uta Priss. Formal concept analysis in information science. *Annual Review of Information Science and Technology*, 40:521–543, 2006.

[Rot95] Emma Rothschild. What is security? the quest for world order. *Daedalus*, 124(3):53–99, June 1995.

[SGF02] Gary Stoneburner, Alice Goguen, and Alexis Feringa. *Risk Management Guide for Information Technology Systems*. NIST National Institute of Standards and Technology, July 2002.

[Ull83] Richard H. Ullman. Redefining security. *International Security*, 8(1):pp. 129–153, 1983.

[Wol52] Arnold Wolfers. National security as an ambiguous symbol. *Political Science Quarterly*, 67(4):481–502, 1952.

CHAPTER

6 Framework Analysis

> *"If you want total security, go to prison. There you're fed, clothed, given medical care and so on. The only thing lacking... is freedom."*
>
> **— Dwight D. Eisenhower**

Contents

This chapter analyzes the features of the framework obtained. After that, a definition of security is proposed. Finally, the framework is applied to several securities. Computer security, therefore, becomes another security of all possible.

In this chapter, based on the analysis of the framework, we propose a definition of computer security. This definition is inclusive, in order that it can be applied to many scenarios. Besides, the way the framework and the definition is obtained allow deep conceptual work.

6.1 Security Definition

It may seem unrealistic to use constructions of the security concept from the field of International Relations for use in computer security. As the concept of security, in this field, has a long history and experience, it can be extremely helpful to discover the common ground. Besides, "Without a precise definition of what security means and how a computer can behave, it is meaningless to ask whether a particular computer system is secure." [Lan81].

Indeed, there is no unified definition of computer security. Therefore, there are two approaches. One of them is focused on the lack of definition, is stated for a long time. In 1981, Carl E. Landwehr highlighted in his paper *Formal Models for Computer Security* that there is no precise security definition in the field of computer security. Anderson [And03] claims for "a new definition of information security". His paper detailed the flaws and why another definition is needed. Basie et al. [vSvS05] proposed to call it business security instead of information security. The other one is focused on looking for common knowledge outside Computer Science [Nis05].

Computer security that mainly protects the information, is revealed as an inadequate concept. Situations such as the following are not covered by current computer security definitions.

- Ariadne 501 exploded 40 seconds after takeoff [Lev04].
- Mars Polar Lander crashed into the surface of Mars at speed 22 m/s [Lev04].
- Radiation. Radiation management involves computer systems [irp00].

6.1.1 Definition

Based on the obtained model, security can be defined in terms of the key elements. Currently, the accepted security definition in the International Relations field is the one proposed by Wolfers [Wol52] "Security in an objective sense, measures the absence of threats to acquire values, in a subjective sense, the absence of fear that such values will be attacked.”.

Computer Security definitions are mainly operational [Lan01] [Sch08]. Because computing has now exceeded purely technical aspects and it is currently involved in most aspects of social issues, security definition should be enlarged in order consider the new scenario of information technology.

Therefore, from the obtained framework, we can formulate a general definition of the security concept.

Security. Identification of threats in a context in order that a set of agents, by means of policies, protect the desired values during a period of time.

6.2 Context or Referent

Due to the subjectivity of the concept, there is not much point to speak about security without a context. One can speak, for example, on information security, national security, legal safety and personal safety even. Hence, it is important to limit the scope (context) of defined security.

6.2.1 Context level

As important as the context is the level at which it applies. Emma Rothschild [Rot95] stated that security notion is currently extended. Therefore, security can be grouped into at least four levels (Table 6.1).

Context level
Individual
Organizational or group
National or state
International or global

Table 6.1 – Context level.

6.3 Value as the central element

The Oxford English Dictionary defines value as “The regard that something is held to deserve; the importance, worth, or usefulness of something”. The idea behind value, in this sense, is for what one should spend time, efforts, economical resources, material resources, human resources and is willing to renounce somethings even.

The hexagon security framework uses *value* in the same sense. Hence, values are the elements one is interested to protect. Values are, in the framework, the center of security. If one knows *what* to protect then *how* to protect is followed. The policies, resources and measures are meaningless without the knowledge on what one is interested in protecting.

Besides, values by themselves are subjective. Indeed, protecting the same value has different resources according to the level it is applied. Table 6.2 outlines an example with information as a value.

	Individual	**Organizational**	**National**	**Global**
Information	- lock - safe ...	- cryptography - backups - off-line data ...	- department - cryptography - backups ...	- multiple copies - cryptography - agreements ...

Table 6.2 – Example of resources from a value

Threats depend on values. As policies, measures, providers and resources are threat dependent, there is no security at all without at least one value. The most important in the framework is setting the values because the remaining elements rely on them.

6.4 Modeling time in Security

Our framework unveils time as a key element in security. In order to implement time, a model of the behavior of security depending on time is required. The time model is not part of the security framework and therefore, our proposal to modeling security is made in the applied part of this research in chapter 8, entitled “Security Level Time Function”. The model proposed is lightweight, capable and easy to be implemented and designed in such a way that could be used as a security metric.

6.5 Relations between concepts

The Security Hexagon Model representation places concepts (Table 5.3) in its vertex and relations in the edges (Figure 6.1).

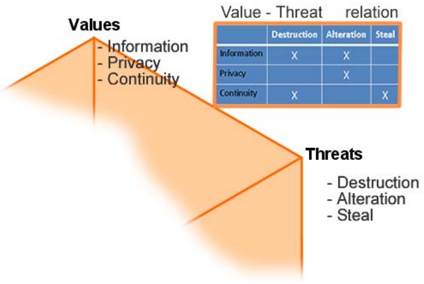
- Vertex: Concepts

  Concepts are defined by the model. Any concept, indeed, is a list or a set of elements related to the defined security (Figure 6.1).

- Edges: Relations

  Edges represent how are related the two concepts that are in the vertex. There is a relation between any both concepts and the relation is constructed by means of a variation of a formal concept (Figure 6.1).

Chapter 7 (Formal Security Model) describes in detail how the concept and the relations are described, constructed and how to operate with the elements.



	Destruction	Alteration	Steal
Information	X	X	
Privacy		X	
Continuity	X		X

Figure 6.1 – 2 vertex with concepts and relation at egde.

6.6 The role of Cryptography

Cryptography is the field of cryptology, which handles encryption techniques designed to alter the messages in order to make them unintelligible to unauthorized recipients. The sender hides or encrypts the message before transmitting it so that only authorized recipient can decipher it.

6.6.1 Societal value of Cryptography

With advances in Information Technology and Communications, cryptography has become essential due to the huge amount of communications and information that is being transmitted over networks and the content thereof. Hence, currently, financial transactions, medical records and private information exchange are usual and, in this sense, cryptography plays a key role in society by acting as a guarantor for the protection of social values such as privacy or wellness.

Moreover, cryptography currently has expanded its role in society and provides new elements of security. Authentication (to be sure about the sender and receiver), confidentiality (to know the message has not been seen by anyone else), Integrity (to be sure that the message has not been altered) and non-repudiation (to know by the sender that the recipient has seen the message).

6.6.2 Cryptography within Information Security

Cryptography becomes paramount when related to information security. In addition to cryptography protecting information during communications, it is also capable of protecting it against attacks that aim to access information. Therefore, any information repository such as cloud systems, network shares, flash or hard drives, tape systems or backup systems rely on cryptography to ensure that their content can only be accessed by the authorized entities.

6.6.3 Cryptography, the Swiss army knife in security framework

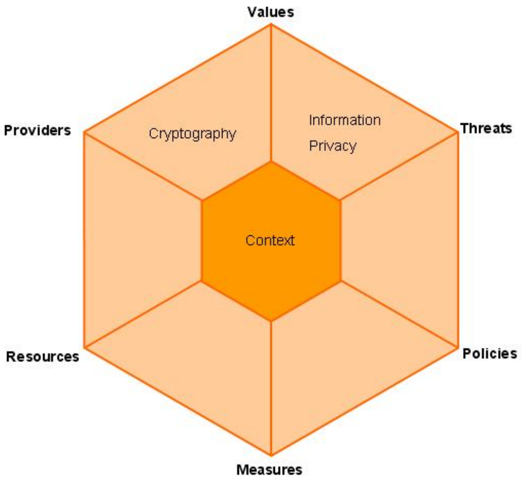


Figure 6.2 – Cryptography in Hexagon model.

Cryptography is currently considered the most important key element in computer security. In this framework, cryptography could be located as a provider a policy or a measure. Besides cryptography is close related to the remaining elements. When one of the values to protect is also privacy, cryptography plays a central role (Figure 6.2).

In the scenario that any security have to protect information and privacy, cryptography must be present in the security model because some threats will be information theft, privacy theft, information alteration, information destruction or information disclosure. Depending on the security design, cryptography can be a provider, a policy or some measures even. Whatever role cryptography plays, it cannot be excluded.

Besides, there are several other security scenarios that safety design involves communication between entities and the information is required neither not to be intercepted nor altered. In all of them cryptography must be present as a provider, policy or measure.

Finally, cryptography should be an element of security in any context level. For example, if one of the values to protect is information then cryptography is one element regardless of the context (individual, organizational, national or global).

6.7 Computer Security in security framework

This section analyzes how computer security fits into the proposed framework.

6.7.1 Values in computer security

Our framework of security involves just 2 values in computer security (Table 6.3).

Context	Value	Threat	Police
Computer security	Information Privacy		

Table 6.3 – Model values.

- **Information**. This value is the one that everybody is concerned about. We have to protect information from unauthorized use, access, alteration or destruction and assure information is used correctly from the authorized persons in the appropriate moments.
- **Privacy**. This value associated to computer security includes the need to protect information that is considered belonging to a person. This person has rights over this information and the system has to assure those rights.

6.7.2 CIA triad

The CIA triad, into this model, protects information against the threats of interruption, interception, modification and fabrication by means of policies, confidentiality, integrity and availability (Figure 6.3). Therefore, CIA triad becomes policies in order to protect information value.

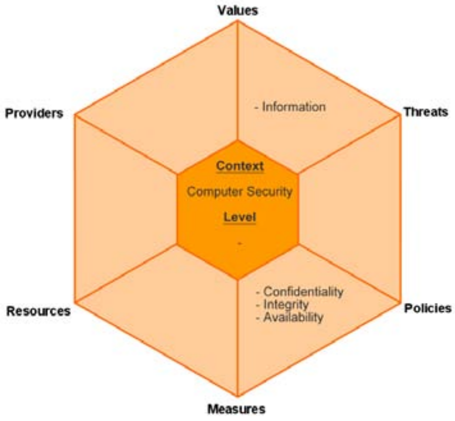


Figure 6.3 – Computer Security Hexagon model.

6.7.3 Computer Security context level

Computer security elements vary depending on the level considered.

6.7.3.1 Individual

A research study with Dr. Stephen Cheskiewicz was carried out about people's concerns about Internet security. A survey was published in SurveyMonkey®[11] in two languages (English and Spanish) and spread around the world. A total of 1622 answered surveys from a wide range of people were obtained. From some questions of the research we have achieved the values of computer security at an individual level. What people perceive as values to protect in computer security are privacy, children and personal economy. A more detailed description of this part of the survey appears in **Appendix F**. Surprisingly; people perceived the threats what Wolfers [Wol52] defined in 1952. Those things that "degrade" the quality of life (Figure 6.4).

6.7.3.2 Organizational

An analysis of the framework was made within a high technology computer institution (detailed in chapter entitled "Case Study"). The research with the CSUC institution raised another important value in order to be considered at this level; continuity. This value, associated to computer security, is mainly known as a disaster recovery plan (DRP) or business continuity plan (BCP). The value assures

[11] http://www.surveymonkey.com

the continuity of the organization in a catastrophic scenario (Figure 6.5).

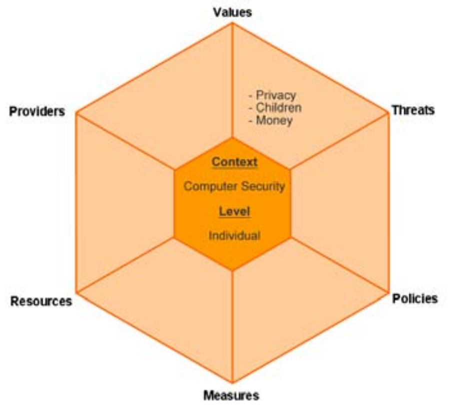


Figure 6.4 – Computer Security Hexagon model at individual level.

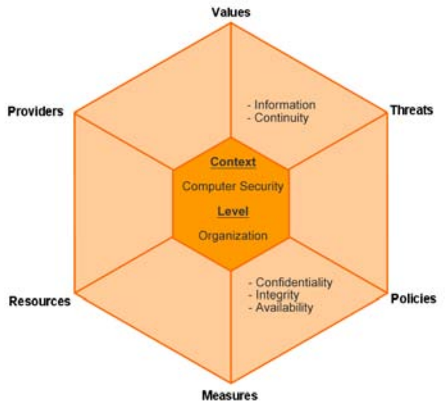


Figure 6.5 – Computer Security Hexagon model at organization level.

6.8 Security models in security framework

To verify that the framework fits in many scenarios, various existing securities are reviewed to see how they work. The proposed framework is generic enough to be used to model most securities. Security is subjective and all definitions share some elements and relations that could fit into the model as shown in table 6.4. Attending on the most important element, the values, the following securities are modeled:

Context	Value	Threat	Policy
Human Security [UND94]	People	Seven categories of threats	Human development Security Council ...
National Security [Wol52]	Sovereignty Independence Territorial Integrity	Terrorism Espionage War Natural disasters ...	Foreign Homeland Border Critical Infrastructure ...
Food Security [WFC74]	Food	Availability Access	Availability Access

		Utilization Stability	Use
Information security InfoSec	Information	Confidentiality breach	Bell-Lapadula Chinese wall
Information security InfoSec	Information	Integrity breach	Biba Chinese wall Clark-Wilson
CIA triad	Information	Interruption Interception Modification Fabrication	Confidentiality Integrity Availability
NIST SP800-30 [SGF02]	Information	Threat list	Confidentiality Integrity Availability
ISO 27000 [ISO13]	Information	Threat list	Confidentiality Integrity Availability
OCTAVE [SEI01]	Information	Threat list	Risk management
MEHARI [CLU10]	Information	Threat list	Confidentiality Integrity Availability
MAGERIT [Min06]	Information	Threat list	Confidentiality Integrity Availability

Table 6.4 – Model comparison.

6.8.1 Human security

The human security concept appeared in 1994 in the United Nations Program for Development [UND94]. Could be defined as "...the need to protect the free development of individuals in areas where human rights are threatened and violated" [Oro06]. For human security the value is a person or individual (Figure 6.6). The scope is huge and should include threats in seven areas always referred to people:

- Economic security
- Food security
- Health security
- Environmental security
- Personal security
- Community security
- Political security

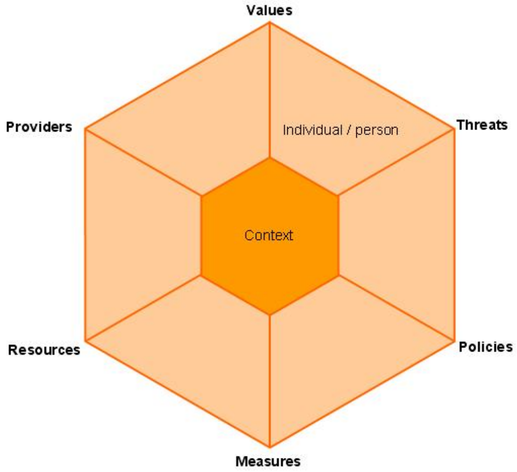


Figure 6.6 – Human Security Hexagon model.

6.8.2 National Security

There are several definitions of National Security, but one of them also uses the term *value* to refer the core elements to protect "the ability of a nation to protect its internal values from external threats" [BB66].

This security has the level defined (state) and the values to protect, sovereignty, independence and territorial integrity. Security, therefore, is set as shown in Figure 6.7.

6.8.3 Food Security

The World Food Summit of 1996 defined food security as existing "when all people at all times have access to sufficient, safe, nutritious food to maintain a healthy and active life" [Dec96]. For food security the value is food and health, and the level is the state (Figure 6.8).

6.8.4 Information related securities

The remaining securities in Table 6.4 have information as the value to protect. The majority of risk and information security methodologies have a threat list that exhibits the threats they are expected to mitigate.

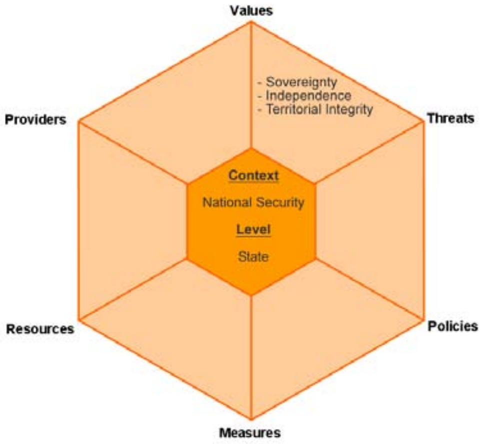


Figure 6.7 – National Security Hexagon model.

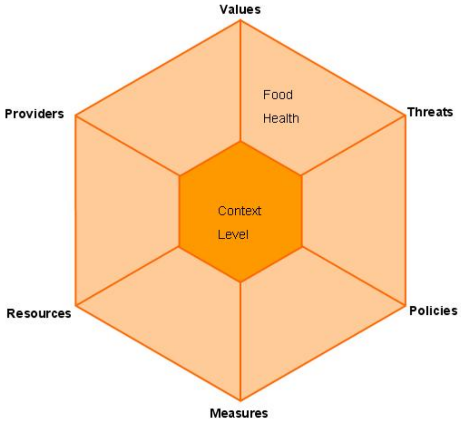


Figure 6.8 – Food Security Hexagon model.

6.9 Conclusions

This chapter has analyzed the framework obtained. Time appears as a key element. A definition of security is proposed and finally, the framework is applied to model several securities. According to the securities reviewed, one can conclude that computer security is only one type of security focused on protecting mainly the value of information. Depending on the context, another values could raise such as privacy or continuity.

The framework proposed does not invalidate previous work created in the field of security, but it gives an integrative framework that allows inclusion to everything done, as well as to work with new items in order to analyze security deeply.

Bibliography

[And03] James M. Anderson. Why we need a new definition of information security. *Computers & Security*, 22(4):308 – 313, 2003.

[BB66] P. G. Bock and Morton Berkowitz. The emerging field of national security. *World Politics*, 19(1):pp. 122–136, 1966.

[CLU10] France CLUSIF, Club de la Sécurité de l'Information Français. Mehari: Information risk analysis and management methodology. Norm, Club de la Sécurité de l'Information Français, 2010.

[Dec96] Rome Declaration. Rome declaration on world food security and world food summit plan of action. November 13 1996. Available online at http://www.fao.org/docrep/003/w3613e/w3613e00.htm (May 21, 2015).

[irp00] *REAC/TS Radiation Accident Registry: Update of Accidents in the United States*. IRPA, International Radiation Protection Association, 2000.

[ISO13] Switzerland ISO, Geneva. Information security management systems. Norm ISO 27001:2013, International Organization for Standardization, 2013.

[Lan81] Carl E. Landwehr. Formal models for computer security. *ACM Computing Surveys*, 13:247–278, 1981.

[Lan01] Carl E. Landwehr. Computer security. *International Journal of Information Security*, 1:3–13, 2001.

[Lev04] Nancy G. Leveson. The role of software in spacecraft accidents. *AIAA Journal of Spacecraft and Rockets*, 41:564–575, 2004.

[Min06] Ministerio de Administraciones Públicas. *Magerit-versión 2. Metodología de Análisis y Gestión de riesgos de los sistemas de información.* Ministerio de Administraciones Públicas. España., June 2006.

[Nis05] Helen Nissenbaum. Where computer security meets national security. *Ethics and Inf. Technol.*, 7:61–73, June 2005.

[Oro06] Gabriel Orozco. El concepto de la seguridad en la teoría de las relaciones internacionales. *Revista CIDOB d'Afers Internacionals*, (72):161–180, 2006.

[Rot95] Emma Rothschild. What is security? the quest for world order. *Daedalus*, 124(3):53–99, June 1995.

[Sch08] Bruce Schneier. The psychology of security. January 18 2008. Available online at http://www.schneier.com/essay-155.html (September 27, 2010).

[SEI01] United States SEI, Carnegie Mellon University. Operationally critical threat, asset, and vulnerability evaluation. Norm, Software Engineering Institute (SEI), 2001.

[SGF02] Gary Stoneburner, Alice Goguen, and Alexis Feringa. *Risk Management Guide for Information Technology Systems*. NIST National Institute of Standards and Technology, July 2002.

[UND94] UNDP. *HDR 1994 - New Dimensions of Human Security*. Human Development Report Office (HDRO), United Nations Development Programme (UNDP), 1994.

[vSvS05] Basie von Solms and Rossouw von Solms. From information security to...business security? *Computers & Security*, 24(4):271 – 273, 2005.

[WFC74] World Food Conference, Rome, 5 to 16 november 1974. Communication from the Commission to the Council. sec (74) 4955 final 9 December, 1974.

[Wol52] Arnold Wolfers. National security as an ambiguous symbol. *Political Science Quarterly*, 67(4):481–502, 1952.

CHAPTER

7

Formal Security Model

"If you think technology can solve your security problems, then you don't understand the problems and you don't understand the technology."

— Bruce Schneier

Contents

This chapter details how the framework obtained could be expressed in formal notation and, based on that, how a security object is constructed.

The framework obtained in chapter 6 is suitable to be expressed in formalized notation. The fundamentals of formal context analysis (FCA) have been introduced in section 3.2. Therefore, definitions, properties, relations and operations of the framework security model are explained.

7.1 Definitions

Definition 1 (security schema)**.** A *security schema* $\mathbb{C}$ is a 6-tuple $(\mathbb{V}, \mathbb{T}, \mathbb{P}, \mathbb{I}, \mathbb{M}, \mathbb{S})$ of concepts, where:

- $\mathbb{V} \quad is\ a\ finite\ set\ (of\ value\ names) \quad \mathbb{V} = \{v_1, v_2, ..., v_{n_1}\}$
- $\mathbb{T} \quad is\ a\ finite\ set\ (of\ threat\ names) \quad \mathbb{T} = \{t_1, t_2, ..., t_{n_2}\}$
- $\mathbb{P} \quad is\ a\ finite\ set\ (of\ provider\ names) \quad \mathbb{P} = \{p_1, p_2, ..., p_{n_3}\}$
- $\mathbb{I} \quad is\ a\ finite\ set\ (of\ policy\ names) \quad \mathbb{I} = \{i_1, i_2, ..., i_{n_4}\}$
- $\mathbb{M} \quad is\ a\ finite\ set\ (of\ measure\ names) \quad \mathbb{M} = \{m_1, m_2, ..., m_{n_5}\}$
- $\mathbb{S} \quad is\ a\ finite\ set\ (of\ resource\ names) \quad \mathbb{S} = \{s_1, s_2, ..., s_{n_6}\}$

$\mathbb{V}, \mathbb{T}, \mathbb{P}, \mathbb{I}, \mathbb{M}, \mathbb{S}$ are the concepts related to security and v, t, p, i, m, s are the elements of the concepts respectively.

A *formal context* $\mathcal{K}$ is a triple $\mathcal{K} = (O, A, I)$ where O and A are sets and I is a relation between O and A. The elements of O are called the objects and the elements of A are called the attributes of the context. Formally it can be regarded as a subset of the cartesian product (incidence relation), i.e. $I \subseteq O \times A$.

Definition 2 (relation between two concepts)**.** A Security schema $\mathbb{C}$ is made by 6 concepts and their relations. The relations raise between any two concepts and they are very similar to formal contexts. For our purposes, *the relation between two concepts* is a binary relation made using two sets $\mathbb{X}$ and $\mathbb{Y}$ of $\mathbb{C}$. $\mathbb{X}$ and $\mathbb{Y}$ are concepts and the relation involves two concepts instead of objects and attributes. Formal contexts are represented graphically by means of a table. Formally the relation is expressed using the $\times$ operator or writing only the sets. For example, the formal context $K = (A, B, I)$ could be expressed as $\mathbb{A}\mathbb{B}$ or $\mathbb{A} \times \mathbb{B}$.

7.2 Security Schema as a Graph: Security Graph

Definition 3 (security graph)**.** A *security graph* is a graph where the nodes are the sets and the edges are the relation between two nodes. The nodes represent the concepts and the edges represent the formal context between two concepts.

7.2.1 Security Tree

In graph theory, a tree is defined as an undirected graph in which any two nodes are connected by exactly one path, all nodes are connected and the tree does not have cycles. Trees are graphs that connect all vertices using the smallest possible number of edges. For a N nodes tree, the number of edges is $N-1$.

Definition 4 (security tree)**.** A *security tree* is a graph with $N=6$ that satisfies the conditions of a tree.

7.2.1.1 Number of security trees

The total number of security trees is N^{N-2}.

7.2.1.2 Minimum number of edges of a security tree

We are interested in determining the minimum number of edges necessary to create a security tree. The problem is known as Minimum Spanning Tree (MST) and could be resolved with several methods. The most known are kruskal [Kru56] and Prim [Pri57] algoritms. For a valid MST, the edge number have to be equal to the number of vertices minus one. A security tree is $N=6$ and the minimal number of edges is $N-1=5$.

7.2.2 Labeling edges, multigraph

Every edge has two nodes. Each node represents a concept, and the edge the relation between both. If concept $V=\{v_1,v_2,v_3\}$, concept $T=\{t_1,t_2,t_3,t_4\}$ and its relation $V\times T=\{(v_1,t_1),(v_1,t_4),(v_2,t_2),(v_3,t_3),(v_3,t_4)\}$, the resultant graph is a labeled multigraph (Figure 7.1).

For simplicity, it is represented as the operation of two concepts (Figure 7.2), but indeed the relation is fully represented by the whole relation. The graph representation is shown in Figure 7.3.

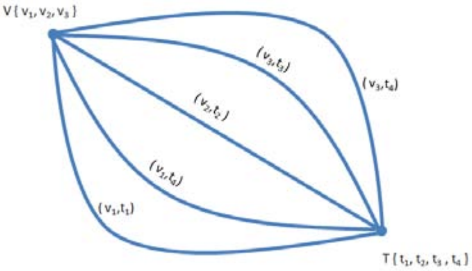


Figure 7.1 – Labeled Multigraph.

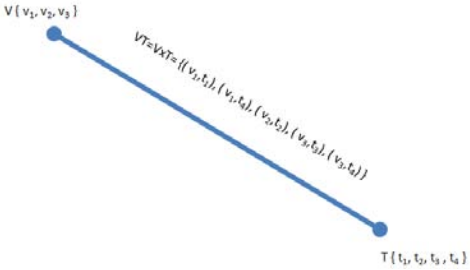
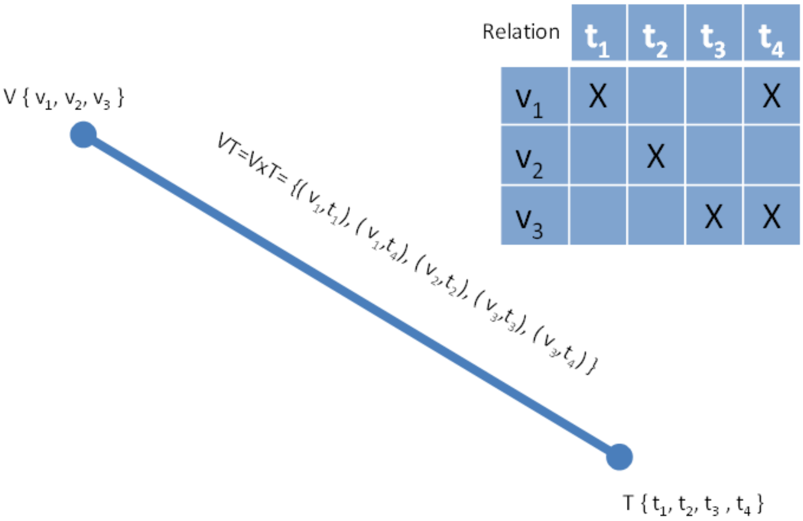


Figure 7.2 – Labeled graph.



Relation	t_1	t_2	t_3	t_4
v_1	X			X
v_2		X		
v_3			X	X

Figure 7.3 – Labeled graph and operation.

7.3 Relations

The Security Hexagon Model is based on concepts and relations underlying the notion of security. According to the definition of security graph, a complete graph with 6 vertices has a total number of edges of $\binom{6}{2} = 15$ (Figure 7.4). This number is the total number of relations. In Figure 7.5 all the relations with the name used is drawn.

7.3.1 Primary relations and elected MST

Primary relations are defined as the minimum number of binary relations among sets defined in security schema necessary to create, by means of inferring, all the possible 15 relations. This is a minimum spanning tree (MST) problem, and the solution for 6 vertices is 5 edges, that in the security graph are relations. We name at that relations *primary relations*. From those, it is possible to reach any

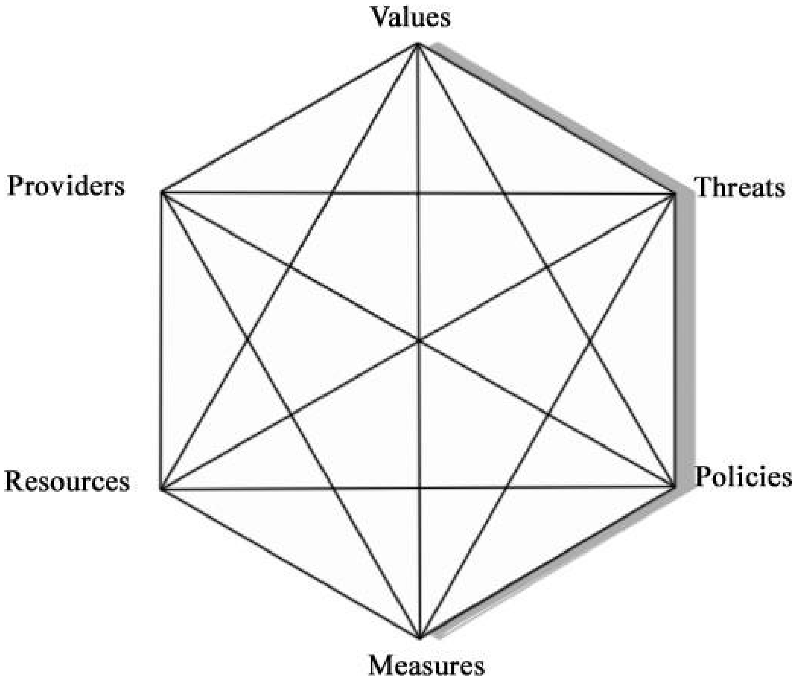


Figure 7.4 – Security hexagon with all relations.

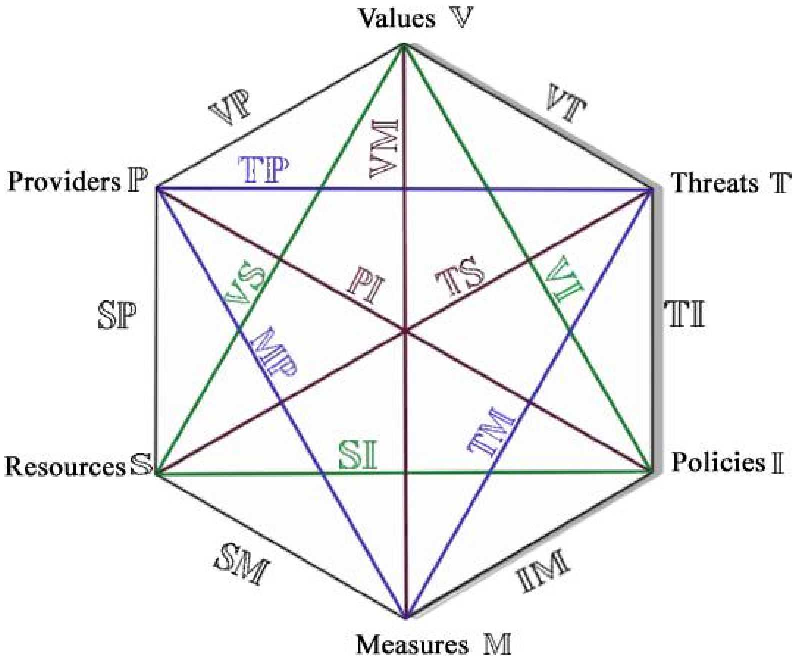


Figure 7.5 – Security hexagon with all relations named.

other. The primary relations chosen are the ones which have been obtained in the construction of the security framework in chapter 5 (Figure 5.5 and Figure 5.7). The figures describe 6 and 8 relations respectively. Only 5 relations are needed for constructing an MST and, therefore, the primary relations defined (Figure 7.6) are “threats to values”, “policies to threats”, “provider to threats”, “measures of policies” and “resources of measures” (Figure 7.9).

The relations and therefore the graph is not directed. For example, the “threats to values” $\mathbb{T} \times \mathbb{V}$ relation is the same as “values of threats” $\mathbb{V} \times \mathbb{T}$ and could be used and represented in both ways (detailed in section 7.3.2).

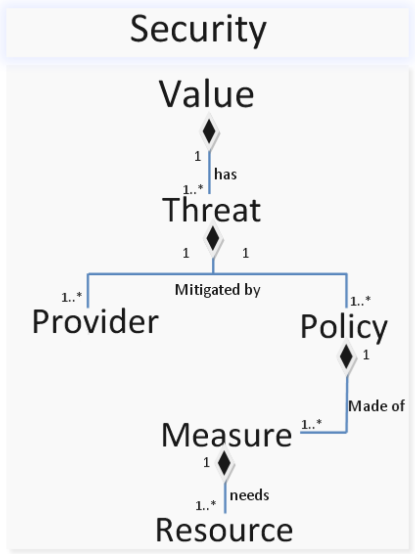


Figure 7.6 – Primary relations of the MST tree chosen.

The creation of a security schema implies to find all the relations (the complete graph and its relations). The following sections use the MST chosen to make the complete graph and relations, but it is possible to choose any other minimum spanning tree to create the security schema. For example the one drawn in Figure 7.10. Not all 5-edges election are MST trees. For example, in Figure 7.11 the set of primary relations chosen don’t allow to achieve all of the relations.

Definition 5 (Threats to Values)**.** Let $\mathbb{VT} = (\mathbb{V}, \mathbb{T}, I)$ be a binary relation. Value $v \in \mathbb{V}$ is related to threat $t \in \mathbb{T} \iff$ value v is threaten by t. It is expressed as $vIt,\ \mathbb{VT}\ \ or\ \ \mathbb{V} \times \mathbb{T}$.

Definition 6 (Policies to Threats)**.** Let $\mathbb{IT} = (\mathbb{I}, \mathbb{T}, I)$ be a binary relation. Policy $i \in \mathbb{I}$ is related to threat $t \in \mathbb{T} \iff$ policy i acts over t. It is expressed as $iIt,\ \mathbb{IT}\ \ or\ \ \mathbb{I} \times \mathbb{T}$.

Definition 7 (Provider to Threats)**.** Let $\mathbb{PT} = (\mathbb{P}, \mathbb{T}, I)$ be a binary relation. Provider $p \in \mathbb{P}$ is related to threat $t \in \mathbb{T} \iff$ provider p inhibits somehow threat t. It is expressed as $pIt,\ \mathbb{PT}\ \ or\ \ \mathbb{P} \times \mathbb{T}$.

Definition 8 (Measures of Policies)**.** Let $\mathbb{MI} = (\mathbb{M}, \mathbb{I}, I)$ be a binary relation. Measure $m \in \mathbb{M}$ is related to policy $i \in \mathbb{I} \iff$ policy i is made using measure m. It is expressed as $mIi,\ \mathbb{MI}\ \ or\ \ \mathbb{M} \times \mathbb{I}$.

Definition 9 (Resources of measures). Let $\mathbb{MS} = (\mathbb{M}, \mathbb{S}, I)$ be a binary relation. Measure $m \in \mathbb{M}$ is related to resource $s \in \mathbb{S} \iff$ resource s is used in measure m. It is expressed as $mIs,\ \mathbb{MS}\ or\ \mathbb{M} \times \mathbb{S}$.

7.3.2 Transitivity and Transposition properties in relations

- Transposition. Given a relation $K = (\mathbb{X}, \mathbb{Y}, I) \in \mathbb{C}$, the sets $\mathbb{X}$ and $\mathbb{Y}$ could be placed in rows or columns interchangeably.

  For example, a relation $R = (A, B, I)$ with $A = \{A1, A2, A3\}$ and $B = \{B1, B2, B3\}$ could express the relation I graphically with A and in rows or columns and consequently B in columns or rows respectively (Figure 7.7). As a relation $R = (A, B, I)$ is also expressed as $\mathbb{AB}$ or $\mathbb{A} \times \mathbb{B}$, the order of the sets is meaningless and hence $\mathbb{AB} = \mathbb{BA}$ or $\mathbb{A} \times \mathbb{B} = \mathbb{B} \times \mathbb{A}$.

- Transitivity. Two relations $K_1 = (A_1, B_1, I_1), K_2 = (A_2, B_2, I_2)$ could be combined to form a new relation $K_3 = (A_3, B_3, I_3)$ if one of the sets is the same in both relations:

  $$A_1 = A_2\ \ or\ \ A_1 = B_2\ \ or\ \ B_1 = A_2\ \ or\ \ B_1 = B_2$$

  The new relation is made by the no common set of the both relations (Figure 7.8).

  $$K_1 = (A, B, I_1) \quad K_2 = (B, C, I_2) \quad \longrightarrow K_3 = (A, C, I_3)$$

  $$aI_1b\ \ and\ \ bI_2c \quad \longrightarrow \quad aI_3c$$

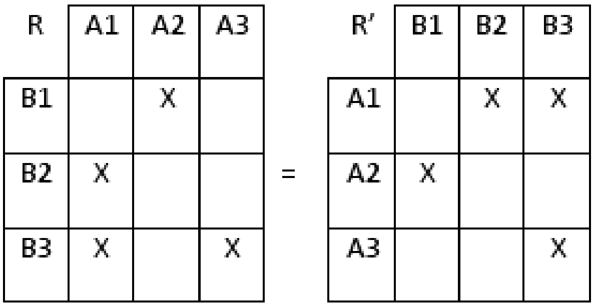
  Transitivity is expressed using the $\circ$ operator. The preceding relations and operations, hence, are written as:

  $$K_1 = (A, B, I_1) \quad \longrightarrow K_1 = A \times B\ \ or\ \ K_1 = AB$$

  $$K_2 = (B, C, I_2) \quad \longrightarrow K_2 = B \times C\ \ or\ \ K_2 = BC$$

  $$K_3 = (A, C, I_3) \quad \longrightarrow K_3 = A \times C\ \ or\ \ K_3 = AC$$

  Consequently:



R	A1	A2	A3
B1		X	
B2	X		
B3	X		X

=

R′	B1	B2	B3
A1		X	X
A2	X		
A3			X

Figure 7.7 – Transposition.

$$K_3 = K_1 \circ K_2$$

$$K_3 = (A \times B) \circ (B \times C) = A \times C$$

$$K_3 = AB \circ BC = AC$$

Transitive operation is extended to n relations $K_1, K_2, ..., K_n$. The new relation K_{n+1} has the two non shared sets and the result of applying transitivity with the relations. For example, with 3 relations:

$$K_1 = (A, B, I_1) \quad K_2 = (B, C, I_2) \;\; K_3 = (C, D, I_3) \longrightarrow K_4 = (A, D, I_4)$$

$$aI_1b \;\; and \;\; bI_2c \;\; and \;\; cI_3d \quad \longrightarrow \quad aI_4d$$

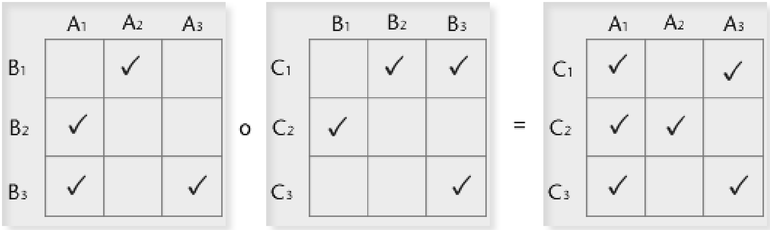
The new relation K_4 is expressed as:

$$K_4 = K_1 \circ K_2 \circ K_3$$

$$K_4 = (A \times B) \circ (B \times C) \circ (C \times D) \longrightarrow K_4 = A \times D$$

7.3.3 Deduced relations

Deduced relations are the remaining edges of the 6-vertex complete graph that does not belong to the chosen MST. As the minimal vertex number for the security graph is 5 and the total number or relations is $\binom{6}{2} = 15$, then, the remaining 10 relations must be deduced. The number of possible MST is N^{N-2} and therefore



$$
\begin{array}{c|c|c|c|}
 & A_1 & A_2 & A_3 \\ \hline
B_1 & & \checkmark & \\ \hline
B_2 & \checkmark & & \\ \hline
B_3 & \checkmark & & \checkmark \\ \hline
\end{array}
\circ
\begin{array}{c|c|c|c|}
 & B_1 & B_2 & B_3 \\ \hline
C_1 & & \checkmark & \checkmark \\ \hline
C_2 & \checkmark & & \\ \hline
C_3 & & & \checkmark \\ \hline
\end{array}
=
\begin{array}{c|c|c|c|}
 & A_1 & A_2 & A_3 \\ \hline
C_1 & \checkmark & & \checkmark \\ \hline
C_2 & \checkmark & \checkmark & \\ \hline
C_3 & \checkmark & & \checkmark \\ \hline
\end{array}
$$

Figure 7.8 – Transitive.

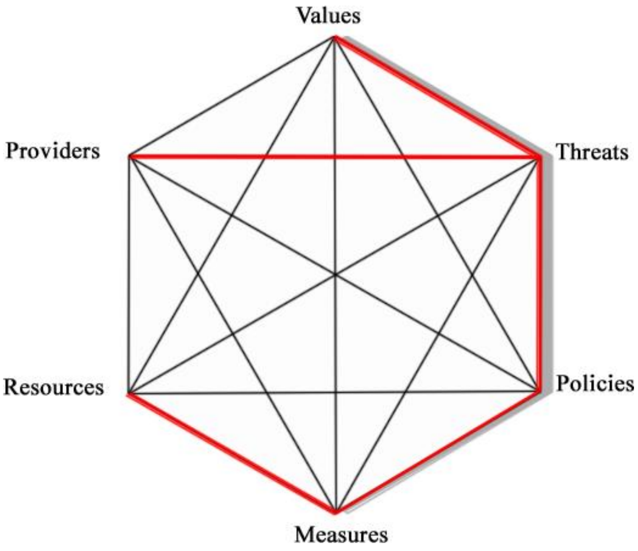


Figure 7.9 – Primary and deduced relations.

the deduced relations depends on the MST (primary relations) used. Those 10 relations are deduced from the primary ones.

In order to obtain the remaining relations from the primary relations, transitive and transposition properties in all the sets of security schema $\mathbb{C}$ are used. Primary relations are shown in Table 7.1 and the deduced ones in Table 7.2. Expressed graphically (Figure 7.9), we can realize that from the chosen MST, the primary relations (edges) allow to reach the remaining.

Relation	Description
$\mathbb{VT}$	Values are threaten by a set of threats
$\mathbb{TI}$	Threats are mitigated by a set of policies
$\mathbb{IM}$	Policies are implemented by a set of measures
$\mathbb{MS}$	Measures need a set of resources
$\mathbb{PT}$	Providers "provide" security to a number of threats

Table 7.1 – Primary relations.

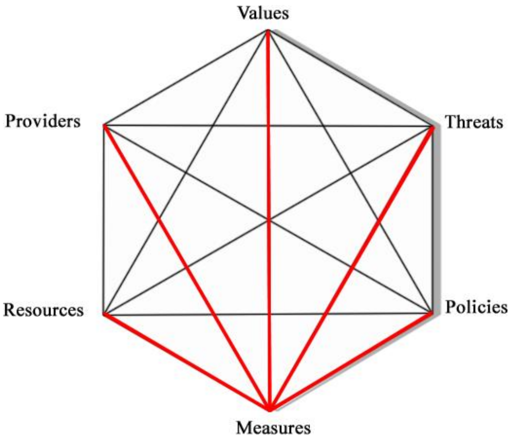


Figure 7.10 – Set of primary relations.

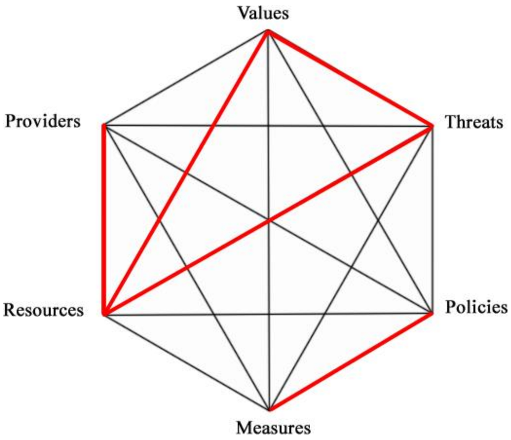


Figure 7.11 – Another set of primary relations.

Relation	Deduced by
$\mathbb{VI} =$	$VT \circ TI$
$\mathbb{PI} =$	$PT \circ TI$
$\mathbb{VM} =$	$VI \circ IM = VT \circ TI \circ IM$
$\mathbb{VP} =$	$VT \circ TP$
$\mathbb{TM} =$	$TI \circ IM$
$\mathbb{SI} =$	$SM \circ MI$
$\mathbb{TS} =$	$TI \circ IS = TI \circ IM \circ MS$
$\mathbb{MP} =$	$MT \circ TP = TI \circ IM \circ TP$
$\mathbb{SP} =$	$ST \circ TP = TI \circ IS \circ TP = ST \circ TP = TI \circ IM \circ MS \circ TP$
$\mathbb{VS} =$	$VT \circ TS = VT \circ TI \circ IS = VT \circ TI = VT \circ TI \circ IM \circ MS$

Table 7.2 – Deduced relations.

Definition 10 (Policies of Values)**.** Let $\mathbb{VI} = (\mathbb{V}, \mathbb{I}, I)$ be a binary relation. Value $v \in \mathbb{V}$ is related to policy $i \in \mathbb{I} \iff$ policy i is used in protecting value v. It is expressed as $vIi,\ \mathbb{VI}\ or\ \mathbb{V} \times \mathbb{I}$.

Definition 11 (Providers of policies)**.** Let $\mathbb{PI} = (\mathbb{P}, \mathbb{I}, I)$ be a binary relation. Provider $p \in \mathbb{P}$ is related to policy $i \in \mathbb{I} \iff$ provider p acts in policy i. It is expressed as $pIi,\ \mathbb{PI}\ or\ \mathbb{P} \times \mathbb{I}$.

Definition 12 (Measures of values)**.** Let $\mathbb{MV} = (\mathbb{M}, \mathbb{V}, I)$ be a binary relation. Measure $m \in \mathbb{M}$ is related to value $v \in \mathbb{V} \iff$ measure m is needed to protect value v. It is expressed as $mIv,\ \mathbb{MV}\ or\ \mathbb{M} \times \mathbb{V}$.

Definition 13 (Values of providers)**.** Let $\mathbb{VP} = (\mathbb{V}, \mathbb{P}, I)$ be a binary relation. Value $v \in \mathbb{V}$ is related to provider $p \in \mathbb{P} \iff$ provider p protects value v. It is expressed as $vIp, \;\mathbb{VP} \;\; or \;\; \mathbb{V} \times \mathbb{P}$.

Definition 14 (Threats of measures)**.** Let $\mathbb{TM} = (\mathbb{T}, \mathbb{M}, I)$ be a binary relation. Threat $t \in \mathbb{T}$ is related to measure $m \in \mathbb{M} \iff$ threat t is mitigated by means of measure m. It is expressed as $tIm, \;\mathbb{TM} \;\; or \;\; \mathbb{T} \times \mathbb{M}$.

Definition 15 (Resources of policies)**.** Let $\mathbb{SI} = (\mathbb{S}, \mathbb{I}, I)$ be a binary relation. Resource $s \in \mathbb{S}$ is related to policy $p \in \mathbb{P} \iff$ resource s is used in policy i. It is expressed as $sIi, \;\mathbb{SI} \;\; or \;\; \mathbb{S} \times \mathbb{I}$.

Definition 16 (Resources of threats)**.** Let $\mathbb{ST} = (\mathbb{S}, \mathbb{T}, I)$ be a binary relation. Resource $s \in \mathbb{S}$ is related to threat $t \in \mathbb{T} \iff$ resource s is used to mitigate threat t. It is expressed as $sIt, \;\mathbb{ST} \;\; or \;\; \mathbb{S} \times \mathbb{T}$.

Definition 17 (Measures of providers)**.** Let $\mathbb{MP} = (\mathbb{M}, \mathbb{P}, I)$ be a binary relation. Measure $m \in \mathbb{M}$ is related to provider $p \in \mathbb{P} \iff$ measure m is used by provider p. It is expressed as $mIp, \;\mathbb{MP} \;\; or \;\; \mathbb{M} \times \mathbb{P}$.

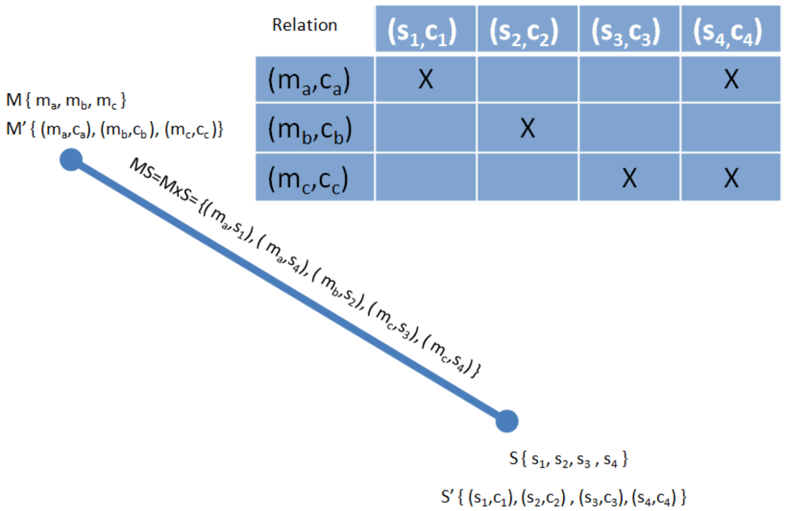
Definition 18 (Resources of providers)**.** Let $\mathbb{SP} = (\mathbb{S}, \mathbb{P}, I)$ be a binary relation. Resource $s \in \mathbb{S}$ is related to provider $p \in \mathbb{P} \iff$ resource s is needed by provider p. It is expressed as $sIp, \;\mathbb{SP} \;\; or \;\; \mathbb{S} \times \mathbb{P}$.

Definition 19 (Values of resources)**.** Let $\mathbb{VS} = (\mathbb{V}, \mathbb{S}, I)$ be a binary relation. Value $v \in \mathbb{V}$ is related to resource $s \in \mathbb{S} \iff$ value v is protected with resource s. It is expressed as $vIs, \;\mathbb{VS} \;\; or \;\; \mathbb{V} \times \mathbb{S}$.

7.4 Knowledge extraction: Measuring cost

The deduced security model includes two measures in every concept; cost and degree (Figure 5.4). To include cost in the security hexagon model, a real number is associated to each concept of $\mathbb{C}$. As the primary relations are hierarchically ordered (Figure 7.6), the cost for all concepts and elements are deduced by specifying only the cost of every resource $s \in \mathbb{S}$ (Figure 7.12).

Definition 20 (Cost)**.** Cost expresses in a quantitative form the “effort” related to a concept or element. Indeed, this number could express the financial cost, effort, hours or whatever. The point is to be consistent in what expresses that value. The cost of an element is the total amount of the costs of other elements in concepts that are related. For example, the cost of a measure is obtained by


M { m_a, m_b, m_c }
M′ { (m_a,c_a), (m_b,c_b), (m_c,c_c)}
MS=MxS= {(m_a,s_1), (m_a,s_4), (m_b,s_2), (m_c,s_3), (m_c,s_4)}
S { s_1, s_2, s_3 , s_4 }
S′ { (s_1,c_1), (s_2,c_2) , (s_3,c_3), (s_4,c_4) }

Relation	(s1,c1)	(s2,c2)	(s3,c3)	(s4,c4)
(ma,ca)	X			X
(mb,cb)		X		
(mc,cc)			X	X

Figure 7.12 – Cost associated to resource concept.

means of the $\mathbb{MS}$ relation (Figure 7.12). The same occurs with the remaining concepts. Hence, the *security schema* $\mathbb{C}$ is extended to $\mathbb{C}'$. Therefore, $\mathbb{C}'$ is a 6-tuple $(\mathbb{V}', \mathbb{T}', \mathbb{P}', \mathbb{I}', \mathbb{M}', \mathbb{S}')$ and the sets are defined in the following paragraphs.

- **Resource cost.** The cost of a resource is a number associated to a resource. Given $\mathbb{S}$ a finite set (of resource names) $\mathbb{S} = \{s_1, s_2, ..., s_{n_6}\}$, then $\mathbb{S}'$ is defined as: $\mathbb{S}' = \{(s_1, c_1), (s_2, c_2), ..., (s_{n_6}, c_{n_6})\}$ with $(s_i, c_i) \in \mathbb{S}' \mid s_i \in \mathbb{S}$, $c_i \in \mathbb{R}$, $i = 1, .., n_6$

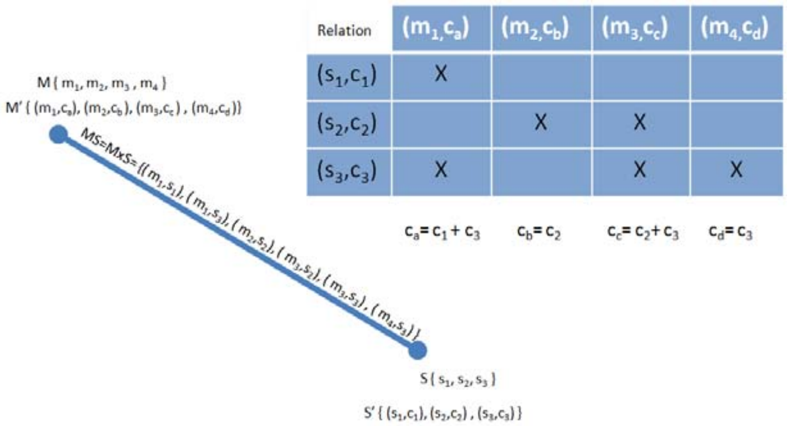
- **Measure cost.** It is the cost of a measure deduced by means of $M \times S$ relation. The cost of a measure is based on the used resources. Given $\mathbb{M}$ a finite set (of measure names) $\mathbb{M} = \{m_1, m_2, ..., m_{n_5}\}$, then $\mathbb{M}'$ is defined as: $\mathbb{M}' = \{(m_1, cm_1), (m_2, cm_2), ..., (m_{n_5}, cm_{n_5})\}$ with $(m_i, cm_i) \in \mathbb{M}' \mid m_i \in \mathbb{M}$, $cm_i \in \mathbb{R}$, $i = 1, .., n_5$

  The cost of a measure m is the total amount of costs of the resources which have a relation with a measure m_i (Figure 7.13). Hence, the cost cm_i associated to measure m_i is deduced by:

$$cm_i = \sum_{j=1}^{n_6} c_j \mid m_i I s_j \quad (s_j, c_j) \in \mathbb{S}' \tag{7.1}$$

For example, given $\mathbb{S}'$, $\mathbb{M}'$ and $M \times S$ relation :

$$\mathbb{S}' = \{(s_1, c_1), (s_2, c_2), (s_3, c_3)\}$$


M { m1, m2, m3 , m4 }
M' { (m1,ca), (m2,cb), (m3,cc) , (m4,cd)}
MS=MxS={(m1,s1), (m1,s3), (m2,s2), (m3,s2), (m3,s3), (m4,s3)}
S { s1, s2, s3 }
S' { (s1,c1), (s2,c2) , (s3,c3) }
ca= c1 + c3
cb= c2
cc= c2+ c3
cd= c3

Relation	(m_1,c_a)	(m_2,c_b)	(m_3,c_c)	(m_4,c_d)
(s_1,c_1)	X			
(s_2,c_2)		X	X	
(s_3,c_3)	X		X	X

Figure 7.13 – Cost of measures.

$$\mathbb{M}' = \{(m_1, c_a), (m_2, c_b), (m_3, c_c), (m_4, c_d)\ \}$$

	m_1	m_2	m_3	m_4
s_1	✓			
s_2		✓	✓	
s_3	✓		✓	✓

Table 7.3 – $M \times S$ relation.

$M \times S'$ and $(M \times S)'$ are graphically expressed in Table 7.4 and Table 7.5. The cost of the measures is:

$$c_a = c_1 + c_3$$

$$c_b = c_2$$

$$c_c = c_2 + c_3$$

$$c_d = c_3$$

- **Policy cost.** It is the cost of a policy deduced by means of $I \times S$ relation. The cost of a policy is based on the used resources. Given $\mathbb{I}$ a finite set (of policy names) $\mathbb{I} = \{i_1, i_2, ..., i_{n_4}\ \}$, then $\mathbb{I}'$ is defined as: $\mathbb{I}' = \{(i_1, ci_1), (i_2, ci_2), ..., (i_{n_4}, ci_{n_4})\ \}$ with $(i_i, ci_i) \in \mathbb{I}' \mid i_i \in \mathbb{I}\ \ ,\ \ ci_i \in \mathbb{R}\ ,\ \ i = 1, .., n_4$

  The cost of a policy i is the total amount of costs of the resources which

		m_1	m_2	m_3	m_4
s_1	c_1	✓			
s_2	c_2		✓	✓	
s_3	c_3	✓		✓	✓

Table 7.4 – $M \times S'$ relation.

		m_1	m_2	m_3	m_4
s_1	c_1	✓			
s_2	c_2		✓	✓	
s_3	c_3	✓		✓	✓
		c_a	c_b	c_c	c_d

Table 7.5 – $(M \times S)'$ relation.

have a relation with a policy i_i. Hence, the cost ci_i associated to policy i_i is deduced by:

$$ci_i = \sum_{j=1}^{n_6} c_j \mid i_i I s_j \quad (s_j, c_j) \in \mathbb{S}' \tag{7.2}$$

The cost of a policy could also be deduced by means of the $I \times M$ relation, because $I \times S = I \times M \circ M \times S$.

- **Provider cost**. It is the cost of a provider deduced by means of $P \times S$ relation. The cost of a provider is based on the used resources. Given $\mathbb{P}$ a finite set (of provider names) $\mathbb{P} = \{p_1, p_2, ..., p_{n_3}\}$, then $\mathbb{P}'$ is defined as: $\mathbb{P}' = \{(p_1, cp_1), (p_2, cp_2), ..., (p_{n_3}, cp_{n_3})\}$ with $(p_i, cp_i) \in \mathbb{P}' \mid p_i \in \mathbb{P}$, $cp_i \in \mathbb{R}$, $i = 1, .., n_3$

  The cost of a provider p is the total amount of costs of the resources which have a relation with a provider p_i. Hence, the cost cp_i associated to provider p_i is deduced by:

$$cp_i = \sum_{j=1}^{n_6} c_j \mid p_i I s_j \quad (s_j, c_j) \in \mathbb{S}' \tag{7.3}$$

  The cost of a provider could also be deduced by means of the $P \times M$ relation, because $P \times S = P \times M \circ M \times S$

- **Threat cost**. It is the cost of a threat deduced by means of $T \times S$ relation. The cost of a threat is based on the used resources. Given $\mathbb{T}$ a finite set (of threat names) $\mathbb{T} = \{t_1, t_2, ..., t_{n_2}\}$, then $\mathbb{T}'$ is defined as: $\mathbb{T}' = \{(t_1, ct_1), (t_2, ct_2), ..., (t_{n_2}, ct_{n_2})\}$ with $(t_i, ct_i) \in \mathbb{T}' \mid t_i \in \mathbb{T}$, $ct_i \in \mathbb{R}$, $i = 1, .., n_2$

  The cost of a threat t is the total amount of costs of the resources which have a relation with a threat t_i. Hence, the cost ct_i associated to threat t_i is deduced by:

$$ct_i = \sum_{j=1}^{n_6} c_j \mid t_i I s_j \quad (s_j, c_j) \in \mathbb{S}' \tag{7.4}$$

The cost of a threat could also be deduced by means of the $T \times I$ relation, because $T \times S = T \times I \circ I \times M \circ M \times S$.

- **Value cost**. It is the cost of a value deduced by means of $V \times S$ relation. The cost of a value is based on the used resources. Given $\mathbb{V}$ a finite set (of value names) $\mathbb{V} = \{v_1, v_2, ..., v_{n_1}\}$, then $\mathbb{V}'$ is defined as: $\mathbb{V}' = \{(v_1, cv_1), (v_2, cv_2), ..., (v_{n_1}, cv_{n_1})\}$ with $(v_i, cv_i) \in \mathbb{V}' \mid v_i \in \mathbb{V}$, $cv_i \in \mathbb{R}$, $i = 1, .., n_1$

  The cost of a value v is the total amount of costs of the resources which have a relation with a value v_i. Hence, the cost cv_i associated to value v_i is deduced by:

$$cv_i = \sum_{j=1}^{n_6} c_j \mid v_i I s_j \quad (s_j, c_j) \in \mathbb{S}' \tag{7.5}$$

  The cost of a value could also be deduced by means of the $V \times T$ relation, because $V \times S = V \times T \circ T \times I \circ I \times M \circ M \times S$.

- **Security cost**. It is deduced as the total of every resource cost. Given

$$\mathbb{S}' = \{(s_1, c_1), (s_2, c_2), ..., (s_{n_6}, c_{n_6})\}$$

  The cost of security CS is deduced by:

$$CS = \sum_{j=1}^{n_6} c_j \mid (s_j, c_j) \in \mathbb{S}' \tag{7.6}$$

7.5 Knowledge extraction: Measuring security

The security model deduced includes two measures in every concept; cost and degree (Figure 5.4). To include the degree of security or level in the security hexagon model, three time points are associated to each concept of $\mathbb{C}$. As the primary relations are hierarchically ordered (Figure 7.6), the security degree for all concepts and elements are deduced by specifying only the time points of every measure $m \in \mathbb{M}$. Hence, the *security schema* $\mathbb{C}$ is extended to $\mathbb{C}''$. Therefore, $\mathbb{C}''$ is a 6-tuple $(\mathbb{V}'', \mathbb{T}'', \mathbb{P}'', \mathbb{I}'', \mathbb{M}'', \mathbb{S}'')$ and the sets are defined in the following paragraphs.

7.5.1 Security level

As stated by Wolfers [Wol52], security is a matter of degree. Thereby, security could raise or fall for a number of reasons. Without external inputs, security decreases as time increases. Hence, we define the security level as the level of security of a concept in a time instant. The value depends on how the security function is modeled, but the time points of the concepts, and thus their security functions could be deduced.

According to the following definitions, the security function is the outcome of modeling the behavior of security in time. This section only introduces the definitions in order to describe formally the level of security that could be obtained. Our implementation of the security function is developed in the applied part of the research (chapter 8 - Security Level Time Function).

Definition 21 (Time instant)**.** One discrete point on time axis. Granularity of a time instant depends on its use. Could be a date, could be minutes or even milliseconds if necessary. It depends on the model. A time instant is expressed as t_i and it is a real number $t_i \in \mathbb{R}$

Definition 22 (Time interval)**.** Time interval is a set of time instants. Indeed, a time interval is the set of discrete time points between two time instants. A time interval is expressed as Ti and it is a pair of time time instants (t_0, t_1) and denoted as $T_i \subset \mathbb{R}^2$.

Definition 23 (Time range)**.** As security is a time dependent function, there is a need to quantify some relevant intervals of time. The first one when security is good. The second one when security is right. The third when security is really risky and finally when there is no longer security. Graphically time intervals are shown in Figure 8.10. Hence, there are four time intervals T_0, T_1, T_2, T_3 and thereby three time instants t_0, t_1, t_2 are needed to represent the intervals. A time range is a vector of three time instants (t_0, t_1, t_2) and denoted as $T^r \subset \mathbb{R}^3$. A time range $T^r(t_0, t_1, t_2)$ is composed by four time intervals as shown in Table 7.6.

$T_0 = (-\infty, t_0]$	Security is good
$T_1 = (t_0, t_1]$	Security is right
$T_2 = (t_1, t_2]$	Security is risky
$T_3 = (t_2, \infty)$	There is no security

Table 7.6 – Time intervals.

Definition 24 (Security level function)**.** Security is a time function associated to every element or set of the security schema $\mathbb{C}$. The function depends on:

- Any element v, t, p, i, m, s of the concepts $\mathbb{V}, \mathbb{T}, \mathbb{P}, \mathbb{I}, \mathbb{M}, \mathbb{S}$, denoted by α.
- A time range T^r which represents four time intervals T_0, T_1, T_2, T_3 expressed by three time instants t_0, t_1, t_2.
- A time instant t.

Thereby, security level $S = f(t, \alpha, T^r)$. As T^r is associated to the element α, indeed, security level depends on the time and the element α considered. The value could be expressed as a percentage or a number between $[0, 1]$.

$$S = f(\alpha, t) \longrightarrow [0, 1]$$

Definition 25 (Threshold)**.** *Threshold* is defined as the security level that changes a concept or element from secure to insecure state (Figure 8.9). Threshold is denoted as λ and $\lambda \in [0, 1]$

From the definitions explained, we can deduce several security levels. The security level of a policy in a given time instant is dependent on the security levels of the measures that constitutes the policy; the $\mathbb{IM}$ relation. The same occurs with the remaining concepts. Concepts and relations are hierarchically ordered (Figure 5.5), hence providing the security level of the measures the remaining security level of all elements of $\mathbb{C}$ could be deduced using the security hexagon relations. In order to deduce security level, a time range T^r is associated to every concept of the model.

- **Security level of measures**. The security level of a measure is a security level function linked to a measure m and a time range T^r.

$$\mathbb{M} \;\; a\ finite\ set\ (of\ measure\ names) \quad \mathbb{M} = \{m_1, m_2, ..., m_{n_5}\}$$

$$\mathbb{M}''\ \ is\ \ defined\ \ as:\ \ \mathbb{M}'' = \left\{(m_1, T_1^r), (m_2, T_2^r), ..., (m_{n_5}, T_{n_5}^r)\right\}$$

$$with \quad m'' = (m, T^r) \in \mathbb{M}'' \mid m \in \mathbb{M} \ \ , \ \ T^r \in \mathbb{R}^3$$

The security function $S_m(t)$ of a measure m is:

$$S_m(t) = f(m'', t) \longrightarrow [0, 1]$$

- **Security level of Policies.** It is a security level function linked to a policy i and a time range T^r.

$$\mathbb{I} \quad is\ a\ finite\ set\ (of\ policy\ names) \quad \mathbb{I} = \{i_1, i_2, ..., i_{n_4}\}$$

$$\mathbb{I}''\ \ is\ \ defined\ \ as:\ \ \mathbb{I}'' = \left\{(i_1, T_1^r), (i_2, T_2^r), ..., (i_{n_4}, T_{n_4}^r)\right\}$$

$$with \quad i'' = (i, T^r) \in \mathbb{I}'' \mid i \in \mathbb{I} \ \ , \ \ T^r \in \mathbb{R}^3$$

The security function $S_i(t)$ of a policy i is:

$$S_i(t) = f(i'', t) \longrightarrow [0, 1]$$

Time range T^r could be deduced for every policy by means of $I \times M$ relation. For a policy α the time points $t_{\alpha_0}, t_{\alpha_1}, t_{\alpha_2}$ are deduced as:

$$t_{\alpha_0} = min\,\{t_{i_0} \mid iIm\ \ \forall i\} \tag{7.7}$$

$$t_{\alpha_1} = \{t \in [0, 1] \mid S_i(t) =\ \lambda\ \ and\ \ t_{\alpha_0} \leq t \leq t_{\alpha_2}\} \tag{7.8}$$

$$t_{\alpha_2} = max\,\{t_{i_2} \mid iIm\ \ \forall i\} \tag{7.9}$$

For example, given

$$\mathbb{I}'' = \{(i_1, T_1^r), (i_2, T_2^r), (i_3, T_3^r)\ \}$$

$$\mathbb{M}'' = \{(m_1, T_a^r), (m_2, T_b^r), (m_3, T_c^r), (m_4, T_d^r)\ \}$$

$$and\ \ I \times M\ \ relation\ \ (Table\ 7.7):$$

$I'' \times M$ and $(I \times M)''$ are graphically expressed in Table 7.8 and Table 7.9

	m_1	m_2	m_3	m_4
i_1	✓			
i_2		✓	✓	
i_3	✓		✓	✓

Table 7.7 – $I \times M$ relation.

		m_1	m_2	m_3	m_4
i_1	T_1^r	✓			
i_2	T_2^r		✓	✓	
i_3	T_3^r	✓		✓	✓

Table 7.8 – $I'' \times M$ relation.

		m_1	m_2	m_3	m_4
i_1	T_1^r	✓			
i_2	T_2^r		✓	✓	
i_3	T_3^r	✓		✓	✓
		T_a^r	T_b^r	T_c^r	T_d^r

Table 7.9 – $(I \times M)''$ relation.

respectively. In Table 7.10 the time range elements are drawn. Thereby, the time range of the policies are deduced:

$$with \;\; T_a^r \;\; (t_{a_0}, t_{a_1}, t_{a_2}) \;\; then$$

$$t_{a_0} = min\,\{t_{1_0}, t_{3_0}\}$$

$$t_{a_1} = \{t \in [0,1] \mid S(t) \;\; = \;\; \lambda \;\; and \;\; t_{a_0} \leq t \leq t_{a_2}\}$$

$$t_{a_2} = max\,\{t_{1_2}, t_{3_2}\}$$

And the same for T_b^r, T_c^r, T_d^r

		m_1	m_2	m_3	m_4
i_1	$(t_{1_0}, t_{1_1}, t_{1_2})$	✓			
i_2	$(t_{2_0}, t_{2_1}, t_{2_2})$		✓	✓	
i_3	$(t_{3_0}, t_{3_1}, t_{3_2})$	✓		✓	✓
		(t_{a0}, t_{a1}, t_{a2})	(t_{b0}, t_{b1}, t_{b2})	(t_{c0}, t_{c1}, t_{c2})	(t_{d0}, t_{d1}, t_{d2})

Table 7.10 – $(I \times M)''$ relation.

- **Security level of Providers**. It is a security level function linked to a provider p and a time range T^r.

$$\mathbb{P} \quad is \; a \; finite \; set \; (of \; provider \; names) \quad \mathbb{P} = \{p_1, p_2, ..., p_{n3}\}$$

$$\mathbb{P}'' \ \ is \ \ defined \ \ as: \ \ \mathbb{P}'' = \{(p_1, T^r_1), (p_2, T^r_2), ..., (p_{n3}, T^r_{n3}) \ \}$$

$$with \quad p'' = (p, T^r) \in \mathbb{P}'' \mid p \in \mathbb{P} \ \ , \ \ T^r \in \mathbb{R}^3$$

The security function $S_p(t)$ of a provider p is:

$$S_p(t) = f(p'', t) \longrightarrow [0, 1]$$

Time range T^r could be deduced for every provider by means of $P \times M$ relation. For a provider α the time points $t_{\alpha_0}, t_{\alpha_1}, t_{\alpha_2}$ are deduced as:

$$t_{\alpha_0} = min\{t_{p_0} \mid pIm \ \ \forall p\} \tag{7.10}$$

$$t_{\alpha_1} = \{t \in [0, 1] \mid S_p(t) = \ \lambda \ \ and \ \ t_{\alpha_0} \leq t \leq t_{\alpha_2}\} \tag{7.11}$$

$$t_{\alpha_2} = max\{t_{p_2} \mid pIm \ \ \forall p\} \tag{7.12}$$

- **Security level of Threats**. It is a security level function linked to a threat h and a time range T^r.

$$\mathbb{T} \quad is \ a \ finite \ set \ (of \ threat \ names) \quad \mathbb{T} = \{h_1, h_2, ..., h_{n_2}\}$$

$$\mathbb{T}'' \ \ is \ \ defined \ \ as: \ \ \mathbb{T}'' = \{(h_1, T^r_1), (h_2, T^r_2), ..., (h_{n_2}, T^r_{n_2})\}$$

$$with \quad h'' = (h, T^r) \in \mathbb{T}'' \mid h \in \mathbb{T} \ \ , \ \ T^r \in \mathbb{R}^3$$

The security function $S_h(t)$ of a threat h is:

$$S_h(t) = f(h'', t) \longrightarrow [0, 1]$$

Time range T^r could be deduced for every threat by means of $T \times M$ relation. For a threat α the time points $t_{\alpha_0}, t_{\alpha_1}, t_{\alpha_2}$ are deduced as:

$$t_{\alpha_0} = min\{t_{h_0} \mid hIm \ \ \forall h\} \tag{7.13}$$

$$t_{\alpha_1} = \{t \in [0,1] \mid S_h(t) = \ \lambda \ \ and \ \ t_{\alpha_0} \leq t \leq t_{\alpha_2}\} \tag{7.14}$$

$$t_{\alpha_2} = max\,\{t_{h_2} \mid hIm \ \ \forall h\} \tag{7.15}$$

- **Security level of Values.** It is a security level function linked to a value v and a time range T^r.

$$\mathbb{V} \quad is\ a\ finite\ set\ (of\ value\ names) \quad \mathbb{V} = \{v_1, v_2, ..., v_{n_1}\}$$
$$\mathbb{V}'' \ \ is \ \ defined \ \ as: \ \ \mathbb{V}'' = \big\{(v_1, T_1^r), (v_2, T_2^r), ..., (v_{n_1}, T_{n_1}^r)\big\}$$
$$with \quad v'' = (v, T^r) \in \mathbb{V}'' \mid v \in \mathbb{V} \ \ , \ \ T^r \in \mathbb{R}^3$$

The security function $S_v(t)$ of a value v is:

$$S_v(t) = f(v'', t) \longrightarrow [0,1]$$

Time range T^r could be deduced for every value by means of $V \times M$ relation. For a value α the time points $t_{\alpha_0}, t_{\alpha_1}, t_{\alpha_2}$ are deduced as:

$$t_{\alpha_0} = min\,\{t_{v_0} \mid vIm \ \ \forall v\} \tag{7.16}$$

$$t_{\alpha_1} = \{t \in [0,1] \mid S_v(t) = \ \lambda \ \ and \ \ t_{\alpha_0} \leq t \leq t_{\alpha_2}\} \tag{7.17}$$

$$t_{\alpha_2} = max\,\{t_{v_2} \mid vIm \ \ \forall v\} \tag{7.18}$$

- **Security level.** It is a security level function linked to the security schema $\mathbb{C}$ with a time range T^r. It is deduced by means of $\mathbb{V}$ set.

$$\mathbb{V} \quad is\ a\ finite\ set\ (of\ value\ names) \quad \mathbb{V} = \{v_1, v_2, ..., v_{n_1}\}$$
$$\mathbb{C}'' \ \ is \ \ defined \ \ as: \ \ \mathbb{C}'' = \{\mathbb{C}, T^r\} \ \ with \ \ T^r \in \mathbb{R}^3$$

The security function $S(t)$ of $\mathbb{C}$ is:

$$S(t) = f(t) \longrightarrow [0,1]$$

Time range T^r could be deduced for security schema $\mathbb{C}$ by means of V''. Time points $t_{\alpha_0}, t_{\alpha_1}, t_{\alpha_2}$ are deduced as:

$$t_{\alpha_0} = min\,\{t_{v_0} \;\; \forall v\} \tag{7.19}$$

$$t_{\alpha_1} = \{t \in [0,1] \mid S(t) = \;\lambda \;\; and \;\; t_{\alpha_0} \leq t \leq t_{\alpha_2}\} \tag{7.20}$$

$$t_{\alpha_2} = max\,\{t_{v_2} \;\; \forall v\} \tag{7.21}$$

7.5.2 Protection

Protection is referred as the elements used in securing concepts and consequently to values. Hence, protection is a subset of the sets in security schema $\mathbb{C}$. The subset obtained is denoted by Γ. Γ is related to what mitigate and which elements are used to mitigate. For example, threat protection is related to measures. The notation is a subscript for the elements used.

- **Threat protection**. The set of measures or resources used in order to mitigate that a threat becomes real. The outcome is a set of measures or resources. Subscript M denotes the measures concept and subscript S denotes the resources concept. Therefore $\Gamma_M(t) \subseteq \mathbb{M}$ and $\Gamma_S(t) \subseteq \mathbb{S}$.

$$\Gamma_M(t) = \{m \in \mathbb{M} \mid mIt \;\; \forall m\} \quad t \in \mathbb{T} \tag{7.22}$$

$$\Gamma_S(t) = \{s \in \mathbb{S} \mid sIt \;\; \forall s\} \quad t \in \mathbb{T} \tag{7.23}$$

- **Value protection**. The set of measures or resources used in order to mitigate a value could be in danger. The outcome is a set of measures or resources. Subscript M denotes the measures concept and subscript S denotes the resources concept. Therefore $\Gamma_M(v) \subseteq \mathbb{M}$ and $\Gamma_S(v) \subseteq \mathbb{S}$.

$$\Gamma_M(v) = \{m \in \mathbb{M} \mid mIv \;\; \forall m\} \quad v \in \mathbb{V} \tag{7.24}$$

$$\Gamma_S(v) = \{s \in \mathbb{S} \mid sIv \;\; \forall s\} \quad v \in \mathbb{V} \qquad (7.25)$$

7.5.3 Risk

In our framework, risk is defined as the set of threats linked to value. Risk is denoted by Φ and therefore $\Phi \subseteq \mathbb{T}$.

$$\Phi(v) = \{t \in \mathbb{T} \mid tIv \;\; \forall t\} \quad v \in \mathbb{V} \qquad (7.26)$$

7.6 Procedure / methodology

In order to create a security object from the framework model, several steps are necessary (Table 7.11).

Main steps involve identifying the context and level; the elements of every concept set and create the primary relations. After that, infer the remaining relations. Next, provide cost, time instants to elements and define the threshold values. Finally infer the remaining knowledge (Figure 7.14).

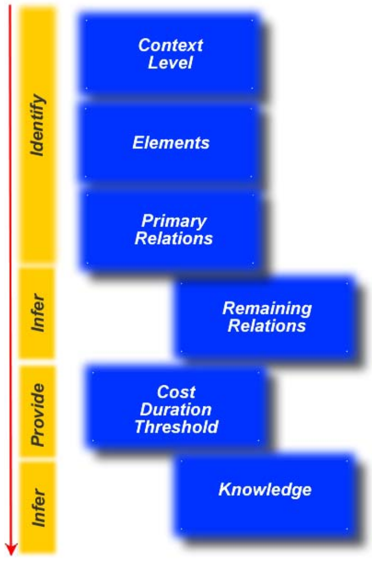


Figure 7.14 – Main steps.

Step	Define	Populate	Infer	
Concepts				
1	Context			

2	Values			$\mathbb{V} = \{v_1, v_2, ..., v_{n_1}\}$
3	Threats			$\mathbb{T} = \{t_1, t_2, ..., t_{n_2}\}$
4	Policies			$\mathbb{I} = \{i_1, i_2, ..., i_{n_3}\}$
5	Providers			$\mathbb{P} = \{p_1, p_2, ..., p_{n_4}\}$
6	Measures			$\mathbb{M} = \{m_1, m_2, ..., m_{n_5}\}$
7	Resources			$\mathbb{S} = \{s_1, s_2, ..., s_{n_6}\}$
Primary relations				
8		Threats of Values		$V \times T$
9		Policies of Threats		$I \times T$
10		Measures of Policies		$I \times M$
11		Resources of Measures		$M \times S$
12		Policies of Threats		$P \times T$
Deduced relations				
13			VI	$VT \circ TI$
14			PI	$PT \circ TI$
15			VM	$VI \circ IM = VT \circ TxI \circ IxM$
16			VP	$VT \circ TP$
17			TM	$TI \circ IM$
18			SI	$SM \circ MI$
19			TS	$TI \circ IS = TI \circ IM \circ MS$
20			MP	$MT \circ TP = TI \circ IM \circ TP$
21			SP	$ST \circ TP = TI \circ IS \circ TP$
22			VS	$VT \circ TS = VT \circ TI \circ IS$
23		*Add cost to resources*		
24		*Add time points to measures*		
25	*Define threshold values*			
26			*Infer security functions*	

Table 7.11 – Methodology.

In Table 7.12 same steps are shown in compact format.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	29	20	21	22
Define	D	V	T		I		P		M		S											
Populate				VxT		IxT		PxT		IxM		MxS										
Infer													VxI	PxI	VxM	VxP	TxM	SxI	TxS	MxP	SxP	VxS

Table 7.12 – Methodology, compact format.

7.7 Examples

Two examples are shown. The first one is an algebraic example and the second an example with real values. The complete development, based on Table 7.11 are in **Appendix B** and **Appendix C** respectively. The second example deduces security level curves based on time. The security level function model to obtain the curves is fully developed in next chapter.

7.8 Conclusions

This framework could be expressed in a formal notation (algebraic). This allows a more careful study of its possibilities and the development of a systematic methodology in order to create and manipulate security objects.

Additionally, the object security obtained allows the extraction of different security measures.

Bibliography

[Kru56] J. B. Kruskal. On the Shortest Spanning Subtree of a Graph and the Traveling Salesman Problem. In *Proceedings of the American Mathematical Society, 7*, 1956.

[Pri57] R. C. Prim. Shortest connection networks and some generalizations. *The Bell Systems Technical Journal*, 36(6):1389–1401, 1957.

[Wol52] Arnold Wolfers. National security as an ambiguous symbol. *Political Science Quarterly*, 67(4):481–502, 1952.

Part IV

Framework Implementation and Case Study

CHAPTER

8

Security Level Time Function

“Security is an attempt to try to make the universe static so that we feel safe.”

— Anne Wilson Schaef

Contents

As stated, time is the key to the security framework. This chapter models the perceived security level as a function of time, setting the shape, behavior and composition of several security functions. This lightweight security model is the basis for the implementation of the security hexagon model.

A new approach for measuring security is proposed. A lightweight model to quantify how much security a system has is developed and, for this purpose, a function is defined in order to systematically represent how security-insecurity is perceived. The function defined shares several characteristics with a reliability function. The model is based on the characteristics of perceived security level over time and how it is related to human perception. The security level function is parameterized based on a minimal quantity of known data. The value obtained is a security metric and can be considered as an indicator for quantifying the security level and predicts how it will change over time.

In order to obtain the model, first the basic assumptions that we believe should have a security model of this kind are exposed. From these elements a model is designed considering that it has to be easy to evaluate the security at any instant of time. A predictive security is modeled, based on inferring the degree of knowledge on security and the behavior of the security over time. The model could be combined in order to form bigger and more complex systems. The proposal is based on security perception, and the model is intended to infer the security level. The metric, therefore, is a subjective leading metric.

An example of how the model could be used is fully developed in **Appendix G**.

8.1 Measuring Security

There is no doubt that measuring security is important in order to predict future situations and take the appropriate countermeasures in advance. Measuring security means knowing how secure a system is. In this paper, a system is understood as a set of elements related to each other. Unfortunately, “what can’t be measured, can’t be managed” and, therefore, a set of metrics are necessary.

A metric is “a quantitative measure of the degree to which a system, component, or process possesses a given attribute” [oEE90]. The purpose of a metric is to measure a set of attributes in order to be used as evidence of the effectiveness of an object (a program, process, etc). This information is intended to be used to facilitate decision-making and improve performance and accountability. The *rule of thumb* for a good metric is that it should be SMART, i.e. specific, measurable, attainable, repeatable, and time-dependent [Pay06]. Security metrics are mainly statistical in nature, thus a metric implies a measurement, and is defined as “a quantitatively expressed reduction of uncertainty based on one or more observations" [Hub10].

The measurement process of a metric is characterized by five activities [Pre09], Formulation, Collection, Analysis, Interpretation and Feedback. Formulation is about designing and creating the metrics and measurements. Collection is about obtaining the metrics and measurements from the system. Analysis is related to work with the metrics to get meaning. Interpretation is about connecting the data with the reality that it represents. Feedback is the last step to apply the outcome.

8.1.1 Metric Classification

Besides what the metric measures, we also have to take into account how it measures it. Therefore, a metric could be intrinsic or relative, static or dynamic, quantitative or qualitative [HV08], subjective or objective and leading or lagging. Subjective measures are commonly used in digital image processing [HLER13] [TWC15] [KK10]. Leading measures are referred to as the ability of a metric to predict tendency, and the lagging measures are referred as the ability of a metric to explain the past. A leading or predictive metric reduces the level of uncertainty in risk analysis.

Due to the specificity of a metric, they are only meaningful inside the domain where they have been defined. This is the reason why several types of metrics exist, such as software metrics [FP97], performance metrics [SK08], quality metrics [Sav13] or security metrics [Bay13] [Sav09] [Pay06].

8.2 Security metrics

(SM) Security Metrics are an approach to measuring security. A metric involves two elements, the measure and the reference. Security metrics, therefore, talk about the state or degree of safety relative to a reference point. SM don't tell anything about the actions to take nor the organization as a whole. SM have become a standard term in the context of Information Technology (IT) [Sav13]. They are used to provide security by offering evidence to engineering, risk and security management. Unfortunately, "Security cannot be measured as a universal concept due to the complexity, uncertainty, non-stationarity, limited observability of operational systems" [Sav13]. SM therefore should be considered as tools that facilitate decision making [Wan05]. SM provide trends over time in order to take improvement actions and, consequently, in terms of subjective metrics it is better to refer to them as indicators of security strength. Despite all efforts, “measurement results only provide a rough estimate of the reality” [Sav10]. Besides, the goal of achieving security is fuzzy because “obtaining a high level of security” is meaningless and subjective and therefore there are several difficulties to face.

8.2.1 Collecting measures

One of the main problems of SM is that measures can be collected from “multiple layers of the IT stack (network, system, OS, application and service, etc.)” [BMGS09]. As a result, the quantity and range of gathered data are huge. To make them meaningful is an enormous task, specially taking into account the relations among them. More metrics don’t mean better security control. The ones chosen have to support strategic decisions and have to provide information of present and future situations.

A metric, can be very simple to obtain, such as the number of viruses or very complicated, such as the quantity of unauthorized traffic in the network. Despite this, some criteria has to be used in order to ensure a metric is good enough [Sav13]. It is very difficult to find useful metrics. Security metrics are negative, and this implies that the less we detect, the more secure we are. This fact could lead to a false feeling of security if there is a long period with and absence of incidents. Security incidents cannot be measured in a positive manner, like blood sugar levels. Besides, the goal of security is to achieve the securest possible scenario, but indeed this is a subjective goal with vague meaning compared with a goal such as “achieving an audience of 3,000 spectators in a play”, for example.

8.2.2 Lagging vs leading metrics

Security Metrics could be mainly a) process-based or b) lagging indicators. The former measure activities or procedures as part of a control such as access restrictions. The main advantage of leading metrics is their ability to model behavior and, consequently, “...to support strategic security decisions, e.g. in terms of security policy changes and security investments” [BMGS09]. The latter, lagging indicators, are used in order to “measure the effect of the control activity in the data and detect occurrences of errors that may have already been introduced in the system” [BMGS09]. Lagging indicators are widely used because they are easier to collect. They might be automatically gathered as part of an IT system.

8.2.3 Objective vs subjective metrics

The goal of securing a system is subjective but SM are mainly objective. This means they are the result of gathering information and constructing indicators, measurements and dashboards. This design gives us the feeling that we have the system under control. Unfortunately, security often has unexpected outcomes. For example, the invested efforts are useless when facing zero-day vulnerabilities because they cannot be avoided. Based on this, a DRP (disaster recovery plan) has to be planned or, otherwise, assume this can happen.

There are arguments for and against subjective measures. On the one hand, they are easy to gather, they can capture knowledge and "they can presumably capture some of the implicit events that objective measures alone cannot" [JM14]. On the other hand, subjective measures have been shown to suffer from many systematic biases and they are often expressed in ordinal scales [JM14]. However, subjective and objective measures are not mutually exclusive and both should co-exist in a security design.

Subjective measures are being used in fields of economy [JM14] and computer graphics [HLER13] [TWC15] [KK10]. They make sense when data collecting are difficult to obtain; when the associated concepts are difficult to be measured, i.e. "measuring corruption, happiness, racism, consumer satisfaction, or sexual behavior" [JM14]; or when they are vaguely defined. Subjective measures can be collected from survey questions or by some sort of assessing made by experts.

Perceptions of security in human activities such as scuba diving or climbing are clearly subjective. Perceptions greatly vary depending on the person. In the IT field there are also several subjective measures like the "usability" concept for example. Indeed measurements such as the "degree of understanding of security issues among computer users, remain somewhat subjective"[Pay06].

8.3 Security approaches and elements

A model is a mental construction in order to have a better understanding of the real world. In general, a model consists of one or several inputs, a way to process the inputs and an output of expected results. A model could be made by aggregation of several sub models (Figure 8.1). The components of our model are the approach taken on security, the time factor, the perception of security and how the expected outcome has to be.

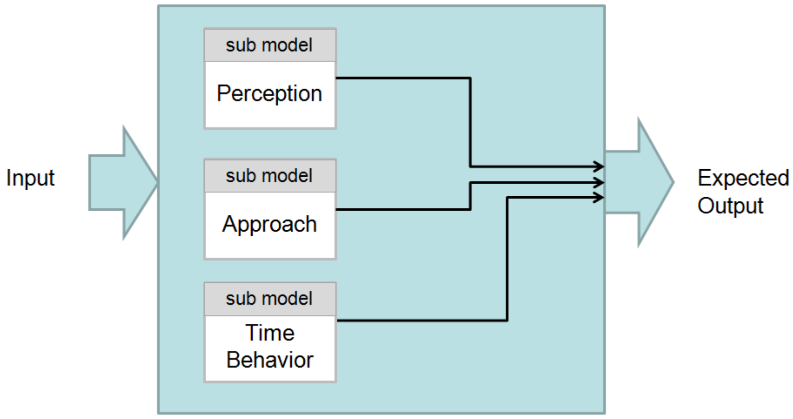


Figure 8.1 – Security model.

8.3.1 Security approaches

Security could be viewed and analyzed from many perspectives; therefore there are several approaches to the concept of security.

- **Security as a state**. This is the traditional way of measuring security. From time to time an audit on security is performed. As a result, a "security state" is obtained along with a set of actions to implement in order to achieve the desired security. This approach implies that security is seen as a state to be achieved and, therefore, a set of goals are supposed to be attained in order to carry the situation to a secure state. The level or strength of a system is in relation with the measures taken, which are mainly objective [AA14]. This view of security could have no relation with the feeling of security for experts or people working with it. Besides, this approach implies that in order to obtain more security, more resources are needed.

- **Security as a process**. This approach considers that security has to be tracked periodically. The process is mainly based on the Demming wheel. The security state is evaluated, the weaknesses are identified, the proper changes are designed and implemented and the process starts again. Techniques such as continuous auditing (CA) in order to get the "security state" are used [ZUL04] [ABKV06]. More time invested in security implies a better scenario. Under the point of view of a process, that means doing the damming circle as often as possible.

- **Security as a matter of degree**. As stated by Wolfers [Wol52], security is a matter of degree. That means that from “complete security” to “no security at all”, there exist all kind of in-between securities. From this perspective, there is no complete security to achieve but a level of security that is considered as appropriate. In order to keep the system under this range, security has to be tracked periodically and the behavior of security needs to be described. Security, in this scenario, implies that the expected quantity of security obtained heavily depends on the security elements being considered and their relations.

8.3.2 Time behavior and model

Time in Security Metrics is usually discrete, although time is indeed continuous. As a result, metrics are lagging and it is hard to foresee the security behavior. Continuous auditing techniques applied to Security Metrics involve a lot more time measurement points and the use of complex statistical techniques. If time is a main element, the behavior has to be modeled using functions. In this scenario, the security behavior could be modeled in order to know its predicted level. Metrics, therefore, become real time dependent predictive metrics.

8.3.2.1 Time modeling

As stated by [BGD11] “time must be modeled with appropriate granularity to provide temporal object access” (Figure 8.2). The following characteristics are defined.

- **Time instant**. One discrete point on time axis. Granularity of a time instant depends on its use. Could be a date, could be minutes or even milliseconds if necessary. It depends on the model. A time instant is expressed as t_i.
- **Time interval**. Time interval is a set of time instants. Indeed, a time interval is the set of discrete time points between two time instants. A time interval is expressed as T_i.

Two time intervals could be [BGD11]:

- **Disjoint**. T_i and T_j are disjoint if $T_i \cap T_j = 0$.
- **Overlapping**. Two time intervals T_i and T_j are overlapping if $T_i \cap T_j \neq 0$.
- **Content**. A time interval T_i is contained in another interval T_j if $T_i \subseteq T_j$.

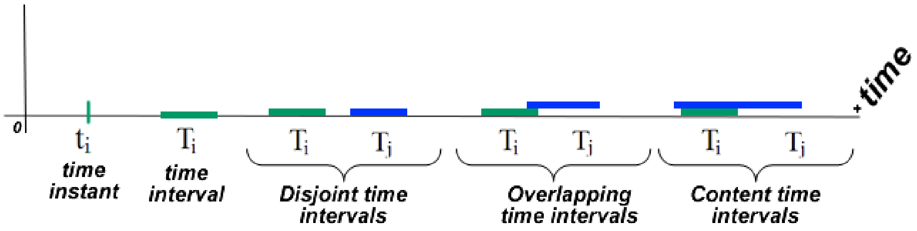


Figure 8.2 – Time point and intervals.

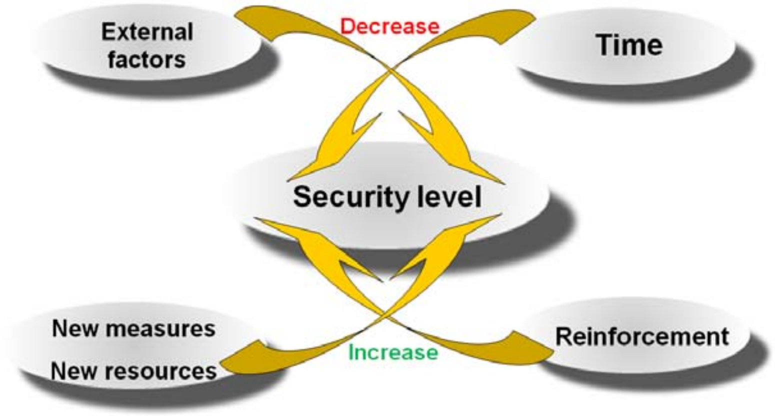


Figure 8.3 – Factors that modify perceived security level.

8.3.3 Perceived security modeling

The same scenario can be perceived by two people differently. Thus, the associated level of security fundamentally depends on the person. As a consequence the security level is subjective for persons and organizations. Security is neither perceived as static nor erratic. Security degree changes according to several factors that decrease or increase the perceived security level (Figure 8.3). Basically:

- **Perceived security changes over time.**

  There are a lot of examples surrounding us that prove that security is dynamic and, therefore, perceived as time-dependent. A climbing rope is less secure as time passes whether used or not. The same is applied to elevators, escalators, cars, information systems and one's health even. If we do nothing to keep security level in an acceptable state then the perception of security decreases over time.

- **Perceived security changes by external factors.**

External factors are those that make us change our perception of the security level, make it downgrading, with absolutely no control over them by us. These factors can be isolated in time or continuous. For example, at the beginning, the 9/11 attacks changed the personal safety level of the American population making the Administration take countermeasures. In the same way the Charlie Hebdo attack in France changed the perception of personal security of European people. European States have enforced this perception of security with measures like a European agreement.

Looking at individual level, our perception of personal safety decreases if we are informed about structure problems in our house.

- **Perceived security changes by reinforcement.**

  As security tends to degrade, a regular review of the elements that create our security feeling is necessary. The action makes that our perception of security increases. For example, computer antiviruses and operating systems are updated periodically in order to keep the computers secure. This updating, indeed, is a security reinforcement. People check their health periodically (reinforcement) to feel healthy. In security terms they need to be sure their "health level" is good. The same reasoning applies to a car or a house. When we think that any of them have changed or have become somehow "insecure", we check them (we apply reinforcement). If the verification fails, several actions to correct the situation are taken. This way, we feel "secure" for some more time.

- **Perceived security changes by adding new elements.**

  Adding new elements in a scenario increases the perceived security level. For example, adding a new router on a network or adding new control rules in the firewall. In terms of personal safety, the acquisition of a new insurance makes us feel safer.

8.3.4 Expected outcome

The expected outcome has to tell the current perceived level of security and foresee its behavior. In this sense the expected output is a predictive, subjective security metric. The desirable characteristics of the metric are:

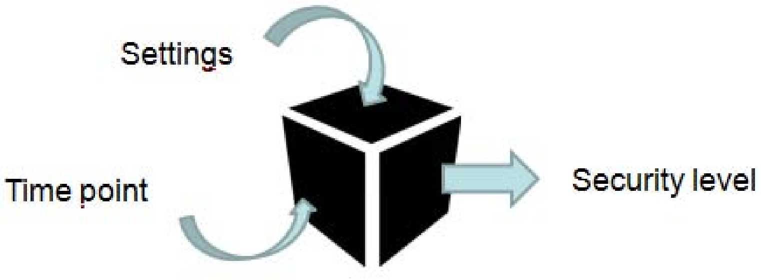


Figure 8.4 – Lightwheight Security Model.

- Output range: A value that indicates the level of security. The gray scale or security level is formalized by an interval of values [0,1] with 0 meaning “no security at all” and 1 meaning “completely secure”.
- Time dependability: For the modeled system, security changes over time and the model has to be able to tell the expected security level.
- Subjectivity: The defined security model implies that security is mainly subjective.

8.4 Lightweight Security Model

Simulation and modeling are used to study how a specific system works in order to predict behavior in a set of different conditions. By applying modeling and simulation in the area of security the aim is to explore how security varies according to elements, time and relationships involved. Our study is carried out using a set of initial assumptions necessary for the model, followed by the detailed description of the elements that constitute the model. A sketch of the black box model is drawn in Figure 8.4.

8.4.1 Model assumptions

Our aim is for the model to offer an answer to some of the following questions in a system. What degree or level of security does it have now? What degree or level of security will it have within 3 months? When could its security become “unsafe”?.

- **Simplicity**

  The model has to be easy to calculate and able to give results from a small quantity of initial data. Besides, a complex model doesn’t guarantee that the outcome will be easily understood.

- **Lightwheight**

  In a heavyweight security model it is hard to get data, implement and analyze the outcome. A lightweight model sacrifices accuracy in order to obtain quicker outcomes, simplicity, easiness of data gathering and simpler calculations.

- **Subjective**

  Objective measurement is usually hard to get and needs continuous revisions. On the one hand, time and effort are significant. For this reason, a serious setback of these models is to infer future behavior. On the other hand, data gathering is a very important step and it is a key element that determines the behavior of the whole model.

  Therefore, data has to be easy to obtain. Subjective measurements are easier to gather and it is possible to model future behavior. This approach conforms to a set of less accurate measures than the objective ones. The key point in this measurement is based on the perception of the security modeler. The security modeler, based on experience, has knowledge of the elements involved and its perception can be as good as an objective measure and much easier to collect. Besides, improving accuracy is easier.

- **Ongoing refinement**

  The model allows for ongoing refinement. From the starting point, in successive iterations, the accuracy will improve.

- **Time dependence**

  In order to support ongoing refinement and to provide predictive outcomes, time is key in this model. Methodologies related to risk analysis and security do not take into account the time factor in measurements. In Deming wheels, OCTAVE, Magerit or MEHARI time is perceived implicitly but not as an integral part of the process.

  Our proposal implies time as a fundamental element and consequently security level changes over time. It makes sense because security, without external factors, is a monotonically decreasing function. Security value in a time point is equal to or greater than at a later time point. It is expressed as:

$$S(t) >= S(t+1) \ \ \forall t$$

- **Variable granularity**

  A system is specified by the modeler. For this reason, a system could be a process, a program, a computer, a network or an entire airplane even. The goal is to model the expected behavior of a system regardless of its size (Figure 8.5). The granularity and the security metric obtained "...should be at least at a level where adequate decision-making based on them is possible" [Sav13].

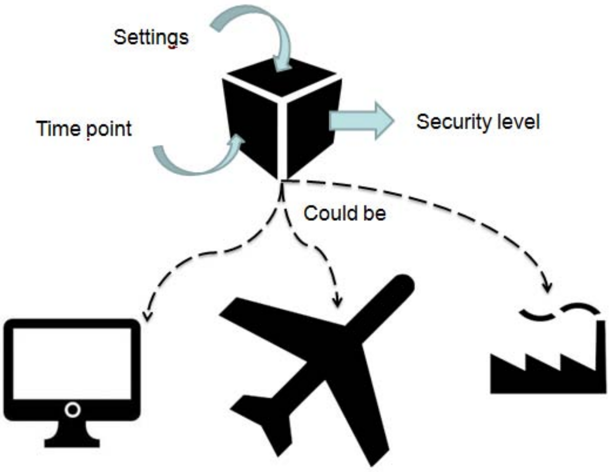


Figure 8.5 – Variable granularity.

8.4.2 Subjective security values

The information provided by the security modeler has to be minimal. In the proposal, just three time points are requested. These time points express when security is considered good, correct and bad or obsolete (Table 8.1). A time point could be any value. For simplicity reasons, non-negative real numbers are considered. There are several ways to convert a time point (dates or hours) into a real number. ISO 8601 or the use of Julian Day Number (JDN) provide algorithms for this conversion. In this sense, one talks of time in terms of $\mathbb{R}^+$.

8.4.3 Formal time model

Time is the key to our model. Hence, the perceived security level changes over time and due to several factors (Figure 8.3). Let $S(t)$ be the function that shows

Qualitative perception	*Quantitative value*
good	t_0
correct	t_1
obsolete	t_2

Table 8.1 – Time points.

the perceived security level in time point t. In order to be a security function $S(t)$ exhibits the following criteria.

Definition. A *security function* $S(t)$ is a function $S : \mathbb{R}^+ \longrightarrow \mathbb{R}$ with codomain $\mathbb{R} \in [0, 1]$. For any t, the function $S(t)$ represents the level of perceived security in instant t. Some properties of the function are:

1. S is continuous in $\mathbb{R}^+$ and undefined in $\mathbb{R}^-$.

2. The codomain $\mathbb{R}$ could be any values. For simplicity reasons, our proposal is to choose a range of values between $[0, 1]$. This implies the security function represents a probability.

3. S requires three given points t_0, t_1, t_2 with $t_0 < t_1 < t_2$ and its images $S(t_0), S(t_1), S(t_2)$ with $S(t_0) > S(t_1) > S(t_2)$ as shown in Figure 8.6.

4. S is a monotonically decreasing function. That is to say $f(x) >= f(y) \ \ \forall x <= y$. Hence, the security function is :

$$S(t) >= S(t+1) \ \ \forall t$$

5. $\lim_{t \to 0} S(t) = 1$ and $\lim_{t \to +\infty} S(t) = 0$.

6. The function shape is similar to an exponential. $S(t) \approx \beta^{-t\gamma}$

8.4.3.1 Security function and reliability function

The security function defined has some common points with a reliability function. The reliability function, also known as the survival function, gives the probability of an item to be operative on time t. The security function gives the probability of the defined system to be secure on time t. Therefore, reliability functions meet the 1), 4), 5) and 6) properties of the security function. Therefore, the shape of the security function has to be similar to the one of the reliability function.

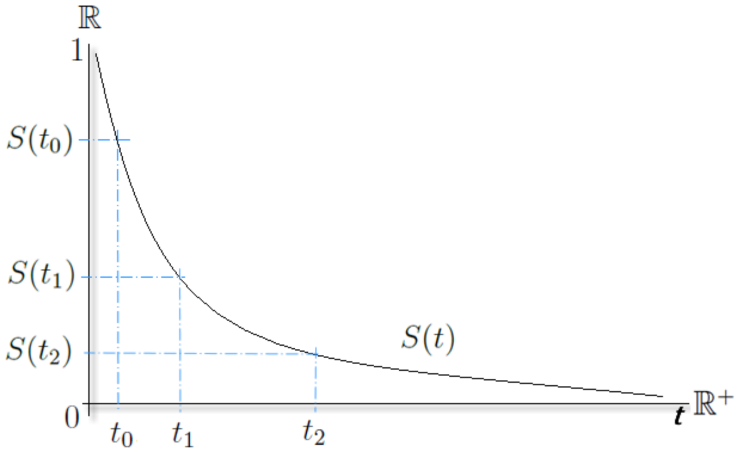


Figure 8.6 – Security over time.

8.4.3.2 Security function construction

Despite the fact that both functions are similar, it seems difficult to find an expression that matches the security function with the reliability function. A reliability function is $R(t) = e^{-\lambda t}$. The shape of this function and the security function $S(t)$ are similar, but the security function has three given points $(t_0, S(t_0)), (t_1, S(t_1))$ and $(t_2, S(t_2))$. That makes it difficult to find a reliability curve that meets this condition.

Hence, in order to describe the function, it is possible to fit a function using straight lines or splines (Figure 8.7). As monotonicity is one of the constraints of the function, a cubic hermite spline interpolation is useful [Sar02] [WA99] for this purpose.

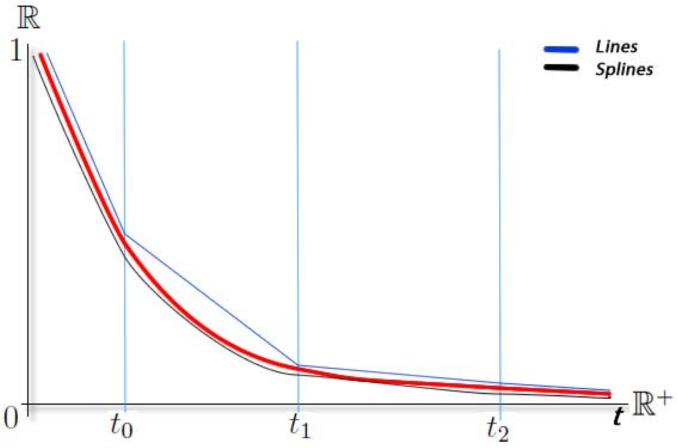


Figure 8.7 – Security function fitted using lines and splines.

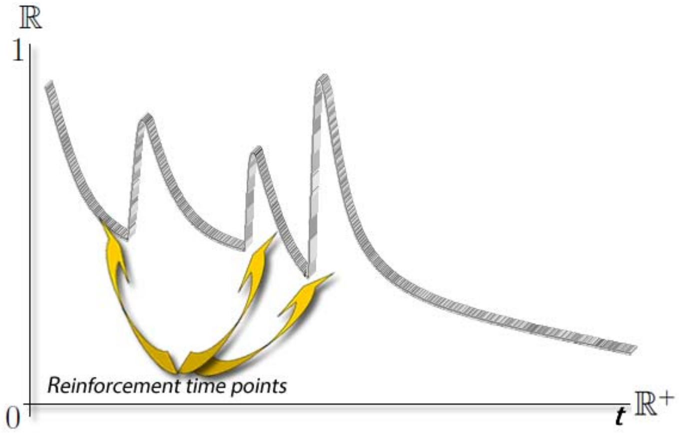


Figure 8.8 – Reinforcement time points.

8.4.3.3 Security level reinforcement

As security is not a static process, indeed it degrades over time when no action is addressed to enforce it. The security level increases due to the factors reviewed in section 8.3.3. As a result of several reinforcements over time, the security function is sawtooth shaped (Figure 8.8).

The criteria to apply reinforcement are decided by the security modeler, but it is reasonable to choose a time point close to the one when the system becomes obsolete, the t_2 time point.

An example of this behavior could be the antivirus protection of a computer. If the antivirus is not updated, then the perception of security for that computer decreases. The regular updating (reinforcement) of an antivirus makes the perception of security increase. The criteria used by the antivirus companies are twofold. Firstly, on a daily basis and secondly, when the security suddenly decreases due to the appearance of a new dangerous virus. In this scenario the update (reinforcement) is made several times in a day.

8.4.4 Security Threshold and intervals

The security function defined requires three given points $(t_0, S(t_0)), (t_1, S(t_1))$ and $(t_2, S(t_2))$. The following three thresholds are defined based on these values (figure 8.9).

- **Good threshold** THg. Is the value of the security level when the system is perceived as amply protected. THg is the image of t_0. $THg = S(t_0)$.

- **Correct threshold** THc. Is the value of the security level when the system is perceived as protected, though some incident might compromise it. THc is the image of t_1. $THc = S(t_1)$.

- **Bad threshold** THb. Is the value of the security level when the system is perceived as unprotected. THb is the image of t_2. $THb = S(t_2)$.

These threshold values define four intervals (figure 8.9).

- **Upper bound interval**. The security level values over the good threshold.

- **Secure interval**. The security level values between good and correct thresholds.

- **Insecure interval.** The security level values between correct and bad thresholds.

- **Exposed interval**. The security level values under the bad threshold.

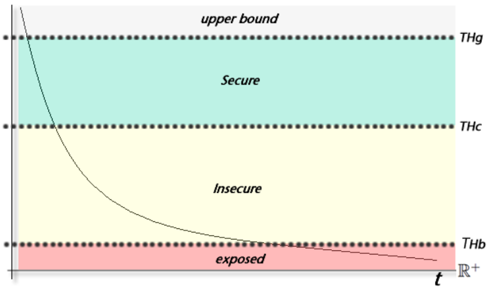


Figure 8.9 – Security levels.

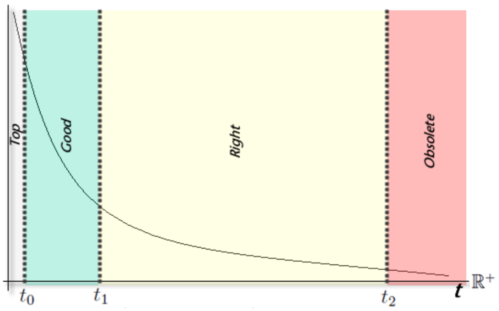


Figure 8.10 – Security time intervals.

The security function reaches a threshold value (TH_g,TH_c and TH_b) in certain time instants. Thus three time instants and four intervals are defined (figure 8.10):

- **Good instant.** It is defined as the time instant when security function reaches its maximum level (the upper bound). Its value is the given t_0 time point.

- **Correct instant.** It is defined as the time instant when security function changes the state from secure to insecure. Its value is the given t_1 time point.

- **Obsolete instant.** It is defined as the time instant when security function changes the state from insecure to exposed. Security becomes “out of date” and there is, indeed, no security. Its value is the given t_2 time point.

- **Top or upper bound interval.** It is defined as the interval of time before security function changes the state from upper bound to secure.

- **Good interval.** It is defined as the interval of time before security function changes the state from secure to insecure.

- **Right interval.** It is defined as the time interval between good and obsolete intervals.

- **Obsolete interval.** It is defined as the interval of time after security function changes the state from right to obsolete.

The time and threshold intervals are drawn in Figure 8.11. From all possible intervals just three make sense, and they are the intervals in which the security function has to fit (Figure 8.11).

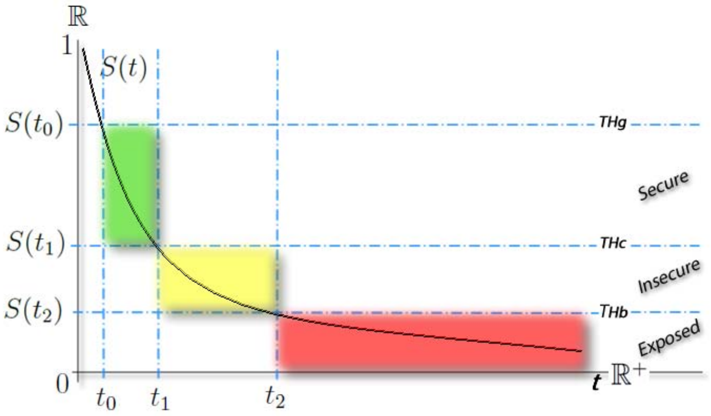


Figure 8.11 – Intervals and thresholds.

8.4.5 Security model obtained

The security model is formed with a set of values that describe the behavior and the security function. The model is graphically described as shown in Figure 8.12.

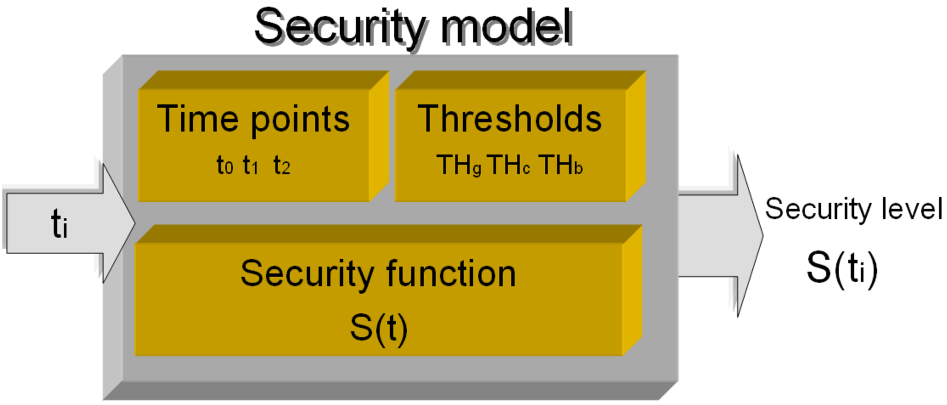


Figure 8.12 – Model.

8.5 Composition of systems

The security level function can be modeled as a block diagram (Figure 8.12). Hence, a graphical analysis technique to show the connections of the systems that corresponds to their logical relation is possible using reliability block diagrams (RBD) [Kim11]. RDBs organize the systems in parallel, series or as a combination. The logical design of the system is made based on the knowledge of the security modeler who determines if the security elements act in a series, in parallel or in a mixed way.

8.5.1 Series

When n subsystems compose a system, we say that they are logically connected in series when the expected security level function is an average of all security level functions (Figure 8.13 and formula 8.27). A restricted behavior can be used if the system is considered obsolete when at least one of the subsystems becomes obsolete. According to this behavior, the expected security level function is 0 if any subsystem is obsolete or the average otherwise (formula 8.28).

$$S(t) = \sum_{i=1}^{n} \frac{S_i(t)}{n} \tag{8.27}$$

$$S(t) = \begin{cases} 0 & if\ \exists j \mid S_j(t) = 0\ (j = 1...n) \\ \sum_{i=1}^{n} \frac{S_i(t)}{n} & otherwise \end{cases} \tag{8.28}$$

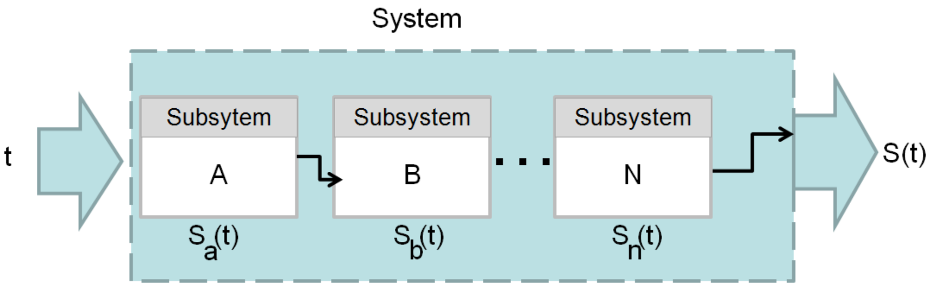


Figure 8.13 – System with n subsystems in series.

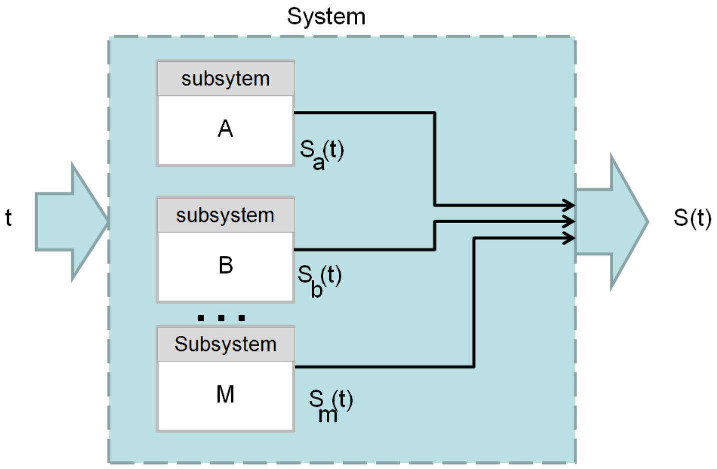


Figure 8.14 – System with m subsystems in parallel.

8.5.2 Parallel

With m subsystems that compose a system, we say that they are logically connected in parallel when the expected security level function is the minimum value of all security level functions (Figure 8.14 and formula 8.29). According to the parallel definition, it only models the restricted behavior. When any subsystem becomes obsolete, the entire system is obsolete.

$$S(t) = minimum\{S_i(t),\ \forall i \in 1...m\} \tag{8.29}$$

8.6 Conclusions

A lightweight model of measuring the security level of a system based on how security is perceived has been introduced. Security, hence, can be addressed from a subjective perspective. As a result, the lightweight model proposed is simple to apply and introduces a predictive metric. As time is revealed as key to security models, the chapter describes how security and time are related and propose a model for its interrelation and behavior. As the security level function mainly depends on time, it constitutes a time security metric also. The security function modeled and the reliability function share some common characteristics.

The model constitutes a basis for larger security models and metrics. Security is deduced based on the security level of an isolated system and the modeling of the composition of those systems. A bigger system could be modeled with several modeled security subsystems in order to obtain the security function of that system. Besides, a set of subsystems could constitute a system.

Bibliography

[AA14] Jaafar Almasizadeh and Mohammad Abdollahi Azgomi. Mean privacy: A metric for security of computer systems. *Computer Communications*, 52(0):47 – 59, 2014.

[ABKV06] Michael Alles, Gerard Brennan, Alexander Kogan, and Miklos A. Vasarhelyi. Continuous monitoring of business process controls: A pilot implementation of a continuous auditing system at siemens. *International Journal of Accounting Information Systems*, 7(2):pp. 137 – 161, 2006. 2005 Research Symposium on Integrity, Privacy, Security & Trust in an IT Context.

[Bay13] Jennifer L. Bayuk. Security as a theoretical attribute construct. *Computers & Security*, 37(0):pp. 155 – 175, 2013.

[BGD11] Padmalochan Bera, Soumya Kanti Ghosh, and Pallab Dasgupta. A wlan security management framework based on formal spatio-temporal rbac model. *Security and Communication Networks*, 4(9):981–993, 2011.

[BMGS09] Yolanta Beres, Marco Casassa Mont, Jonathan Griffin, and Simon Shiu. Using security metrics coupled with predictive modeling and simulation to assess security processes. In *ESEM Empirical Software Engineering and Measurement, 2009*, pages 564–573, 2009.

[FP97] Norman E. Fenton and Shari Lawrence Pfleeger. *Software Metrics: A Rigorous and Practical Approach*. PWS Publishing Company, 1997.

[HLER13] B. Hemery, H. Laurent, B. Emile, and C. Rosenberger. Parametrization of an image understanding quality metric with a subjective evaluation. *Pattern Recognition Letters*, 34(5):pp. 511 – 518, 2013.

[Hub10] Douglas W. Hubbard. *How to Measure Anything: Finding the Value of Intangibles in Business*. John Wiley & Sons, 2. auflage edition, 2010.

[HV08] Elaine Hulitt and Rayford B. Vaughn. Information system security compliance to fisma standard: A quantitative measure. In *IMCSIT*, pages 799–806. IEEE, 2008.

[JM14] Salar Jahedi and Fabio Mendez. On the advantages and disadvantages of subjective measures. *Journal of Economic Behavior & Organization*, 98(0):97 – 114, 2014.

[Kim11] Man Cheol Kim. Reliability block diagram with general gates and its application to system reliability analysis. *Annals of Nuclear Energy*, 38(11):2456 – 2461, 2011.

[KK10] T. Kimoto and F. Kosaka. A subjective image quality metric for bit-inversion-based watermarking. In *Picture Coding Symposium (PCS), 2010*, pages 458–461, Dec 2010.

[oEE90] The Institute of Electrical and Electronics Engineers. IEEE standard glossary of software engineering terminology. *IEEE Std 610.12-1990*, pages pp.1–84, Dec 1990.

[Pay06] S. C. Payne. A guide to security metrics. *SANS institute*, 2006.

[Pre09] Roger S. Pressman. *Software Engineering: A Practitioner's Approach*. McGraw-Hill, 7th edition, April 2009.

[Sar02] M Sarfraz. Some remarks on a rational cubic spline for the visualization of monotonic data. *Computers & Graphics*, 26(1):193 – 197, 2002.

[Sav09] Reijo Savola. A security metrics taxonomization model for software-intensive systems. *JIPS*, 5(4):pp. 197–206, 2009.

[Sav10] Reijo Savola. On the feasibility of utilizing security metrics in software-intensive systems. *IJCSNS International Journal of Computer Science and Network Security*, pages pp. 230–239, Jan 2010.

[Sav13] Reijo M. Savola. Quality of security metrics and measurements. *Computers & Security*, 37(0):pp. 78 – 90, 2013.

[SK08] Antonino Sabetta and Heiko Koziolek. *Dependability Metrics*, volume 4909 of *Lecture Notes in Computer Science*, chapter Performance Metrics in Software Design Models, pages 219–225. Springer-Verlag Berlin Heidelberg, 2008.

[TWC15] Fakhri Torkhani, Kai Wang, and Jean-Marc Chassery. Perceptual quality assessment of 3d dynamic meshes: Subjective and objective studies. *Signal Processing: Image Communication*, 31(0):pp. 185 – 204, 2015.

[WA99] G. Wolberg and I. Alfy. Monotonic cubic spline interpolation. In *Computer Graphics International, 1999. Proceedings*, pages 188–195, 1999.

[Wan05] Andy Ju An Wang. Information security models and metrics. In *Proceedings of the 43rd annual Southeast regional conference - Volume 2*, ACM-SE 43, pages 178–184, New York, NY, USA, 2005. ACM.

[Wol52] Arnold Wolfers. National security as an ambiguous symbol. *Political Science Quarterly*, 67(4):481–502, 1952.

[ZUL04] Urko Zurutuza, Roberto Uribeetxeberria, and Jesus Lizarraga. secuaudit: Continuous computer security auditing experiences. In *In Proceedings of the IADAT International Conference on Telecommunications and Computer Networks (TCN 2004)*, 2004.

CHAPTER

9

Framework Implementation

“A false sense of security is the only kind there is.”

—Michael Meade

Contents

The security framework, named security hexagon, was designed with the intention to be implemented. This chapter explains how this implementation has been carried out.

The security framework was designed and developed in order to be able to be implemented. To carry out the implementation, different techniques are used for every part. Application design is made by means of Knowledge Engineering. As the application needs to store knowledge, therefore, the Knowledge storage of the model is an ontology. Finally, the programming language is Java. Indeed modeling security using ontologies is an active field [FPM09] [BLVG^{+}08] [FW06].

These techniques are chosen by several reasons:

- Knowledge engineering along with ontologies are chosen because they are based on concepts and relations. One of the great advantages in using Knowledge Engineering relies on the possibility to infer new knowledge based on the existing one.

- This thesis has proposed a methodology and some security metrics. From the information provided to the framework, many other elements, information and knowledge can be deducted automatically. Knowledge sharing among concepts is very important, so an ontology makes this kind of operation easier. Therefore a lightweight ontology is used.

- The security framework is a generic security model that can be used in several knowledge areas. Technological ones such as computer security and non technological ones such as sociology or international relations. This feature of the model involves developing the application in a widely usable programming environment. Therefore, portability and multiplatform are important design goals to achieve. The application has to be as much autonomous as possible. Because of all that, finally, the prototype is made with Java and the ontology with flat files, which ensures portability and compatibility in almost any operating system.

- The aim is to make the software application with the ability to automate the maximum possible outcomes of the security model to show that the application is able to generate useful information for a security administration.

9.1 Primary relations

The security hexagon security model is based on concepts and relations underlying the notion of security. The primary and deduced relations as well as operations are defined in the framework (chapter 7). As all the relations could be deduced from the primary ones, just the primary relations need to be introduced.

9.2 Modeling security function

The hexagon security models security level based on the idea that security is a monotone decreasing function. In other words, security decreases over time. The model also defines three states which are secure, insecure and obsolete (Figure 9.1). The details of the security function are described in chapter 8 (Security Level Time Function)

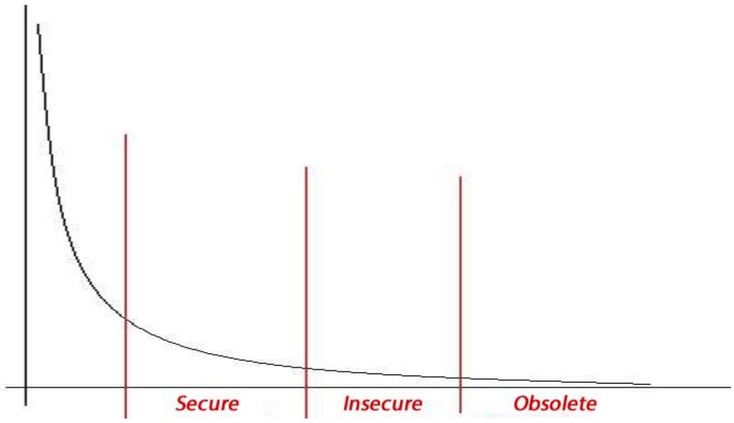


Figure 9.1 – Security function time points.

9.2.1 Function models

Section 8.4.3.2 details how to create a security function. Two algorithms for secure level calculation are implemented. In the initial settings of the application either can be chosen to be used.

- Simple

  The value depends on the three defined time points, and three values. Each one for any time interval. Before the first time point, security is always the maximum, and after the last time point, security is always the obsolete value (Figure 9.2). The outcome into the application is shown in Figure 9.3.

- Complex

  In order to fit the security curve, straight lines are used as explained in section 8.4.3.2 (Figure 9.4). Figure 9.5 shows this modeling into the application.

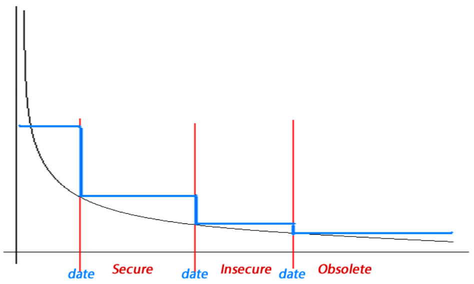


Figure 9.2 – Simple: security function time points.

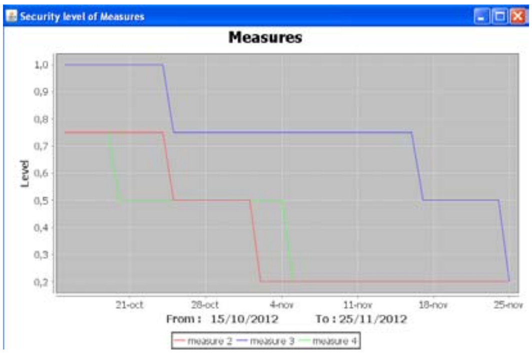


Figure 9.3 – Application: security function time points.

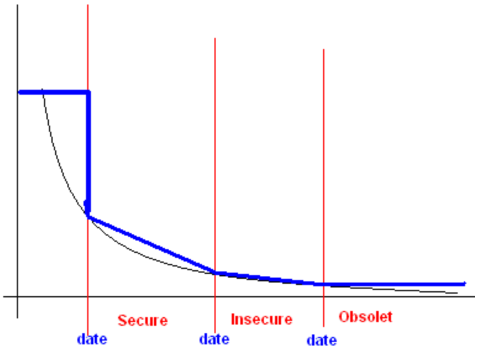


Figure 9.4 – Complex: security function time points.

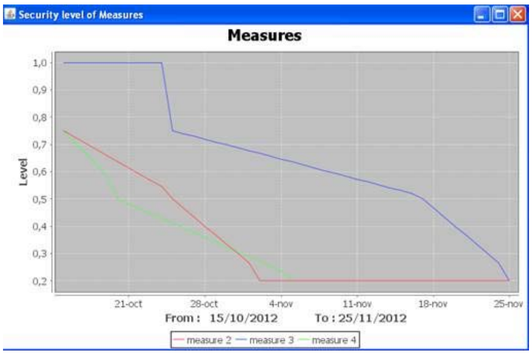


Figure 9.5 – Application: security function time points.

9.2.2 Combining security functions

The composition of several elements to get the level of security are implemented using the formulas described in section 8.5. Figure 9.6 and Figure 9.7 show an example of the curve for the two models.

The restricted behavior detailed in section 8.5 is also implemented. Therefore, if any element becomes obsolete its security level drops below the obsolete threshold because all the system is compromised no matter the security level of the others elements. It's the known model of "security is as week as the weakest element". In the initial settings of the application there is an option to choose this behavior.

9.3 Application architecture

From the user's point of view, the architecture is shown in Figure 9.8. There is a knowledge storage and a GUI interface. The model can do a lot of more things

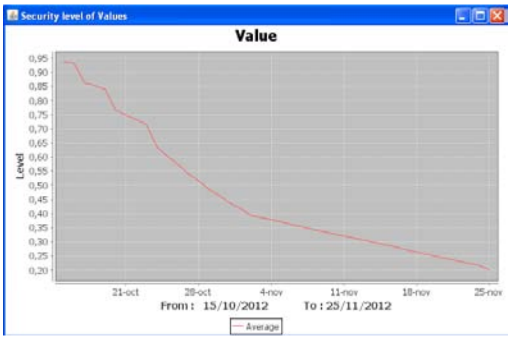


Figure 9.6 – Security function without restrict modeling.

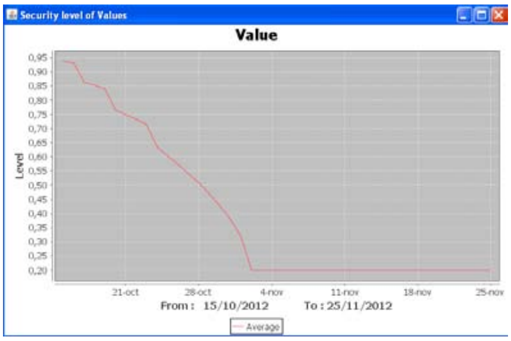


Figure 9.7 – Security function with restrict modeling.

such as a knowledge inference and alert detection, which appears in Figure 9.8 as modules. Theses modules obtain the information from the ontology in order to be proactive to warn the user for security troubles.

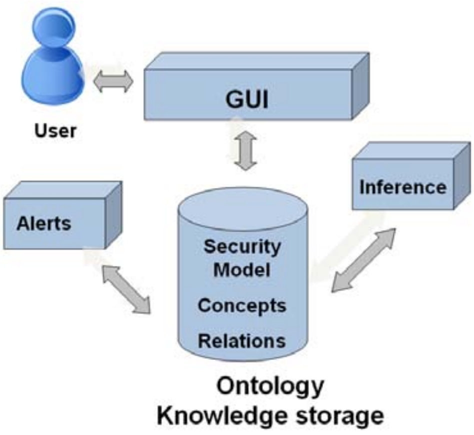
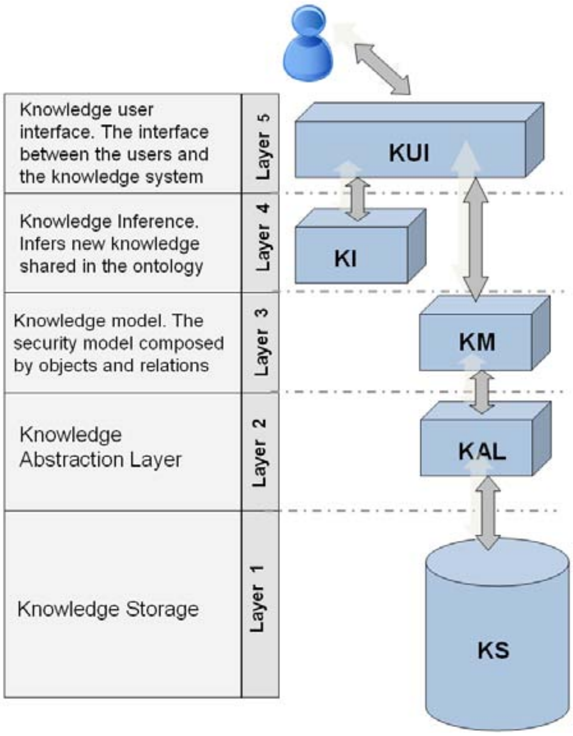


Figure 9.8 – Application architecture.


Knowledge user interface. The interface between the users and the knowledge system

Knowledge Inference. Infers new knowledge shared in the ontology

Knowledge model. The security model composed by objects and relations

Knowledge Abstraction Layer

Knowledge Storage

Figure 9.9 – Application layered architecture.

Indeed, the application is constructed in layers, allowing layers to be implemented in several ways. A scheme is shown in Figure 9.9.

- **Layer 1:** Ontology could be implemented in several ways. Once the ontology is conceptually designed, the knowledge could be stored by means of knowledge tools such as protégé[12] or a relational database.

- **Layer 2**: In order to provide independence between the ontology and the classes that manage the knowledge, there is middleware that handles the communication between the ontology and classes.

  That layer is Ontology dependent, thus, if the Knowledge is stored in different knowledge storage, this layer need to be rewritten. It's named KAL (Knowledge Abstraction Layer)

- **Layer 3**: Is composed by the objects and classes and relations. The objects make the requests to the KAL layer, and thus they don't know how the knowledge is stored. Classes and relations indeed are the implementation of the hexagon security model.

- **Layer 4**: Inference Module. From the stored knowledge, new knowledge could be obtained. This layer takes care of that. Additionally, it controls that starting knowledge is consistent and so the knowledge inferred. This layer, therefore, makes requests and store knowledge into the knowledge storage.

- **Layer 5**: The user, by means of a graphical interface, talks to the objects and their relations. The user is no aware on how the knowledge is implemented

9.4 Knowledge storage

The repository stores knowledge, but an ontology could be, indeed, implemented in several ways. Once the ontology is conceptually designed, the knowledge could be stored by means of knowledge tools such as Jena®[13], protégé and make queries in SPARQL[14] or a relational database.

Mapping between RDBMS and ontologies is an active field [DCES04] [LW07] [AI07] [MCBV12]. Mainly, using a relational storage, a concept becomes a table and the individuals become records in the table. These records need a key that allow representing relations between concepts. The relation between concepts,

[12] A free, open-source ontology editor and framework for building intelligent systems. http://protege.stanford.edu/

[13] (registered trademark of Apache Software). A free and open source Java framework for building Semantic Web and Linked Data applications. https://jena.apache.org/

[14] SPARQL is a query language and a protocol for accessing ontologies. http://www.w3.org/TR/rdf-sparql-query/

therefore, is expressed by a table containing a record for any individual of the two tables which are related.

9.5 Operating

In order to use the ontology and application, its basic steps are drawn in Figure 9.10.

- Populate concepts (create individuals).
- Populate relations.
- Infer knowledge from Ontology.
- Request knowledge from Ontology.

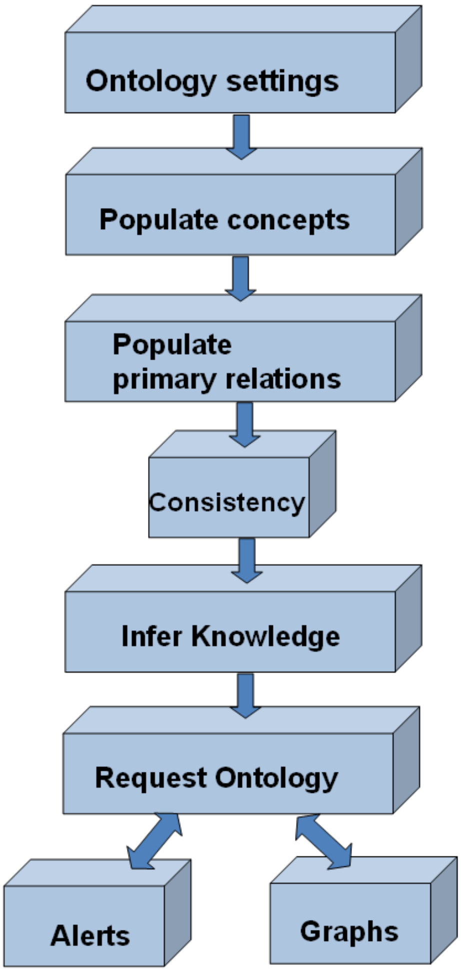
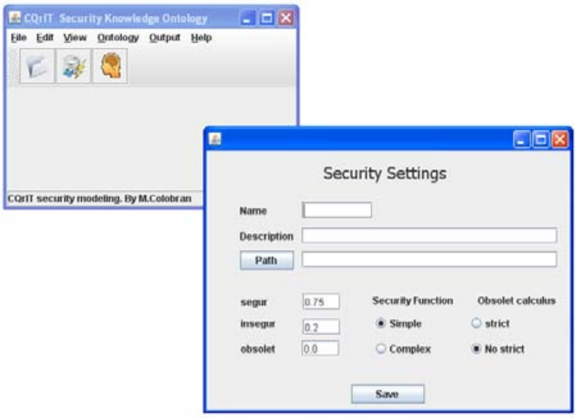


Figure 9.10 – Main steps using application.

9.6 Creating a security object

9.6.1 Initial settings

The application needs some initial parameters (Figure 9.11).


CQrIT Security Knowledge Ontology
File Edit View Ontology Output Help
CQrIT security modeling. By M.Colobran
Path
Save

Security Settings

Name

Description

segur 0.75

insegur 0.2

obsolet 0.0

Security Function: ● Simple ○ Complex

Obsolet calculus: ○ strict ● No strict

Figure 9.11 – Security settings.

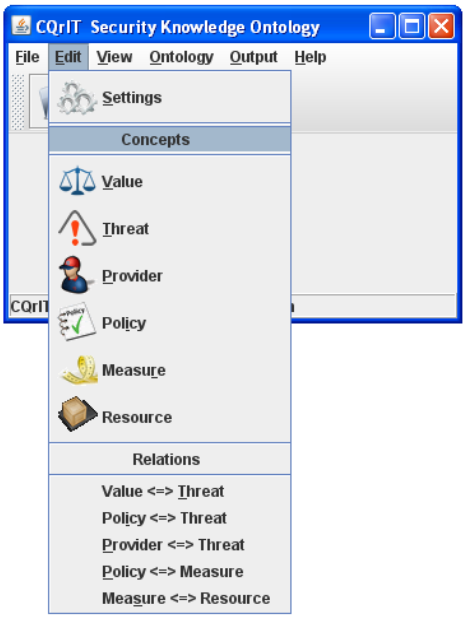


Figure 9.12 – Concepts and primary relations.

- Name and Description are the identification by the user of the security ontology.

- Path. Is the directory to store the ontology.

- The security value levels to consider the security elements as secure, insecure and obsolete.

- Security Function. As explained, two ways to calculate security level. Simple is worst than complex but simple is much more easy to calculate.

- Obsolete calculus. Selecting *strict* instructs the application to consider obsolete the security level in a time point if any of the elements is obsolete.

9.6.2 Ontology data population

As described before, the first step is filling the concepts with individuals. The concepts in application are the ones defined in security hexagon model. Concepts don't need to be filled one after other. Indeed it's a refinement process. At any moment the ontology could be saved, and concepts could be added or removed later (Figure 9.12).

In chapter 7, primary and deduced relations are explained. The security hexagon security model describes 15 possible relations. By means of ontology engineering and the transitive property of the relations, just filling the primary relations, the remaining ones could be deduced. Primary relations in the application are the ones chosen in section 7.3.1, $V \times T$, $T \times I$, $I \times M$, $M \times S$, and $P \times T$.

9.6.3 Integrity check

Once concepts and primary relations are introduced or loaded, integrity check takes care to validate the ontology is neither inconsistent nor corrupted. Integrity check reviews that any element belongs to at least one concept and the relations between objects are really applied to existing individuals.

9.6.4 Managing knowledge

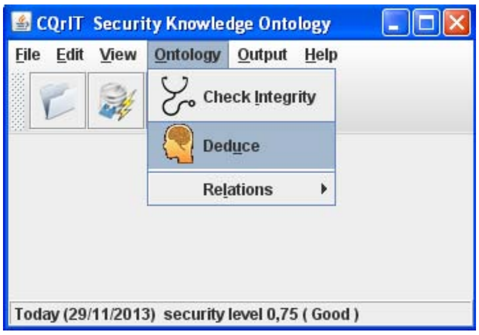
One of the major advantages in using hexagon security model is the ability to infer new knowledge based on the existing one. Knowledge inference (Figure 9.13) infers all the relations between concepts based on knowledge introduced. Knowledge inference is capable of:

- Create new objects.
- Create new relations.
- Populate individuals by inferring new knowledge.
- Populate relations by inferring new instances of object relations.
- Deduce the cost for any object in the ontology.
- Deduce the security level for any object in the ontology.

9.6.5 Security reports

Security reports of the application are graphical. A report could be requested using a defined or automatically calculated period of time (Figures 9.14 and 9.15).

Based on security values, an alert system is implemented (Figure 9.16). Alerts are important because they inform us of the elements of the ontology that are obsolete. Those elements represent a security breach. The right status of the ontology is not to have any elements in this state. An alert is an advice to review



Today (29/11/2013) security level 0,75 (Good)

Figure 9.13 – Inferring knowledge.

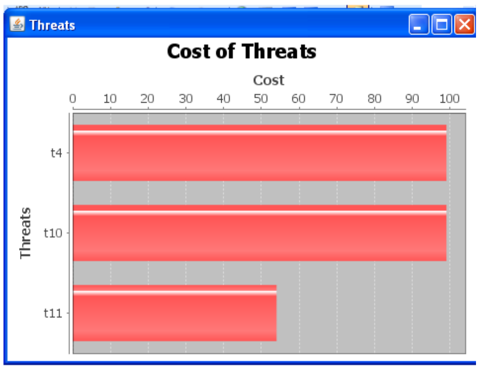


Figure 9.14 – Cost output.

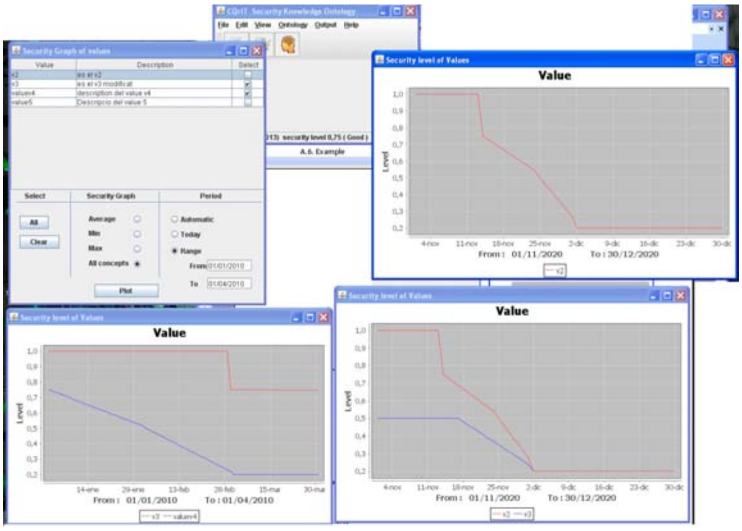


Figure 9.15 – Security level output.

the object, take actions and later update the time periods for that element. The action of change the time periods of one or more elements overcome the problem because the ontology is inferred again.

9.7 Conclusions

In this chapter a real design and implementation of the security hexagon model, proposed in chapter 5, is carried out. In order to construct it, several factors have been considered such as portability, simplicity. The level time function proposed in chapter 8 is also implemented. The application constructed is, indeed, a proof of concept.

The next chapter details how this implementation of the security model is used in a real situation.

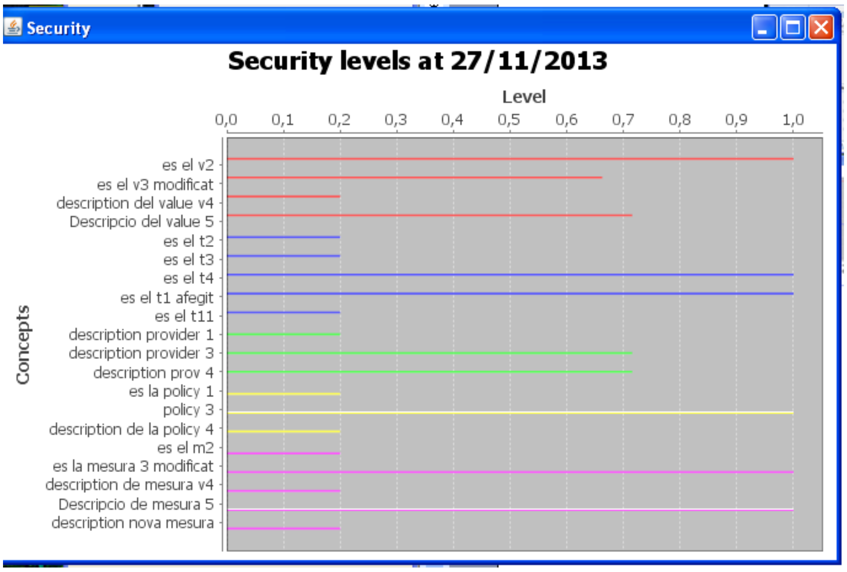


Figure 9.16 – Alerts.

Bibliography

[AI07] Sören Auer and Zachary G. Ives. Integrating ontologies and relational data. Technical Report MS-CIS-07-24, University of Pennsylvania Department of Computer and Information Science Technical, 11 2007.

[BLVG+08] Carlos Blanco, Joaquin Lasheras, Rafael Valencia-García, Eduardo Fernández-Medina, Ambrosio Toval, and Mario Piattini. A systematic review and comparison of security ontologies. In *Proceedings of the 2008 Third International Conference on Availability, Reliability and Security*, ARES '08, pages 813–820, Washington, DC, USA, 2008. IEEE Computer Society.

[DCES04] Souripriya Das, Eugene Inseok Chong, George Eadon, and Jagannathan Srinivasan. Supporting ontology-based semantic matching in {RDBMS}. In Mario A. Nascimento, M. Tamer Özsu, Donald Kossmann, Renée J. Miller, José A. Blakeley, and Berni Schiefer, editors, *Proceedings 2004 {VLDB} Conference*, pages 1054 – 1065. Morgan Kaufmann, St Louis, 2004.

[FPM09] Stefan Fenz, Thomas Pruckner, and Arman Manutscheri. Ontological mapping of information security best-practice guidelines. In *Business*

Information Systems, 12th International Conference on Business Information Systems, BIS 2009. Springer Berlin Heidelberg, 4 2009.

[FW06] Stefan Fenz and Edgar Weippl. Ontology based it-security planning. In *PRDC*, pages 389–390. IEEE Computer Society, 2006.

[LW07] Ki Jung Lee and Taeg Keun Whangbo. Semantic mapping between rdbms and domain ontology. In *Computers and Communications, 2007. ISCC 2007. 12th IEEE Symposium on*, pages 1007–1012, 2007.

[MCBV12] Carmen Martinez-Cruz, IgnacioJ. Blanco, and M.Amparo Vila. Ontologies versus relational databases: are they so different? a comparison. *Artificial Intelligence Review*, 38(4):271–290, 2012.

CHAPTER

10

Case Study

> **“I am regularly asked what the average Internet user can do to ensure his security. My first answer is usually ’Nothing; you’re screwed’.”**
>
> **—Bruce Schneier**

Contents

This chapter describes how the implementation is used in a real scenario. A cornerstone Institution which offers several computing services to universities, the Consorci de Serveis Universitaris de Catalunya (CSUC).

Some data and elements have been changed or removed in order to protect the privacy of CSUC Institution. The outcome, therefore, is not completely real, but it still constitutes a very good example.

We applied to a very important computer center in Barcelona (CSUC) to test the model. The collaboration with the Center has been excellent and the whole process was done in conjunction with them. Actually they are currently using the application in order to have a big overview of their security status.

10.1 CSUC Institution

The Consorci de Serveis Universitaris de Catalunya (CSUC)[15] was created in 1991 by the Fundació Catalana per a la Recerca i la Innovació[16] as a public consortium formed by the Generalitat of Catalonia and the ten Catalan universities (UB, UAB, UPC, UPF, UdG, URV, UdL, UOC, URL, UVic-UCC) with the collaboration of CSIC[17].

CSUC is one of the flagships in IT. It manages infrastructures based on information and communications technology to serve research and development undertaken by companies and institutions that require high performance computing. CSUC aims at offer a range of services to scientific institutions. Its activity is focused in several areas:

- Provide systems for scientific computing, both academic and industrial.
- Provide all the IT security mechanisms, relieving institutions from this burden.
- Supply communication networks along with its elements such as CATNIX (the interchange node of data traffic in the Catalan territory.)
- Consolidate university services in order to increase their efficiency and reduce costs bundling services.
- Reduce power consumption and CO_2 generation using low power and high efficiency information technologies.

[15] Consortium of University Services of Catalonia. http://www.csuc.cat

[16] Catalan Foundation for Research and Innovation (FCRI)

[17] Higher Council for Scientific Research (CSIC)

- Store portals and repositories for university information (TDX, RECERCAT, RACO MDX ...) and data storage also.
- Services related to electronic administration such as digital certification or electronic voting systems.
- Promote the use and benefits of these technologies.
- The operation and maintenance of the entire infrastructure.

10.1.1 Aims and study scope

The institution services catalog is large. This study focuses only on security of a part of the e-Administration services (Figure 10.1)[18].

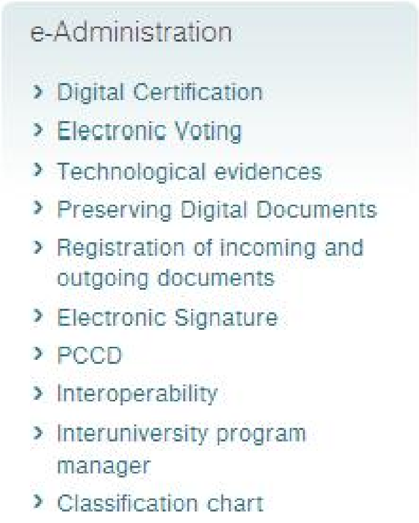

e-Administration
› Digital Certification
› Electronic Voting
› Technological evidences
› Preserving Digital Documents
› Registration of incoming and outgoing documents
› Electronic Signature
› PCCD
› Interoperability
› Interuniversity program manager
› Classification chart

Figure 10.1 – e-Administration Services.

10.2 Applying the model

The aim of the study is to verify that the hexagon security model and its implementation can be modeled in a real environment. In the first interview there was an agreement to make the study over a small set of services with similar characteristics. The services chosen, in addition to being functionally similar, share some

[18] http://www.csuc.cat/en/e-administration

hardware or software resources. The data gathering phase was made through interviews. The schematics produced was checked before the analysis phase. All the process was supervised by CSUC engineers.

10.2.1 Model granularity

The design of the Security Hexagon Model permits to vary the granularity of the elements and systems involved in the analysis. A variable granularity allows to better describe the organization of the institution and, hence, some parts are considered at system-level (for example authentication) and others at element level (for example backup). From the point of view of the model, both are considered resources and consequently independent of the level.

10.2.2 Implementation steps

The information provided to the model are instances of concepts and relations among these concepts and instances. The steps to follow are shown in Table 10.1 and graphically in Figure 10.2.

	Description
Step 1	Identify security by means of defining range
Step 2	Identify values (could be done at the end)
Step 3	Identify security policies
Step 4	Identify, for each policy, providers, measures and resources.
Step 5	Create the primary relations of each policy
Step 6	Join concept objects of all policies
Step 7	Assign cost to resources
Step 8	Assign time points to measures
Step 9	Create primary relations
Step 10	Infer secondary relations

Table 10.1 – Steps.

Steps 1 to 3 are performed at the same time. Steps 4 and 5 are the more complex, because they imply identification of elements and relations. The study will be based on policies; therefore, it is useful to make a diagram for each policy identified including a set of relations.

Steps 6, 7, 8 and 9 are performed when all elements and relations have been described. Finally, step 10 is based on information created in step 6 and is made

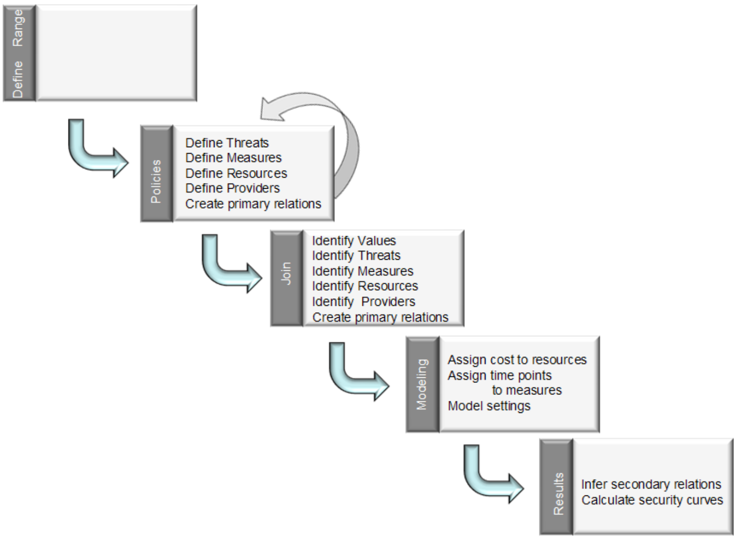


Figure 10.2 – Steps.

by the security modeler.

10.2.3 Services analysis

For each service, we proceeded making a logical description, elements involved and how they relate. It was taken into account:

- Hardware elements.
- Software elements.
- infrastructure.
- The set of possible threats that the service could suffer.

Study is restricted only to those services considered most critical. Each service, therefore, shall be analyzed as a security policy (Table 10.2).

10.3 Physical architecture

The physical structure of e-Administration (Figure 10.3) is composed by several elements. In order to make the analysis, these elements are grouped by function. Therefore, mainly, the physical structure is composed by:

- Router. This system provides communication of the entire infrastructure to public network (Internet). Indeed it's a whole DMZ, but for our purposes we refer to it as *router*. The router is a shared resource for all services.

	Description
Electronic Voting	The electronic voting platform (e-Vot) makes it possible to carry out elections and consultations electronically and to incorporate all of the universities' electoral models. http://www.csuc.cat/en/e-administration/electronic-voting
Technological Evidences	The e-logs platform is a custody solution of digital evidences. It acts as a trusted third party responsible for the custody of generated evidences by other actors, and which are collected through different harvesters. http://www.csuc.cat/en/e-administration/technological-evidences
Preserving Digital Documents	The introduction of electronic, or digital, documents created the need to preserve these digital objects in a way that guaranteed their integrity, confidentiality, and accessibility in the long term, while maintaining their legal validity. http://www.csuc.cat/en/e-administration/preserving-digital-documents

Table 10.2 – Services Analyzed

- Servers. Every service is running (either physically or logically) into its own server.
- Backup system. In order to protect data, applications and to achieve continuity, there is a backup system. Backup is a shared resource for all services.
- Authentication. Every service has its own authentication system (either physically or logically) for information protection and access control.

10.4 Logical architecture

Logical architecture, for simplicity and efficiency, is very similar in all the services analyzed. Figure 10.4 shows a general outline of the services.

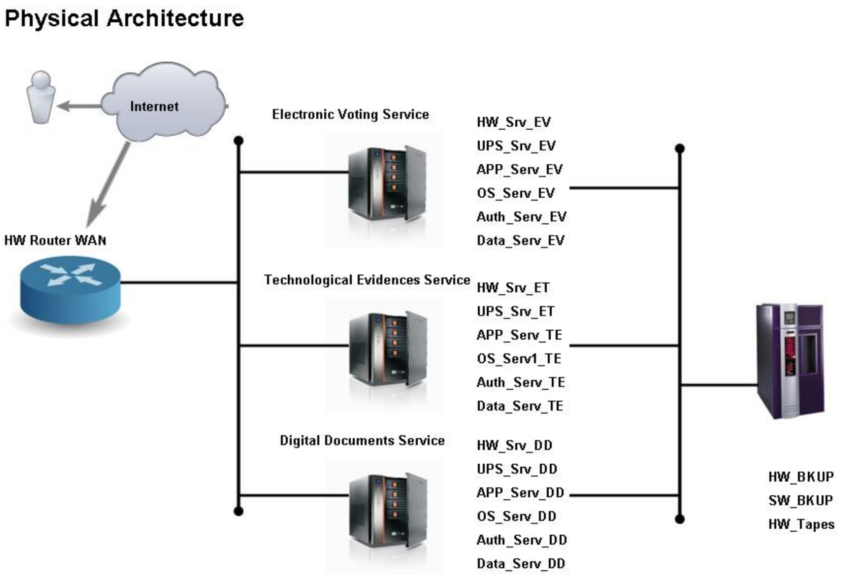


Figure 10.3 – Physical Architecture.

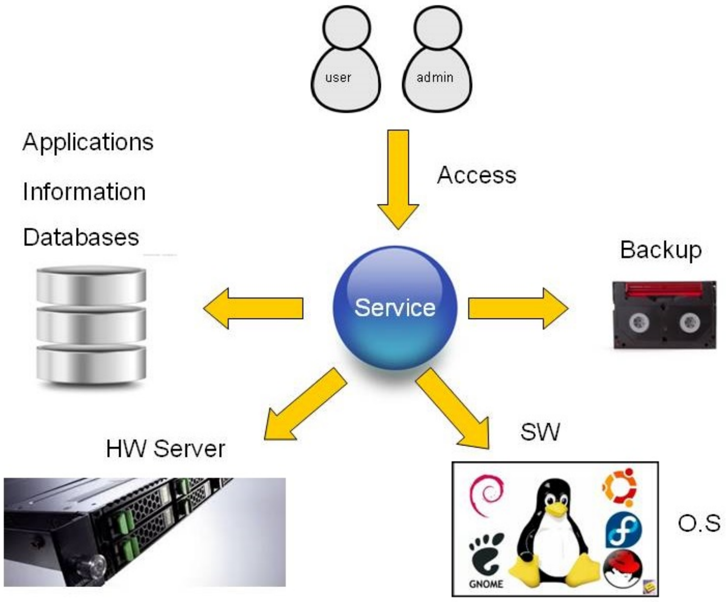


Figure 10.4 – Logical Architecture.

10.4.1 Electronic Voting

10.4.1.1 Logical scheme of service and elements

The logical architecture of electronic voting service is identified with hardware and software elements involved. Figure 10.5 shows how they are related.

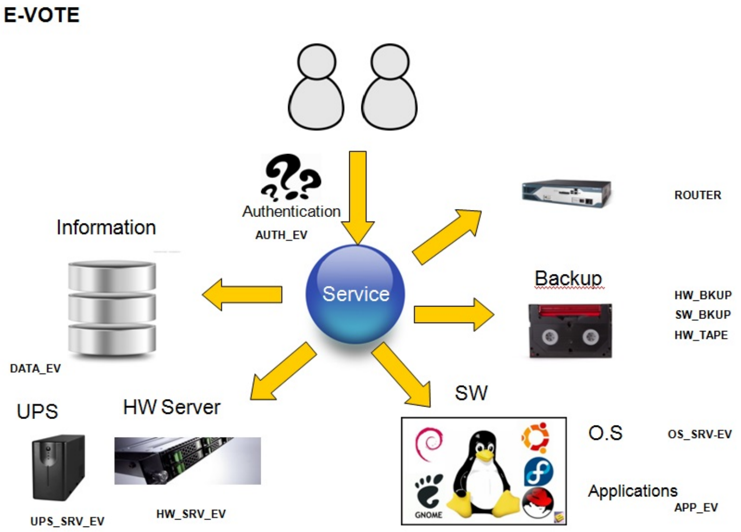


Figure 10.5 – “electronic voting” logical architecture.

10.4.1.2 Security policy

Based on the schema, the concepts and relationships of the service are identified. An identification of threats, measures, security providers and resources as well as the relationship between them is made **(Appendix E, Figure E.1)**.

10.4.1.3 Element identification

The list of concepts and instances of identified concepts is made **(Appendix E, Figure E.2)**.

10.4.1.4 Creation of primary relations

Based on preceding information, tables of relations among the elements is constructed. All these tables and relations could be reviewed in **Appendix E, Figure E.3**.

10.4.2 Technological Evidences

10.4.2.1 Logical scheme of service and elements

The logical architecture of technological evidences service is identified with hardware and software elements involved. Figure 10.6 shows how they are related.

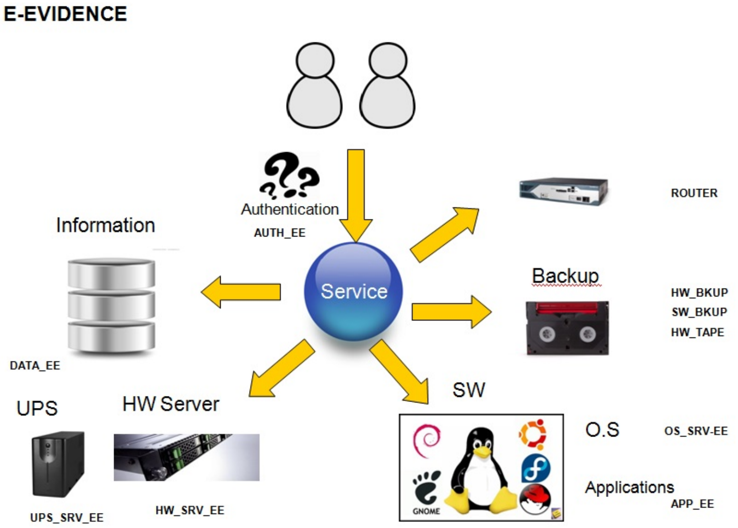


Figure 10.6 – “technological evidences” logical Architecture.

10.4.2.2 Security policy

Based on the schema, the concepts and relationships of the service are identified. An identification of threats, measures, security providers and resources as well as the relationship between them is made **(Appendix E, Figure E.4)**.

10.4.2.3 Element identification

The list of concepts and instances of identified concepts is made **(Appendix E, Figure E.5)**.

10.4.2.4 Creation of primary relations

Based on preceding information, tables of relations among the elements is constructed. All those tables and relations appear in **Appendix E, Figure E.6**.

10.4.3 Preserving digital documents

10.4.3.1 Logical scheme of service and elements.

The logical architecture of preserving digital documents service is identified with hardware and software elements involved. Figure 10.7 shows how they are related.

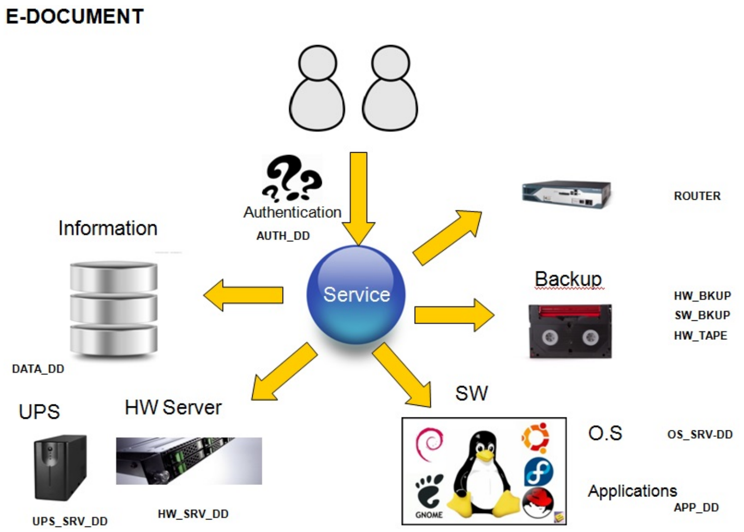


Figure 10.7 – “Preserving digital documents” logical Architecture.

10.4.3.2 Security policy

Based on the schema, the concepts and relationships of the service are identified. An identification of threats, measures, security providers and resources as well as the relationship between them is made **(Appendix E, Figure E.7).**

10.4.3.3 Element identification

The list of concepts and instances of identified concepts is made (**Appendix E, Figure** E.8).

10.4.3.4 Creation of primary relations

Based on preceding information, tables of relations among the elements is constructed. All those tables and relations could be reviewed in **Appendix E, Figure** E.9.

10.5 Model construction

In this step, primary concepts and relations are made. Interviews are carried in a policy base in order to identify the elements. As we are making the study based on services and policies, the list of concept elements and primary relation tables is constructed at this point.

10.5.1 List of concept elements

Two values are raised from the study.

- **Data protection**. Every ICT analysis makes information as a value to protect.
- **Continuity**. Because of the critical work of the institution, it's a big concern to provide the services that are offered to the academic community even in a failure situation. Therefore, continuity is considered as value to be considered within the institution.

		Values
✓	V1	Information
✓	V2	Continuity

Table 10.3 – Values.

The complete lists of elements identified could be reviewed in **Appendix E**.

Values	**Appendix E, Table E.1**
Threats	**Appendix E, Table E.2**
Providers	**Appendix E, Table E.3**
Policies	**Appendix E, Table E.4**
Measures	**Appendix E, Table E.5**
Resources	**Appendix E, Table E.6**

Table 10.4 – Concept elements list.

10.5.2 Primary relations

Once all elements and relations are identified, the primary relations could be constructed. The five primary relations are Value-Threats, Policy-Threats, Policy-

Measures, Provider-Threats and Resources-Measures. The whole details of the relations could be reviewed in **Appendix E**.

Value-Threats	**Appendix E, Table E.7**
Policy-Threats	**Appendix E, Table E.8**
Policy-Measures	**Appendix E, Table E.9**
Provider-Threats	**Appendix E, Table E.10**
Resources-Measures	**Appendix E, Table E.11**

Table 10.5 – Primary relations

10.5.3 Assessment and time points

Resources and time points are assessed based on the following criteria (Tables 10.6 and 10.7).

- CSUC criteria of assessment is the *cost+effort* to keep running properly.
- Total cost of resources is calculated over 100. Therefore it could be seen as a percentage.

10.5.3.1 Assessment

The assessment of resources is made with the following values:

Resource	Description	Value
HW_BKUP	HW Backup	15.0
SW_BKUP	SW Backup	2.0
HW_TAPE	Tape sets	4.0
HW_RTR	Router	10.0
HW_SRV_EV	Server EV	10.0
OS_SRV_EV	O.S. Server EV	2.0
UPS_SRV_EV	UPS Server EV	3.0
$DATA\_EV$	EV Service Data	2.0
APP_EV	Application service	5.0
$AUTH\_EV$	Authentication	1.0
HW_SRV_EE	Server EE	10.0
OS_SRV_EE	O.S. Server EE	2.0
UPS_SRV_EE	UPS Server EE	3.0
$DATA\_EE$	EE Service Data	2.0
APP_EE	Application service	5.0
$AUTH\_EE$	Authentication	1.0
HW_SRV_DD	Server DD	10.0
OS_SRV_DD	O.S. Server DD	2.0
UPS_SRV_DD	UPS Server DD	3.0
$DATA\_DD$	DD Service Data	2.0
APP_DD	Application service	5.0
$AUTH\_DD$	Authentication	1.0

Table 10.6 – Resources assessment.

Time points (secure, insecure and obsolete) are introduced in measures. The following values are introduced with the supposition that in October the first, every element is in its right state.

Resource	Description	Date	Date	Date
M_PF	Power Failure	01/10/2014	01/04/2015	01/10/2015
M_AC	Access Control	01/06/2014	01/12/2014	01/06/2015
M_DL	Data Loss	01/10/2014	01/11/2014	01/12/2014
M_HF	HW Failure	01/03/2014	01/03/2015	01/03/2016
M_SF	SW Failure	01/10/2014	01/04/2015	01/10/2015
M_APP_EV	e-vote application	01/08/2014	01/10/2014	01/12/2014
M_APP_EE	e-evidence application	01/10/2014	01/12/2014	01/02/2015
M_APP_DD	e-document application	01/10/2014	01/02/2015	01/04/2015

Table 10.7 – Security time point of measures.

10.6 Infer knowledge

The hexagon security model infers secondary relations in a systematic way. The java® implementation of the model allows us to obtain the information.

10.6.1 Initial parameters

10.6.1.1 Curve values

Secure, insecure and obsolete values are considered to be:

Secure	0.75
Insecure	0.40
Obsolete	0.10

Table 10.8 – Security setting values.

10.6.1.2 Model used

The preferred model is the strict one because the institution considers risky that any element could be compromised. Despite of that, both models are analyzed in order to determine how security falls over time.

10.6.2 Secondary relations

Once all information is introduced in the model, all secondary relations are inferred by the application. All of them could be reviewed in **Appendix E, Figures E.10** to **E.19**.

10.7 Analysis and results

After data gathering, interviews and the construction of primary relations, all the data are introduced into the application. The Security Hexagon Model deduces the remaining knowledge, which is used to make the analysis. The analysis is performed using both the strict and no strict modeling.

- **Security level.** The security level is correct. This is an expected outcome. A very important feature of the model is that it allows knowing how quick the security level decreases and how many times it takes the system to become insecure (using strict and non strict modeling). The security level is plotted with the simple (Figure 10.8) and complex modeling (Figure 10.9).

- **Cost**. Cost, in the case study, is referred to an assessment of the elements made by the institution. As the backup system is present in all policies, it becomes a critical system and the one highest assessed.
- **Reinforcement**. The application shows when the system becomes insecure and, therefore, it permits to know when reinforcement has to be made and on what elements. Besides, this information is very useful because it is now possible to program in advance the technical stops.
- **Dependency**. Backup system is revealed as the element which the whole system mainly relies on.
- **Obsolete time points**. There are several resources with their time points associated. As the obsolete time points are spread in time, this produces that the whole system changes to an insecure state often.

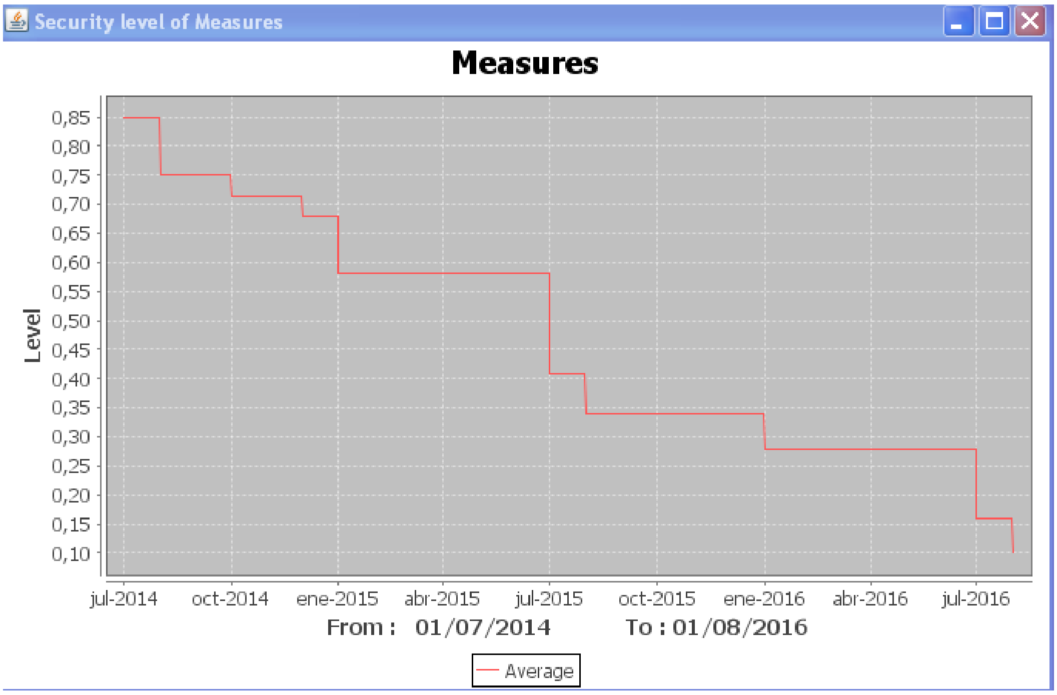


Figure 10.8 – Security curve using simple modeling.

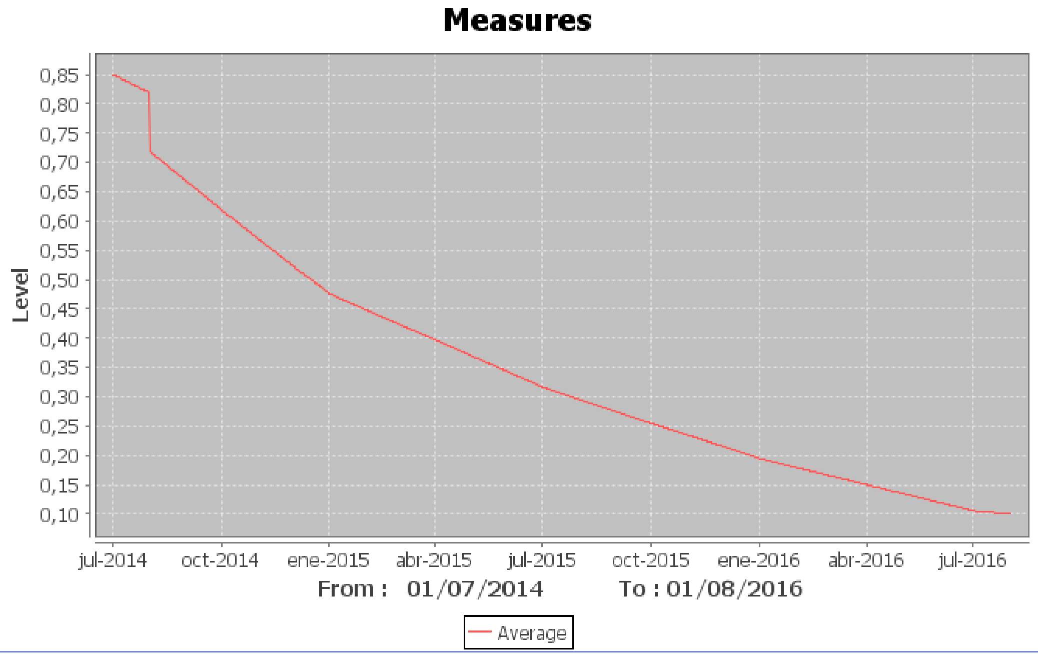


Figure 10.9 – Security curve using complex modeling.

10.7.1 Recommendations

From the analysis, the following recommendations are proposed:

- **Reinforcement**. Ideally it should be made in low activity time periods because it minimizes the impact of technical stops, systems shutdowns and inconvenience to users. Therefore, the obsolete time points have to be as close as possible. In order to achieve it, a realignment is highly recommended.

  In order to reach this state, some reinforcements before the scheduled time (obsolete time point) of some elements have to be performed. After any reinforcement, the knowledge base needs to be update in order to know the next reinforcement time point.

- **Security improvement**. With the information provided, any improvement has to be focused in continuity value. The analysis suggests that this value falls faster than the information value.

- **Backup**. Because of its dependency with the remaining elements, backup reliability and performance have to be checked often. If any unexpected situation occurs, the whole system could be affected.

- **Dependancy**. The high dependence of the whole system in the backup system could be problematic. It's recommended to look for other elements that reduce this strong dependence. For example the incorporation of more redundancy elements in the system could help minimize the impact of problems in the security system.

Part V

Results and Conclusions

11 Conclusions

“You only live once, but if you do it right, once is enough.”

—Mae West

Contents

Main contributions of the research and possible lines for future work are outlined.

An alternative to the traditional concept of "security" is proposed. Security is perceived in a very different manner by individuals, groups or states despite the underlaying notion is the same. Therefore, the proposal explores security as a conceptual object. A methodology for exploring the underlying elements in a concept and the relationships between them is developed and applied to the concept of security.

In order to create the security framework, the starting point was a few works that conceive security notion as a container. Until the elements are not instantiated, there is no specific security. The study, therefore, started with a systematic review of the security concept into the fields that has been working on it. A novel method of conceptual analysis (KBCA) and its extension to include several sources (E-KBCA) is created and the knowledge of the systematic review is extracted by means of this method.

The concept of security, based on the knowledge obtained, is modeled using knowledge engineering, ontology engineering and the principle that it is a containless meaning concept. The result is a flexible and generic security framework called hexagon security model and a definition of security. The framework models an abstract concept and consequently requires specific elements to produce "security". The proposed definition is extremely flexible and can be adapted to any field. Security definition, hence, becomes the specification of the elements that are essential to have *security*.

Finally a proof of concept, a Java® application, is made in order to verify the framework could be used. Additionally, it is applied in a flagship computing institution.

As computer security is an integral part of many social aspects, a purely technical definition is inadequate. Security is a multidimensional concept and computer security has to be part of the existing securities because currently it is not just a technical issue but a social one.

The goals of this research were to model security, to propose an integrative model in order to join as much securities as possible and to provide a definition. Additionally a security metric and a model implementation are obtained.

10.8 Contributions

The following summarizes the main contributions of the thesis, dividing them into two groups. The ones related to methods, models and implementations and the ones related to evaluations, assessments or points of view on security.

1) A conceptual analysis method is created (KBCA I E-KBCA).

2) A generic framework to modeling security is done. Indeed a metamodel with several metrics. As a high level framework, it is not in conflict with other security methods.

3) A methodology for the creation and management of the concept of security.

4) A new security definition is proposed.

5) A formal description of the security concept.

6) A model of perceived security level depending on time and its implementation.

7) A java® implementation of the model using ontologies.

8) Computer security, in this framework, has been integrated within the existing securities.

a) Time is revealed as a key element in modeling Security.

b) Computer security is one more of all possible securities and need to be reviewed because currently is not a just a technical issue but a social one.

c) There are many securities and any change in the security object is in fact a new security specification.

d) The security framework, indeed, represents a higher order model (a meta-model).

10.9 Limitations

The proposed framework is not free of lacks.

- The framework makes a first attempt to integrate security, language and metrics.
- Because of the subjective component of security or new situations outside the security environment, the specification could suffer major changes often. Security is not a static idea but a very dynamic one.
- The subjective component has been reduced as much as possible, but as occurs in all human analysis is hard to remove it completely. Because security is related to the way it is perceived, there is no objective security. Thus, most probably the same scenario analyzed by other persons leads to a slightly different specification.

10.10 Future Work

- This dissertation opens different directions and work in the field of computer security.
- Research on security in a rigorous, systematic and integrated way is just the beginning of a long journey. Further work in this area should extend the security model creating new objects as well as modifying the existing ones.
- There is also a long way to systematize policies, measures and protocols through a software tool that allows a more accurate control and measurement.
- The application prototype could be the core of a bigger system. A graphical front-end will simplify the management of the security object.
- The application prototype reveals that granularity is important, and the outcome of on security could be the input of another, creating a hierarchic security. A graphical application managing all this complexity could be useful.
- The algebraic expression of security opens a workspace for modeling security.

- More research is also needed to discover in which areas this methodology is useful and what changes or improvements would need to adapt to these new scenarios.

- Most security models include knowledge such as vulnerability, risk or assets. Because of this study started from a very different point of view, more work in order to integrate those models with the framework is needed.

- This work does not mean that the concept of security is exhausted. Security is a live concept and changes and surely the model described could be insufficient in a future and requires to be expanded somehow.

Part VI

Appendixes

APPENDIX A

KNOWLEDGE EXTRACTION

Using the KBCA methodology, a concept map and an UML diagram from every source is obtained. This appendix draws the knowledge extraction from the most relevant articles.

National Security as an Ambiguous Symbol by Arnold Wolfers

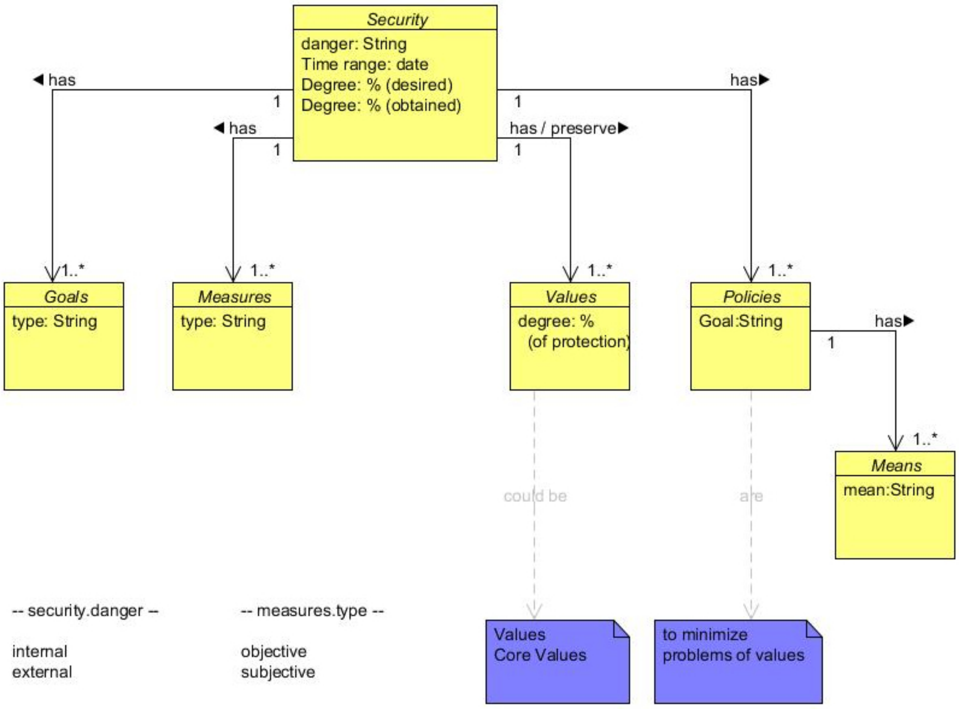


Figure A.1 – UML class diagram.

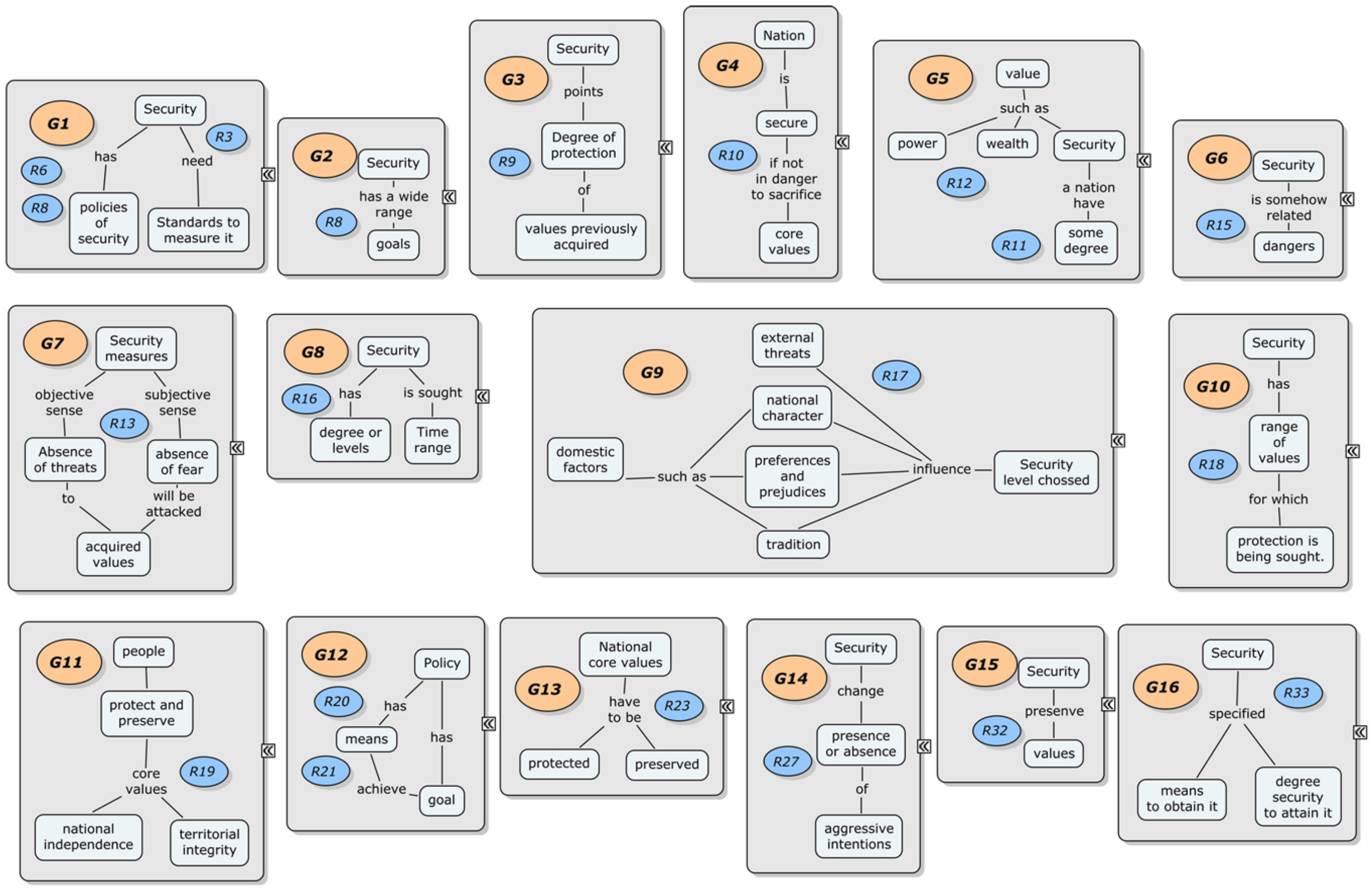
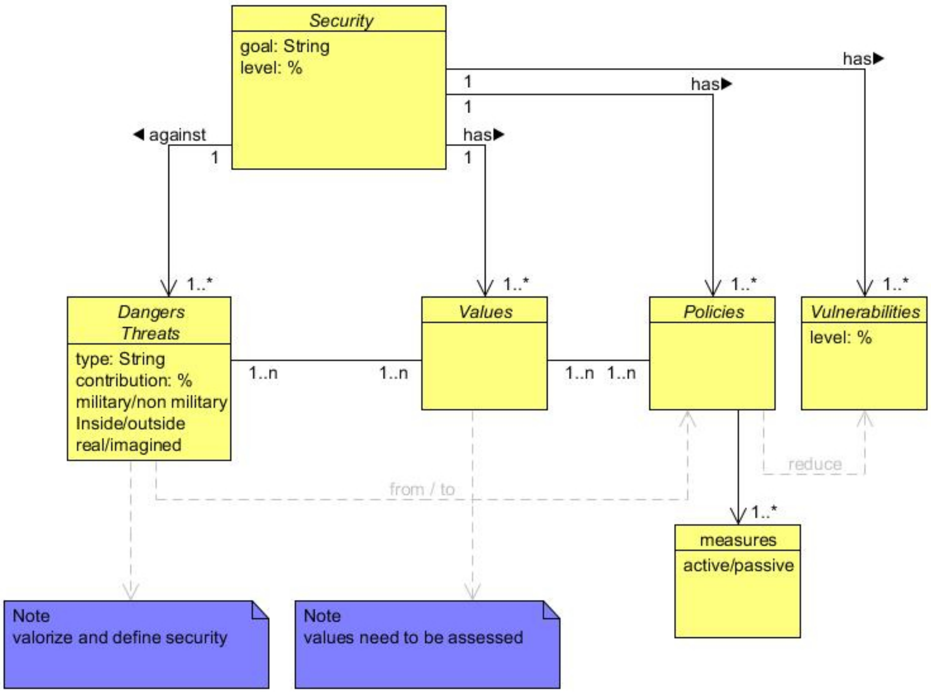


Figure A.2 – Concept map of security concept.

Redefining Security by Richard H. Ullman



-- Threat --

nuclear weapons
external wars
internal rebellions
blockades
raw material shortages
catastrophic natural disasters
...

-- measures --

conservation
alternate energy sources
military
...

-- whom --

Goverment
private entities
non gubernamental entities
persons
groups
corporations

Figure A.3 – UML class diagram.

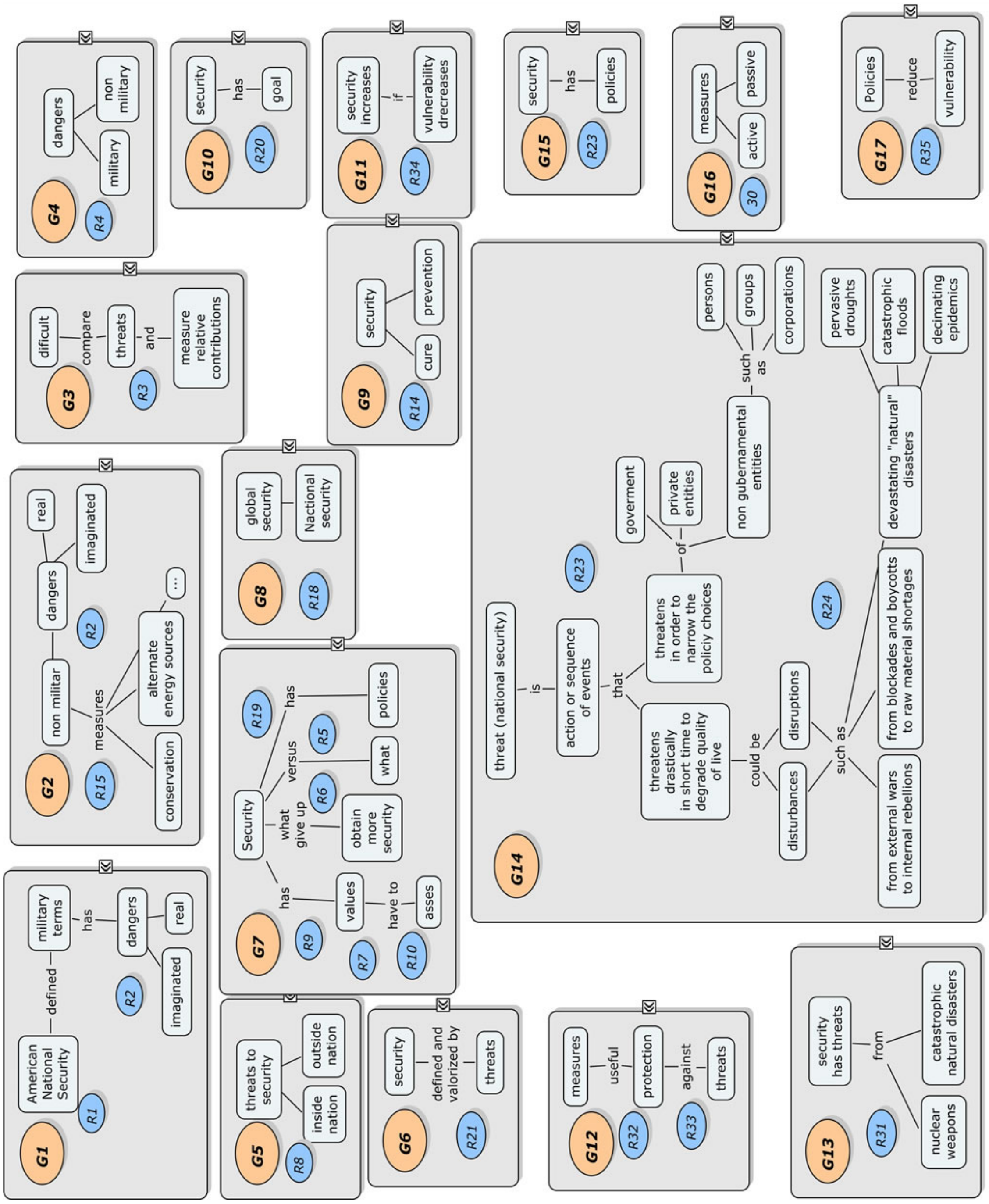


Figure A.4 – Concept map of security concept.

The concept of security by David A. Baldwin

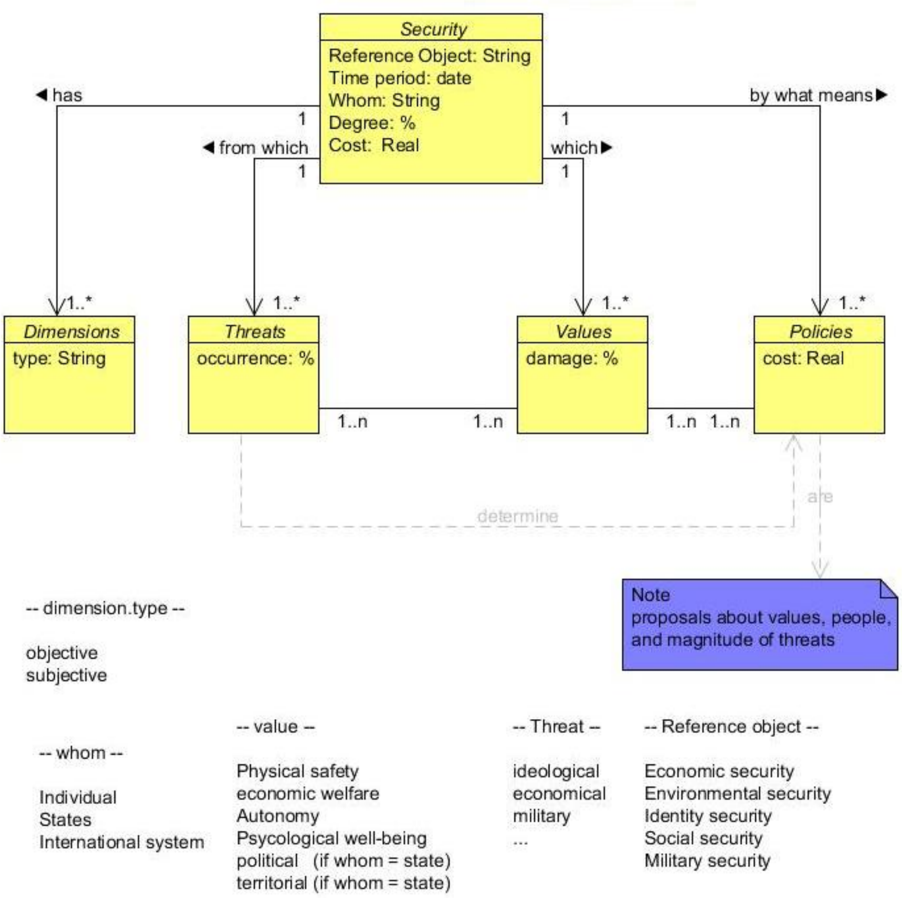


Figure A.5 – UML class diagram.

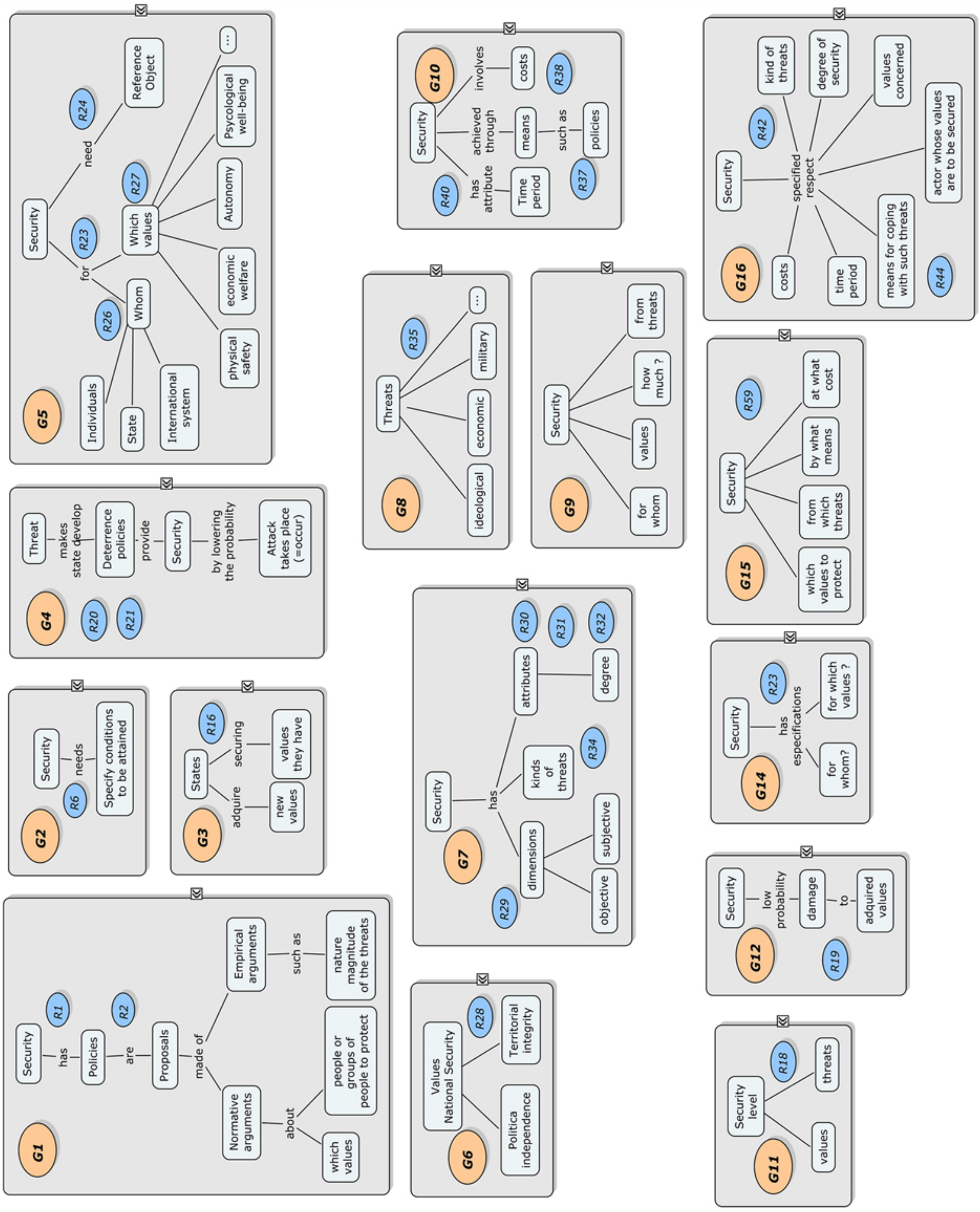
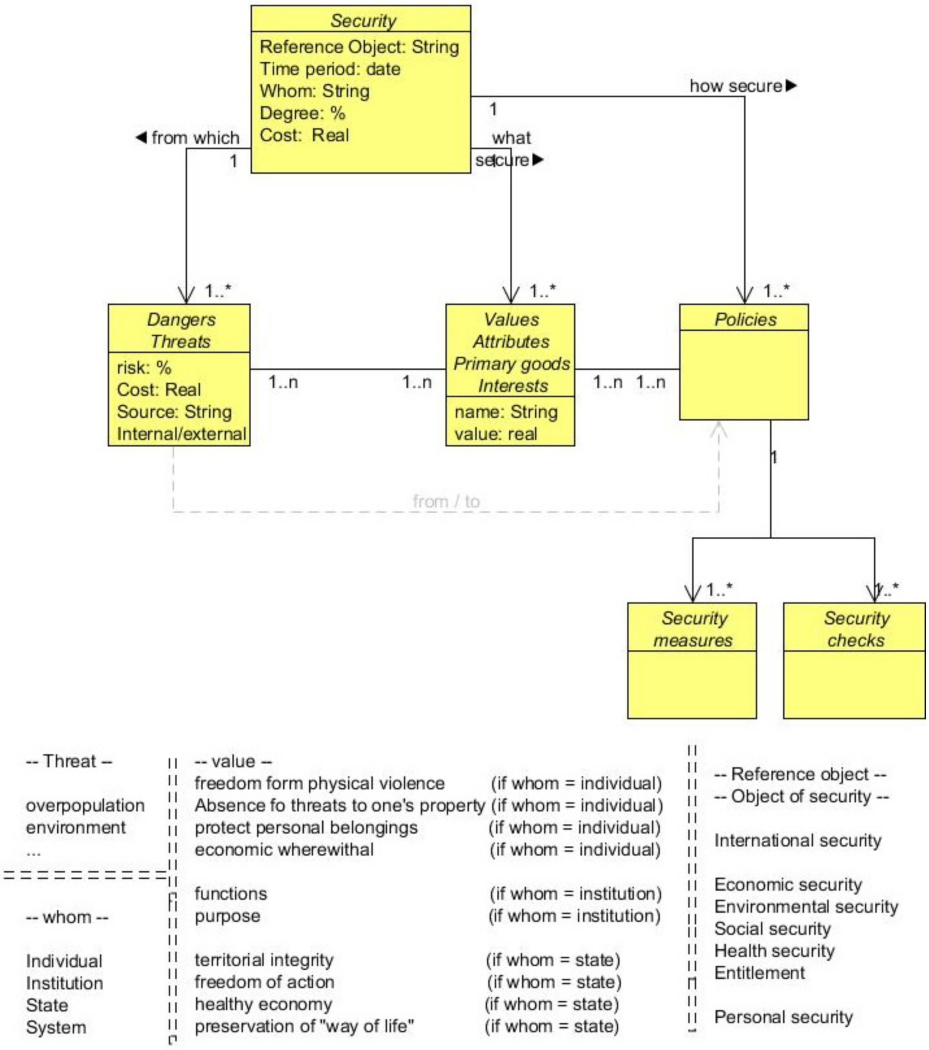


Figure A.6 – Concept map of security concept.

The concept of security by P.E. Digeser



-- Threat --

overpopulation
environment
...

-- whom --

Individual
Institution
State
System

-- value --

freedom form physical violence	(if whom = individual)
Absence fo threats to one's property	(if whom = individual)
protect personal belongings	(if whom = individual)
economic wherewithal	(if whom = individual)
functions	(if whom = institution)
purpose	(if whom = institution)
territorial integrity	(if whom = state)
freedom of action	(if whom = state)
healthy economy	(if whom = state)
preservation of "way of life"	(if whom = state)

-- Reference object --
-- Object of security --

International security

Economic security
Environmental security
Social security
Health security
Entitlement

Personal security

Figure A.7 – UML class diagram.

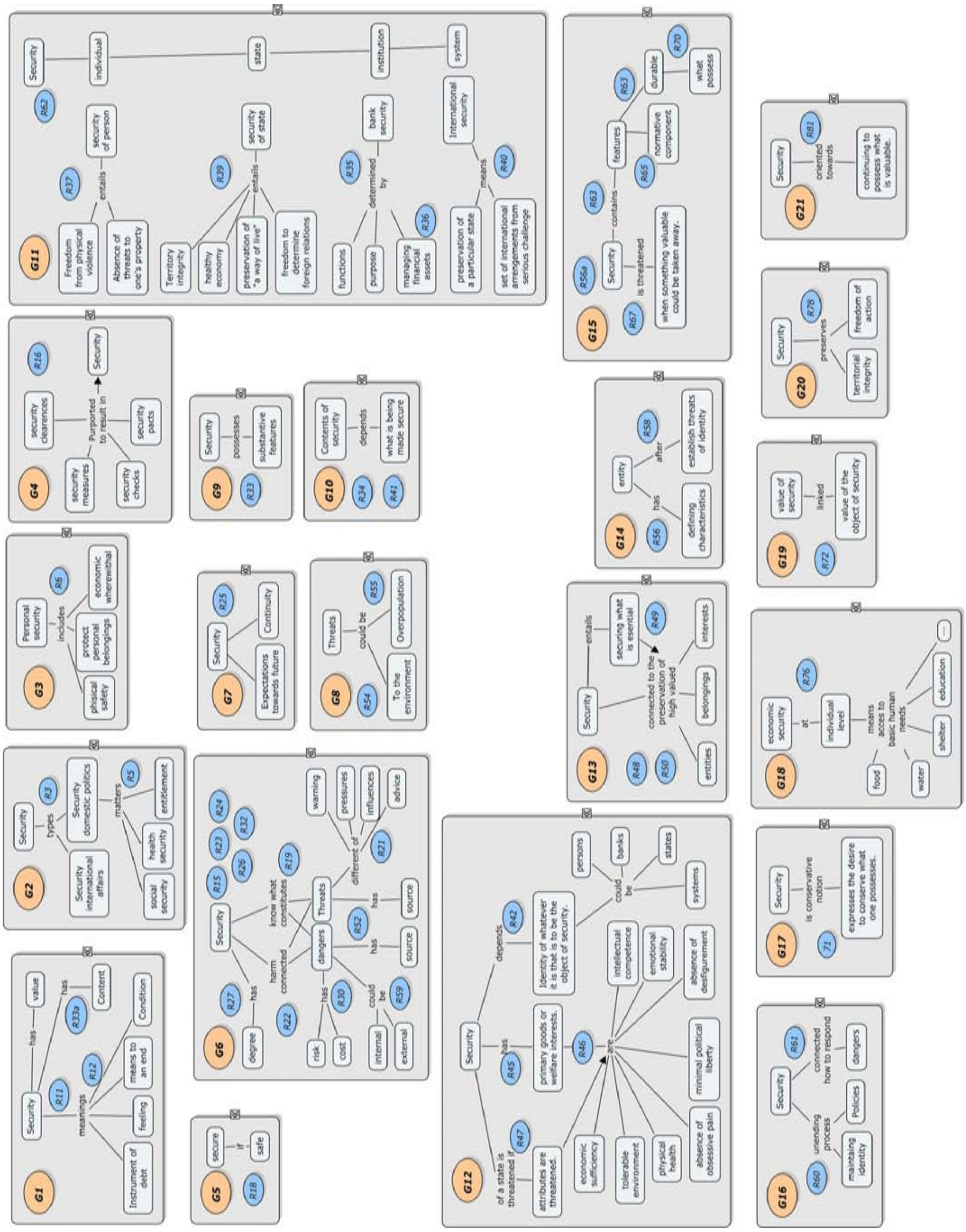


Figure A.8 – Concept map of security concept.

Security as an Analitycal Concept by Czeslaw Mesjasz

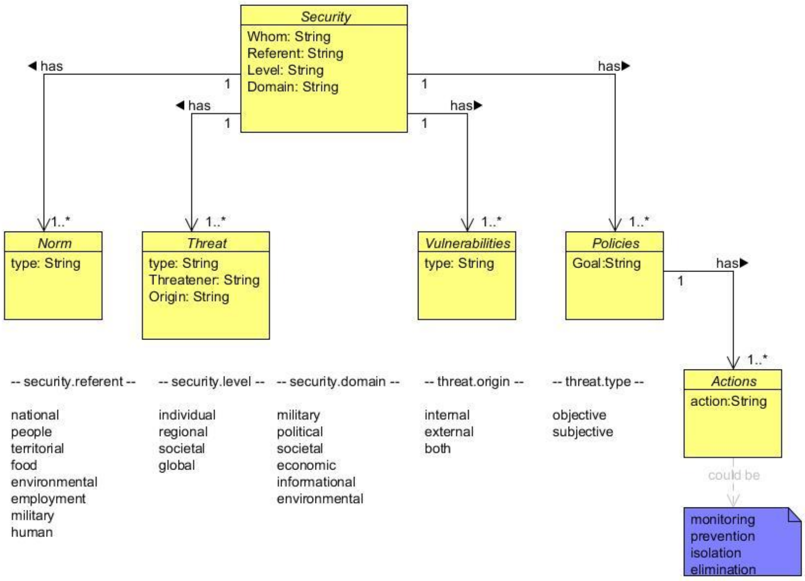


Figure A.9 – UML class diagram.

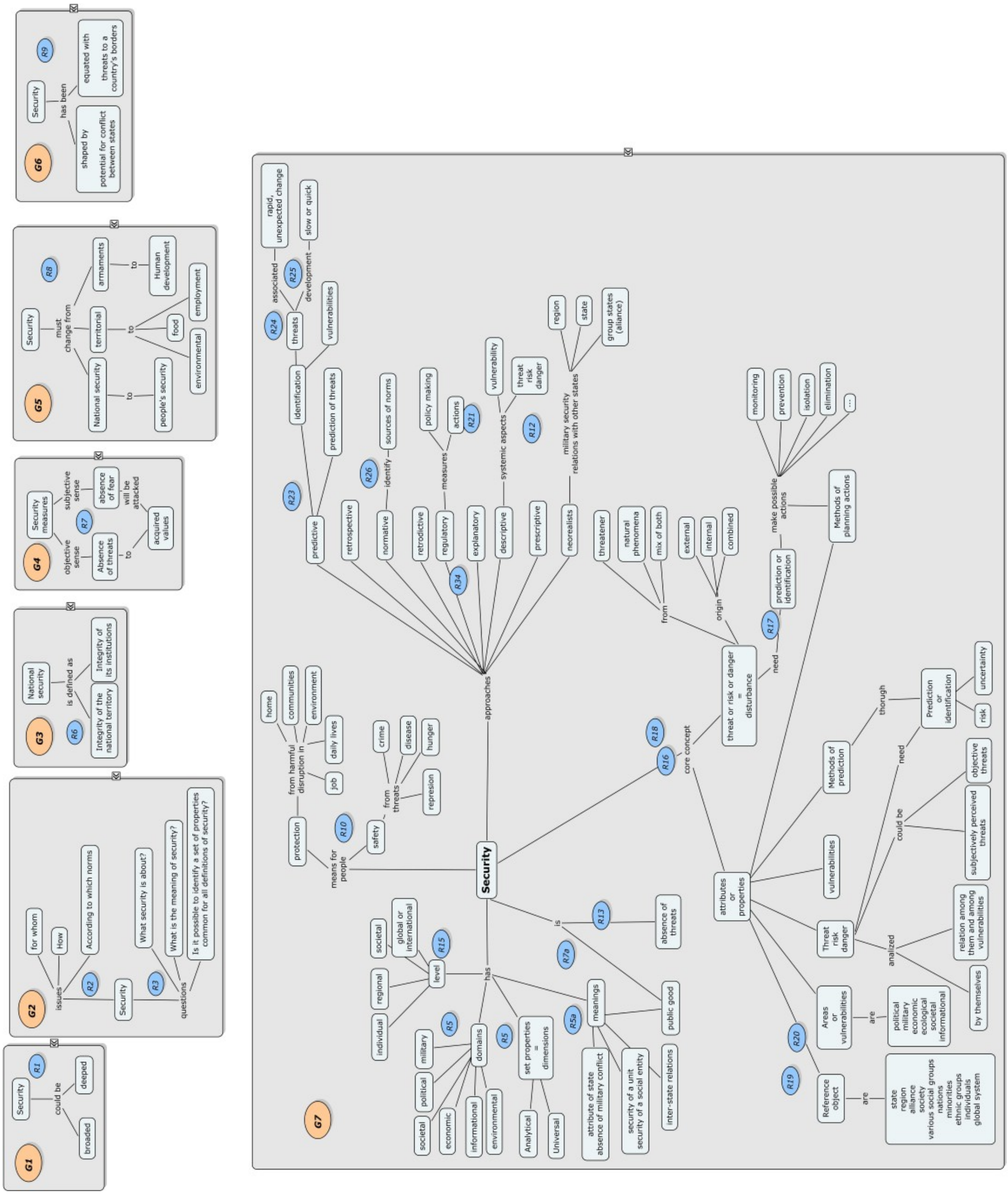


Figure A.10 – Concept map of security concept.

What Is Security by E. Rothschild

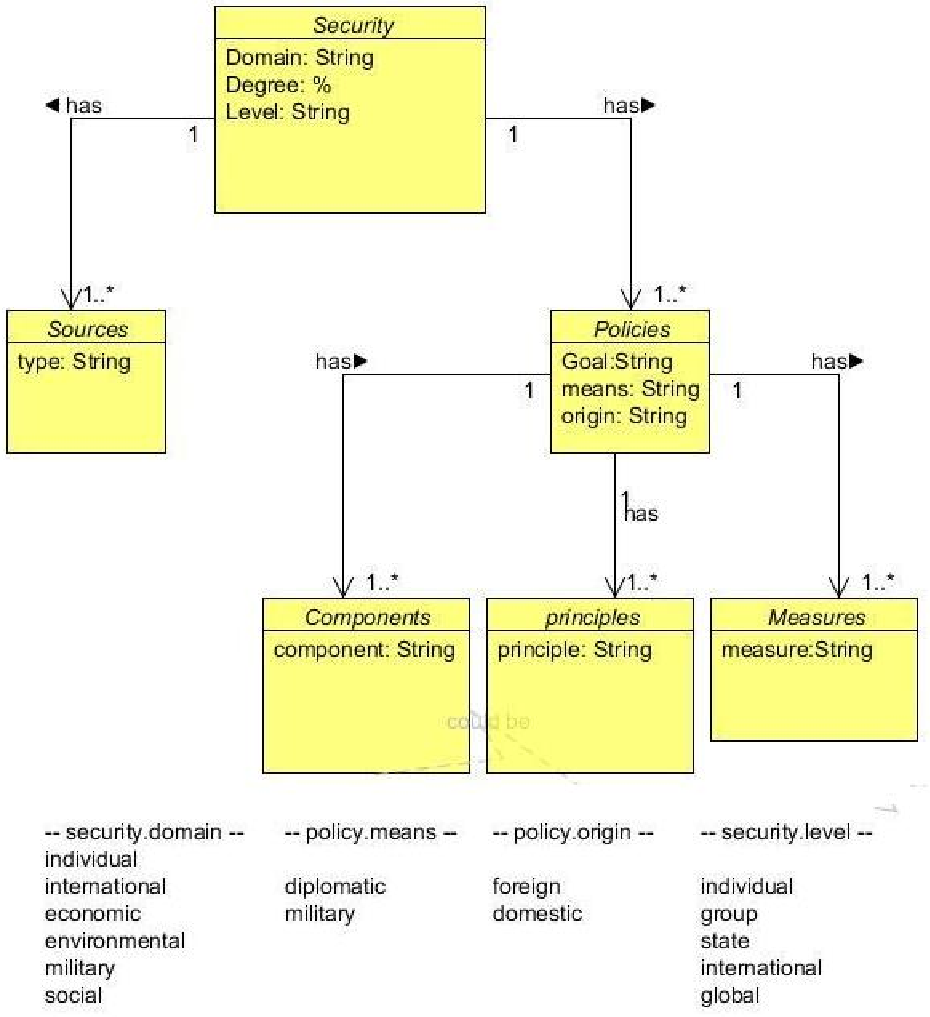


Figure A.11 – UML class diagram.

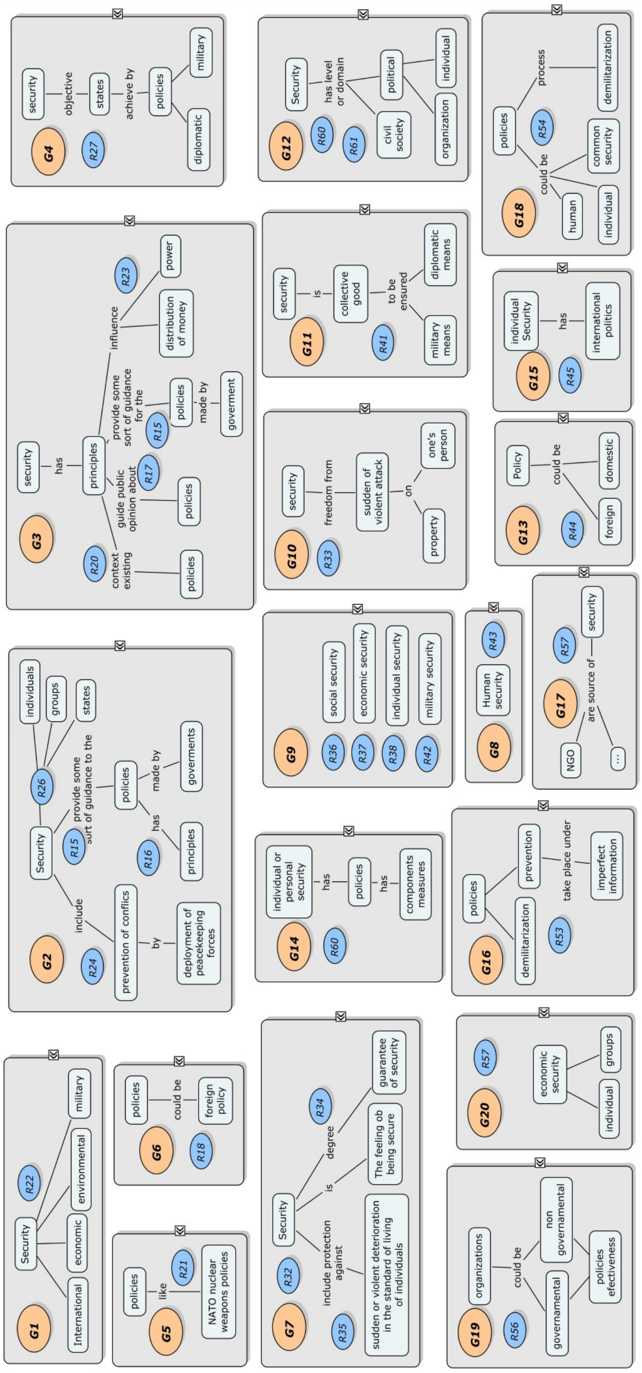


Figure A.12 – Concept map of security concept.

APPENDIX B

ALGEBRAIC EXAMPLE

In order to create a security from the framework model, we have to follow several steps (Table B.1).

Step	Define	Populate	Infer	
Concepts				
1	Context			
2	Values			$\mathbb{V} = \{v_1, v_2, ..., v_n\}$
3	Threats			$\mathbb{T} = \{t_1, t_2, ..., t_n\}$
4	Policies			$\mathbb{I} = \{i_1, i_2, ..., i_n\}$
5	Providers			$\mathbb{P} = \{p_1, p_2, ..., p_n\}$
6	Measures			$\mathbb{M} = \{m_1, m_2, ..., m_n\}$
7	Resources			$\mathbb{S} = \{s_1, s_2, ..., s_n\}$
Primary relations				
8		Threats of Values		VxT
9		Policies of Threats		IxT
10		Measures of Policies		IxM
11		Resources of Measures		MxS
12		Policies of Threats		PxT
Inferred relations				
13			VxI	$VxT * TxI$
14			PxI	$PxT * TxI$

15			VxM	$VxI * IxM = VxT * TxI * IxM$
16			VxP	$VxT * TxP$
17			TxM	$TxI * IxM$
18			SxI	$SxM * MxI$
19			TxS	$TxI * IxS = TxI * IxM * MxS$
20			MxP	$MxT * TxP = TxI * IxM * TxP$
21			SxP	$SxT * TxP = TxI * IxS * TxP$
22			VxS	$VxT * TxS = VxT * TxI * IxS$
23		*Add cost to resources*		
24		*Add time points to measures*		
25	*Define threshold values*			
26			*Infer security functions*	

Table B.1 – Methodology.

B.1 Define concepts and sets

Security $\mathbb{C}$ is defined as a tuple $(\mathbb{V}, \mathbb{T}, \mathbb{P}, \mathbb{I}, \mathbb{M}, \mathbb{S})$, where the sets have the following elements:

- $\mathbb{V} \quad is\ a\ finite\ set\ (of\ value\ names) \quad \mathbb{V} = \{v_1, v_2\ \}$
- $\mathbb{T} \quad is\ a\ finite\ set\ (of\ threat\ names) \quad \mathbb{T} = \{t_1, t_2, t_3, t_4\ \}$
- $\mathbb{P} \quad is\ a\ finite\ set\ (of\ provider\ names) \quad \mathbb{P} = \{p_1, p_2\ \}$
- $\mathbb{I} \quad is\ a\ finite\ set\ (of\ policy\ names) \quad \mathbb{I} = \{i_1, i_2, i_3, i_4, i_5\ \}$
- $\mathbb{M} \quad is\ a\ finite\ set\ (of\ measure\ names) \quad \mathbb{M} = \{m_1, m_2, m_3, m_4, m_5\ \}$
- $\mathbb{S} \quad is\ a\ finite\ set\ (of\ resource\ names) \quad \mathbb{S} = \{s_1, s_2, s_3, s_4, s_5, s_6\ \}$

B.2 Define primary relations

Primary relations are shown in Tables B.2,B.3,B.4,B.5,B.6 :

VT	t_1	t_2	t_3	t_4
v_1	✓	✓		✓
v_2		✓	✓	

Table B.2 – Value / Threat relation.

IT	t_1	t_2	t_3	t_4
i_1	✓	✓		
i_2		✓		
i_3	✓			✓
i_4			✓	
i_5	✓			

Table B.3 – Policy / Threat relation.

IM	m_1	m_2	m_3	m_4	m_5
i_1	✓				
i_2		✓			
i_3	✓				✓
i_4			✓		
i_5				✓	

Table B.4 – Policy / Measures relation.

MS	m_1	m_2	m_3	m_4	m_5
s_1	✓	✓			
s_2			✓		
s_3			✓		
s_4				✓	
s_5		✓			✓
s_6	✓				

Table B.5 – Measures / Resources relation.

PT	t_1	t_2	t_3	t_4
p_1	✓			✓
p_2		✓	✓	

Table B.6 – Provider / Threat relation.

B.3 Deduce inferred relations

The inferred relations are shown in Tables B.7, B.8, B.9, B.10, B.11, B.12, B.13, B.14, B.15, B.16.

VI	i_1	i_2	i_3	i_4	i_5
v_1	✓	✓	✓		✓
v_2	✓	✓		✓	

Table B.7 – Values /Policy relation.

PI	i_1	i_2	i_3	i_4	i_5
p_1	✓		✓		✓
p_2	✓	✓		✓	

Table B.8 – Providers /Policy relation.

VM	m_1	m_2	m_3	m_4	m_5
v_1	✓	✓		✓	✓
v_2	✓	✓	✓		

Table B.9 – Values /Measures.

VP	p_1	p_2
v_1	✓	✓
v_2		✓

Table B.10 – Values / Providers.

TM	m_1	m_2	m_3	m_4	m_5
t_1	✓			✓	✓
t_2	✓	✓			
t_3			✓		
t_4	✓				✓

Table B.11 – Threats /Measures.

SI	i_1	i_2	i_3	i_4	i_5
s_1	✓	✓	✓		
s_2				✓	
s_3				✓	
s_4					✓
s_5		✓	✓		
s_6	✓		✓		

Table B.12 – Resources /Policies.

TS	t_1	t_2	t_3	t_4
s_1	✓	✓		✓
s_2			✓	
s_3			✓	
s_4	✓			
s_5	✓	✓		✓
s_6	✓	✓		✓

Table B.13 – Threats / Resources.

MP	m_1	m_2	m_3	m_4	m_5
p_1	✓			✓	✓
p_2	✓	✓	✓		

Table B.14 – Policies /Measures.

SP	s_1	s_2	s_3	s_4	s_5	s_6
p_1	✓			✓	✓	✓
p_2	✓	✓	✓		✓	✓

Table B.15 – Resources /Policies.

VS	s_1	s_2	s_3	s_4	s_5	s_6
v_1	✓			✓	✓	✓
v_2	✓	✓	✓		✓	✓

Table B.16 – Resources /Values.

B.4 Add cost to resources

Giving a cost to resources (Table B.18), the associated cost to all elements are drawn in Table B.17

B.5 Add time points to measures

Giving time points to measures (Table B.20) and defining the threshold values (Table B.19), the security curves could be deduced. Table B.21 shows result based on the simple security function model.

	Element	Cost
$Values$		
	v_1	24.0
	v_2	24.0
$Threats$		
	t_1	24.0
	t_2	19.0
	t_3	5.0
	t_4	19.0
$Providers$		
	p_1	24.0
	p_2	24.0
$Policies$		
	i_1	12.0
	i_2	8.0
	i_3	19.0
	i_4	5.0
	i_5	5.0
$Measures$		
	m_1	12.0
	m_2	8.0
	m_3	5.0
	m_4	5.0
	m_5	7.0
$Resources$		
	$Introduced$	

Table B.17 – Associated cost of resources.

Cost	
s_1	1.0
s_2	2.0
s_3	3.0
s_4	5.0
s_5	7.0
s_6	11.0

Table B.18 – Associated cost of resources.

	$Secure$	$Insecure$	$Obsolete$
	0.75	0.2	0.0

Table B.19 – Threshold values.

IM	$Secure$	$Insecure$	$Obsolete$
m_1	01/01/2014	01/02/2014	01/03/2014
m_2	01/02/2014	01/03/2014	01/04/2014
m_3	01/03/2014	01/04/2014	01/05/2014
m_4	01/04/2014	01/05/2014	01/06/2014
m_5	01/05/2014	01/06/2014	01/07/2014

Table B.20 – Time points.

IM	*Secure*	*Insecure*	*Obsolet*
Security	31/01/2014	03/06/2014	01/07/2014
Values			
v_1	31/01/2014	02/06/2014	01/07/2014
v_2	31/01/2014	02/04/2014	01/05/2014
Threats			
t_1	31/01/2014	02/06/2014	01/07/2014
t_2	31/01/2014	02/03/2014	01/04/2014
t_3	01/03/2014	02/04/2014	01/05/2014
t_4	31/01/2014	02/06/2014	01/07/2014
Providers			
p_1	31/01/2014	02/06/2014	01/07/2014
p_2	31/01/2014	02/06/2014	01/05/2014
Policies			
i_1	01/01/2014	01/02/2014	01/03/2014
i_2	01/02/2014	01/03/2014	01/04/2014
i_3	31/01/2014	02/06/2014	01/07/2014
i_4	01/03/2014	02/04/2014	01/05/2014
i_5	01/04/2014	01/05/2014	01/06/2014
Measures			Introduced

Table B.21 – Time Curves.

APPENDIX C

HOME SECURITY EXAMPLE

Suppose we are interested in defining and improving our home security. In order to create a security from the framework model, we have to follow several steps (Table C.1).

Step	**Define**	**Populate**	**Infer**	
Concepts				
1	Context			
2	Values			$\mathbb{V} = \{v_1, v_2, ..., v_{n_1}\}$
3	Threats			$\mathbb{T} = \{t_1, t_2, ..., t_{n_2}\}$
4	Policies			$\mathbb{I} = \{i_1, i_2, ..., i_{n_3}\}$
5	Providers			$\mathbb{P} = \{p_1, p_2, ..., p_{n_4}\}$
6	Measures			$\mathbb{M} = \{m_1, m_2, ..., m_{n_5}\}$
7	Resources			$\mathbb{S} = \{s_1, s_2, ..., s_{n_6}\}$
Primary relations				
8		Threats of Values		VxT
9		Policies of Threats		IxT
10		Measures of Policies		IxM
11		Resources of Measures		MxS
12		Policies of Threats		PxT
Inferred relations				
13			VxI	$VxT * TxI$

14			PxI	$PxT * TxI$
15			VxM	$VxI * IxM = VxT * TxI * IxM$
16			VxP	$VxT * TxP$
17			TxM	$TxI * IxM$
18			SxI	$SxM * MxI$
19			TxS	$TxI * IxS = TxI * IxM * MxS$
20			MxP	$MxT * TxP = TxI * IxM * TxP$
21			SxP	$SxT * TxP = TxI * IxS * TxP$
22			VxS	$VxT * TxS = VxT * TxI * IxS$
23		*Add cost to resources*		
24		*Add time points to measures*		
25	*Define threshold values*			
26			*Infer security functions*	

Table C.1 – Methodology.

C.1 Define elements

To identify concepts, sets and primary relations, a list of what and how to protect is made (Tables C.2,C.3,C.4 and C.5).

$i1$	$Internet\ \ privacy$ $By\ \ means\ \ of:$
	Antivirus. Renew each year Firewall Automatic Updates

Table C.2 – Internet privacy.

The relation between the policies and the rest of elements could be seen in Figures C.1 , C.2 , C.3 and C.4.

C.2 Define concepts and sets

For simplicity, the names are labeled. This makes the table representation easier. In this scenario, security $\mathbb{C}$ is defined as a tuple $(\mathbb{V}, \mathbb{T}, \mathbb{P}, \mathbb{I}, \mathbb{M}, \mathbb{S})$, where the sets

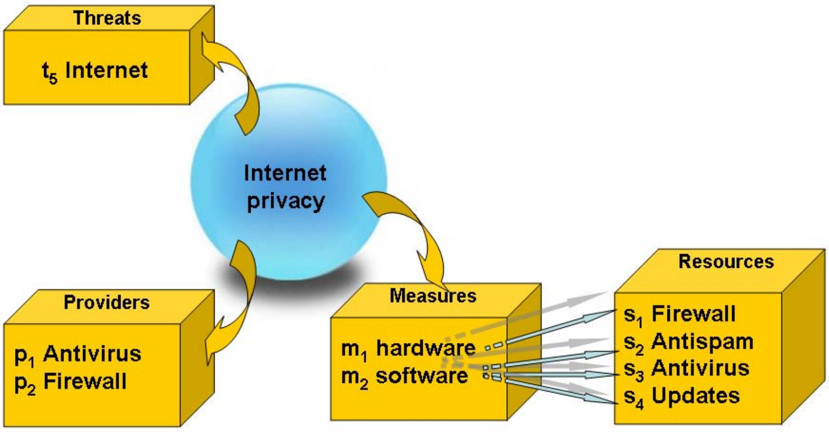


Figure C.1 – Elements in policy i_1.

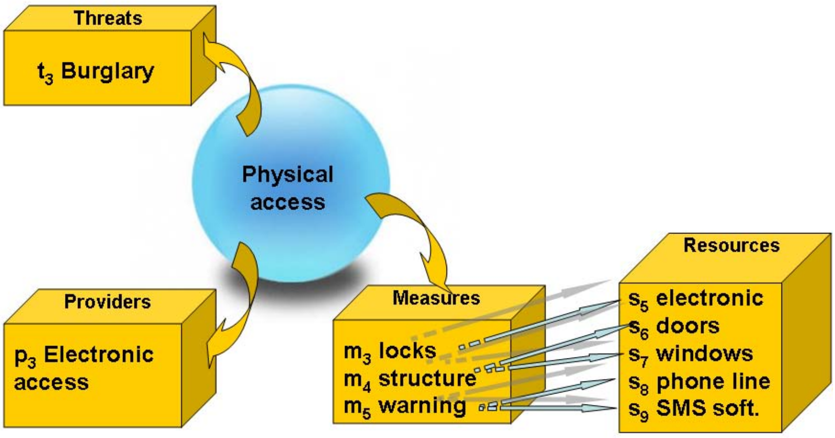


Figure C.2 – Elements in policy i_2.

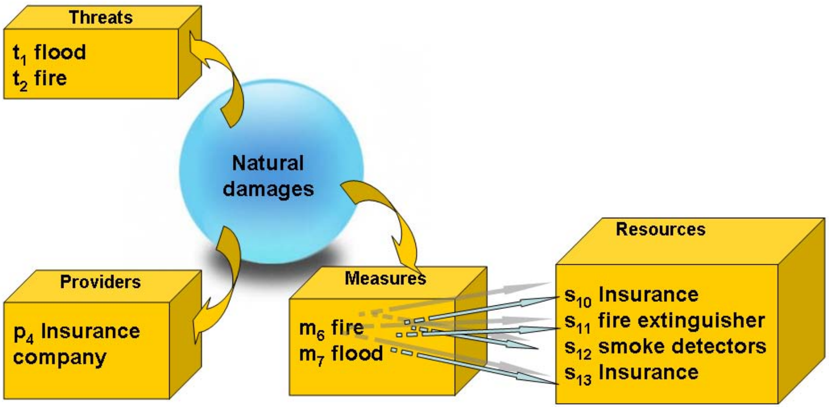


Figure C.3 – Elements in policy i_3.

$i2$	$Physical\ \ access$ $By\ \ means\ \ of:$
	Keys with high difficulty to copy Doors reinforcement Windows reinforcement Notify owners through SMS

Table C.3 – Internet privacy.

$i3$	$Natural\ \ damages$ $By\ \ means\ \ of:$	
	flood	
		Insurance
	fire	
		Fire detectors Fire extinguisher Insurance

Table C.4 – Internet privacy.

$i4$	$Blackout$ $By\ \ means\ \ of:$
	Emergency lights Power generator SMS warning to owner

Table C.5 – Internet privacy.

have the following elements:

- $\mathbb{C}\ \ is\ defined\ as\ \ \ \mathbb{C} = \{HomeSecurity\}$
- $\mathbb{V}\ \ is\ a\ finite\ set\ (of\ value\ names)\ \ \ \mathbb{V} = \{v_1 : security, v_2 : privacy\ \}$
- $\mathbb{T}\ \ is\ a\ finite\ set\ (of\ threat\ names)\ \ \ \mathbb{T} = \{t_1 : flood, t_2 : fire, t_3 : burglary, t_4 : electricalfailure, t_5 : Internet\ \}$
- $\mathbb{I}\ \ is\ a\ finite\ set\ (of\ policy\ names)\ \ \ \mathbb{I} = \{i_1 : Internet\ \ privacy, i_2 : physical\ \ acces, i_3 : natural\ \ damages, i_4 : blackout\ \}$
- $\mathbb{P}\ \ is\ a\ finite\ set\ (of\ provider\ names)\ \ \ \mathbb{P} = \{\ p_1 : antivirus, p_2 : firewall, p_3 : access\ \ control, p_4 : insurance\ \ company, p_5 : emergency\ \ system, p_6 :$

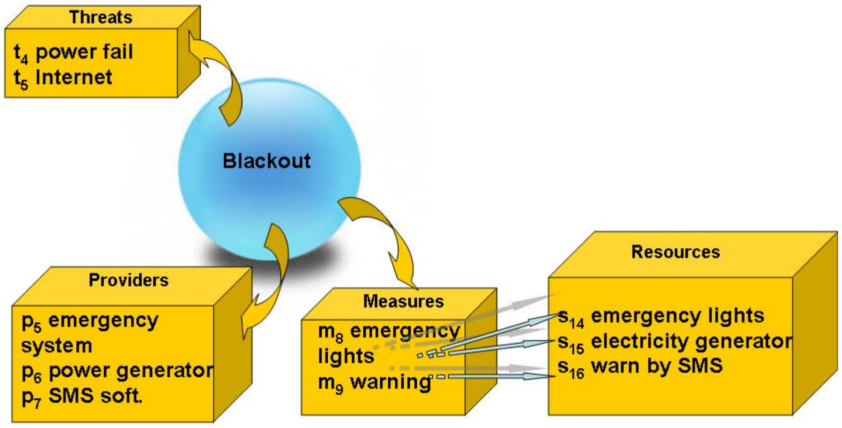


Figure C.4 – Elements in policy i_4.

$generator, p_7 : SMS\ \ software\,\}$

- $\mathbb{M}\quad is\ a\ finite\ set\ (of\ measure\ names)\qquad \mathbb{M} = \{m_1 : hardware, m_2 : software, m_3 : locks, m_4 : structure, m_5 : warning, m_6 : fire, m_7 : flood, m_8 : emergency\ \ lights, m_9 : sms\,\}$
- $\mathbb{S}\quad is\ a\ finite\ set\ (of\ resource\ names)\qquad \mathbb{S} = \{s_1 : firewall, s_2 : antispam, s_3 : antivirus, s_4 : updates, s_5 : electronic\ \ system, s_6 : doors, s_7 : windows, s_8 : phone\ \ line, s_9 : SMS\ \ software, s_{10} : fire\ \ insurance, s_{11} : fire\ \ extinguisher, s_{12} : fumes\ \ detectors, s_{13} : flood\ \ insurance, s_{14} : emergency\ \ lights, s_{15} : generator, s_{16} : SMS\,\}$

C.3 Define primary relations

Primary relations are shown in Tables C.6,C.7,C.8,C.9,C.10 :

VT	t_1	t_2	t_3	t_4	t_5
v_1	✓	✓	✓	✓	
v_2			✓		✓

Table C.6 – Value / Threat relation.

IT	i_1	i_2	i_3	i_4
t_1			✓	
t_2			✓	
t_3		✓		
t_4				✓
t_5	✓			✓

Table C.7 – Policy / Threat relation.

It's easy to construct the $I \times P$ table. From this, by means of the operation $P \times T = T \times I \circ P \times I$, the primary relation could be deduced.

IM	m_1	m_2	m_3	m_4	m_5	m_6	m_7	m_8	m_9
i_1	✓	✓							
i_2			✓	✓	✓				
i_3						✓	✓		
i_4								✓	✓

Table C.8 – Policy / Measures relation.

MS	s_1	s_2	s_3	s_4	s_5	s_6	s_7	s_8	s_9	s_{10}	s_{11}	s_{12}	s_{13}	s_{14}	s_{15}	s_{16}
m_1	✓															
m_2		✓	✓	✓												
m_3					✓											
m_4						✓	✓									
m_5								✓	✓							
m_6										✓	✓	✓				
m_7													✓			
m_8														✓	✓	
m_9																✓

Table C.9 – Measures / Resources relation.

PT	t_1	t_2	t_3	t_4	t_5
p_1					✓
p_2					✓
p_3			✓		
p_4	✓	✓			
p_5				✓	✓
p_6				✓	✓
p_7				✓	✓

Table C.10 – Provider / Threat relation.

C.4 Deduce inferred relations

The inferred relations are shown in Tables C.11, C.12, C.13, C.14, C.15, C.16, C.17, C.18, C.19, C.20.

VI	i_1	i_2	i_3	i_4
v_1		✓	✓	✓
v_2	✓	✓		✓

Table C.11 – Values /Policy relation.

PI	i_1	i_2	i_3	i_4
p_1	✓			✓
p_2	✓			✓
p_3		✓		
p_4			✓	
p_5	✓			✓
p_6	✓			✓
p_7	✓			✓

Table C.12 – Providers /Policy relation.

VM	m_1	m_2	m_3	m_4	m_5	m_6	m_7	m_8	m_9
v_1			✓	✓	✓	✓	✓	✓	✓
v_2	✓	✓	✓	✓	✓			✓	✓

Table C.13 – Values /Measures.

VP	p_1	p_2	p_3	p_4	p_5	p_6	p_7
v_1			✓	✓	✓	✓	✓
v_2	✓	✓	✓		✓	✓	✓

Table C.14 – Values / Providers.

TM	m_1	m_2	m_3	m_4	m_5	m_6	m_7	m_8	m_9
t_1						✓	✓		
t_2						✓	✓		
t_3			✓	✓	✓				
t_4								✓	✓
t_5	✓	✓						✓	✓

Table C.15 – Threats /Measures.

C.5 Add cost to resources

Giving a cost to resources (Table C.22), the associated cost to all elements is shown in Table C.21

SI	i_1	i_2	i_3	i_4
s_1	✓			
s_2	✓			
s_3	✓			
s_4	✓			
s_5		✓		
s_6		✓		
s_7		✓		
s_8		✓		
s_9		✓		
s_{10}			✓	
s_{11}			✓	
s_{12}			✓	
s_{13}			✓	
s_{14}				✓
s_{15}				✓
s_{16}				✓

Table C.16 – Resources /Policies.

TS	t_1	t_2	t_3	t_4	t_5
s_1					✓
s_2					✓
s_3					✓
s_4					✓
s_5			✓		
s_6			✓		
s_7			✓		
s_8			✓		
s_9			✓		
s_{10}	✓	✓			
s_{11}	✓	✓			
s_{12}	✓	✓			
s_{13}	✓	✓			
s_{14}			✓	✓	
s_{15}			✓	✓	
s_{16}			✓	✓	

Table C.17 – Threats / Resources.

MP	p_1	p_2	p_3	p_4	p_5	p_6	p_7
m_1	✓	✓			✓	✓	✓
m_2	✓	✓			✓	✓	✓
m_3			✓				
m_4			✓				
m_5			✓				
m_6				✓			
m_7				✓			
m_8	✓	✓			✓	✓	✓
m_9	✓	✓			✓	✓	✓

Table C.18 – Policies /Measures.

C.6 Add time points to measures

Giving time points to measures (Table C.24) and defining the threshold values (Table C.23), the security curves could be deduced. Table C.25 shows result based on the simple security function model.

SP	p_1	p_2	p_3	p_4	p_5	p_6	p_7
s_1	✓	✓			✓	✓	✓
s_2	✓	✓			✓	✓	✓
s_3	✓	✓			✓	✓	✓
s_4	✓	✓			✓	✓	✓
s_5			✓				
s_5			✓				
s_6			✓				
s_7			✓				
s_8			✓				
s_9			✓				
s_{10}				✓			
s_{11}				✓			
s_{12}				✓			
s_{13}				✓			
s_{14}	✓	✓			✓	✓	✓
s_{15}	✓	✓			✓	✓	✓
s_{16}	✓	✓			✓	✓	✓

Table C.19 – Resources /Policies.

VS	s_1	s_2	s_3	s_4	s_5	s_6	s_7	s_8	s_9	s_{10}	s_{11}	s_{12}	s_{13}	s_{14}	s_{15}	s_{16}
v_1					✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
v_2	✓	✓	✓	✓	✓	✓	✓	✓	✓					✓	✓	✓

Table C.20 – Resources /Values.

	Element	Cost
$Values$		
	v_1	318.0
	v_2	209.0
$Threats$		
	t_1	120.0
	t_2	120.0
	t_3	67.0
	t_4	131.0
	t_5	142.0
$Providers$		
	p_1	142.0
	p_2	142.0
	p_3	67.0
	p_4	120.0
	p_5	142.0
	p_6	142.0
	p_7	142.0
$Policies$		
	i_1	11.0
	i_2	67.0
	i_3	120.0
	i_4	131.0
$Measures$		
	m_1	1.0
	m_2	10.0
	m_3	7.0
	m_4	24.0
	m_5	36.0
	m_6	83.0
	m_7	37.0
	m_8	84.0
	m_9	47.0
$Resources$		
	$Introduced$	

Table C.21 – Associated cost of resources.

Cost	
s_1	1.0
s_2	2.0
s_3	3.0
s_4	5.0
s_5	7.0
s_6	11.0
s_7	13.0
s_8	17.0
s_9	19.0
s_{10}	23.0
s_{11}	29.0
s_{12}	31.0
s_{13}	37.0
s_{14}	41.0
s_{15}	43.0
s_{16}	47.0

Table C.22 – Associated cost of resources.

	$Secure$	$Insecure$	$Obsolete$
	0.75	0.2	0.0

Table C.23 – Threshold values.

IM	$Secure$	$Insecure$	$Obsolete$
m_1	01/01/2014	01/02/2014	01/03/2014
m_2	01/02/2014	01/03/2014	01/04/2014
m_3	01/03/2014	01/04/2014	01/05/2014
m_4	01/04/2014	01/05/2014	01/06/2014
m_5	01/05/2014	01/06/2014	01/07/2014
m_6	01/06/2014	01/07/2014	01/08/2014
m_7	01/07/2014	01/08/2014	01/09/2014
m_8	01/08/2014	01/09/2014	01/10/2014
m_9	01/09/2014	01/10/2014	01/11/2014

Table C.24 – Time points.

IM	$Secure$	$Insecure$	$Obsolet$
$Security$	29/04/2014	01/09/2014	01/11/2014
$Values$			
v_1	30/04/2014	01/09/2014	01/11/2014
v_2	28/02/2014	01/09/2014	01/11/2014
$Threats$			
t_1	30/06/2014	01/08/2014	01/09/2014
t_2	30/06/2014	01/08/2014	01/09/2014
t_3	31/03/2014	02/06/2014	01/07/2014
t_4	31/08/2014	01/10/2014	01/11/2014
t_5	31/01/2014	01/10/2014	01/11/2014
$Providers$			
p_1	31/01/2014	01/10/2014	01/11/2014
p_2	31/01/2014	01/10/2014	01/11/2014
p_3	31/03/2014	02/06/2014	01/07/2014
p_4	30/06/2014	01/08/2014	01/09/2014
p_5	31/01/2014	01/10/2014	01/11/2014
p_6	31/01/2014	01/10/2014	01/11/2014
p_7	31/01/2014	01/10/2014	01/11/2014
$Policies$			
i_1	31/01/2014	02/03/2014	01/04/2014
i_2	31/03/2014	02/06/2014	01/07/2014
i_3	30/06/2014	01/08/2014	01/09/2014
i_4	31/08/2014	01/10/2014	01/11/2014
$Measures$			Introduced

Table C.25 – Time Curves.

APPENDIX D

KBCA METHOD

Knowledge Based Concept Analysis Method using Concept Maps and UML: Security Notion Case

Miquel Colobran, and Josep M. Basart

***Abstract*—**One of the most ancient humankind concerns is knowledge formalization i.e. what a concept is. Concept Analysis, a branch of analytical philosophy, relies on the purpose of decompose the elements, relations and meanings of a concept. This paper aims at presenting a method to make a concept analysis obtaining a knowledge representation suitable to be processed by a computer system using either object-oriented or ontology technologies. Security notion is, usually, known as a set of different concepts related to "some kind of protection". Our method concludes that a more general framework for the concept, despite it is dynamic, is possible and any particular definition (instantiation) depends on the elements used by its construction instead of the concept itself.

***Keywords*—**Concept analysis, Knowledge representation, Security, UML.

I. Introduction

FORMALIZING knowledge is an ancient problem. In the fourth century BC Aristotle included logic in his philosophical system and then the concept was understood as the intellectual representation of an object. The Aristotelian logic remained almost unchanged until the sixteenth century with the work of Leibniz [1] who began to include symbolic notation in logic. In the early nineteenth century, through the work of authors such as Boole [2], logic is related to mathematics through a mathematical system for modeling logical operations and accordingly a concept is a set of logic notions together with a set of rules. The acquisition of concepts has been a topic of study in psychology [3] and even recently, some computer science works focus on the concept notion [4].

Concept Analysis, a branch of analytical philosophy, aims at decomposing the elements, relations and meanings that compose a concept. There are several methods such as the Wilson's method [5], the Rodgers evolutionary method [6] or the Walker and Avant model [7]. Obtaining the characteristics of a concept is similar to requirement gathering or knowledge elicitation used in Computer Science. Our concept analysis is made with knowledge acquisition with constrains located into the knowledge domain and the knowledge sources. The former is reduced to a concept and the latter appears because of the difficulty to reach experts in the proposed domain.

If Concept Analysis techniques [5],[6],[7] are designed to have a clear and accurate definition of the concept under study. Usually, a concept is taken from a set of sources and, by means of several steps, how it operates and which relations it has with other concepts is revealed. The goal is to obtain a better understanding of the concept. Those techniques are particularly valuable when a concept has more than one meaning. The methods can vary according to the number of steps or the sources used. Some of them are language based and others literature based. The outcome is a language based description. Those methods are stepwise, and any enlargement of the concept or source later made implies redoing the whole analysis. Besides the methods are not suitable to be used in any computational system because they are not formal. There is neither model nor relationship between the elements and there is no detail on the constituents of the elements.

The proposed approach is a 7 step incremental and literature based method aiming at obtaining an outcome suitable to be used in object oriented engineering or ontology technologies. The objective is achieved by means of knowledge elicitation and visual modeling techniques. Knowledge elicitation is used to extract the relevant parts of text related to the concept under study. Concept maps help us to graphically represent the requirements and the Unified Model Language (UML) allows us to show graphically the elements and relations underlying the concept. The outcome reveals the attributes (the value) and behavior (as with what other concepts is related) of these concepts. The resulting graphic (a class diagram) shows these elements. Using UML as a knowledge representation language, further implementation is facilitated. Furthermore, the fact of being incremental allows the enlargement of the model adding new sources, with no need of redoing the former analysis.

This paper is organized as follows: Firstly, an overview of the techniques used to develop the method are introduced briefly after the Introduction. Secondly, the Knowledge Based Concept Analysis (KBCA) method is presented. Thirdly, a case study with the security notion.

II. Techniques Overview

Several techniques, briefly described, are used in order to obtain the proposed method (Fig: 1).

M. Colobran collaborates with the Department of Information and Communication Engineering, Universitat Autonoma de Barcelona, 08193 Bellaterra, Spain (e-mail: miquel.colobran@uab.cat).

Josep M. Basart is with the Department of Information and Communication Engineering, Universitat Autonoma de Barcelona, 08193 Bellaterra, Spain (e-mail: josepmaria.basart@uab.cat).

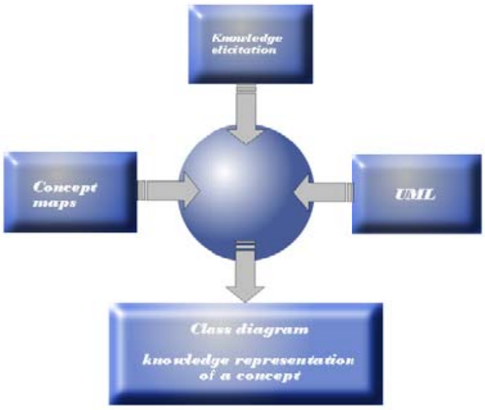


Fig. 1 KBCA Technologies involved

A. Concept Analysis

Concepts are multifaceted, abstract representations of reality [8]. Because of that, the concept analysis deals with its vagueness, ambiguity and context in order to clarify its meaning. The formal theories of concepts tries to systematize the way a concept is described such as Unified Concept Theory [4] or Formal Concept Analysis [9].

Concepts, under the view of knowledge, are a cognitive unit of meaning sometimes defined as a “unit of knowledge” (concept describes an abstract idea). The mental concepts describe a class or category. The grouping process is done by relating the aspects and qualities common to many objects. The set of all concepts gives us a representation of the world. Thus, a search for methods to handling concepts, the elements of concepts and the relations among concepts, is inevitable. Most of the used methods, such as Conceptual Maps [10], Formal Concept Analysis [11], Object Oriented techniques [12],[13],[14] or Knowledge engineering techniques [15] begin at their first stages with some sort of analysis. This analysis, in the knowledge field, is called knowledge acquisition.

B. Knowledge Acquisition

Knowledge acquisition is the process of achieving knowledge from a human expert or a group of experts [16]. The goal in knowledge engineering is the creation of knowledge-based systems (KBS).

Under the view of computer science, knowledge acquisition is a step in knowledge engineering. Broadly speaking, the knowledge life cycle include acquisition, design and implementation. Thus, software engineering and software knowledge have common points [15].

Despite there is a range of knowledge acquisition techniques [17], they deal with particular problems. Getting knowledge is made with informal methods such as interviews, questionnaires or unstructured sources, usually but not necessarily, in text form. Communication appears as a big trouble because experts and knowledge engineers have poor understanding of each other’s knowledge area. Usually, experts are not able to express exactly the knowledge and therefore it is difficult to get an overview of the problem to be solved. Besides, this process has a big quantity of informal knowledge which needs to be classified, organized and formalized somehow. In short, the expert has no knowledge on knowledge engineering and the knowledge engineer has poor knowledge on expert knowledge areas.

Knowledge engineer faces other problems, such as Knowledge validation and Knowledge representation. The former is the way to verify if the knowledge is right understood and the latter the way the knowledge is expressed in order to be used to implement the system. To lighten the problem, as in software engineering, Knowledge elicitation requires tools in order to manage requirements. Usually the tools could be a simple spreadsheet, a database or requirement management system.

Mainly, obtaining requirements or knowledge is based on natural language and this presents unique difficulties [18], [19]. Many of the activities involved are cognitive and require creativity as well as knowledge about information technologies and the application domain. Several tools have appeared in order to ease the problem and try eliminating ambiguities. These try to somehow make an interpretation of natural language in order to apply the heuristics [20]. The purpose of some of these tools is to intend to minimize as much as possible the analyst’s personal influence. These are still leaving the final decision of construction schemes in the analyst hands.

Thus, good knowledge gathering relies on the ability of analysts to interpret the model expressed by the user and then be able to express it in a formalized form. In software engineering, Abbot [21] first proposed a technique to gather requirements from texts. One of the big advantages to work with natural language is that it forces the developer to work on the vocabulary and space of the problem. Knowledge gathering stage, thus, is not rigorous because the natural language is ambiguous.

The final result of the knowledge elicitation step is a requirements knowledge representation. It needs to be simple to understand and formal enough to be used as the input of the knowledge implementation stage. That could be achieved by means of knowledge representation and modeling languages.

C. Mind Maps and Concept Maps

A task of concept classification somehow could be achieved using automated tools. Visual classification tools make it easy to classify objects and organize concepts. Currently, different categories of visual tools can be found [22],[23] such as Mind maps, Conceptual diagrams, Visual metaphors, Tree Maps, Flow Maps or Compare and Contrast Maps.

The most well known techniques are concept maps and mind maps. Concept maps are a way to visualize the mental “map” of concepts and their relationships, as well as the structure and hierarchy of these relationships. One important aspect of concept maps is their ability to show large amounts of information in a compact format. In this context, a concept is defined as “a perceived regularity in events or objects, or records of events or objects, designated by a label” [10].

Mind Mapping is a popular related technique devised by Tony Buzan. He describes mind maps as a net starting with a

International Science Index Vol:7, No:2, 2013 waset.org/Publication/5766

central word or concept and "around the central word you draw the 5 to 10 main ideas that relate to that word. You then take each of those child words and again draw the 5 to 10 main ideas that relate to each of those words" [24].

D. UML as a Knowledge Representation Language

Visual modeling started in Object Oriented software development methodologies and different methodologies for modeling have existed. But with no doubt, the Unified Modeling Language (UML) closed the discussion.

UML is the modeling language for software systems most well known and used today and is a de facto industry standard approved by the OMG (Object Management Group). UML is a set of specifications for object-oriented notation, which are composed of different diagrams that represent different stages of a software project development. UML combines techniques from data modeling, object modeling and component modeling. It can be used with all processes, along the software development life cycle. UML has synthesized the notations of the Booch method [25], the Object-modeling technique (OMT) [26] and Object-oriented software engineering (OOSE) [27] by fusing them into a single, common and widely usable modeling language.

UML is a graphical language for visualizing, specifying, constructing and documenting a system. The language focuses on the representation of a system and tells us how to create and read the models. However, nothing is said about how to create them. The latter is the goal of development methodologies. Some pros of UML could be found in [28]. The UML model consists of three classes of construction blocks, elements, relationships and diagrams. Elements are abstractions of real or fictitious things such as objects or actions. Relationships are the way how elements relate to each other. Diagrams reflect collections of elements along with their relationships.

The class diagram exhibits a set of classes, interfaces and relationships. This is the most common diagram in describing the design of object-oriented systems. In order to properly represent a system, UML offers a wide variety of diagrams to visualize the system from several perspectives and UML 2.0 includes 13 types of diagrams. As the aim of UML is to model any type of systems, not just software, it is also used as a knowledge representation language and the construction of ontologies [29], [30], [31].

III. KBCA

Our proposal uses together knowledge elicitation, concept maps and UML in order to produce a graphical representation of a concept. Knowledge elicitation, with constrains, is used for requirements gathering; concept maps are used to to produce a graphical representation of the requirements and UML is used to draw the final outcome. The method is named as Knowledge Based Concept Analysis (KBCA) of a concept.

A. Concept Analysis and Knowledge Acquisition Restrictions

In concept analysis the work is focused on a previously agreed concept. KBS work is focused on the domain defined at the beginning of the life cycle. Concept analysis ends when the concept is fully described and Knowledge engineering ends when the computational system is constructed. Knowledge engineering life cycle includes an analysis phase, but also has the design and implementation stages. In order to move closer concept analysis and knowledge engineering, the following points need to be considered.

- Knowledge engineering could fit purposes other than creating a computational system.
- Knowledge engineering life cycle involves several steps. Using the ones related to analysis and design, a knowledge model is obtained.
- Knowledge engineering domain is extremely flexible and could be as small as a concept.

In knowledge engineering, if the implementation stage is not done just a knowledge model of the domain is obtained. If the domain is a concept, the analysis and design stages will be focused just on that concept, its attributes and its relations. The result will be the knowledge model of a concept and become a type of concept analysis.

Another restriction is needed. When dealing with a concept, reaching the experts could be difficult or even not possible. Let's suppose a work focused on the Newton's concept of law of universal gravitation or the Descartes concept of mathematics. The concept description should be described on the basis of their writings or the interpretation of these concepts from other people. Thus, the best sources we can achieve are documents.

B. Method in Detail

KBCA consists of seven main steps as shown in Table I and Fig. 2. First three steps belong to the knowledge requirement gathering phase, fourth and fifth steps are the categorization (ordering phase). Sixth step makes the map of ideas/concepts/notions collected, and the last one converts the concept map into a class diagram.

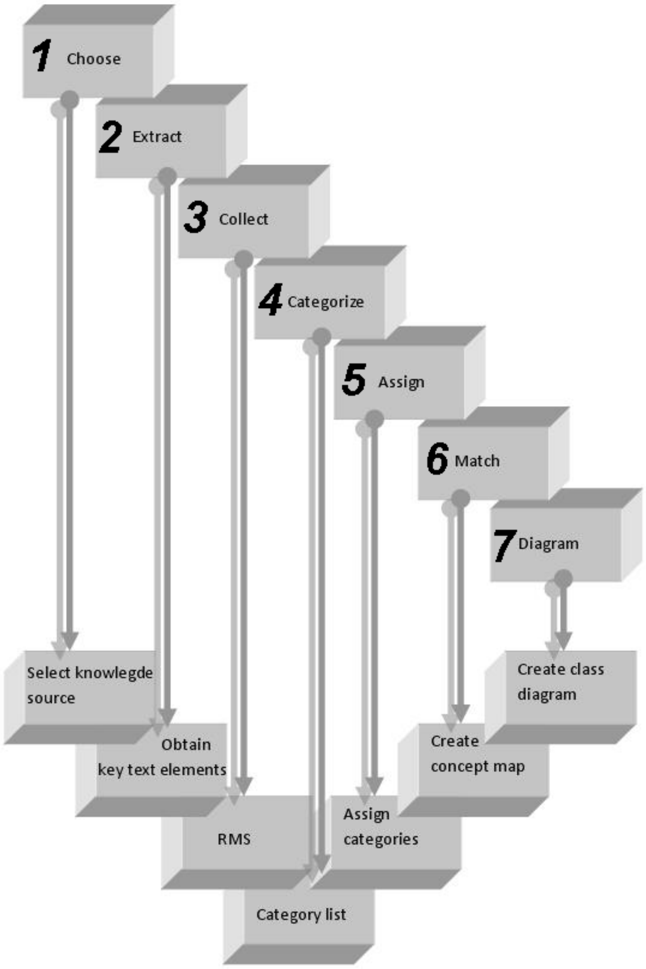


Fig. 2 KBCA Method

The relevant elements and relations are detected in the second step. This is made by emphasizing the important text pieces. That could be made in ways such as changing the text color in a word processor as well as underlining it. From now these chunks of text are called key text elements (KT_i). That could be done in parallel to populate the list or database (step 3). For clarity purposes this has been separated into two steps. This step is the most critical part of KBCA. The rest of steps rely on this one, because the final outcome heavily depends on this one being properly taken. Thus, It implies some kind of subjective component.

The third step implies collecting the information gathered into a list. That list could be made as plain text, spreadsheet, database or even a Requirement Management Software. A simple document could create about 150 key texts. Thus, a kind of mechanical tool is highly recommended. The minimal needed fields are:

- Key text number
- A way of connecting somehow the key text to the document. That could be done in many ways such as writing the key number to the document or even copying the key text into the list.
- The category

TABLE I
KBCA FLOW DIAGRAM

	Stage	Description	
Step 1	Choose	Choose knowledge source	Knowledge Elicitation
Step 2	Extract	Select key text elements	
Step 3	Collect	Insert into database and number	
Step 4	Categorize	Create list of categories	
Step 5	Assign	Assign into categories	
Step 6	Map elements	Create concept map	Concept map
Step 7	Class diagram	Construct class diagram	UML

In the fourth step, after listing, the category list is created. The possibility the category list is known from the beginning exists. Thus, the list could be created any step before. Once again, for clarity purposes, has been placed in that position.

Every requirement matches a category. This is the fifth step. The assignment is needed in order to detect extra or redundant key text. The following steps use the resulting list.

Step 6 is the most difficult part. Any key text is represented into a conceptual map. A number of situations may appear such as two similar key texts that have no relation at all or the coincidence of two key text which reveal that there is no need of rewriting. The rules to operate with key texts could be summarized as shown in table II.

TABLE II
RULES

Rule	Description	Action
$KT_i \neq KT_j \;\; \forall\, j$	A new element	Add to the graphic
$KT_i \supset KT_j$	Includes	No change
$KT_i \subset KT_j$	Included	No change
$KT_i = KT_j$	Same	No change
KT_i *enlarge* KT_j	The key text could add some aspect related to the previous key text	Add
KT_i	Is a relation, not an element	Add

As a result, a concept map diagram is obtained. Thus, there is a bunch of ideas and relations spread on the map.

In the seventh and last step, a conversion of the key text extraction into a more formal graphical language is made, UML. The final result is clearly understood because a standard methodology has been used.

The final diagram needs to express elements or ideas, relations and cardinality.

It is highly recommended to add some extra information at the bottom of the graphics. The reason relies on the fact that some key text elements determine values of the attributes. Thus, the list of the known values is added.

International Science Index Vol:7, No:2, 2013 waset.org/Publication/5766

IV. Case Study of KBCA

Now, the method is applied to security concept. Barry Buzan stated that "security is a underdeveloped concept" [32] in 1983. From then, there is a plethora of work focused on what the security concept is (structure, elements, relations).

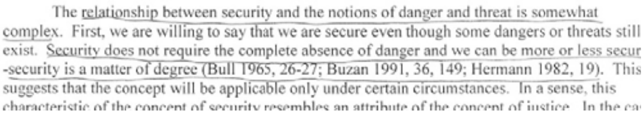
A. Applying KBCA Steps

1) Choose

The article chosen is "The concept of security" by Peter Digeser [33]. The article, unpublished but used with author permission, explores the security concept, its meanings and how it is seen by states and individuals. The work emphasizes that safety is an abstract concept and with no specification the concept is empty of meaning. Just when the elements are filled, security appears as an operational concept.

2) Extract

Text is reviewed and the text key elements are underlined as show in Fig. 3.



The relationship between security and the notions of danger and threat is somewhat complex. First, we are willing to say that we are secure even though some dangers or threats still exist. Security does not require the complete absence of danger and we can be more or less secur -security is a matter of degree (Bull 1965, 26-27; Buzan 1991, 36, 149; Hermann 1982, 19). This suggests that the concept will be applicable only under certain circumstances. In a sense, this characteristic of the concept of security resembles an attribute of the concept of justice. In the ca

Fig. 3 Key text elements underlined in the source

At this step just the text that looks relevant to the concept is chosen. An initial and provisional list of categories is feasible at this point.

3) Collect

A simple database is populated with the key text elements. Fig. 4 shows the database structure and Fig. 5 some key text elements.

During this step, probably, some redundant key text elements could be discovered.

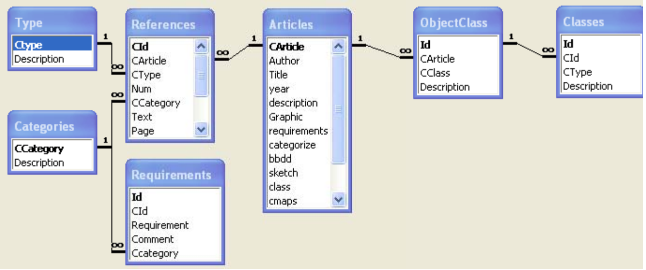
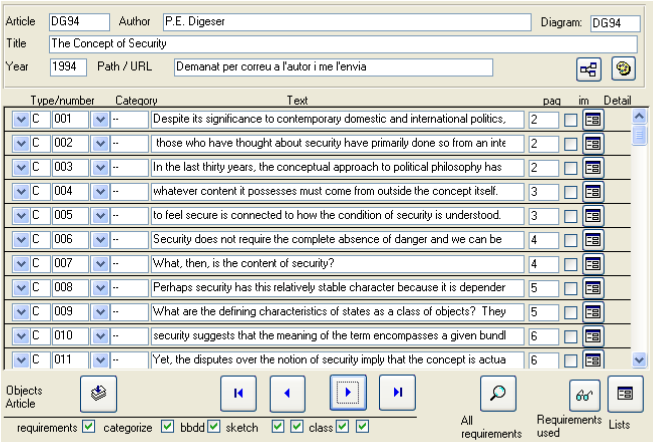


Fig. 4 Database structure



Article DG94 Author P.E. Digeser Diagram: DG94
Title The Concept of Security
Year 1994 Path / URL Demanat per correu a l'autor i me l'envia

Type/number		Category	Text	pag	im	Detail
C	001	--	Despite its significance to contemporary domestic and international politics,	2	□	
C	002	--	those who have thought about security have primarily done so from an inte	2	□	
C	003	--	In the last thirty years, the conceptual approach to political philosophy has	2	□	
C	004	--	whatever content it possesses must come from outside the concept itself.	3	□	
C	005	--	to feel secure is connected to how the condition of security is understood.	3	□	
C	006	--	Security does not require the complete absence of danger and we can be	4	□	
C	007	--	What, then, is the content of security?	4	□	
C	008	--	Perhaps security has this relatively stable character because it is depender	5	□	
C	009	--	What are the defining characteristics of states as a class of objects? They	5	□	
C	010	--	security suggests that the meaning of the term encompasses a given bundl	6	□	
C	011	--	Yet, the disputes over the notion of security imply that the concept is actua	6	□	

Objects Article
requirements ☑ categorize ☑ bbdd ☑ sketch ☑ ☑ class ☑ ☑

Fig. 5 Key text elements

4) Categorize

Once all the requirements are collected, the list of categories needs to be made. Typically there are just a few categories. If previously a provisional list has been made, it is used to make the final one. In this case, the categories listed in table III are discovered.

TABLE III
Security Categories

Categories
National Security
Security – concept
Security – attributes
Security – risk
Security – sort

5) Assign

Assigning a category to a requirement helps later on the graphical stage. We have obtained 81 key text elements. The ones which are useful to our purpose are discovered and categorized (table IV).

TABLE IV
Key Text Categorized

Categories	Quantity
National Security	4
Security – concept	40
Security – attributes	4
Security – risk	2
Security – sort	6

6) Concept Map

This is the most "traditional" step. Once the requirements are collected and organized, we have all the ingredients to create the concept map. This step involves reviewing all the requirements, one by one, in order to raise all the relationships between them and related concepts.

The relations are spread all over the text, and so are into the requirement list. The concept map may contain redundancies, i.e. the same concept appears in different requirements with, apparently, no relation or two concepts are linked together in different places.

International Science Index Vol:7, No:2, 2013 waset.org/Publication/5766

This step is the first one that reduces the amount of information gathered. Using the mentioned rules in table II or even making new ones should be useful to create the concept map.

The outcome is a set of ideas spread onto the canvas. A lot of redundancy is eliminated as shown in Fig. 6.

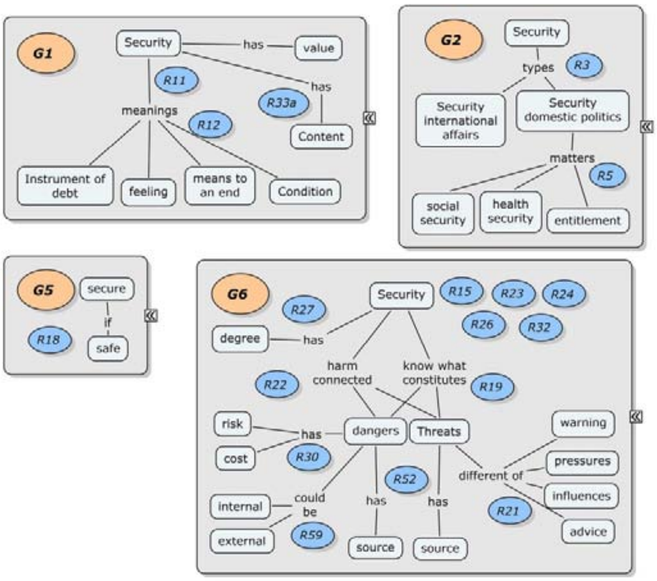


Fig. 6 Some concept map elements of the security concept

7) Class Diagram

This step helps reducing the amount of ideas in the previous stage. The outcome is a class diagram that represents the elements and relations involved (Fig. 7). The class diagram and the elements, using UML terminology, are classes and relations between classes and subclasses.

In order to create the class model, the following actions help.

- Fit each element (concept) in a class box.
- Add the attributes and behavior into the class.
- Create the relationship between elements. Add cardinality.

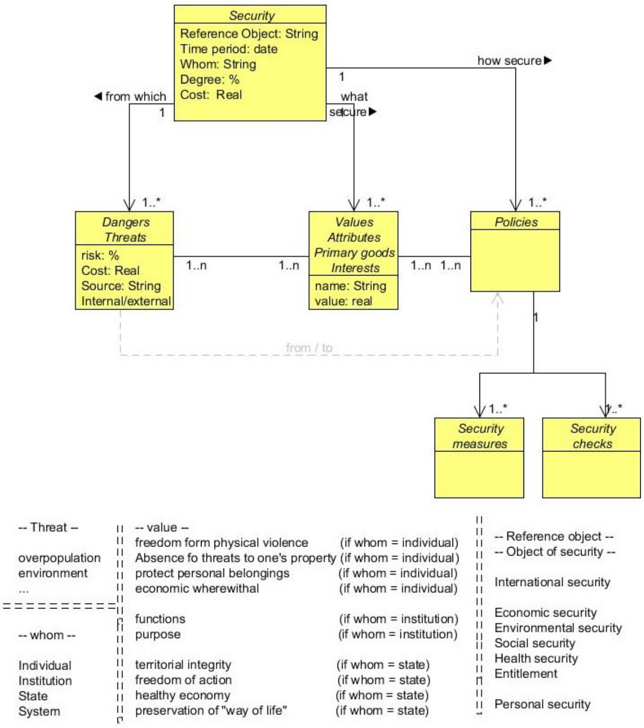


Fig. 7 Class diagram of the security concept

B. Discussion

Several points emerge from this work.

1) Meaning

The knowledge model obtained is a description of the elements, its components and relations among them. Like UML, the model has no meaning by itself. Thus, the case study of security expresses a range of possible definitions of security that are unveiled when the model is instantiated. At this point, when de components have a value, a security definition (class instantiated) appears. That security definition, using the object oriented paradigm, is unique.

2) Incremental Growth

The nature of the method permits an incremental growth of the knowledge model. An iterative process on other sources leads to a bigger and more detailed knowledge model without losing the knowledge acquired from the other sources. Even, new sources produce smaller or no changes because of the model become more complete at every cycle.

3) Uniqueness

As shown, the knowledge security model is meaningless. What if there are two security instances A and B?.

If A = B then all the elements, relations and components are the same, and we can conclude that the security definition is the same.

If A and B are two security objects with B having, for example, a different set of policies or threats, we can conclude that in this scenario, A ≠ B.

Thus, there is no unique security definition. There are just security concept constructions and as many securities as

International Science Index Vol:7, No:2, 2013 waset.org/Publication/5766

different security objects we are able to create. This is the reason why the "definition" of the resulting security is different. Therefore, persons, groups or states perceive different notions of security because the defining elements vary remarkably.

Besides, if we create a different construction of security (from other source for example), all the resulting objects will be different security objects (despite being neither semantically nor in practice incorrect).

4) Security Definitions and Computer Security

From Barry Buzan work [32], a wide range of security definitions are identified. For example, the human security from UNPD [34] or the expanded notion of security stated by Emma Rothschild [35] who argued that security notion is extended in "four main forms". Open questions emerge such as is if all of those securities could be considered a kind of a bigger security model, actually a knowledge security model and how computer security and the existing securities could be peacefully integrated in such model. Because of the fact that computers are social tools, Computer Security needs an inter-disciplinary work in order to become another kind of security.

V. Conclusions Remarks

A methodology for exploring the underlying elements in a concept and the relationships among them is proposed. The outcome is an abstract concept, which requires specific elements to produce "the definition". This definition is extremely flexible and can be adapted to almost any framework of any field.

The knowledge based concept analysis (KBCA) proposed method is based on knowledge engineering, concept maps and UML. It's intended to extract knowledge from any informal source in order to obtain concept class diagram. That outcome could be used in object oriented engineering or knowledge based systems such as ontologies.

Concept analysis can also be made with knowledge elicitation applying some restrictions in the domain and the steps involved. The outcome of design stage in knowledge engineering, when the domain is restricted to one concept, leads to a type of concept analysis. The proposed method is a 7 steps concept analysis and literature based in order to overcome expert elicitation problems.

In the proposed scenario, the knowledge engineering analysis and design stages are focused just on one concept its attributes and its relations. The result is the knowledge model of a concept.

Traditional concept analysis methods are stepwise. Our proposal is incremental, thus enlarge the model is easier. The UML purpose is to model any type of systems (not just software). This language should be understandable to humans and machines and could be used as a knowledge representation language.

KBCA is very systematic. Further implementation of the result, if needed, will be easier because of UML is used. The resulting diagram could be used to check by end-users or documentmakers and even could be used to integrate in bigger projects, related or not with computer software.

Despite we have reduced as much as possible the subjective component, the requirements gathering are a human task and the method still suffers from a subjective component. Thus, most probably the same text analyzed by several people may easily lead to slightly different outcome.

International Relations field has made, in the last decades, a lot of work on the concept and structure of the security notion. Their main concern are the types of securities, the existing relationship between several securities, security policies and, to a lesser extend the semantic notion of security and its consequences on individuals, entities or nationalities. There are no works available in order to link that security with information security in computer science. A generic framework could benefit both fields.

The security concept is meaningless until all the elements are instantiated and the "definition" of security relay on the values instead of the word on its own.

In the case of complex concepts, the review from just a single source of knowledge is clearly insufficient. Therefore, a further work to obtain a class diagram (formalization of a concept) from many sources (formal or informal) is needed. In order to extend the range, other kind of sources such as written documents, voice recordings, pictures and in general any multimedia documents need to be included.

More research is also needed to discover in which areas this methodology is useful and what changes or improvements would be needed to adapt to these new scenarios.

References

[1] L. C. Agrela, "La superacion por Leibniz de la logica aristotelica," Revista Internacional de Filosofía, vol. Suplemento 3,, pp. 67–74, 2010.

[2] P. Jetli, "The Completion of the Emergence of Modern Logic from Boole's The Mathematical Analysis of Logic to Frege's Begriffsschrift," in Logic and Its Applications, ser. Lecture Notes in Computer Science, M. Banerjee and A. Seth, Eds. Springer Berlin Heidelberg, 2011, vol. 6521, pp. 105–123. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-18026-2_10

[3] R. Streveler, T. Litzinger, R. Miller, and P. Steif, "Learning Conceptual Knowledge in the Engineering Sciences: Overview and Future Research Directions," Journal of Engineering Education, vol. 97, pp. 279–294, July 2008.

[4] J. Goguen, "What Is a Concept?" Lecture Notes in Computer Science: Conceptual Structures: Common Semantics for Sharing Knowledge, pp. 52–77, 2005. [Online]. Available: http://dx.doi.org/10.1007/11524564_4

[5] J. Wilson, Thinking with concepts. Cambridge University Press, 1963.

[6] B. L. Rodgers, "Concepts, Analysis, and the Development of Nursing Knowledge: The Evolutionary Cycle." Journal of Advanced Nursing, vol. 14, pp. 330–335, 1989.

[7] K. C. Walker, L.O. Avant, Strategies for theory construction in nursing, 3rd ed. Norwalk, CT: Appleton & Lange, 1995.

[8] V. L. Griffin-Heslin and al., "An analysis of the concept dignity," Accident and Emergency Nursing, vol. 13, pp. 251–257, 2005.

[9] U. Priss, "Formal Concept Analysis in Information Science," Annual Review of Information Science and Technology, vol. 40, pp. 521–543, 2006. [Online]. Available: (http://www.upriss.org.uk/papers/arist.pdf).

[10] J. D. Novak and A. J. Cañas, "The Theory Underlying Concept Maps and How to Construct Them," Technical Report IHMC CmapTools, Tech. Rep. 2006-01, 2006. [Online]. Available: (http://cmap.ihmc.us/Publications/ResearchPapers/TheoryCmaps/TheoryUnderlyingConceptMaps.htm).

[11] B. Ganter and R. Wille, Formal Concept Analysis: Mathematical Foundations, 1st ed. Secaucus, NJ, USA: Springer-Verlag New York, Inc., 1997.

[12] R. S. Pressman, Software engineeering: a practioner's approach, 6th ed. Boston, EUA: McGraw-Hill, 2005.

[13] M. O'Docherty, Object-oriented analysis and design : understanding system development with UML 2.0. John Wiley & Sons, 2005.

[14] P. Coad and E. Yourdon, Object-Oriented Analysis. London: Prentice-Hall, 1991.

[15] R. Studer, R. Benjamins, and D. Fensel, "Knowledge engineering: principles and methods," Data and knowledge engineering, vol. 25, pp. 161–197, 1998.

[16] S. Kendal and M. Creen, An Introduction to Knowledge Engineering, 1st ed. Springer, Oct. 2006.

[17] J. Hua, "Study on Knowledge Acquisition Techniques," in Proceedings of the 2008 Second International Symposium on Intelligent Information Technology Application - Volume 01, ser. IITA '08. Washington, DC, USA: IEEE Computer Society, 2008, pp. 181–185. [Online]. Available: http://dx.doi.org/10.1109/IITA.2008.152.

[18] S. Potter, "A Survey of Knowledge Acquisition from Natural Language," AKT project report Task 1.1.2, 2001.

[19] J. Wang, Y. Wu, X. Liu, and X. Gao, "Knowledge acquisition method from domain text based on theme logic model and artificial neural network." Expert Syst. Appl., vol. 37, pp. 267–275, 2010. [Online]. Available: http://dx.doi.org/10.1016/j.eswa.2009.05.009.

[20] S. P. Overmyer, B. Lavoie, and O. Rambow, "Conceptual Modeling through Linguistic Analysis Using LIDA." in Software Engineering, 2001. ICSE 2001. Proceedings of the 23rd International Conference. IEEE Computer Society, 2001, pp. 401–410.

[21] R. J. Abbott, "Program design by informal English descriptions," Commun. ACM, vol. 26, pp. 882–894, November 1983. [Online]. Available: http://doi.acm.org/10.1145/182.358441.

[22] L. Dillard and B. Myers, "Visual Teaching Tools: Concept Maps," University of Florida, Tech. Rep., May 2008.

[23] M. J. Eppler, "A comparison between concept maps, mind maps, conceptual diagrams, and visual metaphors as complementary tools for knowledge construction and sharing," Information Visualization, vol. 5, no. 3, pp. 202–210, 2006.

[24] T. Buzan and B. Buzan, The Mind Map Book, 2nd ed. London: BBC Books, 1995.

[25] G. Booch, Object-oriented analysis and design with applications (2nd ed.). Redwood City, CA, USA: Benjamin-Cummings Publishing Co., Inc., 1994. [Online]. Available: http://portal.acm.org/citation.cfm?id=174890.

[26] J. Rumbaugh, Object-oriented modeling and design. Prentice Hall, 1991.

[27] I. Jacobson, Object Oriented Software Engineering: A Use Case Driven Approach. Addison-Wesley, 1992.

[28] E. H. Orallo, "El Lenguaje Unificado de Modelado (UML)," Manuales Formativos ACTA, num 26, October 2002.

[29] S. Cranefield and M. K. Purvis, "UML as an Ontology Modelling Language." in In Proceedings of the Workshop on Intelligent Information Integration, 16th International Joint Conference on Artificial Intelligence (IJCAI-99), 1999, pp. 46–53.

[30] A. Felfernig, G. Friedrich, D. Jannach, and M. Zanker, "Configuration Knowledge Representation Using UML/OCL," in "UML" 2002 - The Unified Modeling Language. Springer Berlin / Heidelberg, 2002, pp. 91–108. [Online]. Available: http://dx.doi.org/10.1007/3-540-45800-X_5.

[31] C. W. Chan, "Knowledge and software modeling using UML." Software and System Modeling, vol. 3, no. 4, pp. 294–302, 2004.

[32] B. Buzan, People, States and Fear. Harvester-Wheatsheaf, Brighton, 1983.

[33] P. Digeser, "The Concept of Security," 1994, presented at the Annual Meeting of the American Political Science Association 14 September 1994. Obtained from author. Unpublished.

[34] Undp, HDR 1994 - New Dimensions of Human Security. Human Development Report Office (HDRO), United Nations Development Programme (UNDP), 1994. [Online]. Available: http://EconPapers.repec.org/RePEc:hdr:report:hdr1994.

[35] E. Rothschild, "What is security? the quest for world order," Daedalus, vol. 124, no. 3, pp. 53–99, June 1995.

Miquel Colobran is a doctoral student at the Department of Information and Communication Engineering. His research is in the field of ontologies, security and Social Computing.

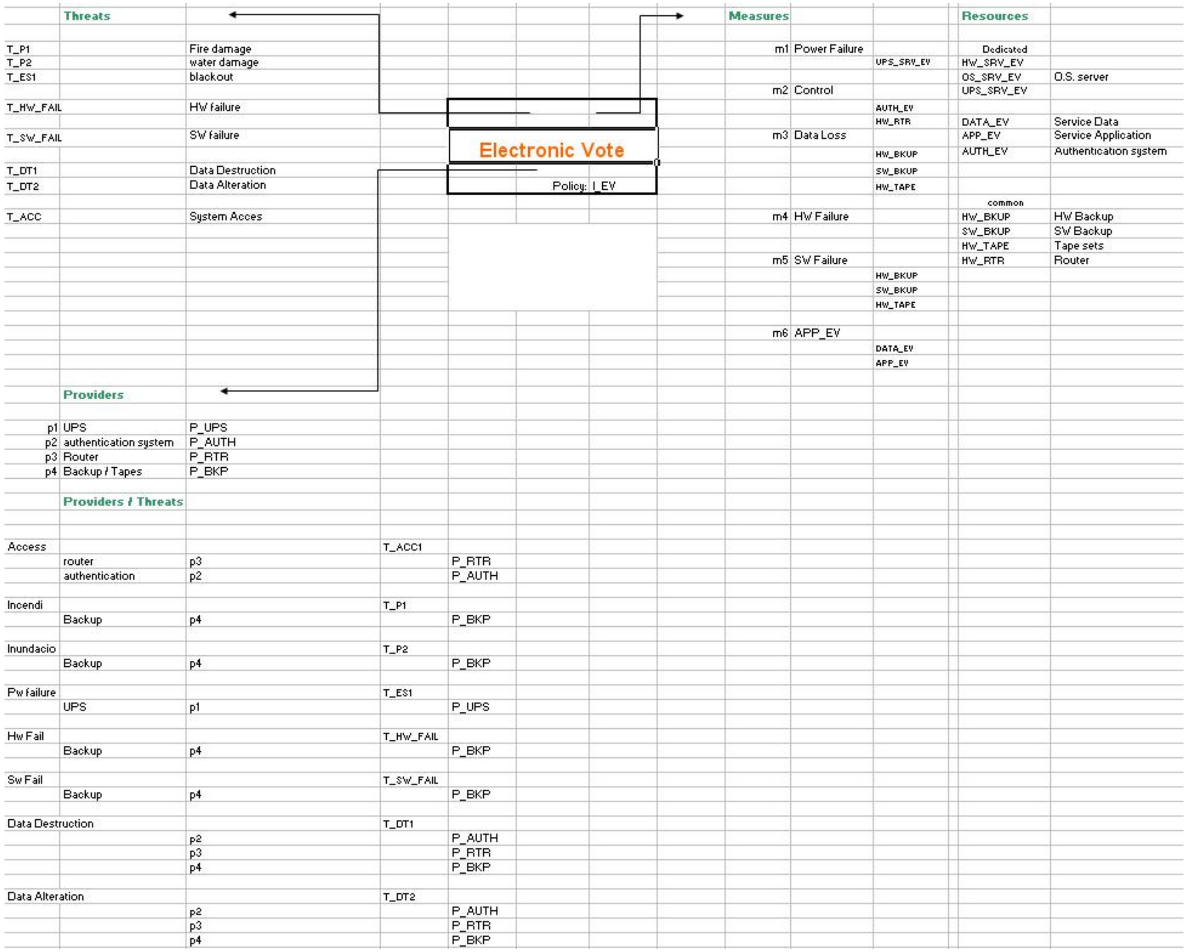
Josep M. Basart is a PHD professor at the Department of Information and Communication Engineering. His research is in the field of Computer Ethics, Engineering Ethics, Applied Ethics and Social Computing.

APPENDIX E

CSUC INSTITUION

Some data and elements have been changed or removed in order to protect the privacy of CSUC Institution. The outcome, therefore, is not completely real, despite it constitutes a very good example.

E.1 Electronic Voting schemes



Threats

T_P1	Fire damage
T_P2	water damage
T_ES1	blackout
T_HW_FAIL	HW failure
T_SW_FAIL	SW failure
T_DT1	Data Destruction
T_DT2	Data Alteration
T_ACC	System Acces

Measures

m1	Power Failure	
		UPS_SRV_EV
m2	Control	
		AUTH_EV
		HW_RTR
m3	Data Loss	
		HW_BKUP
		SW_BKUP
		HW_TAPE
m4	HW Failure	
m5	SW Failure	
		HW_BKUP
		SW_BKUP
		HW_TAPE
m6	APP_EV	
		DATA_EV
		APP_EV

Resources

Dedicated	
HW_SRV_EV	
OS_SRV_EV	O.S. server
UPS_SRV_EV	
DATA_EV	Service Data
APP_EV	Service Application
AUTH_EV	Authentication system
common	
HW_BKUP	HW Backup
SW_BKUP	SW Backup
HW_TAPE	Tape sets
HW_RTR	Router

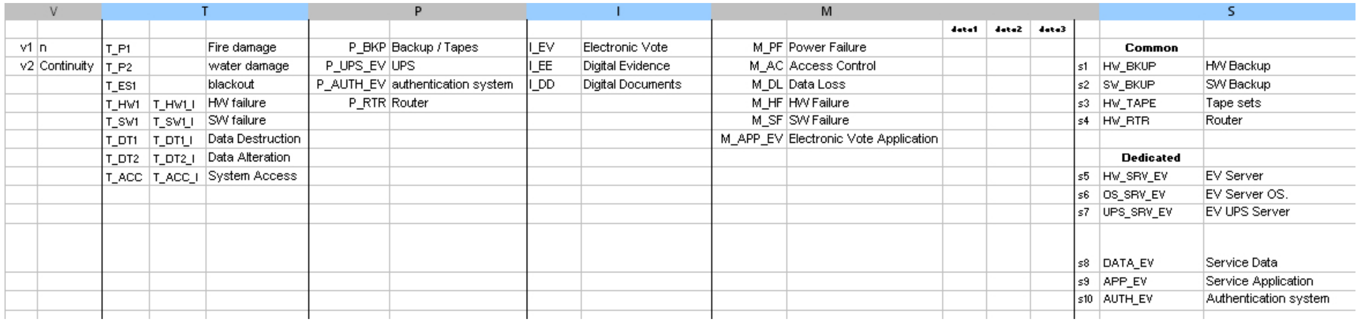
Providers

p1	UPS	P_UPS
p2	authentication system	P_AUTH
p3	Router	P_RTR
p4	Backup / Tapes	P_BKP

Providers / Threats

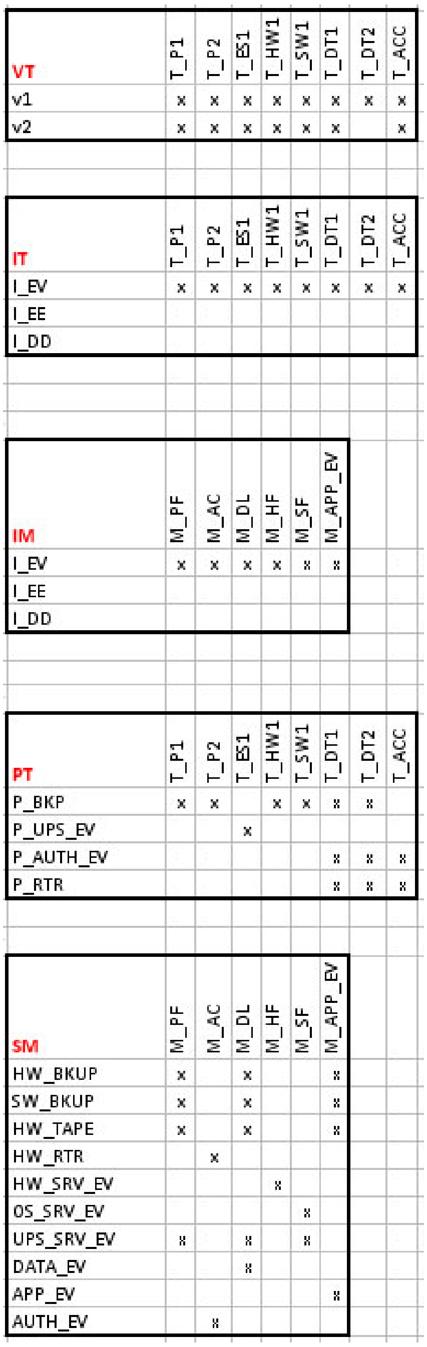
Access			T_ACC1	
	router	p3		P_RTR
	authentication	p2		P_AUTH
Incendi			T_P1	
	Backup	p4		P_BKP
Inundacio			T_P2	
	Backup	p4		P_BKP
Pw failure			T_ES1	
	UPS	p1		P_UPS
Hw Fail			T_HW_FAIL	
	Backup	p4		P_BKP
Sw Fail			T_SW_FAIL	
	Backup	p4		P_BKP
Data Destruction			T_DT1	
		p2		P_AUTH
		p3		P_RTR
		p4		P_BKP
Data Alteration			T_DT2	
		p2		P_AUTH
		p3		P_RTR
		p4		P_BKP

Figure E.1 – “electronic voting” policy.



V		T			P		I		M					S		
											date1	date2	date3			
v1	n	T_P1		Fire damage	P_BKP	Backup / Tapes	I_EV	Electronic Vote	M_PF	Power Failure					**Common**	
v2	Continuity	T_P2		water damage	P_UPS_EV	UPS	I_EE	Digital Evidence	M_AC	Access Control				s1	HW_BKUP	HW Backup
		T_ES1		blackout	P_AUTH_EV	authentication system	I_DD	Digital Documents	M_DL	Data Loss				s2	SW_BKUP	SW Backup
		T_HW1	T_HW1_I	HW failure	P_RTR	Router			M_HF	HW Failure				s3	HW_TAPE	Tape sets
		T_SW1	T_SW1_I	SW failure					M_SF	SW Failure				s4	HW_RTR	Router
		T_DT1	T_DT1_I	Data Destruction					M_APP_EV	Electronic Vote Application						
		T_DT2	T_DT2_I	Data Alteration											**Dedicated**	
		T_ACC	T_ACC_I	System Access										s5	HW_SRV_EV	EV Server
														s6	OS_SRV_EV	EV Server OS.
														s7	UPS_SRV_EV	EV UPS Server
														s8	DATA_EV	Service Data
														s9	APP_EV	Service Application
														s10	AUTH_EV	Authentication system

Figure E.2 – List of concepts, instances and relations.



VT	T_P1	T_P2	T_ES1	T_HW1	T_SW1	T_DT1	T_DT2	T_ACC
v1	x	x	x	x	x	x	x	x
v2	x	x	x	x	x	x		x

IT	T_P1	T_P2	T_ES1	T_HW1	T_SW1	T_DT1	T_DT2	T_ACC
I_EV	x	x	x	x	x	x	x	x
I_EE								
I_DD								

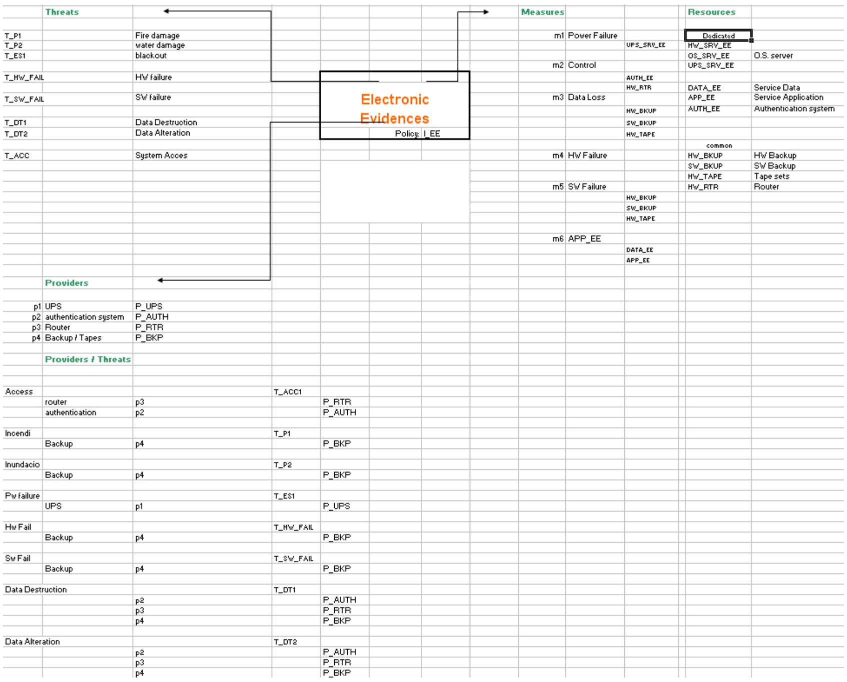
IM	M_PF	M_AC	M_DL	M_HF	M_SF	M_APP_EV
I_EV	x	x	x	x	x	x
I_EE						
I_DD						

PT	T_P1	T_P2	T_ES1	T_HW1	T_SW1	T_DT1	T_DT2	T_ACC
P_BKP	x	x		x	x	x	x	
P_UPS_EV			x					
P_AUTH_EV						x	x	x
P_RTR						x	x	x

SM	M_PF	M_AC	M_DL	M_HF	M_SF	M_APP_EV
HW_BKUP	x		x			x
SW_BKUP	x		x			x
HW_TAPE	x		x			x
HW_RTR		x				
HW_SRV_EV				x		
OS_SRV_EV					x	
UPS_SRV_EV	x		x		x	
DATA_EV			x			
APP_EV						x
AUTH_EV		x				

Figure E.3 – Primary relations in electronic voting service.

E.2 Technological evidences schemes



Electronic Evidences

Policy: I_EE

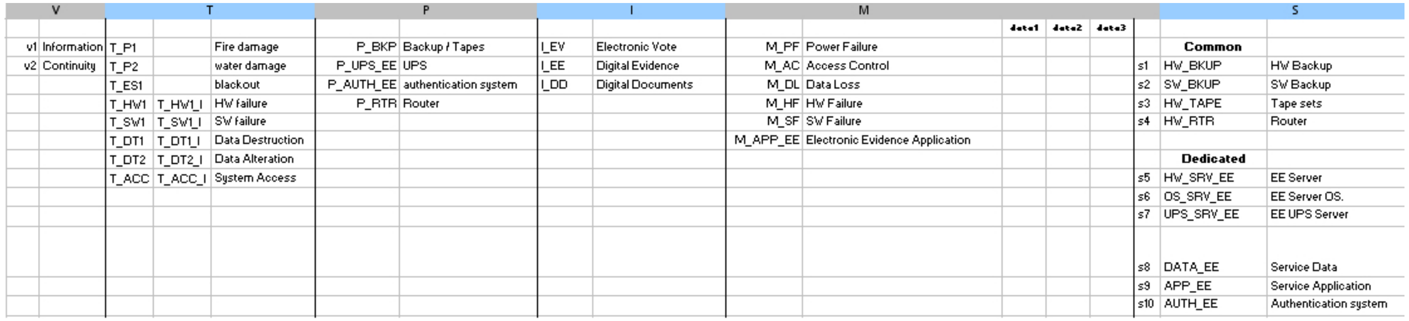
Threats	
T_P1	Fire damage
T_P2	water damage
T_ES1	blackout
T_HW_FAIL	HW failure
T_SW_FAIL	SW failure
T_DT1	Data Destruction
T_DT2	Data Alteration
T_ACC	System Acces

Measures			Resources	
m1	Power Failure		Dedicated	
		UPS_SRV_EE	HW_SRV_EE	
			OS_SRV_EE	O.S. server
m2	Control		UPS_SRV_EE	
		AUTH_EE		
		HW_RTR	DATA_EE	Service Data
m3	Data Loss		APP_EE	Service Application
		HW_BKUP	AUTH_EE	Authentication system
		SW_BKUP		
		HW_TAPE		
			common	
m4	HW Failure		HW_BKUP	HW Backup
			SW_BKUP	SW Backup
			HW_TAPE	Tape sets
m5	SW Failure		HW_RTR	Router
		HW_BKUP		
		SW_BKUP		
		HW_TAPE		
m6	APP_EE			
		DATA_EE		
		APP_EE		

	Providers	
p1	UPS	P_UPS
p2	authentication system	P_AUTH
p3	Router	P_RTR
p4	Backup / Tapes	P_BKP

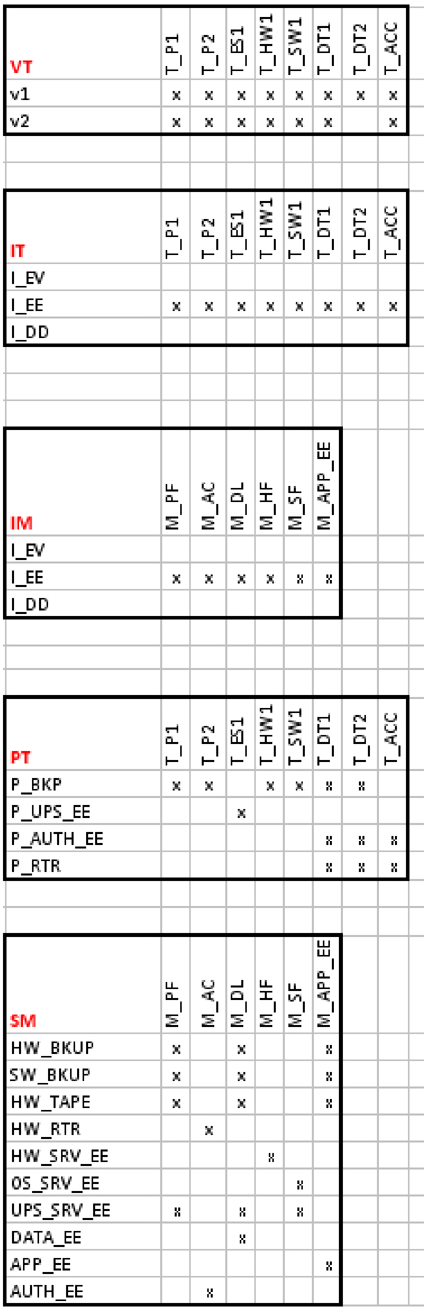
	Providers / Threats			
Access			T_ACC1	
	router	p3		P_RTR
	authentication	p2		P_AUTH
Incendi			T_P1	
	Backup	p4		P_BKP
Inundacio			T_P2	
	Backup	p4		P_BKP
Pw failure			T_ES1	
	UPS	p1		P_UPS
Hw Fail			T_HW_FAIL	
	Backup	p4		P_BKP
Sw Fail			T_SW_FAIL	
	Backup	p4		P_BKP
Data Destruction			T_DT1	
		p2		P_AUTH
		p3		P_RTR
		p4		P_BKP
Data Alteration			T_DT2	
		p2		P_AUTH
		p3		P_RTR
		p4		P_BKP

Figure E.4 – “Electronic evidences” policy.



V		T			P		I		M					S		
											data1	data2	data3			
v1	Information	T_P1		Fire damage	P_BKP	Backup / Tapes	I_EV	Electronic Vote	M_PF	Power Failure					**Common**	
v2	Continuity	T_P2		water damage	P_UPS_EE	UPS	I_EE	Digital Evidence	M_AC	Access Control				s1	HW_BKUP	HW Backup
		T_ES1		blackout	P_AUTH_EE	authentication system	I_DD	Digital Documents	M_DL	Data Loss				s2	SW_BKUP	SW Backup
		T_HW1	T_HW1_I	HW failure	P_RTR	Router			M_HF	HW Failure				s3	HW_TAPE	Tape sets
		T_SW1	T_SW1_I	SW failure					M_SF	SW Failure				s4	HW_RTR	Router
		T_DT1	T_DT1_I	Data Destruction					M_APP_EE	Electronic Evidence Application						
		T_DT2	T_DT2_I	Data Alteration											**Dedicated**	
		T_ACC	T_ACC_I	System Access										s5	HW_SRV_EE	EE Server
														s6	OS_SRV_EE	EE Server OS.
														s7	UPS_SRV_EE	EE UPS Server
														s8	DATA_EE	Service Data
														s9	APP_EE	Service Application
														s10	AUTH_EE	Authentication system

Figure E.5 – List of concepts, instances and relations.



VT	T_P1	T_P2	T_ES1	T_HW1	T_SW1	T_DT1	T_DT2	T_ACC
v1	x	x	x	x	x	x	x	x
v2	x	x	x	x	x	x		x

IT	T_P1	T_P2	T_ES1	T_HW1	T_SW1	T_DT1	T_DT2	T_ACC
I_EV								
I_EE	x	x	x	x	x	x	x	x
I_DD								

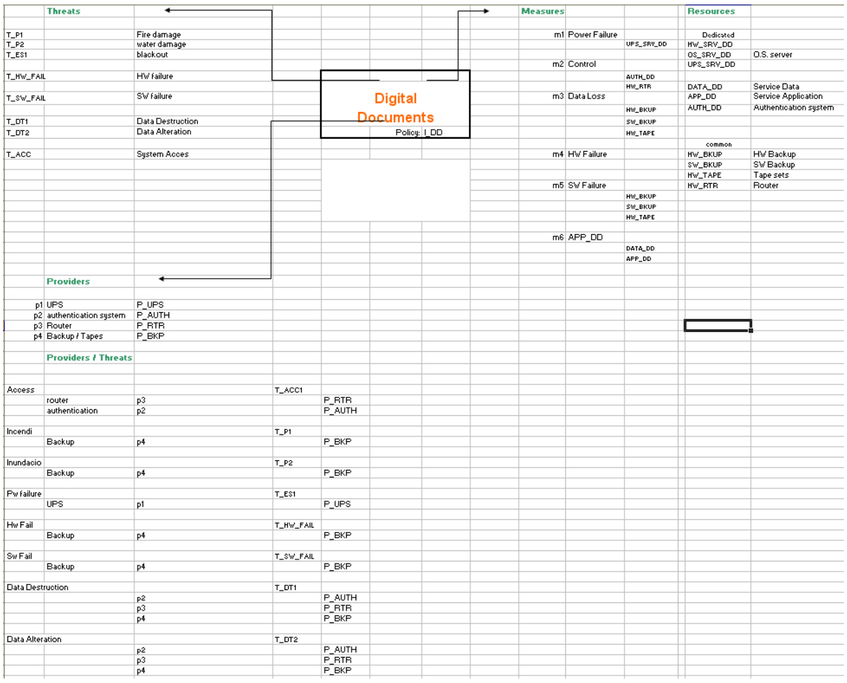
IM	M_PF	M_AC	M_DL	M_HF	M_SF	M_APP_EE
I_EV						
I_EE	x	x	x	x	x	x
I_DD						

PT	T_P1	T_P2	T_ES1	T_HW1	T_SW1	T_DT1	T_DT2	T_ACC
P_BKP	x	x		x	x	x	x	
P_UPS_EE			x					
P_AUTH_EE						x	x	x
P_RTR						x	x	x

SM	M_PF	M_AC	M_DL	M_HF	M_SF	M_APP_EE
HW_BKUP	x		x			x
SW_BKUP	x		x			x
HW_TAPE	x		x			x
HW_RTR		x				
HW_SRV_EE				x		
OS_SRV_EE					x	
UPS_SRV_EE	x		x		x	
DATA_EE			x			
APP_EE						x
AUTH_EE		x				

Figure E.6 – Primary relations in technological evidences.

E.3 Preserving Digital Documents schemes



Digital Documents

Policy: I_DD

	Threats	
T_P1		Fire damage
T_P2		water damage
T_ES1		blackout
T_HW_FAIL		HW failure
T_SW_FAIL		SW failure
T_DT1		Data Destruction
T_DT2		Data Alteration
T_ACC		System Acces

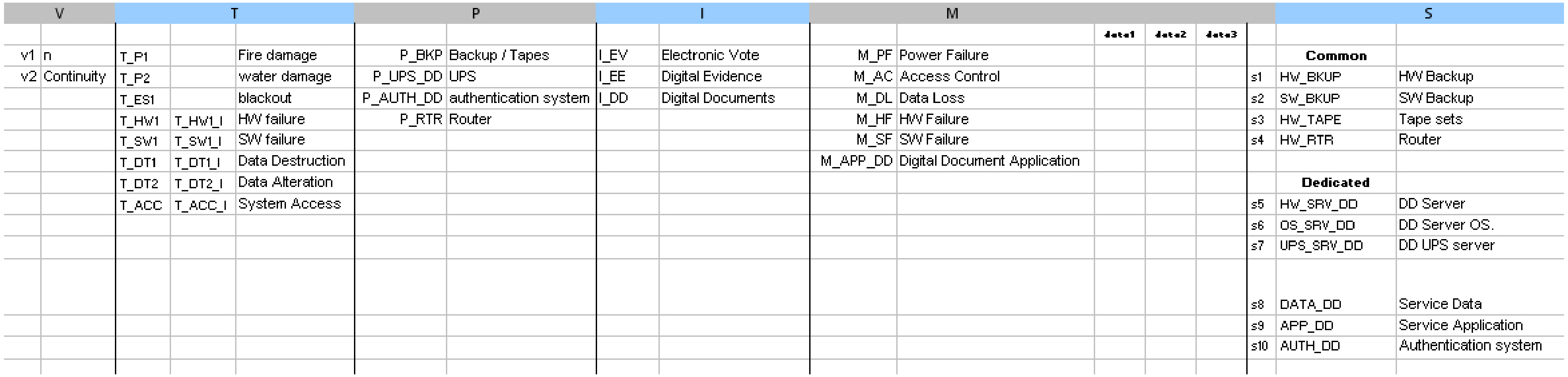
	Measures	
m1	Power Failure	
		UPS_SRV_DD
m2	Control	
		AUTH_DD
		HW_RTR
m3	Data Loss	
		HW_BKUP
		SW_BKUP
		HW_TAPE
m4	HW Failure	
m5	SW Failure	
		HW_BKUP
		SW_BKUP
		HW_TAPE
m6	APP_DD	
		DATA_DD
		APP_DD

Resources	
Dedicated	
HW_SRV_DD	
OS_SRV_DD	O.S. server
UPS_SRV_DD	
DATA_DD	Service Data
APP_DD	Service Application
AUTH_DD	Authentication system
common	
HW_BKUP	HW Backup
SW_BKUP	SW Backup
HW_TAPE	Tape sets
HW_RTR	Router

	Providers	
p1	UPS	P_UPS
p2	authentication system	P_AUTH
p3	Router	P_RTR
p4	Backup / Tapes	P_BKP

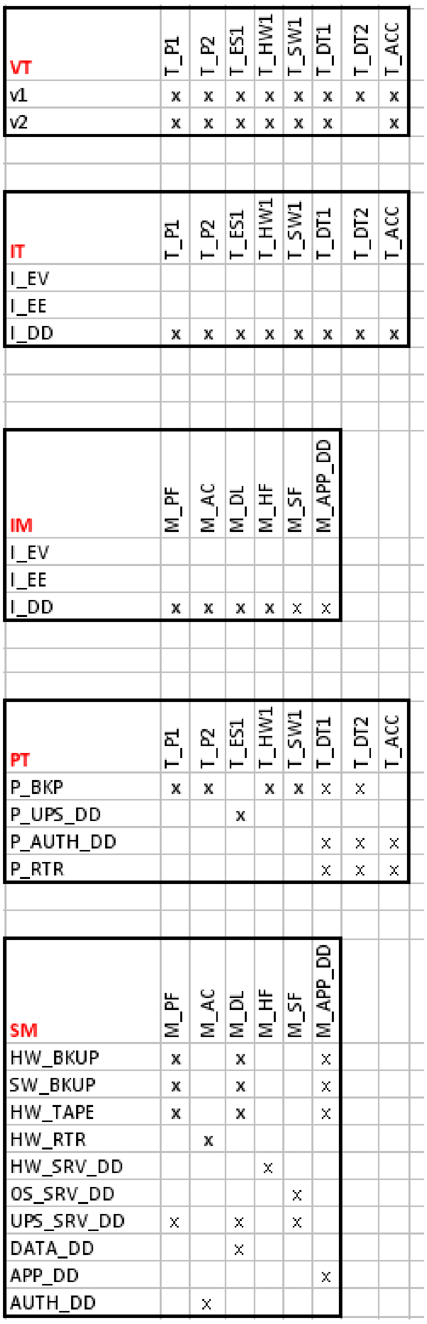
	Providers / Threats			
Access			T_ACC1	
	router	p3		P_RTR
	authentication	p2		P_AUTH
Incendi			T_P1	
	Backup	p4		P_BKP
Inundacio			T_P2	
	Backup	p4		P_BKP
Pw failure			T_ES1	
	UPS	p1		P_UPS
Hw Fail			T_HW_FAIL	
	Backup	p4		P_BKP
Sw Fail			T_SW_FAIL	
	Backup	p4		P_BKP
Data Destruction			T_DT1	
		p2		P_AUTH
		p3		P_RTR
		p4		P_BKP
Data Alteration			T_DT2	
		p2		P_AUTH
		p3		P_RTR
		p4		P_BKP

Figure E.7 – “preserving digital documents” policy.



V		T			P		I		M					S		
											data1	data2	data3			
v1	n	T_P1		Fire damage	P_BKP	Backup / Tapes	I_EV	Electronic Vote	M_PF	Power Failure					**Common**	
v2	Continuity	T_P2		water damage	P_UPS_DD	UPS	I_EE	Digital Evidence	M_AC	Access Control				s1	HW_BKUP	HW Backup
		T_ES1		blackout	P_AUTH_DD	authentication system	I_DD	Digital Documents	M_DL	Data Loss				s2	SW_BKUP	SW Backup
		T_HW1	T_HW1_I	HW failure	P_RTR	Router			M_HF	HW Failure				s3	HW_TAPE	Tape sets
		T_SW1	T_SW1_I	SW failure					M_SF	SW Failure				s4	HW_RTR	Router
		T_DT1	T_DT1_I	Data Destruction					M_APP_DD	Digital Document Application						
		T_DT2	T_DT2_I	Data Alteration											**Dedicated**	
		T_ACC	T_ACC_I	System Access										s5	HW_SRV_DD	DD Server
														s6	OS_SRV_DD	DD Server OS.
														s7	UPS_SRV_DD	DD UPS server
														s8	DATA_DD	Service Data
														s9	APP_DD	Service Application
														s10	AUTH_DD	Authentication system

Figure E.8 – List of concepts, instances and relations.



VT	T_P1	T_P2	T_ES1	T_HW1	T_SW1	T_DT1	T_DT2	T_ACC
v1	x	x	x	x	x	x	x	x
v2	x	x	x	x	x	x		x

IT	T_P1	T_P2	T_ES1	T_HW1	T_SW1	T_DT1	T_DT2	T_ACC
I_EV								
I_EE								
I_DD	x	x	x	x	x	x	x	x

IM	M_PF	M_AC	M_DL	M_HF	M_SF	M_APP_DD
I_EV						
I_EE						
I_DD	x	x	x	x	x	x

PT	T_P1	T_P2	T_ES1	T_HW1	T_SW1	T_DT1	T_DT2	T_ACC
P_BKP	x	x		x	x	x	x	
P_UPS_DD			x					
P_AUTH_DD						x	x	x
P_RTR						x	x	x

SM	M_PF	M_AC	M_DL	M_HF	M_SF	M_APP_DD
HW_BKUP	x		x			x
SW_BKUP	x		x			x
HW_TAPE	x		x			x
HW_RTR		x				
HW_SRV_DD				x		
OS_SRV_DD					x	
UPS_SRV_DD	x		x		x	
DATA_DD			x			
APP_DD						x
AUTH_DD		x				

Figure E.9 – Primary relations in preserving digital documents service.

E.4 List of concept elements

		Values
✓	V1	Information
✓	V2	Continuity

Table E.1 – Values.

		Unintended	Intended
✓ Physical damage			
	Fire	T_P1	
	Water	T_P2	
✓ Essential Services			
	Blackout	T_ES1	
✓ HW			
	Failure	T_HW1	T_HW1_I
✓ SW			
	Failure	T_SW1	T_SW1_I
✓ Data			
	Destruction	T_DT1	T_DT1_I
	Alteration	T_DT2	T_DT2_I
✓ Access			
	System access	T_ACC	T_ACC_I

Table E.2 – Threats.

	Security Providers
	DATA
P_BKP	Backup system
	ELECTRIC POWER
P_UPS_EV	UPS
P_UPS_EE	UPS
P_UPS_DD	UPS
	SYSTEM ACCESS
P_AUTH_EV	Authentication system
P_AUTH_EE	Authentication system
P_AUTH_DD	Authentication system
P_RTR	Router WAN

Table E.3 – Security providers.

		Policies
✓	I_VE	Electronic voting policy
✓	I_EE	Technological evidences policy
✓	I_DD	Preserving digital documents policy

Table E.4 – Policies.

		Measures
✓	M_PF	Power Failure
✓	M_AC	Access Control
✓	M_DL	Data Loss
✓	M_HF	HW Failure
✓	M_SF	SW Failure
✓	M_APP_EV	e-vote application
✓	M_APP_EE	e-evidence application
✓	M_APP_DD	e-document application

Table E.5 – Measures.

		Common
✓	HW_BKUP	Hardware Backup
✓	SW_BKUP	Software Backup
✓	HW_TAPE	Tape sets
✓	HW_RTR	Router
		E-Voting Service
✓	HW_SRV_EV	EV. Hardware Server
✓	OS_SRV_EV	Server OS Software
✓	UPS_SRV_EV	UPS
✓	DATA_EV	Service Data
✓	APP_EV	Service Application
✓	AUTH_EV	Service Authentication
		E-Evidence Service
✓	HW_SRV_EE	EE. Hardware Server
✓	OS_SRV_EE	Server OS Software
✓	UPS_SRV_EE	UPS
✓	DATA_EE	Service Data
✓	APP_EE	Service Application
✓	AUTH_EE	Service Authentication
		E-Document Service
✓	HW_SRV_DD	DD. Hardware Server
✓	OS_SRV_DD	Server OS Software
✓	UPS_SRV_DD	UPS
✓	DATA_DD	Service Data
✓	APP_DD	Service Application
✓	AUTH_DD	Service Authentication

Table E.6 – Resources.

E.5 Primary relations

VT	T_P1	T_P2	T_ES1	T_HW1	T_SW1	T_DT1	T_DT2	T_ACC
v_1	X	X	X	X	X	X	X	X
v_2	X	X	X	X	X	X		X

Table E.7 – Primary VT relation.

IT	T_P1	T_P2	T_ES1	T_HW1	T_SW1	T_DT1	T_DT2	T_ACC
I_EV	X	X	X	X	X	X	X	X
I_EE	X	X	X	X	X	X	X	X
I_DD	X	X	X	X	X	X	X	X

Table E.8 – Primary IT relation.

IM	M_PF	M_AC	M_DL	M_HF	M_SF	M_APP_EV	M_APP_EE	M_APP_DD
I_EV	X	X	X	X	X	X		
I_EE	X	X	X	X	X		X	
I_DD	X	X	X	X	X			X

Table E.9 – Primary IM relation.

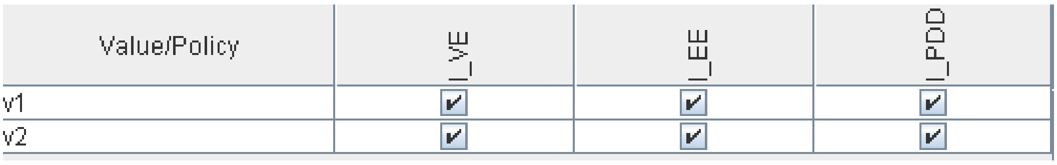
PT	T_P1	T_P2	T_ES1	T_HW1	T_SW1	T_DT1	T_DT2	T_ACC
P_BKP	X	X		X	X	X	X	
P_UPS_EV			X					
P_AUTH_EV						X	X	X
P_UPS_EV			X					
P_AUTH_EV						X	X	X
P_UPS_EV			X					
P_AUTH_EV						X	X	X
P_RTR						X	X	X

Table E.10 – Primary PT relation.

SM	M_PF	M_AC	M_DL	M_HF	M_SF	M_APP_EV	M_APP_EE	M_APP_DD
HW_BKUP	X		X			X	X	X
SW_BKUP	X		X			X	X	X
HW_TAPE	X		X			X	X	X
HW_RTR		X						
HW_SRV_EV				X				
OS_SRV_EV					X			
UPS_SRV_EV	X		X		X			
$DATA\_EV$			X					
APP_EV						X		
$AUTH\_EV$		X						
HW_SRV_EE				X				
OS_SRV_EE					X			
UPS_SRV_EE	X		X		X			
$DATA\_EE$			X					
APP_EE							X	
$AUTH\_EE$		X						
HW_SRV_DD				X				
OS_SRV_DD					X			
UPS_SRV_DD	X		X		X			
$DATA\_DD$			X					
APP_DD								X
$AUTH\_DD$		X						

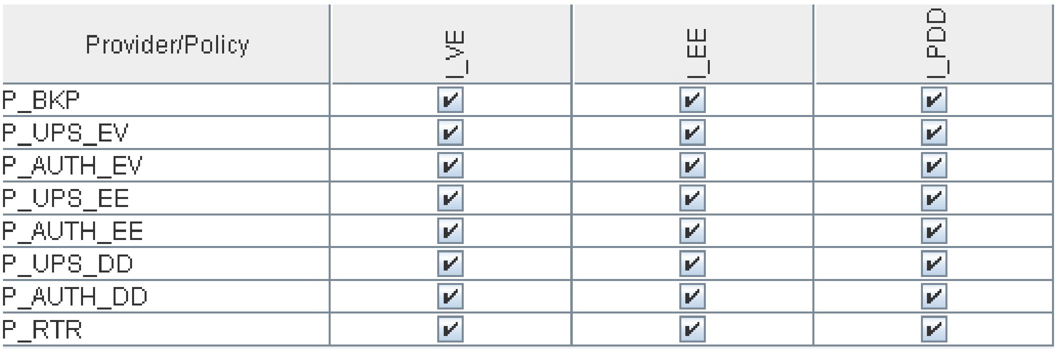
Table E.11 – Primary SM relation.

E.6 Secondary relations inferred



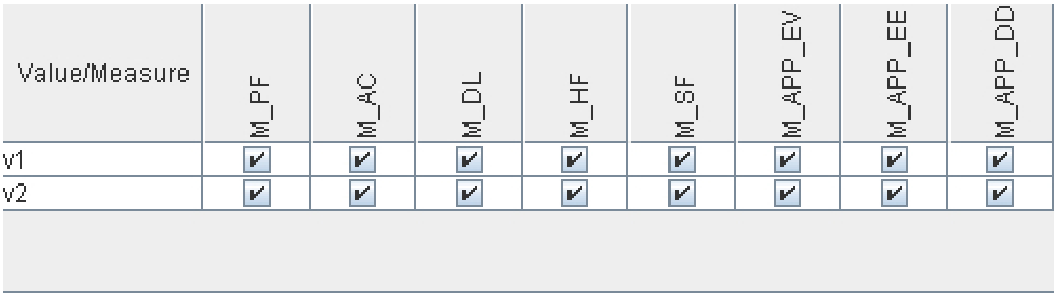
Value/Policy	I_VE	I_EE	I_PDD
v1	☑	☑	☑
v2	☑	☑	☑

Figure E.10 – Value - Policy relation.



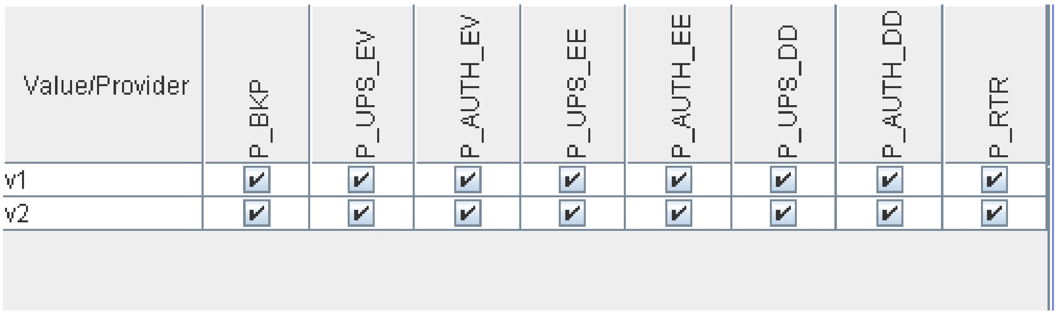
Provider/Policy	I_VE	I_EE	I_PDD
P_BKP	☑	☑	☑
P_UPS_EV	☑	☑	☑
P_AUTH_EV	☑	☑	☑
P_UPS_EE	☑	☑	☑
P_AUTH_EE	☑	☑	☑
P_UPS_DD	☑	☑	☑
P_AUTH_DD	☑	☑	☑
P_RTR	☑	☑	☑

Figure E.11 – Provider - Policy relation.



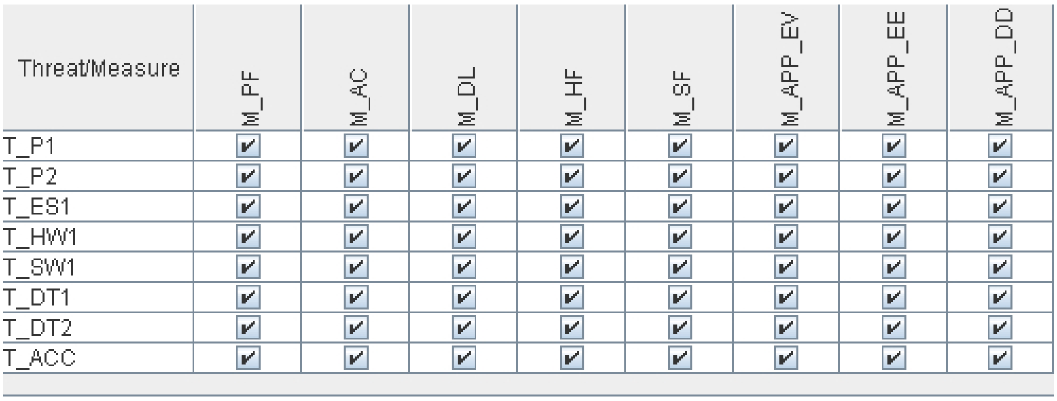
Value/Measure	M_PF	M_AC	M_DL	M_HF	M_SF	M_APP_EV	M_APP_EE	M_APP_DD
v1	☑	☑	☑	☑	☑	☑	☑	☑
v2	☑	☑	☑	☑	☑	☑	☑	☑

Figure E.12 – Value - Measure relation.



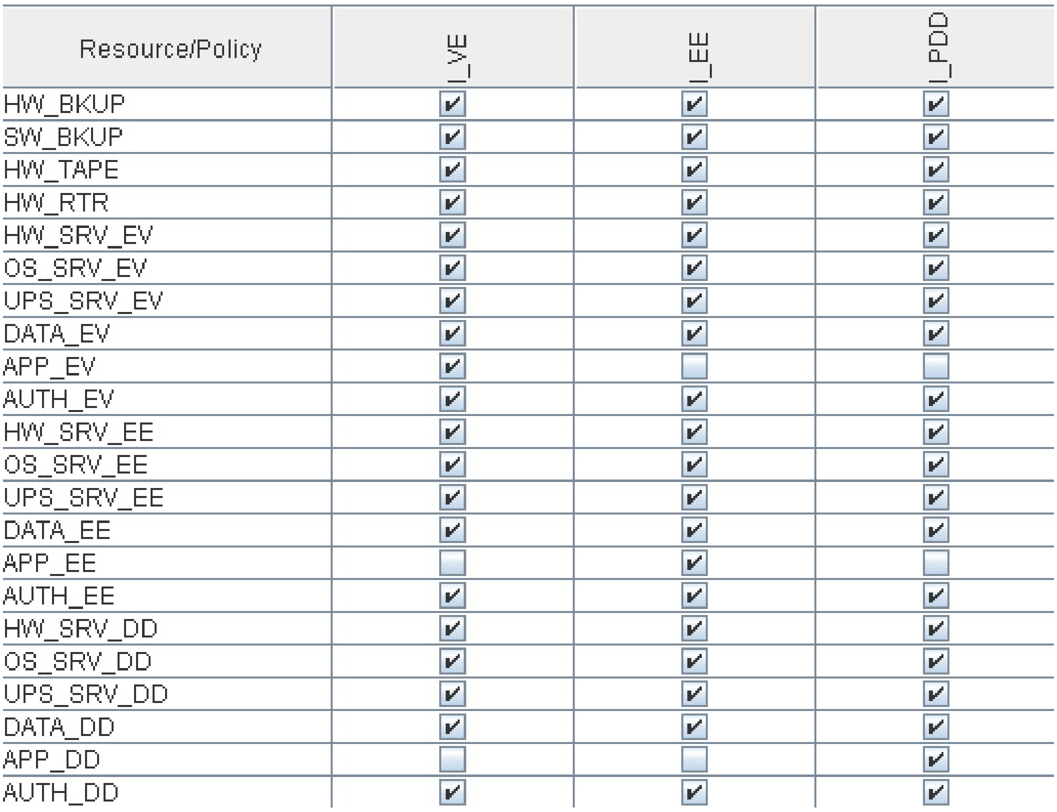
Value/Provider	P_BKP	P_UPS_EV	P_AUTH_EV	P_UPS_EE	P_AUTH_EE	P_UPS_DD	P_AUTH_DD	P_RTR
v1	☑	☑	☑	☑	☑	☑	☑	☑
v2	☑	☑	☑	☑	☑	☑	☑	☑

Figure E.13 – Value - Provider relation.



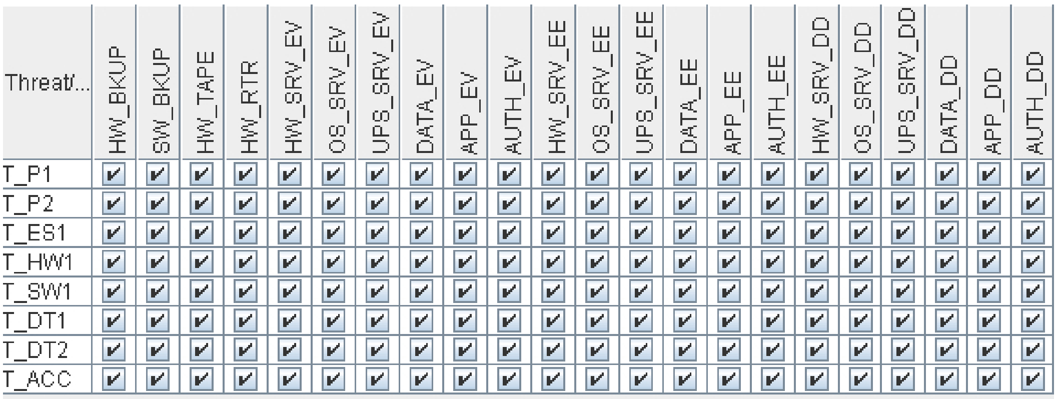
Threat/Measure	M_PF	M_AC	M_DL	M_HF	M_SF	M_APP_EV	M_APP_EE	M_APP_DD
T_P1	☑	☑	☑	☑	☑	☑	☑	☑
T_P2	☑	☑	☑	☑	☑	☑	☑	☑
T_ES1	☑	☑	☑	☑	☑	☑	☑	☑
T_HW1	☑	☑	☑	☑	☑	☑	☑	☑
T_SW1	☑	☑	☑	☑	☑	☑	☑	☑
T_DT1	☑	☑	☑	☑	☑	☑	☑	☑
T_DT2	☑	☑	☑	☑	☑	☑	☑	☑
T_ACC	☑	☑	☑	☑	☑	☑	☑	☑

Figure E.14 – Threat - Measure relation.



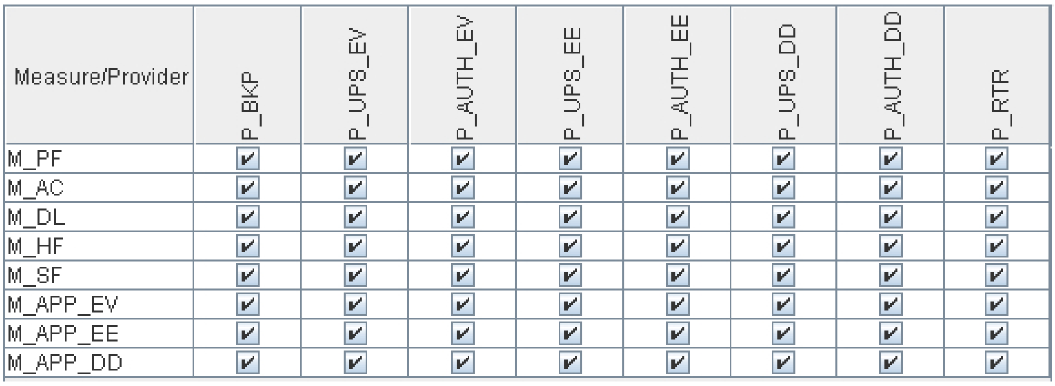
Resource/Policy	I_VE	I_EE	I_PDD
HW_BKUP	☑	☑	☑
SW_BKUP	☑	☑	☑
HW_TAPE	☑	☑	☑
HW_RTR	☑	☑	☑
HW_SRV_EV	☑	☑	☑
OS_SRV_EV	☑	☑	☑
UPS_SRV_EV	☑	☑	☑
DATA_EV	☑	☑	☑
APP_EV	☑	☐	☐
AUTH_EV	☑	☑	☑
HW_SRV_EE	☑	☑	☑
OS_SRV_EE	☑	☑	☑
UPS_SRV_EE	☑	☑	☑
DATA_EE	☑	☑	☑
APP_EE	☐	☑	☐
AUTH_EE	☑	☑	☑
HW_SRV_DD	☑	☑	☑
OS_SRV_DD	☑	☑	☑
UPS_SRV_DD	☑	☑	☑
DATA_DD	☑	☑	☑
APP_DD	☐	☐	☑
AUTH_DD	☑	☑	☑

Figure E.15 – Resource - Policy relation.



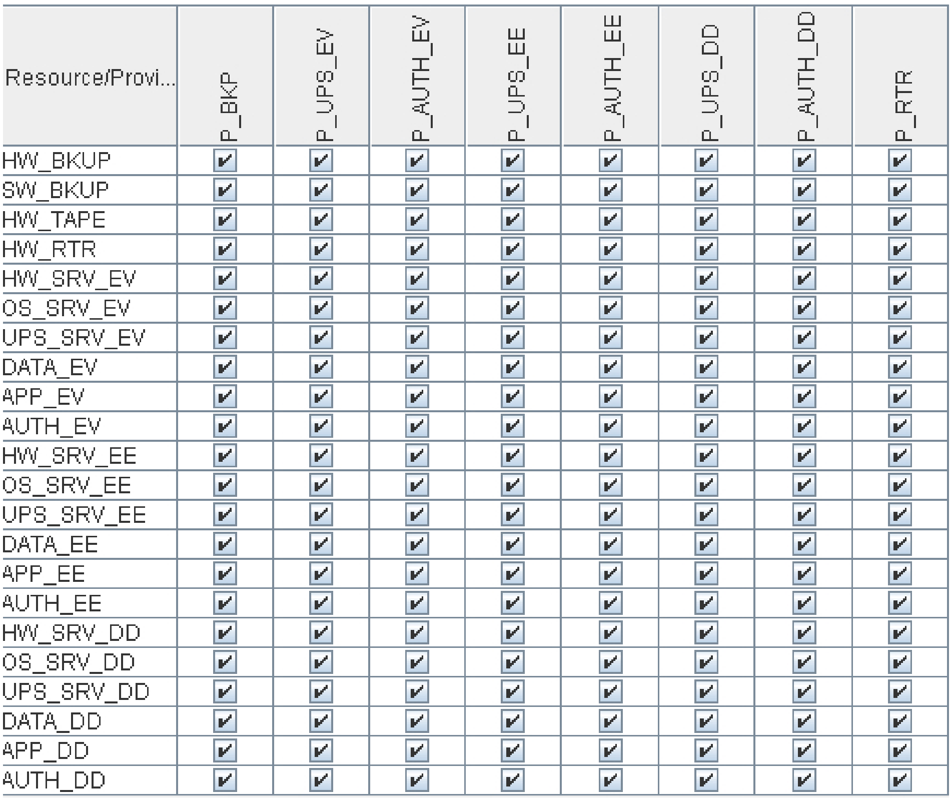
Threat/...	HW_BKUP	SW_BKUP	HW_TAPE	HW_RTR	HW_SRV_EV	OS_SRV_EV	UPS_SRV_EV	DATA_EV	APP_EV	AUTH_EV	HW_SRV_EE	OS_SRV_EE	UPS_SRV_EE	DATA_EE	APP_EE	AUTH_EE	HW_SRV_DD	OS_SRV_DD	UPS_SRV_DD	DATA_DD	APP_DD	AUTH_DD
T_P1	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑
T_P2	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑
T_ES1	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑
T_HW1	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑
T_SW1	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑
T_DT1	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑
T_DT2	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑
T_ACC	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑

Figure E.16 – Threat - Resource relation.



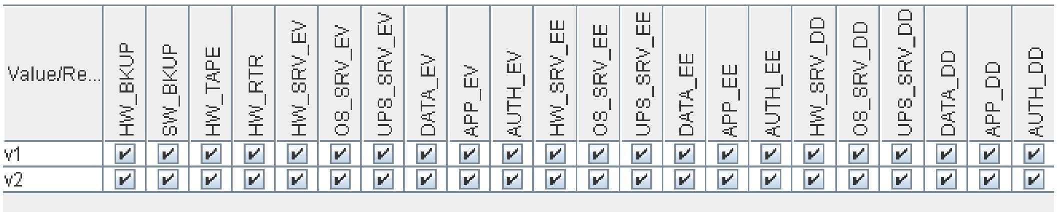
Measure/Provider	P_BKP	P_UPS_EV	P_AUTH_EV	P_UPS_EE	P_AUTH_EE	P_UPS_DD	P_AUTH_DD	P_RTR
M_PF	☑	☑	☑	☑	☑	☑	☑	☑
M_AC	☑	☑	☑	☑	☑	☑	☑	☑
M_DL	☑	☑	☑	☑	☑	☑	☑	☑
M_HF	☑	☑	☑	☑	☑	☑	☑	☑
M_SF	☑	☑	☑	☑	☑	☑	☑	☑
M_APP_EV	☑	☑	☑	☑	☑	☑	☑	☑
M_APP_EE	☑	☑	☑	☑	☑	☑	☑	☑
M_APP_DD	☑	☑	☑	☑	☑	☑	☑	☑

Figure E.17 – Measure - Provider relation.



Resource/Provi...	P_BKP	P_UPS_EV	P_AUTH_EV	P_UPS_EE	P_AUTH_EE	P_UPS_DD	P_AUTH_DD	P_RTR
HW_BKUP	☑	☑	☑	☑	☑	☑	☑	☑
SW_BKUP	☑	☑	☑	☑	☑	☑	☑	☑
HW_TAPE	☑	☑	☑	☑	☑	☑	☑	☑
HW_RTR	☑	☑	☑	☑	☑	☑	☑	☑
HW_SRV_EV	☑	☑	☑	☑	☑	☑	☑	☑
OS_SRV_EV	☑	☑	☑	☑	☑	☑	☑	☑
UPS_SRV_EV	☑	☑	☑	☑	☑	☑	☑	☑
DATA_EV	☑	☑	☑	☑	☑	☑	☑	☑
APP_EV	☑	☑	☑	☑	☑	☑	☑	☑
AUTH_EV	☑	☑	☑	☑	☑	☑	☑	☑
HW_SRV_EE	☑	☑	☑	☑	☑	☑	☑	☑
OS_SRV_EE	☑	☑	☑	☑	☑	☑	☑	☑
UPS_SRV_EE	☑	☑	☑	☑	☑	☑	☑	☑
DATA_EE	☑	☑	☑	☑	☑	☑	☑	☑
APP_EE	☑	☑	☑	☑	☑	☑	☑	☑
AUTH_EE	☑	☑	☑	☑	☑	☑	☑	☑
HW_SRV_DD	☑	☑	☑	☑	☑	☑	☑	☑
OS_SRV_DD	☑	☑	☑	☑	☑	☑	☑	☑
UPS_SRV_DD	☑	☑	☑	☑	☑	☑	☑	☑
DATA_DD	☑	☑	☑	☑	☑	☑	☑	☑
APP_DD	☑	☑	☑	☑	☑	☑	☑	☑
AUTH_DD	☑	☑	☑	☑	☑	☑	☑	☑

Figure E.18 – Resource - Provider relation.



Value/Re...	HW_BKUP	SW_BKUP	HW_TAPE	HW_RTR	HW_SRV_EV	OS_SRV_EV	UPS_SRV_EV	DATA_EV	APP_EV	AUTH_EV	HW_SRV_EE	OS_SRV_EE	UPS_SRV_EE	DATA_EE	APP_EE	AUTH_EE	HW_SRV_DD	OS_SRV_DD	UPS_SRV_DD	DATA_DD	APP_DD	AUTH_DD
v1	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑
v2	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑

Figure E.19 – Value - Resource relation.

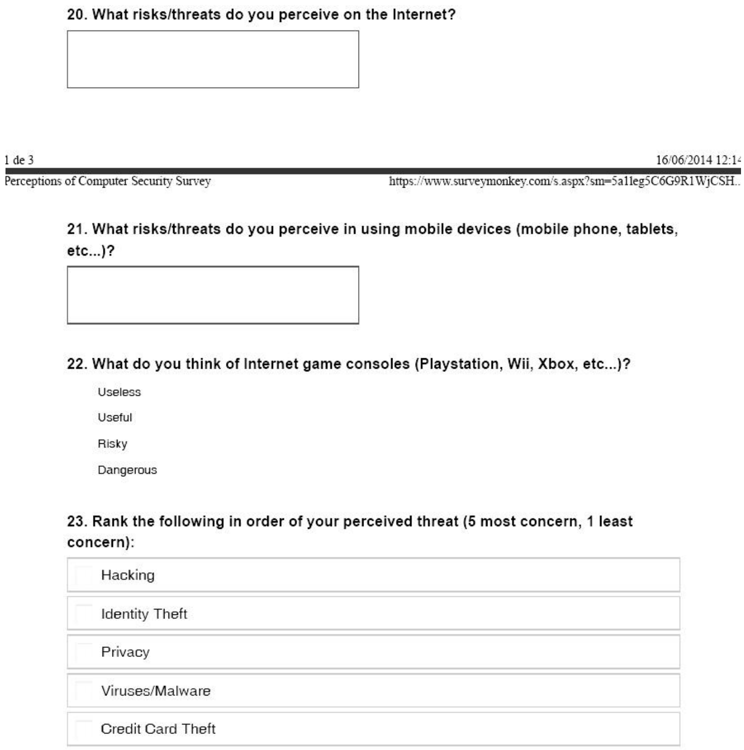
APPENDIX F

PEOPLE PERCEPTION OF SECURITY

A research with Dr. Stephen Cheskiewicz was carried out about people's concern on Internet security. A survey was published in surveymonkey in two languages (English and Spanish) and spread around the world. The result was 1622 answered surveys from a wide range of people.

Part of this survey reveals the most important concerns of people, and therefore, the values to be considered in computer security at individual level.

The questions relevant to individual computer security are number 20 and 23 (Figure F.1).



20. What risks/threats do you perceive on the Internet?

1 de 3 16/06/2014 12:1

Perceptions of Computer Security Survey https://www.surveymonkey.com/s.aspx?sm=5a1leg5C6G9R1WjCSH...

21. What risks/threats do you perceive in using mobile devices (mobile phone, tablets, etc...)?

22. What do you think of Internet game consoles (Playstation, Wii, Xbox, etc...)?

Useless

Useful

Risky

Dangerous

23. Rank the following in order of your perceived threat (5 most concern, 1 least concern):

Hacking

Identity Theft

Privacy

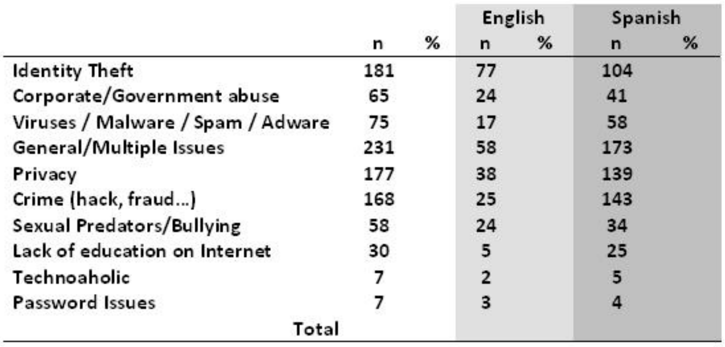
Viruses/Malware

Credit Card Theft

Figure F.1 – Surveymonkey questions.

F.1 Question number 20

That was an open-ended question. The analysis of the answers is made based on coding the answers. We tried to identify trends also. People identify multiple issues, but the most important are related to identity and privacy. On the family sphere, there is a significant number responders that identified sexual and bulling issues related to children (Figure F.2).



			English		Spanish	
	n	%	n	%	n	%
Identity Theft	181		77		104	
Corporate/Government abuse	65		24		41	
Viruses / Malware / Spam / Adware	75		17		58	
General/Multiple Issues	231		58		173	
Privacy	177		38		139	
Crime (hack, fraud...)	168		25		143	
Sexual Predators/Bullying	58		24		34	
Lack of education on Internet	30		5		25	
Technoaholic	7		2		5	
Password Issues	7		3		4	
Total						

Figure F.2 – Analysis of question number 20.

F.2 Question number 23

The analysis of that questions revealed, once again, that people's concern is very similar. Highest perceived threat is hacking and people feel their privacy is in risk with those technologies (Figure F.3).

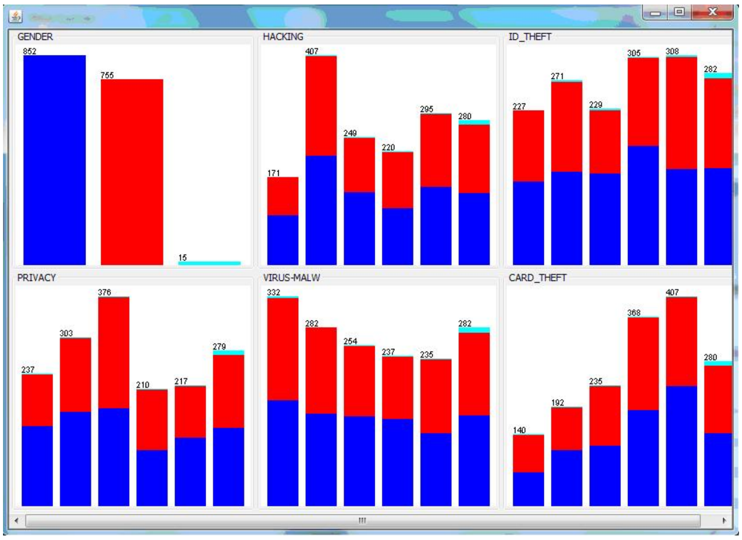


Figure F.3 – Analysis of question number 23.

F.3 Conclusions

The values that a person wants to protect individual level in the field of computer security are different than the ones in computer security applied to an organization. People are concerned the most with privacy and identity thief. Indeed, identity thief, besides its economic side, is close related to privacy.

APPENDIX G

EXAMPLE OF LIGHTWEIGHT SECURITY MODEL

An organization has a IT infrastructure that could be modeled with 3 systems (Figure: G.1). One is the server system, the second system is the set of workstations for users and the third is the connexion of the LAN with Internet. The granularity is not the same. The desktops could be several, while the Internet connection could be a single router or a whole DMZ system. The server, again, could be a single computer or a whole set of servers in some cluster configuration.

For every system described, the time points and threshold values are shown in Table G.1. For simplicity in this example, it is supposed the security level of the three systems is good at January 1st. We want to know how security level change into following months. The values of t_1 and t_f are chosen according to the knowledge and expertise provided by the security modeler.

	t_0	t_1	t_f	THg	THc	THb
Server	Jan 1st	Apr 1st	Jun 1st	0.80	0.40	0.20
Desktop	Jan 1st	Feb 1st	Feb 15th	0.80	0.40	0.20
Router	Jan 1st	Mar 1st	Jun 1st	0.80	0.40	0.20

Table G.1 – Values.

According to the security model, there are 3 security level functions. The security function $S(t)$ is fitted using straight lines (equation G.1). The behavior of all

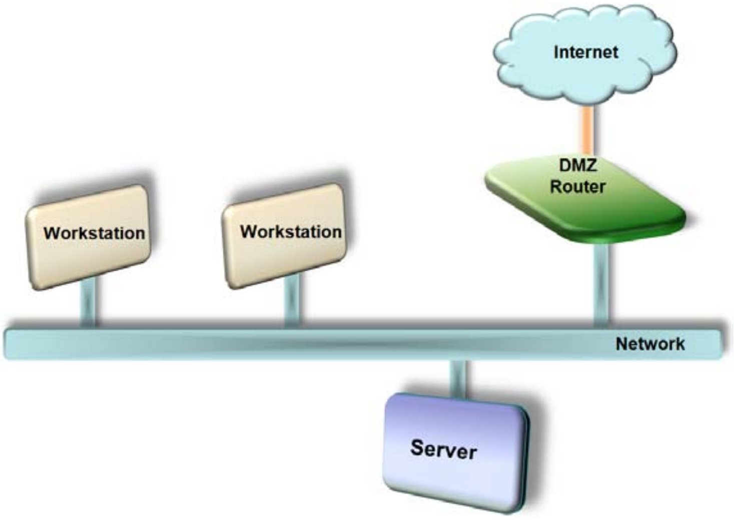


Figure G.1 – IT infraestructure.

systems together for 6 months are shown in Figure G.5. The behavior of a single system is shown in Figures G.2,G.3 and G.4.

$$S(t) = \begin{cases} 1 & t < t_0 \\ THg - \frac{(THg - THc)}{(t_1 - t_0)}(t - t_0) & t \in [t_0, t_1) \\ THc - \frac{(THc - THb)}{(t_f - t_1)}(t - t_1) & t \in [t_1, t_f) \\ THb & t \geq t_f \end{cases} \tag{G.1}$$

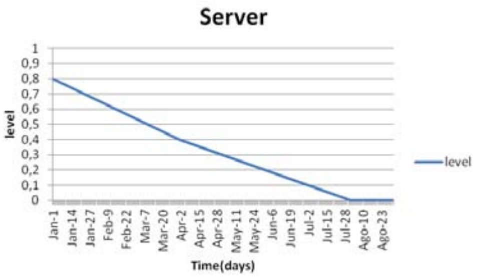


Figure G.2 – Server system security level.

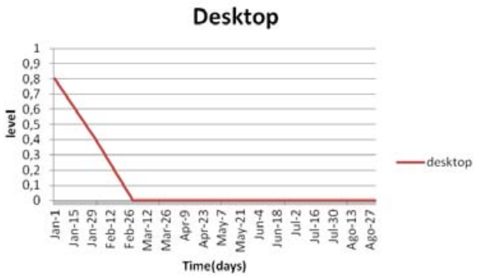


Figure G.3 – Desktop system security level.

According to the settings (Table G.1) the security of desktops falls very quickly. Hence our efforts will focus on that system. The criteria used to make reinforcement is at the time point that security level is close to THb. The security level chosen is 0,3. Therefore, system is checked in dates February 9th and March 17th. The Figure G.6 shows the security level variation for the first four months with the reinforcement made on that dates.

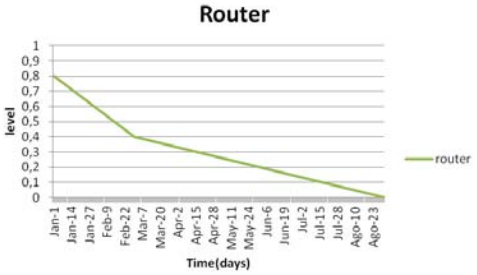


Figure G.4 – DMZ system security level.

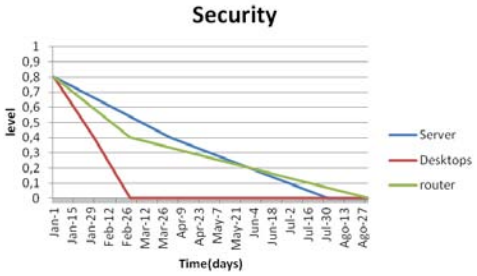


Figure G.5 – Security level.

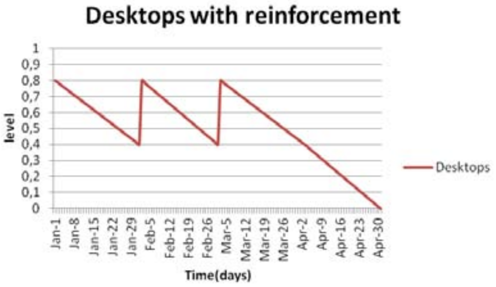


Figure G.6 – Security level with reinforcement.

G.1 Composition

Considering the three subsystems as a whole, the logical connections are shown in Figure G.7. In this scenario, the security level function obtained is drawn in Figure G.8. The security function of the whole system, according the logical design, is obtained applying the formula:

$$S(t) = minimum\left\{\frac{Sa(t)+Sb(t)}{2}, Sc(t)\right\}$$

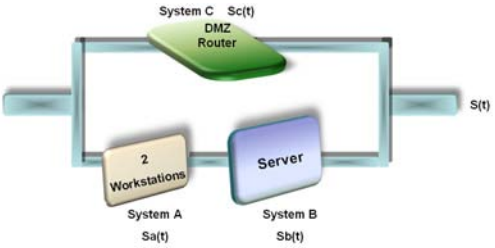


Figure G.7 – Composition of the sub-systems.

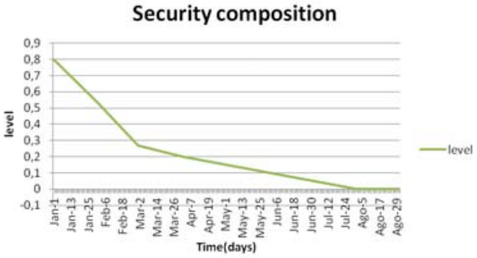


Figure G.8 – Security level of the whole system.

AFTERWORD

If you reached this last page, I am thankful for your interest and patience. Any questions that the research has raised, I'll be glad to try to explain. Please, feel free to contact me anytime at miquel.colobran@uab.cat.

Aquesta memòria ha estat escrita amb LaTeX 2$_\varepsilon$[1] per l'autor[2].

[1] LaTeX 2$_\varepsilon$és una extensió de LaTeX, una col·lecció de macros escrites en TeX.

[2] Usant TeXnicCenter un entorn de desenvolupament – IDE – lliure (http://www.texniccenter.org/)