Aspects and Requirements Domains Applied to Computer Security

Ausberto S. Castro Vera, Luis A. Rivera Escriba, Camila P. G. Morelli
UENF - Universidade Estadual do Norte Fluminense Darcy Ribeiro
Centro de Ciência e Tecnologia, LCMAT, Ciência da Computação
Av Alberto Lamego, 2000 – Parque Califórnia
28013-602 Campos, RJ – Brazil
ascv@uenf.br, rivera@uenf.br, camila.usagi@gmail.com

Abstract—Consider developing safety critical systems is a very complex problem yet. Consider safety aspects in requirements engineering can be an interesting solution. The aim of this paper is to present a methodology of requirements elicitation to development of security computer systems based in aspect-oriented technique. Is show the concepts of requirements engineering, security, requirements domain and aspects and also is shown some examples on how to use this methodology.

Index Terms—Computer security, Requirements Engineering, Requirement domains, requirements, aspects

1 Introduction

Critical systems are these whose failure or malfunction may result in high economic losses, damage to the environment or loss of human life [SOM 07]. Systems building by people, to be used by people, management by people with people informations and with high complexity, naturally represent a critical system. In the "primitive" Software Engineering, early for over thirty years, it was not common to deal with critical systems in developing computer systems. A simple way to check this is by reviewing the content of the basic texts of Software Engineering and Systems Engineering. In the last two decades, however, increasingly has considered the issue of critical systems more deeply in professional journals and inclusion in textbooks chapters dealing with critical systems, safety engineering, reliability engineering, engineering protection, socio-technical systems, risk and cyber risk, risk management, security and auditing, security policy, security engineering software, etc.[ALL 08], [SOM 11], [DEL 11].

With the arrival of the Internet and the rapid growth of information technology, computer systems have become critical systems, mainly because they contain sensitive information from people (personal information form, databases, emails, passwords, etc.). In this context, the concept of security, first was implemented in the maintenance process when the system was already operating. After, has to be considered in the intermediate stages of the development process (design and coding) [BAN 06]. Today, it is imperative to consider this in the early stages of development (analysis and design). For this, the classical methodologies and recent development must adapt and consider the concept of criticality of new computer systems. The purpose of this paper is to contribute with some methodological concepts in the implementation of security properties in the early stages of the process for developing computer systems.

The layout of the paper is as follows: Section 2 presents the related literature concerning Requirements Engineering in Security Engineering. Section 3, presents an overview of the Requirements Domain and Aspects. In Section 4 a proposed methodology of requirements elicitation. Section 5 presents examples of aspects and requirements domains together a case study on an academic ERP, and finally the conclusion in Section 6.

2 Requirements Engineering in Security Engineering

Nowadays, plan, construct, organize, operate and use the computer systems without the safety concept in mind is very high risk, and the implementation of security should be considered a fact, not to be discussed but to be necessarily implemented. Incorporate security into every stage of System Development Life Cycle (SDLC) should be imperative.

Whereas many vulnerabilities of the system may have its origin in the processes and practices used in its creation, software and computer systems engineers, should consider safety practices in the early stages of development: analysis and design, [GOE 06], [BAN 06], [MEA 05], [SOL 13]. This study corresponds to the Requirements Engineering and Security Engineering.

2.1 Requirements Enginnering

To better understand the concept of Requirements Engineering, we need to know two previous definitions: requirement and stakeholder. A **requirement** is *"a condition or capability that must be met or possessed by a system, product, service, result, or component to satisfy a contract, standard, specification, or other formally imposed document. Requirements include the quantified and documented needs, wants, and expectations of the sponsor, customer, and other stakeholders"* [ISO 10]. For example, are

requirements: database centralized, wireless transmission, security, cloud store, etc.

A **stakeholder** is *a person, group of persons or an organization that has an influence (positive or negative), directly or indirectly, on the requirements of a system*. For example, end users, business managers, network administrator, IT department, financial department, etc.

Requirements Engineering is *a systematic and disciplined approach to the specification and management of system requirements with the following goals: a) knowing the relevant requirements, achieving a consensus among the stakeholders about these requirements, documenting them according to given standards, and managing them systematically. b) understanding and documenting the stakeholders' desires and needs, they specifying and managing requirements to minimize the risk of delivering a system that does not meet the stakeholders' desires and needs* [POH 11].

2.2 Security Engineering

Security is a term used for many things (government, people, buildings, computers, etc.) that sometimes involves multidisciplinary skill since encryption algorithms and hardware devices to the knowledge of other areas (economics, applied and organizational psychology, laws), so that according to [JAC 11], its use has turned it into something ambiguous. For example, [WHI 12], define *safety* as *the quality or state of being secure - to be free from danger*. In other words, *protection against adversaries*. Terms such as computer security, information security, information systems security, information assurance, governance of the security, safety engineering, safety engineering information, etc. are defined in different ways. Sometimes, the definition depends on the some acceptable attributes (processing power, usability, operational reliability), other than a kind of perspective (concept, function areas). In this sense, we do not intend to unify definitions, however, use one that serves to Requirements Engineering.

Considering [PFL 07], [ALL 08] and [JAC 11], we define *computer security* as the study and goal of three fundamental aspects of any computer system: confidentiality, integrity and availability. A *secure system* is one that meets the requirements of confidentiality, integrity and availability. The following three concepts define the information system security. The first three concepts are used to the security standard definition of systems, for security of information systems and computational services.

- *Confidentiality*: property of the system (information or software) is protected, not available or disclosed from unauthorized individuals, entities, processes or systems.
- *Integrity*: property or quality of a system to be accurate, complete, incorrupted and free from unauthorized changes during development or execution process.
- *Availability*: property of a system to be operational (usable) and accessible by intended and authorized users (people, systems or processes) without interference or obstruction, and be accessed in the required way.

Additionally, it is considered two complementary concepts associated with the security information system ([ABN 06], [WHI 12], [PFL 07], [HAR 10])

- *Authenticity*: property that determines the truth of an attribute of a given system or entity. It is assurance that an entity (data, message, file, system, process, etc.) comes from announced sources and that was not the target of mutations along a process.
- *Legality*: property through which ensures that an entity is in accordance with the law (rules, standards, regulations, etc.).

According Anderson [AND 08], *Security Engineering* is about building systems to remain dependable in the face of malice, error, or mischance. As a *security engineering discipline*, it focuses on the tools, processes, and methods needed to design, implement, and test complete systems, and to adapt existing systems as their environment evolves.

The major practical problem of Security Engineering for computer systems there is still a clear and well-defined methodology on what to implement, how to implement and when to implement through a complete development process (planning, analysis, design, implementation, maintenance). Due to the complexity and size of the security industry today, explains [SHO 08], the solutions (products and services) are always reactive and not proactive nature. However, changes are happening and in this sense that this proposal seeks to consider the security problem in the analysis process of secure systems development.

3 Requirements Domain and Aspects

In the literature we found that several definitions have been developed to understand the term *aspect*, from the earliest definitions related to aspect-oriented programming, until the latest related requirements analysis. In [BAN 06], a *aspect* in requirements is defined as *a concern that crosscuts requirements artifacts; an aspect in architecture is a concern that crosscuts architectural artifacts.*.

Trying to clarify the different meanings with a focus on Requirements Engineering, Kaindl [KAI 08] defines the term *aspect* as *a means to deal with crosscutting concerns, and not as a synonym of "crosscutting concern"*. Very important here, is the observation that a crosscutting concern identified in a requirements representation is not necessarily a crosscutting concern inside the software, i.e., in design and implementation.

3.1 Requeriments domain

To follow the reasoning of Kaindl [KAI 08], in [CAS 13] is defined a *requirements domain* as a system or set of requirements D that can be decomposed (partitioned) into components D_i, so that:

- D = $\cup_{i=1}^{n} D_i$ (union of components)
- $D_i \cap D_j = \emptyset$ if $i \neq j$ (disjunct components)

Presenting this definition as a mathematical formalism (structure of system or set) is important for the following reasons:

- *Completeness and interdependence*. The domain as the union of components denote the type of interdependency between components (system): the absence of any component of the system turn unfeasible the existence of such system being built.
- *Completeness and well-definedness*. Presents a formal framework well defined (set): all elements defined, well-defined common property or absence of ambiguity.
- *Uniqueness*. Each requirement formally belong to a single component: avoids duplication of requirements in the same domain

As an example of domain requirements we can consider the set of all requirements of any computer system that can be partitioned into the following components: hardware requirements, software requirements, database requirements, documentation requirements, requirements of people, and methodologies requirements.

Requirements domains can be constructed in many ways, for example, from viewpoints [RAS 02], using functional decomposition [YU 04], using tools such as Theme [BAN 04], etc.

3.2 Aspects

To building our methodology to requeriments elicitation applied to computational security, we define a **aspect** as *a requirements set that cuts (crosses, is part of, intersects) all components of a requirements domain* (as illustrated in the Fig.1(a)), i.e. the intersection of the aspect with each of the components of the domain is not empty. The requirements in the elicitation process are grouped into domains that are partitioned into components. In general, from a domain of n components: $C_1, C_2, C_3, ..., C_n$ we can establish an aspect as being subsets not empty of each subcomponent C_i. The Fig.1 b), illustrates the concept of aspect of a way more realistic because each component represent sets of requirements of different natures and sizes, and therefore, the aspects on each component represent different subsets of requeriments

4 The Methodology for Elicitation of Requirements

A critical problem in the early stages of the development of secure systems is how to perform each step of a particular methodology for requirements elicitation that include different security properties.

The definition of requirements domain associated with an aspect brings an implicit methodology as part of the process of requirements elicitation considering only one property or characteristic of the system. This is a security property. This metodology is AORE-type

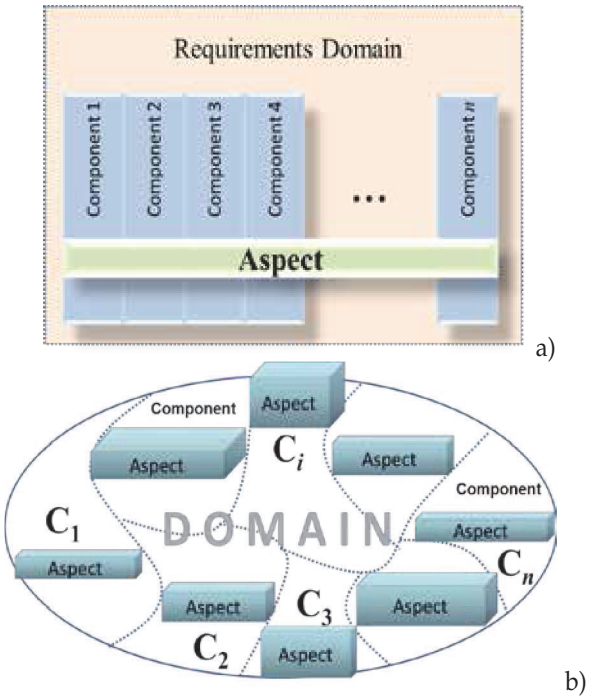

Fig. 1. The concept of *aspect* and *domain*

(Aspect-Oriented Requirements Engineering) [CAS 13]. The methodology (see Fig.2), as a process, includes:

1) determine a transversal aspect (security property),
2) determine a requirements domain (and their components),
3) execute the elicitation of requirements for only the transversal aspect to each component of the domain
4) elicitation of requirements
5) update the requirements library (definition and specification using a requirements management tool)

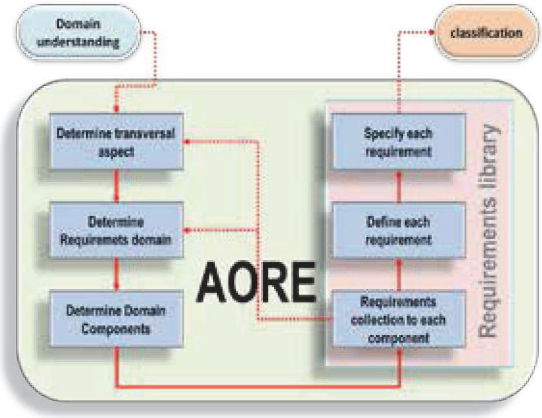

Fig. 2. The AORE-methodolgy

We emphasize here that the transverse aspect of interest is any property, any feature, or item object representing computer security. A property could be integrity or

confidentiality, a feature could be a firewall, a object could be a security camera.

To illustrate the use of the methodology explained, we consider two examples: a standard system computer and a Finacial ERP system. The first (Fig.3 b)), *standard computer system*, consider a requirements domain composed by six core components of the system: hardware (servers, network, terminals), software (operational system, firewall, antivirus), people (access with password, training, ergonometric devices), database (backup, integrity, redundancy control), documents (printed and digital, identification by author) and procedures or methodologies (hierarchy of users, object-oriented development). As second step, is determine a transversal aspect as a set of requirements on security system of each component, i.e. security requirements on hardware, security requirements on software, security requirements on documents, etc. In the third step, are executing the elicitation of requirements in the aspect defined in the previous step and classification of security requirements. In the Financial module of a Enterprise Resource Planning (ERP), requirements domain is formed by the set of requirements on treasury, accounts payable, accounts receivable, financial management, financial investments, financial funding, advancement control and access permission (see Fig.3 b)).

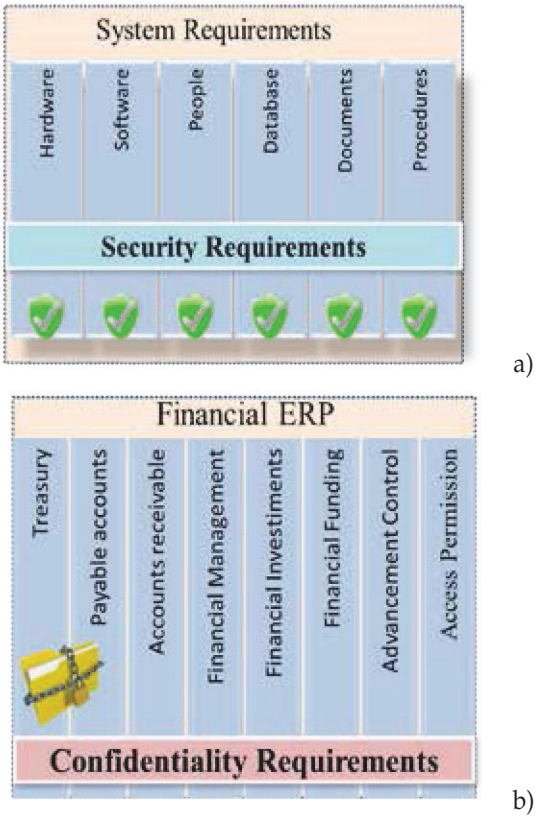

Fig. 3. Examples of requirements domain with a transversal aspect

In the second example, the Financial subsystem in an Enterprise Resource Planning, were identified eight components to form the requirements domain. To the aspect transversal of security it was selected the confidentiality property. The next step (requirements elicitation) was executed was performed by determining all requirements related with the confidentiality property for each component.

- Treasury
  - Access to the system and applications through specific login and password
  - Reports are not available to other departments
- Payable accounts
  - Reports are printed without vendor names, vendor code only
  - Payment dates are not available to other departments
- Accounts receivable
  - Reports available only to the finance department
  - Amounts available only in printed reports
  - Customer list available only by client code (no names)
- Financial Management
  - Confidential management and control of budgets
- Access Permission
  - Visible list of users accessing the module
  - Registration users only for Financial Management

5 Applying the Methodology: Aspects and Requirements Domains

The following are some examples of aspects and requirements domains that may be used to implement engineering requirements considering a safety aspect as the crosscutting concerns in the process of elicitation and analysis requirements.

5.1 Examples of transversal aspects

We can start any secure system design considering as examples of cross-cutting concerns the following: security, confidentiality, integrity, availability, accountability, dependability, reliability, privacy, risk, identification, authentication, authorization, accountability, non-repudiation, cryptography, control, etc..

5.2 Examples of requirements domains with their components

- *System basic components*: hardware, software, people, database, methodologies or procedures
- *Common Body of Knowledge* (CBK): Access control, application and systems development security , business continuity planning and disaster recovery planning, cryptography, information security and risk management, legal, regulations, compliance

and investigations, operational security, physical security, security architecture and models, telecommunications and network security [JAC 11].

- *Applications*: operational system, management system, internet, database,office automation software, etc.
- *Subsystems*: network, database, business management system, financial management, etc.
- *Information systems*: physical security, communications security, emission security, computer security, network security, information security, people security.
- *Information systems*: physical security, network security, server security, application security, data security, system reliability.
- *ISO 27002*: policies, organization, assets, human resources, environment, communications and operations management, access control, acquisition, development and maintenance, incident management, business continuity, flexibility, etc.
- *Processes of information and communication technology*: activities, processes, domains
- *Resources of information and communication technology*: applications, information, infraestructure, people

5.3 Case Study: An Academic ERP

In [MOR 13] was presented a case study based on [CAS 13] about the application of the methodology to security requirements elicitation to develop an Academic Enterprise Resource Planing (AERP) system using domains and some security properties as aspects. Was chosen as aspect the confidentiality property. The full system have four domains (see Fig. 4): Academic Management, Technological Management, Enterprise Management, and Strategic Management. Each domain can be decomposed into subdomains in a top-down way.

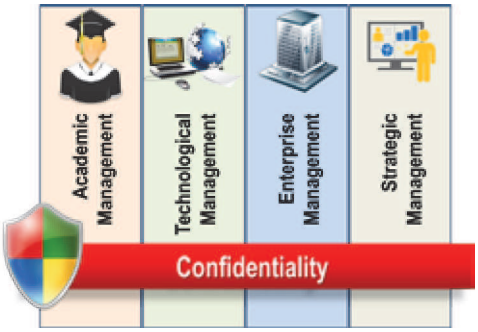

Fig. 4. Academic ERP domain

The Academic Management domain have four subdomains: Undergraduate, Graduate, EAD and Library, as show in Fig. 5.

Finally, the Undergraduate domain is decomposed into five atomic components: Student, Teacher, Administrative Staff, Student Assistance and Academic Record (see Fig. 6). At this point we can now apply the AORE

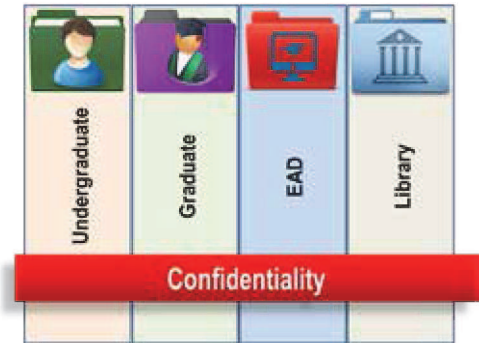

Fig. 5. Academic Management domain

methodology for requirements elicitation with the property of confidentiality as a cross-cutting issue. In a first step was considered approximately ten requirements on confidentiality for each component.

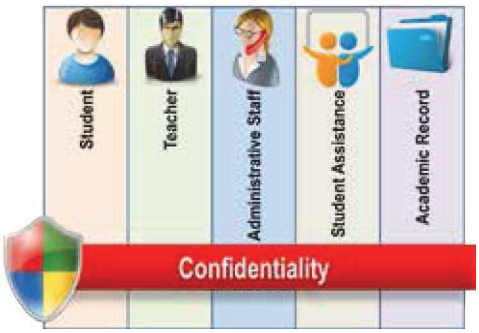

Fig. 6. Undergraduate domain

The requirements library, modelling and documentation of Academic ERP was made with Visual Paradigm (Requiremet Grid, class diagram and report).

6 Conclusion

Here was presented an overview of a methodology for requirements elicitation and classification within the unified process of Requirements Engineering and Security of computer systems, emphasizing the concepts of requirements domains and aspects. This work is part of a research project being developed at UENF together with a tool to requirements management (FGR)[GUA 14], and a library of standard requirements for development of secure systems.

References

[ABN 06] ABNT NBR ISO/IEC 27001. Information Technology - Security Techniques - Information security management systems - Requirements. Technical report, Associação Brasileira de Normas Técnicas, 2006. One citation in section 2.

[ALL 08] Allen, J. H., Barnum, S., Ellison, R. J., McGraw, G., and Mead, N. R. Software Security Engineering. Addison-Wesley, 2008. 2 citations in sections 1 and 2.

[AND 08] Anderson, R. Security Engineering - A Guide to Building Dependable Distributed Systems. Wiley Publishing, Inc., 2nd edition, 2008. One citation in section 2.

[BAN 04] Baniassad, E. and Clarke, S. Theme: An approach for aspect-oriented analysis and design. In Society, I. C., editor, ICSE 04 - Proceedings of the 26th International Conference on Software Engineering, pages 158 - 167, 2004. One citation in section 3.

[BAN 06] Baniassad, E., Clements, P. C., AraÂ´ujo, J., Moreira, A., Rashid, A., and Tekinerdogan, B. Discovering early aspects. IEEE Software, 2006. 2 citations in sections 1 and 2.

[CAS 13] CASTRO VERA, A. S., Engenharia de Requisitos e Segurança: uma metodologia orientada a aspectos. (Notas de Pesquisa). UENF-CCT-LCMAT-CC. 2013 3 citations in sections 2, 3, and 5.

[DEL 11] Deloitte. Raising the bar - 2011 TMT Global Security Study - Key Findings. Technical report, Deloitte Touche Tohmatsu Limited, 2011. One citation in section 1.

[GOE 06] Goertzel, K. M., Winograd, T., McKinley, H. L., Holley, P., and Hamilton, B. A. Security in the Software Lifecycle: Making application development processes - and the software produced by them - more secure. Technical report, U.S. Departament of Homeland Security, 2006. One citation in section 1.

[GUA 14] GUALHANO, M. A., and Castro Vera, A. S., and Medeiros Junior, R. C., Biblioteca de Requisitos sobre propriedades de segurança em Sistemas Computacionais. ENCompIF 2014 - II Encontro Nacional de Computação dos Institutos Federais, XXXIV Congreso da Sociedade Brasileira de Computação, Brasilia, DF, 2014. One citation in section 5.

[HAR 10] Harris, S. CISSP All-in-One Exam Guide. McGraw Hill, 5th edition, 2010. One citation in section 2.

[ISO 10] ISO-IEC-IEEE 24765. Systems and software engineering - vocabulary. Technical report, ISO-IEC-IEEE Computer Society, 2010. One citation in section 1.

[JAC 11] Jacobs, S. Engineering Information Security: The Application of Systems Engineering Concepts to Achieve Information Assurance. Wiley-IEEE Press, 1st edition, 2011. 2 citations in sections 2 and 5.

[KAI 08] Kaindl, H. What is an aspect in aspect-oriented requirements engineering? In: Halpin, T., Krogstie, J., and Proper, E., editors, Proceedings of EMMSAD 2008, Thirteenth International Workshop on Exploring Modeling Methods for Systems Analysis and Design, pages 164 - 170, 2008. One citation in section 2.

[MEA 05] Mead, N. R., Hough, E. D., and II, T. R. S. Security Quality Requirements Engineering (SQUARE) Methodology. Technical report, Carnegie Mellon Software Engineering Institute, 2005. One citation in section 1.

[MOR 13] MORELLI, C. P. G., and Castro Vera, A. S., Aplicação de uma metodologia AORE para elicitação de Requisitos de Segurança, Trabalho de Conclusão de Curso Ciência da Computação, UENF-CCT-LCMAT-CC, Campos, RJ, 2013 One citation in section 5.

[PFL 07] Pfleeger, C. P. and Pfleeger, S. L. Security in Computing. Pearson Education Prentice Hall, 4th edition, 2007. One citation in section 2.

[POH 11] Pohl, K. and Rupp, C. Requirements Engineering Fundamentals: a study guide for the Certified Professional for Requirements Engineering Exam. Rocky Nooc Inc, 2011. One citation in section 2.

[RAS 02] Rashid, A., Sawyer, P., Moreira, A., and Araújo, J. Early aspects: a model for aspect-oriented requirements engineering. In Proceedings of the IEEE Joint International Conference on Requirements Engineering (RE 02), pages 199 - 202, 2002. One citation in section 3.

[SHO 08] Shostack,A., Stewart, A., A New School of Information Security, Addison-Wesley Professional, 1th edition, 2008 One citation in section 2.

[SOL 13] Solinas, M., Antonelli, L., Fernández, E., Software Secure Building Aspects in Computer Engineering, IEEE Latin America Transactions, Vol. 11, No. 1, Feb. 2013. One citation in section 1.

[SOM 07] Sommerville, I., Engenharia de Software, 8ed., São Paulo: Pearson Prentice Hall, 2007. One citation in section 1.

[SOM 11] Sommerville, I., Engenharia de Software, 9ed., São Paulo: Pearson Prentice Hall, 2011. One citation in section 1.

[WHI 12] Whitman, M. E. and Mattord, H. J. Principles of Information Security. Course Technology CENGAGE Learning, 4th edition, 2012. One citation in section 2.

[YU 04] Yu, Y., do Prado Leite, J. C. S., and Mylopoulos, J. From goals to aspects: discovering aspects from requirements goal models. In Proceedings of the 12th IEEE Joint International Conference on Requirements Engineering (RE 04), pages 38 - 47, 2004. One citation in section 3.