International Journal of
Emerging Research in Management &Technology
ISSN: 2278-9359

Research Article | **January 2013**

Requirements Reuse for Improving Information Systems Security

Venkata Kiran Maram*
Dept of Computer Applications
CMJ University
Bangalore, India

Dr. Mohammad Sharif K B
Dept of Computer Applications
CRD Business School
Bangalore, India

Badri H.S
Dept of Computer Applications
Presidency College
Bangalore, India

Abstract:

Information systems security issues have usually been considered only after the system has been developed completely, and rarely during its design, coding, testing or deployment. However, the advisability of considering security from the very beginning of the system development has recently begun to be appreciated, and in particular in the system requirements specification phase. We present a practical method to elicit and specify the system and software requirements, including a repository containing reusable requirements, a spiral process model, and a set of requirements documents templates. Any information system including these security requirements must therefore pass a risk analysis and management study performed with MAGERIT. The requirements specification templates are hierarchically structured and are based on IEEE standards.

Keywords: Requirements Engineering; Requirements Reuse; Security; Common Criteria Framework; Risk Analysis and Management Methods.

I. INTRODUCTION

There has recently been an increasing interest in information systems security issues. For instance, PITAC (President's Information Technology Advisory Committee) [1] has stated the need for a scalable information infrastructure, that is, for techniques which ensure that the United States information infrastructure -including communications systems, the Internet, large data repositories, and other emerging systems- is reliable and secure and can grow smoothly to accommodate the massive numbers of new users and applications requiring high bandwidth which are anticipated over the next two decades.

Nowadays, information systems are vulnerable to many threats, such as new viruses (e.g. worm viruses) that propagate through Internet; the threats brought about by unacceptable employees use of the Internet resources (such as non-business activities, accidental or deliberate disclosure of sensitive information, and hacking) [2]; failure to observe the personal data privacy laws (leading, in our country, to fines of up to $700,000 or to important administrative sanctions [3, 4]); even strikes or loss of key personnel are threats that information systems need to be able to cope with. In addition, the general acceptance of e-commerce and the digital signature to perform administrative and commercial transactions means that the security of information systems needs to be ever more reliable. A security breakdown can result in very serious problems for an organization: for example, a recent survey on 1,000 UK organizations [5] shows that the occurrence of a security failure in a business with no contingency plan leads to its shutdown in 80% of the cases, and 60% of UK businesses have suffered an important security breach in the last two years.

A large number of methods and regulations concerning the security of organizations have appeared in response to such a situation. Of particular importance are ISO 15408, Common Criteria Framework (CCF) [6], and the following national risk analysis and management methods: CRAMM in the UK (CCTA −Carmarthenshire College Of Technology and Art− Risk Analysis and Management Method) [7], MARION in France (Méthode d'Analyse de Risques Informatiques et d'Optimisation par Niveau) [8], and MAGERIT (Metodología de Análisis y GEstión de RIesgos del MinisTerio de Administraciones Públicas) [9], which is the Spanish Public Administration's adaptation of CCF.

In the information systems development field, requirements form the foundation for the rest of the software development process, since building high-quality requirements specification is essential to ensure that the product satisfies the users' needs [10-12]. However, drawing up a specification of quality requirements is a difficult task. According to the IEEE 830-1998 standard [13], a requirement of quality is that it be correct, unambiguous, complete, consistent, ranked for importance and/or stability, verifiable, modifiable, and traceable. We agree with Mili et al. [14] that "there exist few alternatives but software reuse as the (only) realistic approach to bring about the gains of productivity and quality that the software industry needs". It has been agreed (see, for example, [12, 15]) that the benefits of reuse are greater when the abstraction level is increased and not only code, but also designs and specifications, are reused.

The usual practice during the early steps in information systems development has been to pay attention to such aspects as reliability, availability or integrity, while security issues have usually been considered only once the system has been deployed, or at best, during the system design, coding or testing [9, 16].

This approach complements SIREN (SImple REuse of software requiremeNts), a general purpose RE method based on requirements reuse that we are currently defining. As this paper explains, the inclusion of requirements from the repository controls risk levels within the information system to be built, so that the information system will fulfill the demands of risk analysis performed with MAGERIT. How this approach has been applied to a real case study concerning an information system in our regional government is shown in this paper.

II. REASONS BEHIND THE STUDY

This research is based on the lessons learned during a risk analysis and management project developed with MAGERIT in our regional government. Our research offers a combination of RE, reuse, and information systems security in order to improve information systems quality and productivity.

A. RE and Security.

The essential reason behind this proposal is the need to include security explicitly in information systems development, as of the RE process, and thus improve several information systems quality attributes, namely security, safety, reliability, and robustness. Sommerville [12] claims that a high percentage of system malfunctions are the result of errors in specification rather than design. For example, the specification may be incomplete in that it does not describe the behavior of the system required in some critical situations. In a study of errors in embedded systems, Lutz [17] concludes that "(...) difficulties with requirements are the key root cause of the safety-related software errors which have persisted until integration and system testing".

Although the widespread use of Internet has given rise to very active, multidisciplinary research concerning security issues, and researchers in RE are beginning to focus on e-commerce and other Internet issues, we agree with Antón and Earp [16] that there remains a need to apply proven RE methods in this field. Some interesting approaches are the following:

- The integration of SSADM (Structured Systems Analysis and Design Method) [18] with CRAMM [7], and Métrica 3 [19] with MAGERIT (Métrica 3 is the standard development method of the Spanish public administration and is similar to SSADM). These approaches relate standard information systems development methods (including requirements specification) to standard risk analysis and management methods, and thus address the entire system-development life cycle. They do not, however, provide a reusable requirements repository to support reuse of security safeguards during the RE process. This criticism also concerns PFIRES (Policy Framework for Interpreting Risk in eCommerce Security) [20], a non-standard framework for risk analysis in e-commerce which addresses the entire system-development life cycle. The plan phase of PFIRES includes a requirements definition step, but it does not provide specific guidance on how to translate the security recommendations into requirements (safeguards), as does, for example, the integration between Métrica 3 and MAGERIT.

- First Chung [21] and later Antón and Earp [16] focused their previously proposed goal-based RE frameworks on the security field (with the latter specifically addressing e-commerce security issues and built on the PFIRES approach). We think that these approaches can be complementary to ours: as we explain below in this paper, we use MAGERIT in the first place to perform a risk analysis, and then we manage risk by reusing MAGERIT-compatible security requirements. Analogously, we could first specify the goals of the system to be developed by means of one of the goal-based frameworks mentioned before, and then operationalize the goals by making use of our security requirements, and thus reuse generic security design knowledge from MAGERIT. Like this, we would be able to specify the relationships between the security needs and the other functional and non-functional requirements.

- Jones et al. [22] are carrying out a project aimed at defining a framework for trust requirements in e-business. This framework contains three major elements: stakeholders in e-business, a list of common e-business process models, and a conceptual model of the elements involved in an e-business transaction. These authors, moreover, seek to develop a list of generic requirements for e-business to be adapted to specific projects by means of the framework. The purpose of this work is very similar to the research that we present in this paper, but the work focuses specifically on e-business, while our work is focused on information systems security in general. Their research is still at an early stage, and they do not present a list of requirements, but a list of categories of generic requirements.

Security standards do not adequately address the RE process either. We consider that a significant part of the study of Fenton and Neil [23] on lacks of safety related standards is also applicable to security standards (in particular, to ISO 15408 Common Criteria Framework). In particular, we agree with Fenton and Neil that: i) the requirements imposed by the standards are imprecise; and ii) the standards are too complex and difficult to use. In fact, Fenton and Neil show a process for improving safety-related standards through the improvement in the structure and writing of the requirements

that the standards impose. These authors, moreover, claim that safety standards are conceived to evaluate systems already created, and they do not consider the problem of providing security for new systems.

B . RE process improvement.

Our proposal also addresses the improvement of information systems quality through the improvement of the quality of the RE process. In this section we deal with the improvement of the quality of the RE process by means of the main processes life cycle standards and quality standards. To this extent, neither life cycle processes standards (such as ISO 12207 [24]) nor quality standards (such as ISO 9000 [25]) provide precise guidelines for identifying problems and planning improvements in the life cycle processes of systems and software (in particular, ISO 9000 does not include any section specifically devoted to RE [26]).

In contrast, capability maturity models, such as CMM [27] and ISO 15504/SPICE (Software Process Improvement and Capability determination) [28], do provide guidelines to improve the quality of the life cycle processes. However, once again, they do not cover the RE process [26]. This circumstance led Sommerville and Sawyer to propose a Requirements Engineering Process Maturity (REPM) model [26], which defined a set of good practices in RE, and allowed organizations to be assessed at three levels, namely 1) Initial; 2) Repeatable; and 3) Defined. Level 1 organizations do not have a defined RE process, while level 2 organizations have defined standards for requirements documents and requirements descriptions and have introduced policies and procedures for requirements management. These two levels correspond to CMM levels 1 and 2. Finally, REPM level 3 is assigned to organizations that have a defined RE process model based on good practices and techniques, with an active process improvement program in place and which can objectively estimate new tools and techniques. This third level corresponds to CMM levels 3, 4, and 5.

REPM is based on the incremental adoption of an RE good practice guide [26]. This guide, on which our proposal SIREN is based, contains a spiral process model and ten good practice guidelines

Define a standard document structure
Make the document easy to change
Uniquely identify each requirement
Define policies for requirements management
Define standard templates for requirements description
Use language simply, consistently and concisely
Organize formal requirements inspections
Define validation checklists
Use checklists for requirements analysis
Plan for conflicts and conflict resolution

Table 1 Sommerville and Sawyer's top ten guidelines.

C. MAGERIT.

MAGERIT [9] is the information systems risk analysis and management method of the Spanish public administration, which is compatible with ISO 15408, Common Criteria Framework. In this paper we show how we have translated the security measures stated in MAGERIT into reusable security requirements. MAGERIT is, therefore, the source of our reusable security requirements repository. MAGERIT, moreover, specifies how risk analysis has to be performed before selecting the required security. This section summarizes the basic elements of MAGERIT beginning with a brief description of the evolution of the risk analysis and management methods.

Baskerville [29] studies the evolution of information systems security design methods. The first generation of methods for analyzing risk in information systems appeared in the United States in the early 1970s, based on checklists. From the mid 1980s, there appeared a second generation of more formalized methods headed by the British CRAMM. In the late 1990s, there began to appear a third generation of methods more linked to information systems development methods and closer to security legislation and regulations, that is, 3 generation security methods take into account the personal data protection laws as well as some other regulations in the security field. The new version of CRAMM (integrated with SSADM) and the first version of MAGERIT (integrated with Métrica) pertain to this generation. Organizational problems are considered in the third generation and so security is reconciled with the functionality of the system, since the security aspects are taken into account from the first stages of the development process [29].

MAGERIT is, in fact, only applied to one of the steps in information systems security management: that of risk analysis and management. Starting from the objectives, strategy and policy of security in the existing information systems in the organization, MAGERIT is applied in order i) to study the risks that affect each information system and its environment, and ii) to propose the countermeasures (safeguards) that should be adopted to register, prevent, avoid, reduce and control the risks evaluated. These countermeasures are collected in an information systems security plan which is then put into practice, and followed up during the maintenance stage. The application of MAGERIT is iterative, so that new risk

analyses are periodically performed, and hence new countermeasures are proposed during maintenance.The risk analysis and management model of MAGERIT includes:

- the submodel of elements, providing the basic entities related to the information system risk analysis: assets, threats, vulnerabilities, effects, risks, and countermeasures;
- the submodel of processes, describing the stages in the security project that is to be developed: planning, risk analysis, risk management, and recommendation of countermeasures.

Risk analysis with MAGERIT involves the following major steps: i) identification of the assets of the organization, which are the resources of the information system, and which may either directly belong to the information system or just to the information system environment; ii) study of the vulnerabilities of these assets and the threats to them; iii) estimation of the risk related to these assets, based on the threats and vulnerabilities associated (threats are qualified by the likelihood of their occurring); iv) proposal of the countermeasures managing the risk. Therefore, countermeasures manage threats and are transitively linked to assets.

As indicated at the beginning of this section, we have translated the MAGERIT countermeasures into security requirements. Security requirements are thus linked to assets. Once risk analysis has been performed and we know how the assets are menaced, it is useful for the structure of the security requirements in the repository to conform to the structure of the assets of MAGERIT in order to find the suitable security requirements (countermeasures). It is, therefore, of interest to briefly outline the organization of the assets in MAGERIT (Figure 1).

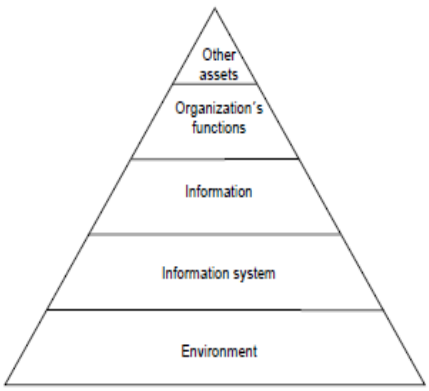

Figure 1. MAGERIT assets hierarchy.

As Figure 1 shows, MAGERIT organizes assets in five layers:

- The information system environment layer includes the facilities containing the information system, such as furniture and supplies.
- The information system layer includes the information related to the software, hardware, and personnel, which make up the information system.
- The information layer includes the information managed by the information system: applications data and metadata (such as data structures and data dictionaries).
- The organization's functions layer justifies the usefulness of the information system, by providing it with a purpose; it describes the assets and services produced by the information system and the users of the information system services.
- Finally, the other assets layer describes assets that are usually intangible, and do not fit into the lower layers, as, for instance, the organization credibility or an individual's privacy.

Each layer can be refined into blocks of homogeneous assets. Besides, each block can be further divided into sub-blocks.

D. The SIREN approach to requirements reuse.

Our proposal, called SIREN (SImple REuse of software requiremeNts), is a method for RE based on requirements reuse. The purpose of development with requirements reuse is to identify descriptions of systems that could be used (either totally or partially) with a minimal number of modifications, thus reducing the total effort of development [15]. In our view requirements reuse can produce more benefit than only design or code reuse because traceability relationships can be established between the high-level requirements of the system and the architecture and implementation which are built from them. Thus, if a system requirement is reused, the subsequent development process can be speeded up. Although a lot of more complex techniques for reusing requirements exist (see the Cybulsky's classification in [15]), our SIREN proposal for requirements reuse emphasizes simplicity and it is therefore more practical.

SIREN also conforms to the most well-known Software Engineering standards for requirements specification. Requirements have a textual format, but can include any kind of objects as complementary information - for instance, tables or schemas of any type.
SIREN encompasses a process model, some guidelines, techniques and tools. The guidelines that SIREN provides consist of a hierarchy of requirements specification documents together with the templates for each document. These serve to structure a reusable requirements repository. Finally, we will present the SIREN process model and a discussion on the tools used to support the process.

III. CONCLUSION

This paper presents a practical approach for managing the security of the information systems from the RE process. We believe it helps in the development of more complete and robust information systems. This approach is a particularization of a general purpose process for requirements reuse called SIREN, which shortens the development process since the analyst starts from a reusable set of requirements. The particularization of SIREN to the security profile has been based on the risk analysis and management method MAGERIT (conforming to IEEE 15408 Common Criteria Framework). SIREN provides a spiral model process based on Kotonya and Sommerville's proposal [10] which is enriched by explicitly taking reuse into account. The immediate benefit for an information system development is that the inclusion of these security requirements ensures that the proper implementation of the system will fulfill the demands of risk analysis performed with MAGERIT.

Together with the defined process, SIREN provides a requirements specification documents hierarchy (system, software and tests) based on the IEEE specification standards. The IEEE 830-1998 standard for SRS has been extended with items from the VOLERE template [11], but maintaining the conformance with the standard. The inclusion of a system tests specification document (SyTS), dealing, in particular, with security, was also found to be of interest, because the SyTS can serve as a checklist which encompasses all the information system features that must be checked for it to be considered secure. The traceability relationships between the requirements of the repository imply inclusive or exclusive dependence relationships. This approach has been tested in a case study in our regional government and the simplicity of the process and the fast implication of the staff involved have been proved.

As mentioned, the requirements imposed by the security standards are generally imprecise. Likewise, we think that some of our security requirements, expressed in natural language, may also be imprecise. As a first step, we consider that we can manage imprecision informally through i) the basic, good practice guidelines [26] regarding requirements writing; ii) the Repository Improvement activity in the SIREN process model and iii) the fit criteria included in the SyTS and STS documents. Nevertheless, we are conscious that complete imprecision management requires more advanced techniques to formalize both functional and non-functional requirements. Two lines of future research to tackle this problem are the following:
Further work is also needed to specify the interactions between security requirements and other types of non-functional requirements, as well as between functional requirements. To this extent, we can adopt a formal framework for integrating goals and goals refinement in requirements models, such as the KAOS methodology (Knowledge Acquisition in Automated Specification of Software) [45], the GBRAM methodology (Goal-Based Requirements Analysis Method) [16], or the NFR methodology (Non-Functional Requirements) [46] in order to capture and evaluate the alternative goals decomposition. Chung, for example, has already adapted the NFR framework to the security field [21].

Inconsistency management is another interesting aspect to be considered. Inconsistency can appear, for example, because the domains and profiles are not mutually exclusive. We think that in well defined profiles such as those coming from MAGERIT and the personal data protection law, inclusive and exclusive dependencies are enough to avoid most of the inconsistencies. Nevertheless, inconsistencies cannot be easily avoided in real applications dealing with multiples and less-formalized perspectives of the system.

Finally, further work will be the link from the SIREN requirements specification to the rest development artifacts (analysis, design, implementation), since we think that the integration of SIREN with the rest of the development process will improve the usefulness of the approach. In this regard, it is necessary to study the combination of SIREN with other methods, such as the Unified Process [47].

ACKNOWLEDGMENT

I am indebted to Dr. Mohammad Sharif Bammanalli for his valuable insights and guidance.

REFERENCES

[1] PITAC. (President's Information Technology Advisory Committee). Interim Report to the President. Setting Federal Research Priorities: Findings and Recommendations.1998. http://www.ccic.gov/ac/interim
[2] Lichtestein, S and Swatman, PMC. Effective Internet Acceptable Usage Policy for Organisations. In. 10th International Conference on Electronic Commerce (Bled'97). Slovenia. 1997. pp. 503-522.

[3] Constitutional Law 15/1999, of December 13, on Protection of private data of individuals in Spain. (Ley Orgánica 15/1999, de 13 de diciembre, de Protección de Datos de Carácter Personal en España). 1999. (In Spanish)
[4] Real Decreto 994/1999, of June 11, in which the Ruling on security measures of automated files containing private data on individuals is passed. (Real Decreto 994/1999, de 11 de junio, por el que se aprueba el Reglamento de medidas de seguridad de los ficheros automatizados que contengan datos de carácter personal). 1999: p. 24241. (In Spanish)
[5] Infosec. Information Security Breaches Survey. 2000. http://www.infosec.co.uk
[6] ISO/IEC Std. 15408. Evaluation Criteria for Information Technology Security. 1999.
[7] CCTA, SSADM-CRAMM Subject Guide for SSADM Version 3 and CRAMM Version 2. Central Computer and Telecommunications Agency, IT Security and Privacy Group, Her Majesty's Government, London. 1991
[8] CLUSIF, MARION version 98. La Commission Méthodes du CLUSIF (Club de la Securité des Systèmes d'Information Français). 1998
[9] MAP. Metodología de Análisis y Gestión de Riesgos del Ministerio de Administraciones Públicas Español, MAGERIT v.1.0. 1996. (In Spanish)
[10] Kotonya, G and Sommerville, I, Requirements Engineering. Processes and Techniques. John Wiley & Sons. 1998
[11] Robertson, S and Robertson, J, Mastering the requirement process. Addison-Wesley. 1999
[12] Sommerville, I, Software Engineering (6th edition). Pearson Education Limited. 2001
[13] IEEE Std 830-1998. Guide to Software Requirements Specifications (ANSI). The Institute of Electrical and Electronics Engineers, Inc. IEEE Software Engineering Standards Collection. 1999.
[14] Mili, H, Mili, F, and Mili, A, *Reusing Software: Issues and Research Directions.* IEEE Transactions on Software Engineering, 1995. 21(6): p. 528-562.
[15] Cybulsky, J and Reed, K. *Requirements Classification and Reuse: Crossing Domains Boundaries*. In. *6th International Conference on Software Reuse (ICSR'2000)*. Springer, Lecture Notes in Computer Science. Viena. 2000. pp. 190-210.
[16] Antón, AI and Earp, JB. *Strategies for Developing Policies and Requirements for Secure Electronic Commerce Systems*. In. *1st ACM Workshop on Security and Privacy in E-Commerce (CCS 2000)*. Athens, Greece. 2000.
[17] Lutz, R. *Analyzing software requirements errors in safety-critical embedded systems*. In. *Requirements Engineering (RE'93)*. San Diego, CA. 1993. pp. 126-133.
[18] Skidmore, S, Farmer, R, and Mills, G, *SSADM version 4 models and methods, second edition*. NCC Blackwell. 1994
[19] MAP. *MÉTRICA versión 3*. Ministerio de Administraciones Públicas. Secretaría de Estado para la Administración Pública. Consejo Superior de Informática. 2001 (In Spanish).
[20] *Policy Framework for Interpreting Risk in eCommerce Security*. Technical Report CERIAS (*Center for Education and Research in Information Assurance and Security*). Purdue University. 1999. http://www.cerias.purdue.edu/techreports/public/PFIRES.pdf
[21] Chung, L. *Dealing with Security Requirements during the development of Information Systems.* In: C. Rolland, F. Bodat, and C. Cauvet (eds.). *5th International Conference on Advanced Information Systems Engineering (CAISE'93)*. Spring Verlag. Paris. Berlin. 1993. pp. 234-251.
[22] Jones, S, Wilikens, M, Morris, P, and Masera, M, *Trust Requirements in e-Business.* Communications of the ACM, 2000. 43(12): p. 81-87.
[23] Fenton, N and Neil, M, *A Strategy for Improving Safety Related Software Engineering Standards.* IEEE Transactions on Software Engineering, 1998. 24(11): p. 1002-1014.
[24] *IEEE/EIA Std 12207.0-1996. Industry Implementation of International Standard ISO/IEC 12207: 1995. Standard for Information Technology- Software Life Cycle Processes*, in *Volume 1: Customer and Terminology Standards*. The Institute of Electrical and Electronics Engineers, Inc. IEEE Software Engineering Standards Collection. 1999.
[25] *ISO 9000 International Standars for Quality Management. 4th edition.* ISO (*International Organization for Standarization*). 1994
[26] Sommerville, I and Sawyer, P, *Requirements Engineering: A good practice guide*. John Wiley & Sons. 1997
[27] Paulk, MC and Weber, CV, *The Capability Maturity Model: Guidelines for Improvement the Software Process*. Addison-Wesley. 1995
[28] El Emam, K, Drouin, J-N, and Melo, W, *SPICE. The Theory and Practice of Software Process Improvement and Capability Determination*. IEEE Computer Society. 1997
[29] Baskerville, R, *Information Systems Design Methods: Implications for Information Systems Development.* Computing Surveys, 1993. 25(4): p. 375-414.
[30] Gabb, A. *The Requirements Spectrum*. In. *First Regional Symposium of the Systems Engineering Society of Australia (SE'98)*. Australia. 1998.
[31] Pressman, RS, *Software Engineering. A practitioner's approach. Fifth edition*. Mc Graw Hill. 2000
[32] *IEEE/EIA Std 12207.1-1997. Guide for Information Technology - Software life cycle processes - Life cycle data*. The Institute of Electrical and Electronics Engineers, Inc. IEEE Software Engineering Standards Collection. 1999.
[33] *IEEE Std 1233-1998. Guide for Developing System Requirements Specifications*. The Institute of Electrical and Electronics Engineers, Inc. IEEE Software Engineering Standards Collection. 1999.

[34] Toval, A, Olmos, A, and Rodero, JA. *The Role of Requirements Reuse in Protecting Personal Data*. In. *5th World Multiconference on Systemics, Cybernetics and Informatics (SCI 2001)*. Orlando, Florida (USA). 2001
[35] Ramesh, B and Jarke, M, Toward reference models for requirements traceability. IEEE Transactions on Software Engineering, 2001. 27(1): p. 58-93.
[36] Nuseibeh, B, Weaving Together Requirements and Architectures. IEEE Computer, 2001. 34(3): p. 115-117.
[37] Vitech Corporation. CORE. 2000. http://www.vtcorp.com
[38] Quality Systems & Software. DOORS. 2000. http://www.qssinc.com
[39] Rational Software. Requisite Pro. 2000. http://www.rational.com
[40] INVESDOC. IECISA. http://www.ieci.es
[41] MAGERIT risk analysis and management in the Regional Information Systems and Telecommunication Office. Contract CARMMA 3142-UMU. CARM (Autonomous Community of Murcia) and the Software Engineering Research Group of the University of Murcia. 1999.
[42] Jaaksi, A, A Method for Your First Object-Oriented Project. Journal of Object Oriented Programming (JOOP), 1998. 10(8): p. 17-25.
[43] Fernández, J and Toval, A. Can Intuition Become Rigorous? Foundations for UML Verification Tools. In: T. FM (eds.). The 11th Int. Sym. on Software Reliability Engineering. IEEE Computer Press. San José (California). 2000. pp. 344-355.
[44] Toval, A and Fernández, J. Improving System Reliability via Rigorous Software Modeling: The UML Case. In. 2001 IEEE Aerospace Conference (track 10: Software and Computing). IEEE Computer Society. Montana, USA. 2001.
[45] Dardenne, A, van Lamsweerde, A, and Fickas, S, Goal-Directed Requirements Acquisition. Science of Computer Programming, 1993. 20: p. 3-50.
[46] Chung, L, Nixon, B, Yu, E, and Mylopoulos, J, Non-Functional Requirements in Software Engineering. Kluwer Academic Publishers. 2000
[47] Jacobson, I, Booch, G, and Rumbaugh, J, The Unified Software Development Process. Addison-Wesley Longman, Inc. 1999
[48] Moros, B, Nicolás, J, García, J, and Toval, A, Combining Formal Specifications with Design by Contract. Journal of Object Oriented Programming (JOOP), 2000. 12(9): p. 16-22.
[49] García, J, Ortín, M, Moros, B, Nicolás, J, and Toval, A. Towards Use Case and Conceptual Models through Business Modeling. In: L.S. Laender AHF, Storey VC (eds.). 19th International Conference on Conceptual Modeling, ER'2000. Springer. Salt Lake City (Utah, USA). 2000. pp. 281-294.