Apéndice A
Socialización de resultados

Bien está lo que bien acaba

Esta tesis presentó sus resultados a la consideración de evaluadores externos. Primero como ponencia en un evento internacional organizado por la IEEE Sección Centro Occidente de México [Serna:2010]; y posteriormente con un artículo publicado en la revista Iteckne de la Universidad Santo Tomás, clasificada en categoria C por Publindex [Serna:2100b].
La Figura A.1 muestra el certificado de participación en la XII Reunión de Otoño de Potencia, Electrónica y Computación - ROPEC'2010 INTERNACIONAL, mientras que la Figura A.2 muestra la primera página del artículo publicado en las memorias de dicho evento.

La Figura A.3 muestra la tabla de contenido de la revista Iteckne donde se evidencia la publicación del artículo, mientras que la Figura A.4 muestra la primera página del artículo que se publicó en dicha revista.

Figura A.1: Certificado como ponente en ROPEC'2010

XII Reunión de Otoño de Potencia, Electrónica y Computación, INTERNACIONAL ROPEC'2010

Representación de requisitos no funcionales en el diagrama de clase

Sergio Serna, Member, IEEE
Centro de Investigación
Instituto Tecnológico Metropolitano
Cra 31 # 54 - 10 Medellín, Colombia
Email: sergioserna@itm.edu.co

Fernando Arango, Non Member, IEEE
Escuela de Sistemas
Facultad de Minas, Universidad Nacional
Cra 80 # 65 - 223 Medellín, Colombia
Email: farango@unal.edu.co

***Resumen*—La representación de requisitos no funcionales en sistemas software, ha presentado múltiples retos a académicos investigadores interesados en el tema. Las cada vez más crecientes exigencias de los sistemas en atributos catalogados, dentro de los que se denominan requisitos no funcionales, como desempeño, seguridad, escalabilidad, entre otros, ha permitido diversos enfoques a la hora de construir los planos software del sistema deseado por el cliente. Este trabajo, se enmarca dentro de la representación de requisitos temporales no funcionales, utilizando el lenguaje de modelado unificado para la construcción de los planos software. El trabajo no trata con requisitos funcionales, ni requisitos no funcionales que no tengan que ver con tiempo. La especificación se hace sobre diagramas de clase UML, no trabaja con otros diagramas ni otros lenguajes de modelado. Se apoya en métodos existentes durante las primeras etapas de desarrollo, lo que le permite elicitar los requisitos e ir llevándolos de manera consistente a través de todos los diagramas del modelo, hasta llegar a un nuevo diagrama de clases. Este nuevo diagrama de clases relaciona elementos del modelo con los del metamodelo, logrando una mayor expresividad y permitiendo tomar decisiones de implementación que antes no era posible en esta etapa del desarrollo. Para lograr esto, es necesario realizar una variante a la semántica del diagrama de clases, permitiendo relacionar metaclases que antes no lo estaban. Igualmente, se introduce una nueva simbología para expresar la nueva metarela-ción presente en el diagrama de clases.**

I. Introducción

El éxito de un sistema software, se mide principalmente por el grado en el cual se cumplieron los propósitos para los cuales fue concebido. Estos propósitos son tomados de los interesados, sin embargo, las necesidades de los interesados pueden ser múltiples y variadas, y con frecuencia contradictorias. Por esto, es indispensable tener una base de requisitos estable antes de comenzar el proyecto, si no es así, lo más probable es que se fracase.

La ingeniería de requisitos (RE - Requirements Engineering) [13] [21] se ha enfocado, fundamentalmente, en la elicitación de los llamados requisitos funcionales (FRs - Functional Requierements) [1], mientras que los requisitos no funcionales (NFRs - Non-Functional Requierements) se han tratado de una forma distinta [6] [25] [3] [7].

Mientras que los FRs describen que debe el sistema o dispositivo hacer, los NFRs se concentran en la manera como el sistema o dispositivo debe acompañarse de tal forma que se satisfagan los FRs [4]. Los NFRs son requisitos muy generales que imponen restricciones a las soluciones de acuerdo a las posibilidades del mundo real, no están confinados en una porción de funcionalidad y probablemente el mismo NFR tiende a ser deseado de manera diferente dependiendo del sistema.

Hay una gran variedad de NFRs, y debe tenerse en cuenta que no existe una definición formal o una lista completa de NFRs, ni hay un esquema de clasificación universal que satisfaga todas las necesidades de los diferentes dominios de aplicación bajo todas las posibles situaciones. Por lo tanto, basado en [6] y [14], se pueden categorizar los NFRs en restricciones de interfaz, de desempeño, operativas, de recursos, de verificación, de documentación, de portabilidad, de seguridad, de calidad, entre otras.

Este artículo se centra en la especificación de los requisitos de desempeño, los cuales incluyen propiedades de tiempo y espacio.

I-A. Desempeño

Para representar el desempeño de un sistema, es necesario tener claridad acerca de lo que significa buen desempeño [18]: un sistema con buen desempeño es aquel que realiza su función en el menor tiempo posible y utilizando la menor cantidad de recursos.

Existen varios conceptos que hacen parte de lo que se denomina desempeño de un sistema, los cuales cada vez pueden ser subdivididos en conceptos más especializados (véase Figura 1), que finalmente pueden ser operacionalizables [6].

En la literatura, existen varias metodologías de trabajo para construir sistemas con buen desempeño, los cuales, dependen fundamentalmente del dominio del sistema que se esté tratando: para cadenas de suministro [24], para sistemas cliente/servidor [19], para sistemas distribuidos [2], para sistemas de información [6] [20], para protocolos de transporte [16], entre otros.

II. Representación de requisitos no funcionales

La representación de NFRs se ha realizado de diversas formas. Uno de los trabajos más destacados tiene que ver con un método gráfico cuyo eje central es el concepto de softgoal, el cual define un marco de trabajo basado en NFRs [6].

Otros métodos de representación de NFRs utilizan ontologías, con estas logran establecer, en algunos casos, relaciones semánticas entre el contexto y los NFRs [24] [15]. En otros casos, los NFRs se expresan independientemente del dominio de aplicación, aunque se asocien con requisitos de QoS [8]. En el mapeo de NFRs directa y explicitamente desde el proceso de ingeniería de requisitos hasta el diseño de la arquitectura, se destacan los trabajos realizados por [25] [23]. Otras técnicas definen propiedades no funcionales relacionadas con un NFR muy importante, y que es el eje

Figura A.2: Primera página del artículo publicado en las memorias de ROPEC'2010

ISSN 1692 - 1798

Carrera 18 No. 9-27
E-mail: iteckne@mail.ustabuca.edu.co
iteckne@gmail.com
Bucaramanga - Santander.
Servicio al Cliente Iteckne
Teléfono 6800 801 Ext. 1411- 1421
Fax 6717067

Contenido

La revista ITECKNE es una publicación científica y tecnológica de la División de Ingenierías de la Universidad Santo Tomás, Seccional de Bucaramanga, que en la actualidad comprende las Facultades de Ingeniería de Telecomunicaciones, Ingeniería Mecatrónica, Ingeniería Industrial y Química Ambiental que tiene por objeto divulgar los resultados de las investigaciones que se realizan en su seno.

Actualmente la revista está indexada en la categoría C del Índice Bibliográfico Nacional Publindex y desde su inicio ha contado con un gran apoyo, no solamente del cuerpo de profesores de la Institución, sino de expertos y científicos de otras instituciones tanto en el ámbito nacional como en el internacional con las que la Universidad Santo Tomás y particularmente la División de Ingenierías ha hecho convenios. Esta valiosa colaboración nos ha permitido estar abiertos a los avances científicos desarrollados en otros países, situación que contribuye a mantenernos actualizados y a la vanguardia en las disciplinas profesionales que conforman la División de Ingenierías.

Figura A.3: Tabla de contenido de la revista Iteckne Vol. 7, No. 1, Julio 2010

Especificación de Requisitos de Desempeño en el Diagrama de Clases

Sergio Serna
Centro de Investigación
Instituto Tecnológico Metropolitano
Cra 31 # 54 - 10 Medellín, Colombia
Email: sergioserna@itm.edu.co

Fernando Arango
Escuela de Sistemas
Facultad de Minas, Universidad Nacional
Cra 80 # 65 - 223 Medellín, Colombia
Email: farango@unal.edu.co

Resumen— La especificación de requisitos no funcionales en sistemas software, ha presentado múltiples retos a académicos investigadores interesados en el tema. Las cada vez más crecientes exigencias de los sistemas en atributos catalogados, dentro de los que se denominan requisitos no funcionales, como desempeño, seguridad, escalabilidad, entre otros, ha permitido diversos enfoques a la hora de construir los planos software del sistema deseado por el cliente. Este trabajo se enmarca dentro de la especificación formal de requisitos temporales no funcionales, utilizando el lenguaje de modelado unificado para la construcción de los planos software. El trabajo no trata sobre requisitos funcionales o no funcionales que no están relacionados con tiempo. La especificación se hace sólo sobre diagramas de clase UML, y se apoya en métodos existentes durante las primeras etapas de desarrollo, lo que le permite elicitar los requisitos e ir llevándolos de manera consistente a través de todos los diagramas del modelo, hasta llegar a un nuevo diagrama de clases. Este nuevo diagrama de clases relaciona elementos del modelo con los del metamodelo, logrando una mayor expresividad y permitiendo tomar decisiones de implementación que antes no era posible en esta etapa del desarrollo. Para lograr esto, es necesario realizar una variante a la semántica del diagrama de clases, permitiendo relacionar metaclases que antes no estaban relacionadas. Igualmente, se introduce una nueva simbología para expresar la nueva metarelación presente en el diagrama de clases.

Palabras clave— Diagrama de clases, Metamodelo, Requisitos no funcionales, UML.

Abstract— Specification of non-functional requirements in software systems represents a challenge to academic researchers interested in this subject. The increasing demand for catalogued attributes, among which are the non-functional requirements (performance, security, scalability and others) has given rise to different approaches to building the software for the system required by the customer. This work deals with the specification of non-functional requirements of timing, using a unified modeling language for building the software plans. The specification is made only on UML class diagrams, and it is supported on existing methods at the early stages of development, in order to elicit requirements for consistency of the diagrams, and reaching a new class diagram. This new class diagram relates elements of the model and metamodel, to achieve a greater expressiveness and enabling decisions for implementation, which formerly were not possible at this stage of development. To achieve this, it was necessary to perform a variant on the semantics of the class diagram. This allowed relating metaclasses that were not related before. A new symbology is also introduced, in order to express the new metarelation that is present in the class diagram.

Keywords— Class diagram, metamodel, non-functional requirements, UML.

INTRODUCCIÓN

El éxito de un sistema software se mide principalmente por el grado en el cual se cumplen los propósitos para los cuales fue concebido. Estos propósitos son tomados de los interesados, sin embargo, las necesidades de los interesados pueden ser múltiples y variadas, y con frecuencia contradictorias. Por esto, es indispensable tener una base de requisitos estable antes de comenzar el proyecto, si no es así, lo más probable es que se fracase.

La ingeniería de requisitos (RE - Requirements Engineering) [1] [2] se ha enfocado, fundamentalmente, en la elicitación de los llamados requisitos funcionales (FRs – Functional Requirements) [3], mientras que los requisitos no funcionales (NFRs - Non-Functional Requirements) se han tratado de una forma distinta [4] [5] [6] [7].

Los NFRs son requisitos muy generales, que imponen restricciones a las soluciones de acuerdo a las posibilidades del mundo real. Por ejemplo, una red de sensores puede requerir que todos sus nodos sean consultados una vez cada cinco segundo. Este requisito es transversal a muchos artefactos del sistema, y de alguna manera restringe la plataforma de ejecución.

Figura A.4: Primera página del artículo publicado en la revista Iteckne Vol. 7, No. 1, Julio 2010

Glosario

Atributo Es información que representa el estado de un objeto.

Característica Cualidad que determina los rasgos de una persona o cosa y la distingue claramente de las demás.

Clasificador Es una clasificación de instancias, el cual describe un conjunto de instancias que tienen un conjunto de características en común. El clasificador es una metaclase abstracta.

Línea de vida Representa uno y sólo un participante en un diagrama de interacción.

Mensaje Permite definir comunicaciones particulares entre líneas de vida de un diagrama de de interacción. La comunicación puede ser una señal, la invocación de una operación, o la creación o destrucción de una instancia.

Método Es la implementación de una operación.

Operación Es una abstracción que indica la declaración de un cierto comportamiento de una clase.

Operacionalización Es la puesta en funcionamiento de un NFR, esto se puede hacer cuando el NFR está lo suficientemente refinado y por lo tanto se puede identificar una posible técnica de desarrollo con el fin de satisfacerlo.

Performance Se traduce al español como desempeño, y define la eficacia de un sistema de cómputo.

Propiedad Es una característica estructural. Puede representar un atributo de un clasificador o un extremo de una asociación.

Restricción Es una condición o limitación expresada en un lenguaje, formal o no, con el fin de dar mayor claridad sobre la semántica de un elemento.

Satisfacibilidad Es una medida cualitativa del grado en el cual un requisito se satisface en el sistema. Cuando se habla de grado de satisfacibilidad, se refiere al hecho de que un requisito no se cumplió completamente, sólo en cierto grado pudo satisfacerse. El nivel de satisfación normalmente no se expresa en forma cuantitativa, sino en forma cualitativa, como parcialmente satisfecho o parcialmente insatisfecho.

Softgoal Es un objetivo que no necesariamente tiene un criterio absolutamente claro de satisfacibilidad. Algunos pueden ser medidos y cuantificados, pero se utilizan sobretodo en enfoques cualitativos. Las relaciones entre softgoal no son tan rígidas como entre los objetivos, relaciones del tipo AND – OR, más bien se enmarcan en relaciones más suaves, que permiten razonar acerca de contribuciones entre ellos. Las contribuciones pueden ser positivas – negativas, o completas – parciales.

Throughput Se traduce al español como rendimiento, y define la cantidad de trabajo, puede ser tareas, métodos, entre otras, que realiza un sistema de cómputo por unidad de tiempo.

Referencias

[Allen:1997] Garlan D. Allen, R. A formal basis for architectural connection. *ACM Transactions on Software Engineering and Methodology*, 6(3):213–249, 1997. cited By (since 1996) 438.

[Anton:1997] Ana I. Anton. *Goal identification and refinement in the specification of software-based information systems*. PhD thesis, Atlanta, GA, USA, 1997.

[Arango:2006] Fernando Arango and Carlos Mario Zapata. *UN-Método para la elicitación de requisitos de software*. Escuela de Sistemas Universidad Nacional de Colombia, Medellín, 2006.

[Beck:2000] K. Beck. *Extreme Programming Explained: Embrace Change*. Addison-Wesley, 2000.

[Bertagnolli:2003] Silvia de Castro Bertagnolli and Maria Lúcia Blanck Lisbôa. The frida model. In *Proc. of Workshop on Analysis Aspect-Oriented Software*, Germany, 2003.

[Boehm:1976] J. R. Brown B. Boehm and M. Lipov. Quantitative evaluation of software quality. In *Proc. 2nd IEEE International Conference on Software Engineering*, pages 592–605, San Francisco, California, United States, 1976.

[Booch:2006] Grady Booch, James Rumbaugh, and Ivar Jacobson. *El lenguaje unificado de modelado : guía del usuario*. Pearson Educación S.A., Madrid, 2 edition, 2006.

[Botella:2001] Pere Botella, Xavier Burgues, Xavier Franch, Mario Huerta, and Guadalupe Salazar. Modelling non-functional requirements. In *Proceedings of Jornadas Ingeniera de Requisitos Aplicados (JIRA)*, Sevilla, Spain, 2001.

[Burge:2002] J. Burge and D. Brown. Nfrs: Fact or fiction? Technical Report WPI-CS-TR-02-01, Computer Science Technical Report Worcester Polytechnic University, 2002.

[Chandrasekaran:2000] B. Chandrasekaran and John R. Josephson. Representing function as effect. In *Engineering with Computers, Special Issue on Computer Aided Engineering*, pages 162–177, 2000.

[Chung:1995] Lawrence Chung and Brian A. Nixon. Dealing with non-functional requirements: Three experimental studies of a process-oriented approach. In *Software Engineering, 1995. ICSE 1995. 17th International Conference on*, pages 25 –25, april 1995.

[Chung:2000] Lawrence Chung, Brian A. Nixon, Eric Yu, and John Mylopoulos. *Non-functional requirements in software engineering*. Kluwer Academic Publishers Group, 2000.

[Coad:1999] Jeff De Luca Peter Coad, Eric Lefebvre. *Java Modeling In Color With UML: Enterprise Components and Process*. Prentice Hall PTR, 1999.

[Cysneiros:2001] Luiz Marcio Cysneiros and Julio César Sampaio do Prado Leite. Using uml to reflect non-functional requirements. In *CASCON '01: Proceedings of the 2001 conference of the Centre for Advanced Studies on Collaborative research*, page 2. IBM Press, 2001.

[Cysneiros:2001a] Luiz Marcio Cysneiros and Julio César Sampaio do Prado Leite. Using the language extended lexicon to support non-functional requirements elicitation, 2001.

[Cysneiros:2004] L.M. Cysneiros and J.C.S. do Prado Leite. Nonfunctional requirements: from elicitation to conceptual models. *Software Engineering, IEEE Transactions on*, 30(5):328–350, May 2004.

[Dai:2005] Lirong Dai and Kendra Cooper. Modeling and analysis of non-functional requirements as aspects in a uml based architecture design. In *SNPD-SAWN '05: Proceedings of the Sixth International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing and First ACIS International Workshop on Self-Assembling Wireless Networks*, pages 178–183, Washington, DC, USA, 2005. IEEE Computer Society.

[Dai:2006] Lirong Dai and Kendra Cooper. Modeling and performance analysis for security aspects. *Science of Computer Programming*, 61(1):58 – 71, 2006. Special Issue on Quality system and software architectures.

[Dardenne:1993] Anne Dardenne, Axel Van Lamsweerde, and Stephen Fickas. Goal-directed requirements acquisition. In *Science of Computer Programming*, volume 20, pages 3–50, 1993.

[Davis:1993] Alan M. Davis. *Software requirements: objects, functions, and states*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1993.

[Fei:2007] Ye Fei and Zhu Xiaodong. An xml-based software non-functional requirements modeling method. In *Electronic Measurement and Instruments, 2007. ICEMI '07. 8th International Conference on*, pages 2–375–2–380, 16 2007-July 18 2007.

[Flake:2002] Stephan Flake and Wolfgang Mueller. An ocl extension for real-time constraints. In *Object Modeling with the OCL, The Rationale behind the Object Constraint Language*, pages 150–171, London, UK, 2002. Springer-Verlag.

[Flake:2002a] Stephan Flake and Wolfgang Mueller. A uml profile for real-time constraints with the ocl. In *UML '02: Proceedings of the 5th International Conference on The Unified Modeling Language*, pages 179–195, London, UK, 2002. Springer-Verlag.

[Freescale:2010] Rabbit A Digi International Brand. Hcs 08 family reference manual. Technical Report Part Number HCS08RMv1, Freescale Semiconductor, May 2007.

[Hamilton:2006] Kim Hamilton and Russell Miles. *Learning UML 2.0*. O'Reilly, April 2006.

[Harrington:1991] J. Harrington. *Business Process Improvement: The Breakthrough Strategy for Total Quality, Productivity and Competitiveness*. McGraw-Hill, 1991.

[Hull:2005] Elizabeth Hull, Ken Jackson, and Jeremy Dick. *Requirements Engineering*. Springer, 2 edition, 2005.

[IEEE1012:2005] IEEE. Ieee std 1012 - 2004 ieee standard for software verification and validation. Technical report, 2005.

[IEEE802.15:2006] IEEE. Part 15.4: Wireless medium access control (mac) and physical layer (phy) specifications for low-rate wireless personal area networks (wpans). Technical report, September 2006.

[IEEE830:1998] IEEE. Ieee recommended practice for software requirements specifications. Technical report, 1998.

[IEEE:1990] Computer Society of the IEEE. Ieee standard glossary of software engineering terminology. Technical Report IEEE Std 610.12-1990, IEEE, 1990.

[ISO/IEC:1999] ISO/IEC. Standards 9126: Information technology - software product evaluation - quality characteristics and guidelines for their use 1991, and 14598: Information technology - software product evaluation: Part 1, general overview; part 4, process for acquirers 1999. Technical report, International Organization for Standardization and International Electrotechnical Commission, 1999.

[ISO:9000] ISO 9000:2005. Quality management systems - fundamentals and vocabulary. Technical report, International Organization for Standardization, 2005.

[Inverardi:1995] Wolf Alexander L. Inverardi, Paola. Formal specification and analysis of software architectures using the chemical abstract machine model. *IEEE Transactions on Software Engineering*, 21(4):373–386, 1995. cited By (since 1996) 73.

[Ishikawa:1986] K. Ishikawa. *Guide to quality control*. Asian Productivity Organization, Tokio, 1986.

[Jacobson:2000] I. Jacobson, G. Booch, and J. Rumbaugh. *El proceso unificado de desarrollo de software*. Pearson Educación S.A., Madrid, 2000.

[Jain:1991] R. Jain. *The Art of Computer Systems Performance Analysis: Techniques for Experimental Design, Measurement, Simulation, and Modeling*. Wiley- Interscience, New York, NY, April 1991.

[Jingbai:2008] Tian Jingbai, He Keqing, Wang Chong, and Liu Wei. A context awareness non-functional requirements metamodel based on domain ontology. In *Semantic Computing and Systems, 2008. WSCS '08. IEEE International Workshop on*, pages 1–7, July 2008.

[Kaiya:2005] H. Kaiya and M. Saeki. Ontology based requirements analysis: lightweight semantic processing approach. pages 223 – 230, sep. 2005.

[Kaiya:2006] H. Kaiya and M. Saeki. Using domain ontology as domain knowledge for requirements elicitation. pages 189 –198, sep. 2006.

[Kotonya:1998] Gerald Kotonya and Ian Sommerville. *Requirements Engineering : Processes and Techniques*. Worldwide Series in Computer Science. John Wiley & Sons, September 1998.

[Lavazza:2005] Luigi Lavazza, Sandro Morasca, and Angelo Morzenti. A dual language approach to the development of time-critical systems. *Electronic Notes in Theoretical Computer Science*, 116:227 – 239, 2005. Proceedings of the International Workshop on Test and Analysis of Component Based Systems (TACoS 2004).

[Lazowska:1984] Edward D. Lazowska, John Zahorjan, G. Scott Graham, and Kenneth C. Sevcik. *Quantitative system performance: computer system analysis using queueing network models*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1984.

[Leite:1993] J.C.S.do.P. Leite and A.P.M. Franco. A strategy for conceptual model acquisition. In Proceedings of IEEE International Symposium on, editor, *Requirements Engineering*, pages 243–246, Jan 1993.

[Luckham:1995] D.C. Luckham, J.J. Kenney, L.M. Augustin, J. Vera, D. Bryan, and W. Mann. Specification and analysis of system architecture using rapide. *Software Engineering, IEEE Transactions on*, 21(4):336 –354, apr. 1995.

[MARTE:2009] Object Management Group. Uml profile for marte: Modeling and analysis of real-time embedded systems, version 1. Technical report, 2009.

[MOF:2002] Object Management Group. Meta-object facility (mof) 1.4 specification. Technical report, April 2002.

[Milner:1982] R. Milner. *A Calculus of Communicating Systems*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 1982.

[Muneton:2008] Andrés Felipe Muñetón Lopera. Obtención automática del cuerpo de los métodos contenidos en los diagramas de comportamiento de uml 2.0 para un lenguaje orientado a objetos. Master's thesis, Escuela de Sistemas, Facultad de Minas, Universidad Nacional de Colombia, 2008.

[Mylopoulos:1992] J. Mylopoulos, L. Chung, and B. Nixon. Representing and using nonfunctional requirements: a process-oriented approach. *Software Engineering, IEEE Transactions on*, 18(6):483–497, Jun 1992.

[Nuseibeh:2000] Bashar Nuseibeh and Steve Easterbrook. Requirements engineering: a roadmap. In *ICSE '00: Proceedings of the Conference on The Future of Software Engineering*, pages 35–46, New York, NY, USA, 2000. ACM.

[OCL:2010] Object Management Group. Object constraint language, v2.2. Technical report, February 2010.

[ODM:2009] Object Management Group. Ontology definition metamodel, version 1.0. Technical report, 2009.

[Oracle:2000] Oracle Corporation. *Oracle Method CDM Quick Tour*. Oracle Corporation, Redwood City, 2000.

[RAPIDE:1997] RAPIDE Design Team. Rapide 1.0 architecture language reference manual. Technical report, Program Analysis and Verification Group, Computer Systems Lab. Stanford University, 1997.

[RAPIDE:1997a] RAPIDE Design Team. Rapide 1.0 language reference manual. Technical report, Program Analysis and Verification Group, Computer Systems Lab. Stanford University, 1997.

[RAPIDE:1997b] RAPIDE Design Team. Rapide 1.0 pattern language reference manual. Technical report, Program Analysis and Verification Group, Computer Systems Lab. Stanford University, 1997.

[Rabbit:2010] Rabbit A Digi International Brand. Rabbit 3000 microprocessor user's manual. Technical Report Part Number 019-0108, Digi International Inc., 2010.

[Robbes:2008] Romain Robbes. *Of change and software*. PhD thesis, Università della Svizzera italiana. Facoltà di scienze informatiche, 2008.

[Robertson:1999] S. Robertson and J. Robertson. *Mastering the Requirements Process*. Addison-Wesley Professional, 1999.

[SPEC:2010] The standard performance evaluation corporation (spec). http://www.spec.org/, January 2010.

[Schauerhuber:2006] A. Schauerhuber, W. Schwinger, E. Kapsammer, W. Retschitzegger, and M. Wimmer. Towards a common reference architecture for aspect-oriented modeling. In *Proceedings of the 8th International Workshop on Aspect-Oriented Modeling (AOM)*. Bonn, Germany, 2006.

[Serna:2010] Sergio Serna and Fernando Arango. Representación de requisitos no funcionales en el diagrama de clases. In *XII Reunión de Otoño de Potencia, Electrónica y Computación - ROPEC'2010 INTERNACIONAL*, volume 1, pages 122–129. IEEE Sección Centro Occidente, México, Noviembre 2010.

[Serna:2010a] Sergio Serna, Fabio León Suárez, Daniel Ocampo Valencia, and Luis Alejandro Álvarez. Ethernet y zigbee para aplicaciones de consumo y administración de la energía. In *Segundo Congreso Virtual Microcontroladores y sus Aplicaciones*, volume 1. Universidad Tecnológica Nacional Argentina, Octubre 2010.

[Serna:2100b] Sergio Serna and Fernando Arango. Especificación de requisitos de desempeño en el diagrama de clases. *Revista Iteckne*, 7(1), Junio 2010.

[Sha:2004] Lui Sha, Tarek Abdelzaher, Karl-Erik Årzén, Anton Cervin, Theodore Baker, Alan Burns, Giorgio Buttazzo, Marco Caccamo, John Lehoczky, and Aloysius K. Mok. Real time scheduling theory: A historical perspective. *Real-Time Systems*, 28(2/3):101–155, November 2004.

[Shaw:1995] DeLine Robert Klein Daniel V. Ross Theodore L. Young David M. Zelesnik Gregory Shaw, Mary. Abstractions for software architecture and tools to support them. *IEEE Transactions on Software Engineering*, 21(4):314–335, 1995. cited By (since 1996) 229.

[Smith:1999] Graeme Smith. *The Object-Z Specification Language*. Advances in Formal Methods , Vol. 1. Springer-Verlag New York, Inc., 1999.

[Sommerville:2007] Ian Sommerville. *Software Engineering*. Pearson Education Limited, 8 edition, 2007.

[Sousa:2004] G. Sousa and J.F.B. Castro. Supporting separation of concerns in requirements artifacts. In *First Brazilian Workshop on Aspect-Oriented Software Development (WASP´04)*, Brazil, 2004.

[Stein:2006] Dominik Stein, Stefan Hanenberg, and Rainer Unland. Expressing different conceptual models of join point selections in aspect-oriented design. In *AOSD '06: Proceedings of the 5th international conference on Aspect-oriented software development*, pages 15–26, New York, NY, USA, 2006. ACM.

[Supakkul:2005] S. Supakkul and L. Chung. A uml profile for goal-oriented and use case-driven representation of nfrs and frs. In *Software Engineering Research, Management and Applications, 2005. Third ACIS International Conference on*, pages 112 – 119, aug. 2005.

[Tonu:2005] S.A. Tonu and L. Tahvildari. Towards a framework to incorporate nfrs into uml models. In *Proceedings of 1st International Workshop on Reverse Engineering to Requirements (RETR'05)*, pages 13–18, held at Carnegie Mellon University, November 2005.

[Tonu:2006] Subrina Anjum Tonu. Incorporating non-functional requirements with uml models. Master's thesis, University of Waterloo, 2006.

[UMLInfra:2009] Object Management Group. Omg unified modeling language (omg uml), infrastructure, v2.2. Technical report, February 2009.

[UMLPSPT:2005] Object Management Group. Uml profile for schedulability, performance, and time specification, version 1.1. Technical report, January 2005.

[UMLSuper:2009] Object Management Group. Omg unified modeling language (omg uml), superstructure, v2.2. Technical report, February 2009.

[VanLamsweerde:1998] A. van Lamsweerde, R. Darimont, and E. Letier. Managing conflicts in goal-driven requirements engineering. *Software Engineering, IEEE Transactions on*, 24(11):908–926, Nov 1998.

[Wang:1999] He X. Deng Y. Wang, J. Introducing software architecture specification and analysis in sam through an example. *Information and Software Technology*, 41(7):451–467, 1999. cited By (since 1996) 27.

[Wehrmeister:2007] M.A. Wehrmeister, E.P. Freitas, C.E. Pereira, and F.R. Wagner. An aspect-oriented approach for dealing with non-functional requirements in a model-driven development of distributed embedded real-time systems. In *Object and Component-Oriented Real-Time Distributed Computing, 2007. ISORC '07. 10th IEEE International Symposium on*, pages 428 –432, may 2007.

[Wieger:2003] Karl Eugene Wiegers. *Software Requirements*. Microsoft Press, Redmond, Washington, 2 edition, 2003.

[Wiegers:1999] Karl Wiegers. Writing quality requirements. *Softw. Dev.*, 7(5):44–48, 1999.

[Xu:2005] L. Xu, H. Ziv, and D. Richardson. Towards modeling non-functional requirements in software architecture. In *Early Aspects 2005: Aspect-Oriented Requirements Engineering and Architecture Design, in conjunction with AOSD'2005*, Chicago, Illinois, USA, March 2005.

[Zapata:2006] F. Arango C. M. Zapata and A. Gelbukh. Pre-conceptual schema: a uml isomorphism for automatically obtaining of uml conceptual schemas. *Research in Computing Sciences*, 19:3–14, 2006.

[Zave:1997] Pamela Zave. Classification of research efforts in requirements engineering. *ACM Comput. Surv.*, 29(4):315–321, 1997.

[ZigBee:2006] ZigBee Alliance. Zigbee specification. Technical report, December 2006.