

JISBD
2009
500a

XIV Jornadas de Ingeniería del Software y Bases de Datos

San Sebastián, 8-11 de septiembre de 2009

Editores:

Antonio Vallecillo

Dept. Lenguajes y Ciencias de la Computación

ETSI Informática

Universidad de Málaga

Bulevar Louis Pasteur, 35. 29071 Málaga. Spain

e-mail: av@lcc.uma.es

Goiuria Sagardui

Departamento de Electrónica e Informática

Escuela Politécnica Superior

Mondragon Unibertsitatea

Loramendi, 4; 20500 Arrasate-Mondragón. Spain

e-mail: gsagardui@eps.mondragon.edu

Filmación e impresión: Gráficas Michelena

Depósito Legal: SS-988-2009

ISBN: 978-84-692-4211-7

Prólogo

Las XIV Jornadas de Ingeniería del Software y Bases de Datos se celebraron en San Sebastián del 8 al 11 de septiembre de 2009, en el incomparable marco del Palacio Miramar, siendo organizadas por la Escuela Politécnica Superior de la Universidad de Mondragón. Como viene ocurriendo desde la edición del 2001, se han celebrado, en paralelo y compartiendo algunos actos, de las IX Jornadas de Programación y Lenguajes (PROLE). Ambos eventos son organizados bajo los auspicios de SISTEDES, la Sociedad de Ingeniería del Software y Tecnologías de Desarrollo de Software.

Este volumen recoge los trabajos seleccionados por el Comité de Programa de JISBD 2009. Este año se han incluido tres tipos distintos de contribuciones en las actas. En primer lugar tenemos los artículos de investigación originales, que describen resultados de investigación o experiencias industriales relativas a los campos de la Ingeniería del Software y de las Bases de Datos. Se recibieron un total de 81 resúmenes preliminares, de los cuales 75 se plasmaron finalmente en artículos. Entre ellos el Comité de Programa decidió seleccionar 22 como artículos largos. Esto ha supuesto un ratio de aceptación del 29 %, lo que demuestra el arduo proceso de revisión y selección al que fueron sometidos los artículos, así como la calidad de los finalmente seleccionados. Además de estos artículos, otros 5 fueron seleccionados para participar en la conferencia como artículos cortos, con la idea de favorecer y estimular el debate científico entre los asistentes y dar cabida a la presentación de trabajos incipientes. Todos los artículos fueron revisados siguiendo un sistema de revisión por pares, y discutidos entre los miembros del Comité.

Además de este tipo de artículos, este año también se ha incluido en las actas los resúmenes de las demostraciones de herramientas presentadas, y de los artículos relevantes ya publicados. Las demostraciones de herramientas son el camino para demostrar la viabilidad práctica de las propuestas teóricas y metodológicas formuladas por los equipos de investigación, y una muestra de lo que pueden aportar a la ciencia y a la industria. En esta ocasión se seleccionaron 23 herramientas para ser brevemente presentadas durante las sesiones, y expuestas durante las Jornadas.

Por otro lado, la madurez de la comunidad JISBD se plasma en un número creciente de trabajos publicados en revistas y congresos de primera línea. Muchos de estos trabajos pasan desapercibidos para la comunidad al no tener un reflejo en las propias Jornadas, más orientadas hacia trabajos emergentes. Por ello, JISBD acoge desde las últimas dos ediciones la presentación de este tipo de publicaciones, con un doble objetivo: por una parte, publicitarlos dentro de las propias jornadas; por otra, ofrecerlos como guía y estímulo al resto de la comunidad. En esta ocasión se incluyen en las actas los resúmenes de los 10 trabajos seleccionados por la organización en esta categoría.

El éxito de la conferencia JISBD también se refleja en el número de eventos que suceden a su alrededor. En primer lugar, JISBD 2009 contó con tres conferenciantes invitados de primer nivel: Don Batory (Department of Computer Sciences, University of Texas at Austin, EE.UU.) que impartió la charla “Stepwise Development of Streaming Software Architectures”; Jean Bézivin (University of Nantes, Francia) con “Advances

in Model Driven Engineering”; y Houari Sahraoui (Universidad de Montreal, Canada), cuya charla fue “Put the Horse before the Cart: Task-Driven Development of Software Visualization Tools”. Nuestro agradecimiento más sincero por su disponibilidad para aceptar la invitación y venir a San Sebastián a impartir sus conferencias.

El programa de JISBD 2009 también incluyó dos tutoriales sobre temas de candente actualidad, como son las líneas de producto y el desarrollo dirigido por modelos.

Otro de los puntos fuertes de JISBD son los Talleres, que conforman punto de reunión obligado para los investigadores que trabajan en algunos temas de especial interés, y que ofrecen un foro de discusión excelente para estimular el debate y la colaboración entre ellos. Este año se ha contado con 6 Talleres, algunos de amplia tradición como ADIS (9ª edición), DSDM (6ª edición) o PRIS (4ª edición) y otros más noveles como son PNIS, ZOCO o WASELF.

Nos gustaría expresar nuestro más sincero agradecimiento a todos los miembros del Comité de Programa por su tiempo y dedicación a la hora de revisar y seleccionar los artículos que fueron finalmente aceptados para su presentación, y que han permitido confeccionar un año más un programa de gran calidad y nivel. Por supuesto, queremos también agradecer a los autores que enviaron artículos a las Jornadas, fueran finalmente aceptados o no, por el esfuerzo realizado y por su contribución al evento.

También queremos agradecer desde aquí al equipo organizador todo su esfuerzo y trabajo. Esto incluye a los organizadores locales de la Universidad de Mondragón, que han permitido hacer realidad esta conferencia, así como a los distintos Coordinadores que se han ocupado de organizar aspectos esenciales como los Talleres (Coral Calero), Demostraciones (Juan de Lara), Tutoriales (Ernest Teniente), Divulgación de Trabajos Relevantes ya Publicados (Belén Vela), Publicidad (Gentzane Aldekoa y José Raúl Romero), Actas (Leire Etxeberria) y Web (Ana Altuna). Nos gustaría también mostrar nuestro agradecimiento al Comité Permanente de las JISBD: primero, por depositar en nosotros su confianza a la hora de presidir el Comité de Programa y organizar la conferencia; y segundo, por su constante apoyo y soporte. Mención especial merece Oscar Díaz, cuyos consejos y ayuda han sido siempre inestimables. También mencionar el sistema de revisión de artículos que hemos utilizado, EasyChair, que fue de gran utilidad y ayuda durante todo el proceso de revisión y para la confección de estas actas.

Nuestro agradecimiento explícito a los patrocinadores del evento que hicieron posible que la conferencia fuera todo un éxito: las empresas Intersystems y Ulma, la Asociación de Técnicos en Informática (ATI), la revista Novática, las Universidades de Mondragón y del País Vasco, Cursos de Verano, el Gobierno Vasco, el Ministerio de Ciencia e Innovación, la caja de ahorros Caja Laboral y la Corporación Mondragón.

Muchas gracias a todos los asistentes a las JISBD 2009, y esperamos verles de nuevo en las próximas JISBD 2010.

San Sebastián, Septiembre 2009

Antonio Vallecillo
Goiuria Sagardui

Comité Ejecutivo

Presidente del Comité de Programa

Antonio Vallecillo (Univ. Málaga)

Presidenta del Comité Organizador

Goiuria Sagardui (Univ. Mondragón)

Coordinadora de Talleres

Coral Calero (Univ. Castilla-La Mancha)

Coordinador de Demostraciones

Juan de Lara (Univ. Autónoma Madrid)

Coordinador de Tutoriales

Ernest Teniente (Univ. Polit. Cataluña)

Coordinadora de Divulgación de Trabajos Relevantes ya Publicados

Belén Vela (Univ. Rey Juan Carlos)

Coordinadores de Publicidad

Gentzane Aldekoa (Univ. Mondragón)

José Raúl Romero (Univ. Córdoba)

Coordinadora de actas

Leire Etxeberria (Univ. Mondragón)

Coordinadora de la Web

Ana Altuna (Univ. Mondragón)

Comité Organizador (Univ. Mondragón)

Goiuria Sagardui

Joseba Andoni Aguirre

Gentzane Aldekoa

Ana Altuna

Merixell Armentia

Lorea Belategui

Itxaso Buruaga

Leire Etxeberria

Osane Lizarralde

Urtzi Markiegi

Xabier Sagarna

Comité de Programa

Abrahão, Silvia (Univ. Polit. Valencia)
Acuña, Cesar (Univ. Rey Juan Carlos)
Aldana, José (Univ. Málaga)
Álvarez, Bárbara (Univ. Polit. Cartagena)
Aramburu, María José (Univ. Jaume I)
Araujo, Joao (Univ. Nova Lisboa)
Ávila, Orlando (Open Canarias)
Bañares, Jose A. (Univ. Zaragoza)
Berlanga, Rafael (Univ. Jaume I)
Bertrand, Enrique (Software AG)
Boronat, Artur (Univ. Leicester)
Brisaboa, Nieves (Univ. Coruña)
Cabot, Jordi (Univ. Oberta Cataluña)
Cachero, Cristina (Univ. Alicante)
Canal, Carlos (Univ. Málaga)
Canos, José Hilario (Univ. Polit. Valencia)
Carretero, Baltasar (T-Systems)
Cavero, José María (Univ. Rey Juan Carlos)
Corchuelo, Rafael (Univ. Sevilla)
Costal, Dolors (Univ. Polit. Cataluña)
Crespo, Yania (Univ. Valladolid)
de la Riva, Claudio (Univ. Oviedo)
Delgado Kloos, Carlos (Univ. Carlos III)
Dolado, Javier (Univ. País Vasco)
Duran, Francisco (Univ. Málaga)
Fdez-Bertoa, Manuel (Univ. Málaga)
Fdez-Medina, Eduardo (Univ. Castilla-La Mancha)
Franch, Xavier (Univ. Polit. Cataluña)
Fredlund, Lars-Ake (Univ. Polit. Madrid)
Fuente, Pablo de la (Univ. Valladolid)
Garbajosa, Juan (Univ. Polit. Madrid)
García Molina, Jesús (Univ. Murcia)
García, Félix (Univ. Castilla-La Mancha)
Gaspar da Silva, Mario (Univ. Lisboa)
Genero, Marcela (Univ. Castilla-La Mancha)
Genova, Gonzalo (Univ. Carlos III)
Gómez, Jaime (Univ. Alicante)
Gonzalez, Daniel (Univ. La Laguna)
Goñi, Alfredo (Univ. País Vasco)
Guerra, Esther (Univ. Carlos III)
Hernández, Juan (Univ. Extremadura)
Hogdson, Peter (Procedimientos Uno)
Irastorza, Arantza (Univ. País Vasco)
Iribarne, Luis (Univ. Almeria)
Iturrioz, Jon (Univ. País Vasco)
Juristo, Natalia (Univ. Polit. Madrid)
Laguna, Miguel A. (Univ. Valladolid)
Llorens, Juan (Univ. Carlos III)
Lopes, Antonia (Univ. Lisboa)
Lopez Cobo, Jose M.(XimetriX)
Lozano, Adolfo (Univ. Extremadura)
Melia, Santiago (Univ. Alicante)
Mena, Eduardo (Univ. Zaragoza)
Moreira, Ana (Univ. Lisboa)
Moreno, Ana María (Univ. Polit. Madrid)
Pelechano, Vicente (Univ. Polit. Valencia)
Pimentel, Ernesto (Univ. Málaga)
Polo, Antonio (Univ. Extremadura)
Quer, Carme (Univ. Polit. Cataluña)
Riquelme, José (Univ. Sevilla)
Rito, Antonio (Univ. Tec. Lisboa)
Roda, José Luis (Univ. La Laguna)
Ruíz, Francisco (Univ. Castilla-La Mancha)
Ruíz-Cortés, Antonio (Univ. Sevilla)
Sagardui, Goiuria (Univ. Mondragón)
Samos, José (Univ. Granada)
Sánchez, Fernando (Univ. Extremadura)
Sánchez, Juan (Univ. Polit. Valencia)
Sánchez, Victor (Open Canarias)
Toro, Miguel (Univ. Sevilla)
Toval, Ambrosio (Univ. Murcia)
Trujillo, Juan Carlos (Univ. Alicante)
Trujillo, Salvador (Ikerlan)
Tuya, Javier (Univ. Oviedo)
Urpí, Toni (Univ. Polit. Cataluña)
Vicente, Cristina (Univ. Polit. Cartagena)

Revisores

Miguel Ángel Martínez
Luz Marina Moreno de Antonio
Isabel Brito
Nuno Cardoso
Dante Carrizo
Ana Cerdeira-Pena
Antonio Corral
Diego Alonso
Javier Cámara
Jose María García
Irene Garrigós
Anna Grimán
Jorge García
Ignacio García
Sergio Ilarri
Ernesto Jimenez
Beatriz Bernárdez
Joaquín Lasheras
Manuel Llavador
Esperanza Manso

Miguel Angel Martinez-Prieto
Enric Mayol
Jose-Norberto Mazon
Fernando Molina
Sonia Montagud
M^a Ángeles Moraga
Isabel A. Nepomuceno-Chamorro
Jesús Pardillo
Juan Ángel Pastor
Jennifer Perez
Javier Pérez
Manuel Resinas
Roberto Rodríguez-Echeverría
Roberto Ruiz
Sergio Segura
Manuel Serrano
Pedro Sánchez
Mari Carmen Otero
Jose Zubcoff

Patrocinadores



Tabla de contenidos

I Conferencias Invitadas

Advances in Model Driven Engineering	3
<i>Jean Bézivin</i>	
Stepwise Development of Streaming Software Architectures	4
<i>Don Batory</i>	
Put the Horse before the Cart: Task-Driven Development of Software Visualization Tools	5
<i>Houari Sahraoui</i>	

II Sesión 1. Verificación y pruebas

Generación de Pruebas Basada en Restricciones para Consultas SQL	9
<i>María José Suárez-Cabal, Claudio de la Riva, Javier Tuya</i>	
Selección de Características para Mejorar los Modelos de Verificación de Información en EAI	21
<i>Iñaki Fernández de Viana, José Luis Arjona, José Luis Álvarez, Pedro Abad</i>	
Análisis de los Efectos de las Políticas de Gestión de la Capacidad de los Servicios en el Cumplimiento de los SLAs utilizando Simulación	33
<i>Elena Orta, Mercedes Ruiz, Miguel Toro</i>	
A tabu search algorithm for structural software testing (<i>Artículo ya publicado</i>) . .	45
<i>Eugenia Díaz, Javier Tuya, Raquel Blanco, José Javier Dolado</i>	
WebAVLTester: Una Herramienta de Pruebas Funcionales Automáticas para Formularios Web (<i>demo</i>)	46
<i>Eugenia Díaz, Marta Fernández de Arriba, Roberto López</i>	
GAMera: una herramienta para la generación y selección mediante algoritmos genéticos de mutantes WS-BPEL (<i>demo</i>)	50
<i>Antonia Estero, Inmaculada Medina, Juan José Domínguez, Lorena Gutiérrez</i>	

III Sesión 2. Gestión de Proyectos

TUNE-UP: Seguimiento de proyectos software dirigido por la gestión de tiempos	57
<i>María Isabel Marante, Patricio Letelier, Francisco Suárez</i>	

Selección de Herramientas para la Gestión de Proyectos de Software en Pequeñas y Medianas Empresas	69
<i>Lornel Rivas, María Pérez, Luis E. Mendoza, Anna Grimán</i>	
TEMPO: Una herramienta para la reutilización efectiva en la ingeniería de procesos software (<i>demo</i>)	81
<i>Orlando Avila-García, Adolfo Sánchez-Barbudo, Víctor Roldán, Carlos González, Antonio Estévez</i>	
Una Aplicación Web de Gestión Empresarial Orientada a Proyectos para PYMEs (<i>demo</i>)	85
<i>Leticia González, Miguel R. Luaces</i>	

IV Sesión 3. Requisitos / Ingeniería del software empírica

Pautas para Agregar Estudios Experimentales en Ingeniería del Software	91
<i>Enrique Fernández, Oscar Dieste, Patricia Pesado, Ramón García-Martínez</i>	
Adapting Software by Identifying Volatile and Aspectual Requirements	103
<i>José M. Conejero, Juan Hernández, Ana Moreira, João Araújo</i>	
A Tool-supported Natural Requirements Elicitation Technique for Pervasive Systems centred on End-users	115
<i>Francisca Pérez, Pedro Valderas</i>	

V Sesión 4. MDE y Transformaciones

On the Refinement of Model-to-Text Transformations	123
<i>Salvador Trujillo, Ander Zubizarreta, Josune de Sosa, Xabier Mendialdua</i>	
Towards Automatic Code Generation for EAI Solutions using DSL Tools	134
<i>Hassan A. Sleiman, Abdul W. Sultán, Rafael Z. Frantz, Rafael Corchuelo</i>	
Reingeniería sobre Almacenes de Datos Seguros aplicando ADM	146
<i>Carlos Blanco, Eduardo Fernández-Medina, Juan Trujillo</i>	
Wires* : A Tool for Orchestrating Model Transformations (<i>demo</i>)	158
<i>José Eduardo Rivera, Daniel Ruiz-González, Fernando López-Romero, José María Bautista</i>	
Gra2MoL: Una Herramienta para la Extracción de Modelos en Modernización de Software (<i>demo</i>)	162
<i>Javier Luis Cánovas, Jesús García</i>	

MoteGen: Una herramienta para el desarrollo de aplicaciones para redes de sensores (<i>demo</i>)	166
<i>Fernando Losilla, Pedro Sánchez, Bárbara Alvarez, Diego Alonso</i>	
Generación de herramientas de modelado colaborativo independientes del dominio (<i>demo</i>)	170
<i>Jesús Gallardo, Crescencio Bravo, Miguel Ángel Redondo</i>	
ModelSET Component Framework: Refinando el Ciclo de Vida de MDA (<i>demo</i>)	174
<i>E. Victor Sanchez, Orlando Avila-García, Pablo J. Hernández, Salvador Martínez, Antonio Estévez</i>	
MOMENT2: EMF Model Transformations in Maude (<i>demo</i>)	178
<i>Artur Boronat, José Meseguer</i>	

VI Sesión 5. BBDDs y Tecnologías de SGBD

Philo: un Sistema Experimental de Gestión de Bases de Datos Distribuido en Memoria de Alto Rendimiento	183
<i>Alejandro Bascuñana, Jesús Javier Arauz</i>	
Microsistemas de Información	195
<i>Jordi Pradel, Jose Raya, Xavier Franch</i>	
Cache-aware load balancing for question answering (<i>Artículo ya publicado</i>)	207
<i>David Dominguez-Sal, Mihai Surdeanu, Josep Aguilar-Saborit, Josep-Lluis Larriba-Pey</i>	
An MDA approach for the development of data warehouses (<i>Artículo ya publicado</i>)	208
<i>Jose-Norberto Mazón, Juan Trujillo</i>	
An Adaptive Mechanism to Protect Databases against SQL Injection	209
<i>Cristian I. Pinzón, Juan F. De Paz, Javier Bajo, Juan M. Corchado</i>	
Un Conjunto de plugins de Eclipse para el Diseño Multidimensional de Almacenes de Datos Dirigido por Modelos (<i>demo</i>)	215
<i>Octavio Glorio, Jesús Pardillo, Paul Hernández, Jose Quinto, Jose-Norberto Mazón, Juan Trujillo</i>	
OOH4RIA Tool: Una Herramienta basada en el Desarrollo Dirigido por Modelos para las RIAs (<i>demo</i>)	219
<i>Santiago Meliá, Jose-Javier Martinez, Álvaro Pérez, Jaime Gómez</i>	

VII Sesión 6. Recuperación de Información, Indexación y BD en Web

Indexación espacial de puntos empleando wavelet trees	225
<i>Nieves R. Brisaboa, Miguel R. Luaces, Gonzalo Navarro, Diego Seco</i>	
Desarrollo de un compresor de textos orientado a palabras basado en PPM	237
<i>Sandra Álvarez, Ana Cerdeira-Pena, Antonio Fariña, Susana Ladra</i>	
Reducción del Tamaño del Índice en Búsquedas por Similitud sobre Espacios Métricos	249
<i>Luis G. Ares, Nieves R. Brisaboa, María F. Esteller, Óscar Pedreira, Ángeles S. Places</i>	
Reorganizing Compressed Text (<i>Artículo ya publicado</i>)	261
<i>Nieves R. Brisaboa, Antonio Fariña, Susana Ladra, Gonzalo Navarro</i>	
Exploiting Pipeline Interruptions for Efficient Memory Allocation (<i>Artículo ya publicado</i>)	262
<i>Josep Aguilar-Saborit, Mohammad Jalali, Dave Sharpe, Victor Muntés-Mulero</i>	

VIII Sesión 7. Líneas de Producto

Towards security requirements management for software product lines: a security domain requirements engineering process (<i>Artículo ya publicado</i>)	267
<i>Daniel Mellado, Eduardo Fernández-Medina, Mario Piattini</i>	
A Software Product Line Definition for Validation Environments (<i>Artículo ya publicado</i>)	268
<i>Belen Magro, Jennifer Perez, Juan Garbajosa</i>	
Realizing Feature Oriented Software Development with Equational Logic: An Exploratory Study	269
<i>Roberto E. Lopez-Herrejon, José Eduardo Rivera</i>	
Revisión Sistemática de Métricas de Calidad para Líneas de Productos Software	275
<i>Sonia Montagud, Silvia Abrahão</i>	
FMT: Una Herramienta de Modelado y Configuración de Líneas de Productos Software para MS Visual Studio (<i>demo</i>)	281
<i>Rubén Fernández, Miguel A. Laguna, Jesús Requejo, Nuria Serrano</i>	
Moskitt FM and FAMA FW: Taking feature models to the next level (<i>demo</i>)	285
<i>Carlos Cetina, Pablo Trinidad, Vicente Pelechano, Antonio Ruiz-Cortés, David Benavides</i>	

IX Sesión 8. Ontologías, Web semántica

WEAPON: Modelo de Workflow con Ontologías para Procesos Administrativos <i>Álvaro E. Prieto, Adolfo Lozano-Tello</i>	291
Populating Data Warehouses with Semantic Data <i>Victoria Nebot, Rafael Berlanga</i>	303
Towards the Semantic Desktop: the seMouse approach (<i>Artículo ya publicado</i>) <i>Jon Iturrioz, Oscar Díaz, Sergio F. Anzuola</i>	315
Logic-based Ontology Integration using ContentMap (<i>demo</i>) <i>Ernesto Jiménez-Ruiz, Bernardo Cuenca, Ian Horrocks, Rafael Berlanga</i>	316
SPARQL Query Splitter: query translation between different contexts (<i>demo</i>) <i>Carlos R. Osuna, David Ruiz, Rafael Corchuelo, José Luis Arjona</i>	320
Annotator: Herramienta para la Anotación Semántica de Islas de Datos Amigables en la Web (<i>demo</i>) <i>José L. Álvarez, José L. Arjona, Agustín Domínguez, Nicolás Amador</i>	324

X Sesión 9. Validación/Modelado Conceptual

Extensión UML para Casos de Uso Reutilizables en entornos Grid Móviles Seguros <i>David G. Rosado, Eduardo Fernández-Medina, Javier López</i>	331
Perfiles UML en el diseño de notaciones visuales <i>Jesús Pardillo, Cristina Cachero</i>	343
Decidable Reasoning in UML Schemas with Constraints (<i>Artículo ya publicado</i>) <i>Anna Queralt, Ernest Teniente</i>	354
Rigorous Software Development Using McErlang (<i>demo</i>) <i>Clara Benac, Lars-Åke Fredlund</i>	355
UMLtoSBVR: Una herramienta para la validación de modelos UML mediante SBVR (<i>demo</i>) <i>Raquel Pau, Jordi Cabot, Ruth Raventós</i>	359
Una Aplicación basada en Eclipse para la Personalización de Aplicaciones Web Dirigida por Modelos (<i>demo</i>) <i>Irene Garrigós, Octavio Glorio, Paul Hernández, Alejandro Mate</i>	363
Takuan: generación dinámica de invariantes en composiciones de servicios web con WS-BPEL (<i>demo</i>) <i>Manuel Palomo, Antonio García, Alejandro Álvarez, Inmaculada Medina</i>	367

XI Sesión 10. Calidad, Medición y Estimación de Productos y Procesos Software

Un análisis de la Calidad en Uso de los Componentes Software utilizando Redes Bayesianas	373
<i>Manuel F. Bertoa, María Ángeles Moraga, M. Carmen Morcillo, Coral Calero</i>	
Validación empírica de medidas para procesos ETL en almacenes de datos	387
<i>Lilia Muñoz, Jesús Pardo, Jose-Norberto Mazón, Juan Trujillo</i>	
ECAPRIS: Metodología ágil de medición de calidad y productividad en PyMEs .	399
<i>Astrid Duque, Joaquín Lasheras, Ambrosio Toval</i>	
A Literature Review for Obtaining PDQM v.2.0	411
<i>Carmen Moraga, María Ángeles Moraga, Coral Calero, Angélica Caro</i>	
Towards a Catalogue of Patterns for defining Metrics over i* Models (<i>Artículo ya publicado</i>)	417
<i>Xavier Franch, Gemma Grau</i>	
SMT: Software Measurement Tool (<i>demo</i>)	418
<i>Beatriz Mora, Francisco Ruiz, Félix García</i>	

XII Talleres

Integración de Aplicaciones e Información Empresarial (ZOCO, 7ª edición)	425
<i>Rafael Corchuelo, David Ruiz, José Luis Álvarez, José Luis Arjona</i>	
Apoyo a la Decisión en Ingeniería del Software (ADIS, 9ª edición)	426
<i>José C. Riquelme, Roberto Ruiz, Daniel Rodríguez</i>	
Pruebas en Ingeniería del Software (PRIS, 4ª edición)	427
<i>Claudio de la Riva, Peter Hodgson, Ewout van Driel, Fergus Flaherty, Juan Garbajosa, Luis Fernández, Macario Polo, Javier Tuya</i>	
Procesos de Negocio e Ingeniería de Servicios (PNIS, 2ª edición)	428
<i>Antonio Ruiz-Cortés, Manuel Resinas, Francisco Ruiz, Félix García</i>	
Desarrollo de Software Dirigido por Modelos (DSDM, 6ª edición)	429
<i>José Raúl Romero, Orlando Avila-García, Vicente Pelechano</i>	
Autonomic and SELF-adaptive Systems (WASELF, 2ª edición)	430
<i>Javier Cámara, Carlos E. Cuesta, Miguel Ángel Pérez-Toledano</i>	

XIII Tutoriales

Análisis en líneas de productos: avances, desafíos y lecciones aprendidas	433
<i>David Benavides, Antonio Ruiz-Cortés, Pablo Trinidad</i>	
Herramientas Eclipse para el Desarrollo de Software Dirigido por Modelos	434
<i>Cristina Vicente, Diego Alonso</i>	
Índice de autores	435

Un análisis de la Calidad en Uso de los Componentes Software utilizando Redes Bayesianas

M.F. Bertoa¹, M.A. Moraga², M.C. Morcillo³, C. Calero²

¹Dept. Lenguajes y Ciencias de la Computación. Universidad de Málaga, Spain

²Dept. Information Technologies and Systems. Universidad de Castilla-La Mancha, Spain

³Dept. Estadística e Investigación Operativa. Universidad de Málaga, Spain

bertoa@lcc.uma.es, mariaangeles.moraga@uclm.es, aixela@uma.es, coral.calero@uclm.es

Resumen. Las propuestas para mejorar la calidad de los productos software se han centrado tradicionalmente en mejorar la Calidad Interna o Externa (independientemente de los posibles contextos de uso), basándose en la idea de que una buena Calidad Externa (según los términos de ISO/IEC 9126) garantiza una buena Calidad en Uso. Este artículo plantea otro camino, cambiando el foco de atención hacia la Calidad en Uso como elemento determinante en el diseño de productos software de alta calidad para usuarios concretos o para seleccionar el producto que mejor se ajuste a las necesidades de un cliente. Nuestra propuesta consiste en realizar un análisis de las relaciones entre la Calidad Externa y la Calidad en Uso para determinar las subcaracterísticas de Calidad Externa que son realmente relevantes para asegurar el nivel requerido de calidad de un producto en un contexto de uso específico. El objetivo es evitar costes superfluos o características irrelevantes para el usuario final que incrementan innecesariamente el coste y el esfuerzo de desarrollo de los productos. En este trabajo proponemos las Redes Bayesianas para modelar esas relaciones y ofrecer un método para definir las de una forma cuantificable. Como ejemplo de uso del método propuesto nos hemos centrado en el dominio de los componentes software para realizar un análisis de su Calidad en Uso.

Keywords: Calidad del software, Calidad en uso, Redes bayesianas, Componentes software.

1 Introducción

Valorar la calidad de productos software es, en general, una tarea difícil y compleja. Un enfoque para simplificar este proceso de evaluación consiste en descomponer la calidad del producto en varios factores o características. Una de las propuestas más extendidas para evaluar la Calidad del software es la definida por la norma ISO/IEC 9126 [5]. Este estándar internacional establece tres niveles (o vistas) en los cuales la calidad del producto software puede ser observada: Interna, Externa y en Uso (Fig. 1).

La Calidad Interna utiliza una aproximación de “caja blanca” del producto software, relacionada principalmente con sus propiedades estáticas y se evalúa durante las fases de diseño y desarrollo. La Calidad Externa usa una propuesta de “caja negra”, valorando el producto software según sus atributos y características externas; es decir, aquellos que pueden ser percibidos por sus usuarios. Finalmente, la Calidad en Uso es

el punto de vista del usuario final cuando el producto software se utiliza en su entorno de trabajo, para llevar a cabo las tareas específicas que el usuario final necesita realizar. Entre estas *Calidades*, o puntos de vista de la Calidad, hay una fuerte relación, como se muestra en la Figura 1.

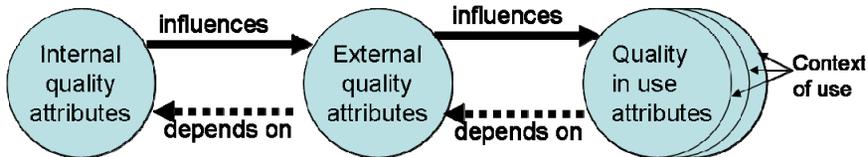


Fig. 1. Relaciones entre la vistas de la Calidad según ISO/IEC 9126 [5].

Hasta ahora, el énfasis de la comunidad de Calidad del Software ha estado puesto en la valoración de la Calidad Interna y Externa de los productos software (véase [1-3, 6, 8, 10, 11]), principalmente porque estas dos vistas están definidas de forma más precisa en la literatura, se pueden identificar más fácilmente y, por tanto, son más fáciles de evaluar.

Esta relación entre la Calidad Externa y en Uso de los productos software parece estar basada en la presunción de que tener un producto con una alta calidad (externa) garantiza un producto con una alta Calidad en Uso. Sin embargo, esto no es necesariamente cierto en una mayoría de situaciones: todos conocemos que un producto con la mejor calidad (externa) no garantiza necesariamente que dicho producto satisfará las necesidades del usuario en su entorno de uso. Especialmente cuando la calidad global (como la percibe el usuario final) está compuesta de muchos factores en conflicto; por ejemplo, un Ferrari no es el mejor coche para ir al trabajo si eres un asistente social que trabaja en un barrio marginal de una gran ciudad.

La Calidad Externa de un producto software es importante, pero en este trabajo cambiamos el foco de atención hacia la Calidad en Uso, ya que la vemos como el factor clave a considerar cuando se diseña un producto para un mercado específico o cuando se selecciona el producto que mejor satisface las necesidades de un cliente.

Hay varias razones que nos han motivado a cambiar la forma de evaluar la calidad de un producto software. En primer lugar, no todas las características de Calidad Externa tienen la misma influencia sobre la Calidad en Uso. Esto, junto con la falsa creencia acerca de la dependencia directa entre la Calidad Externa y la Calidad en Uso, antes mencionada, fuerza muchas veces a sobre-especificar algunos aspectos de los productos, que no son críticos para los usuarios finales con la intención de asegurar un cierto nivel de Calidad en Uso. Esta decisión incrementa innecesariamente los costes y el esfuerzo de desarrollo, sin un efecto proporcional sobre las ventajas que percibe el usuario final.

En segundo lugar, los usuarios normalmente tienden a percibir con igual importancia todas las características de Calidad Externa de un producto software cuando son preguntados acerca de ellos en abstracto; es decir, sin ningún contexto de uso en mente. Sin embargo, cuando se fija un contexto de uso para el producto software, los usuarios son capaces de discriminar las características del producto que realmente les interesan en dicho contexto de aquellas que son deseables pero no imprescindibles desde su punto de vista. De esta forma, la calidad “real” del producto dependerá del contexto de uso; es decir, desde el punto de vista de un coche como una

“caja negra”, el Ferrari es el coche que tiene mejor calidad, sin embargo, esto cambia cuando a ese coche lo ponemos en un contexto de uso determinado.

Este artículo se centra en las relaciones entre la Calidad Externa y la Calidad en Uso de un producto software. Así, se propone un análisis hacia atrás que ayude a seleccionar un grupo reducido de subcaracterísticas de Calidad Externa que sean *verdaderamente relevantes* y que aseguren el nivel de Calidad en Uso requerido, centrándose únicamente en ellas para evitar costes superfluos y aspectos irrelevantes que incrementan innecesariamente el precio final del producto.

Este trabajo analiza las diferentes posibilidades para modelar esas relaciones y proporciona una propuesta que se puede usar en diferentes contextos de uso. Para ello se usan Redes Bayesianas para representar las relaciones entre la Calidad Externa y la Calidad en Uso. Nuestro enfoque propone un modelo para los componentes software dejando para futuros trabajos extender y validar la propuesta para otros dominios diferentes.

Este artículo está organizado como sigue. Tras esta introducción, la sección 2 presenta el modelo de calidad de ISO/IEC 9126, que hemos utilizado como marco de trabajo de calidad en nuestro estudio. A continuación, en la sección 3, se da una breve introducción de las Redes Bayesianas. La sección 4 propone cómo construir una Red Bayesiana para modelar la Calidad en Uso y la sección 5 describe la construcción y utilización de una Red Bayesiana para evaluar la Calidad en Uso de componentes software. Finalmente, la sección 6 discute algunas conclusiones y propone líneas para futuras investigaciones.

2 El modelo de calidad de ISO/IEC 9126

ISO/IEC 9126 [5] propone uno de los modelos de calidad más ampliamente usado, descomponiendo la calidad (global) de un producto software en tres vistas (*Interna*, *Externa* y *En Uso*) y definiendo cada una en términos de características y subcaracterísticas. ISO/IEC 9126 define 6 características para calidad Interna y Externa: *Funcionalidad*, *Fiabilidad*, *Usabilidad*, *Eficiencia*, *Mantenibilidad* y *Portabilidad*. Es de destacar que estas características son las mismas en ambos casos porque el estándar define un paralelismo entre ellas, asumiendo que existe una relación de dependencia una a una (p.e.: la Fiabilidad Interna influye en la Fiabilidad Externa y viceversa). Cada una de las características del modelo de calidad se refina en varias subcaracterísticas para poder realizar un análisis de valoración más preciso.

Respecto a la Calidad en Uso, ISO/IEC 9126 no define un paralelismo similar con la Calidad Externa. De hecho, se definen sólo cuatro características para la Calidad en Uso: *Efectividad*, *Productividad*, *Seguridad* y *Satisfacción*, las cuales no se descomponen en subcaracterísticas. ISO/IEC 9126 no incluye cómo representar la relación de influencia entre las seis características de Calidad Externa y las cuatro de Calidad en Uso, que es precisamente el objetivo de nuestro trabajo.

Finalmente, debemos señalar que ISO está trabajando actualmente en la preparación de una nueva familia de normas ISO 25000 que ha recibido el nombre de SQuaRe (Software QUALity REquirements), que sustituirán entre otros al estándar 9126. Sin embargo, SQuaRe no está aún maduro. Realmente, no hay todavía consenso en el *Working Group* sobre las diferentes vistas de la calidad que quieren distinguir. Por esta causa, hemos decidido basar nuestra propuesta en un estándar internacional

publicado como es ISO/IEC 9126 y esperar a que la propuesta de SQuaRE se establezca. En cualquier caso, la propuesta que presentamos para construir una Red Bayesiana se podría adaptar y construir sin mayor dificultad para el nuevo Modelo de Calidad de SQuaRE, algo que entra en nuestros objetivos de trabajo cuando aparezca publicada una versión de ISO/IEC 25010.

3 Redes Bayesianas

El modelo que vamos a construir tiene en cuenta los problemas usuales que se presentan en la construcción de modelos en la Ingeniería del Software: la recogida de datos necesarios para llevar a cabo el estudio (resultados previos, juicios expertos); estimación de los parámetros intrínsecos del modelo; y datos fiables que nos permitan generalizar las conclusiones que obtengamos.

El primer problema es determinar qué método o herramienta estadística utilizar para realizar un estudio a posteriori entre la Calidad en Uso y la Calidad Externa. Las técnicas clásicas, como el Análisis de Componentes Principales o Modelos Lineales de Regresión, no son las apropiadas para este estudio, pues realizan un análisis clásico, “hacia adelante”, y no a posteriori como nosotros deseamos; también es cierto que los métodos clásicos necesitan una gran cantidad de datos para que dichos análisis sean eficientes y fiables, cosa que no es muy habitual en la construcción de modelos para la Ingeniería del Software.

El siguiente problema es cómo plantear el modelo que explique de forma clara las relaciones e influencias entre las características de la Calidad Externa con las características de la Calidad en Uso. Estas relaciones se pueden plasmar en un grafo dirigido causal.

Por último, hay que decidir cómo medir las relaciones de influencia que se han establecido en el grafo. Ya que los nodos de dicho grafo se pueden considerar variables aleatorias parece que la medida más lógica es medir la influencia de unos nodos sobre otros en términos de probabilidad.

Las Redes Bayesianas se postulan como la mejor herramienta para realizar este estudio. Una Red Bayesiana es un grafo acíclico y dirigido, cuyos nodos son variables de incertidumbre (variables aleatorias) y sus arcos representan la influencia causal entre los nodos. Para cada nodo hay asociada una tabla de probabilidades condicionadas, que explican la influencia causal entre los nodos [7].

Para definir una red bayesiana necesitamos:

1. Dar un conjunto de variables aleatorias (nodos) y las relaciones (influencia causal) entre dichas variables.
2. Construir un grafo con los nodos y las relaciones establecidas entre ellos.
3. Definir tablas de probabilidad condicionada asociadas a cada nodo. Estas tablas determinan el grado de influencia de los nexos del grafo y se utilizan para calcular la distribución de probabilidad de cada nodo en la Red Bayesiana.

A partir de las redes bayesianas podemos: representar las diferentes relaciones entre las características y subcaracterísticas de la Calidad Externa y la Calidad en Uso y el grado de dependencia o influencia entre ellas, incorporando la subjetividad en los criterios de evaluación y la incertidumbre que aparece cuando se establece un criterio

de decisión que engloba a varios criterios de decisión ya establecidos; por ejemplo: los criterios basados en juicios de expertos.

La red así definida, junto con los algoritmos bayesianos de propagación de probabilidades, nos permite hacer inferencia sobre los valores de los nodos de la red a partir de un conjunto de observaciones de evidencias disponibles. Esta inferencia dependerá del objetivo del estudio: o determinar el conjunto de subcaracterísticas de la Calidad Externa que debemos considerar para garantizar un nivel requerido de la Calidad en Uso; o determinar la probabilidad de obtener ciertos resultados en el futuro (inferencia predictiva).

4 Construcción de una red bayesiana para la Calidad en Uso

Nuestro trabajo se basa en la hipótesis de que la Calidad Externa tiene influencia en la Calidad en Uso, que esta influencia se puede modelar mediante una red bayesiana y que esta red nos sirve para realizar un estudio a posteriori sobre el nivel requerido de la Calidad en Externa para asegurar un nivel deseado de la Calidad en Uso.

La forma de modelar las relaciones entre las características de la Calidad Externa y sus subcaracterísticas usando redes bayesianas se muestran en la Figura 2, en la que los nodos de la red son las características y subcaracterísticas, y los nexos representan las relaciones entre ellas.

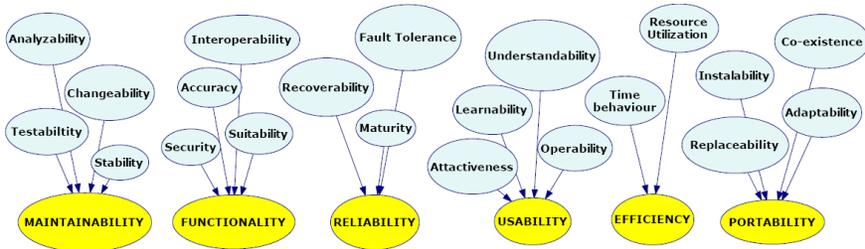


Fig. 2. Características y subcaracterísticas de la Calidad Externa (ISO/IEC 9126)

En segundo lugar modelamos la relación entre la Calidad en Uso y sus cuatro características mediante redes bayesianas. La Figura 3 muestra la red para la Calidad en Uso; hay que recordar ISO/IEC 9126 no define subcaracterísticas para ella.



Fig. 3. Características de la Calidad en Uso (ISO/IEC 9126)

En tercer lugar, necesitamos unir, mediante nexos, estas dos redes para poder definir las relaciones que estamos buscando. Las posibles formas en la que estas dos redes se pueden modelar se describen en un trabajo previo [9], en el que se discuten las distintas opciones para definir las relaciones entre los nodos de las redes de la Calidad Externa y la Calidad en Uso. Básicamente es preciso definir las relaciones entre las subcaracterísticas de la Calidad Externa y las características de la Calidad en

Uso. Estas relaciones determinarán el grado de influencia de cada subcaracterística de la Calidad Externa sobre cada característica de la Calidad en Uso.

Evidentemente se podrían haber definido directamente las relaciones entre los nodos que representan a las características de la Calidad Externa (Mantenibilidad, Funcionalidad, etc.) y los nodos que representan a las características de la Calidad en Uso (Seguridad, Satisfacción, etc.). Pero si hiciésemos esto nos encontraríamos que si una característica de la Calidad Externa está relacionada con una característica de la Calidad en Uso, todas sus subcaracterísticas tendrán algún grado de influencia sobre esa característica de la Calidad en Uso. Además, cada subcaracterística de la Calidad Externa tendrá el mismo grado de influencia sobre una característica de la Calidad en Uso dada, lo que no es correcto.

El enfoque que parece más lógico es relacionar cada subcaracterística de la Calidad Externa con las características de la Calidad en Uso, pero esto produce un número alto de entradas sobre los nodos que representan a las características de la Calidad en Uso. Por este motivo, introducimos nodos sintéticos, práctica común por otra parte, con el fin de simplificar el número de relaciones en la red bayesiana final. Estos nodos se definen, de forma natural, como nodos intermedios entre las subcaracterísticas de cada característica de la Calidad Externa y las características de la Calidad en Uso. Así cada nodo sintético recoge la influencia particular de una característica de la Calidad Externa sobre una característica de la Calidad en Uso.

Como desventaja, el concepto de características de Calidad Externa se pierde. Este inconveniente puede obviarse si hacemos uso de las propiedades y operaciones de las redes bayesianas. De hecho, lo que hacemos es crear cuatro redes individuales, una para cada característica de la Calidad en Uso, y unir las para crear una red completa. Estas cuatro redes son las que constituyen el modelo básico de nuestra propuesta. Una de ellas, la correspondiente a la característica Satisfacción de la Calidad en Uso, es la que se muestra en la Fig. 4.

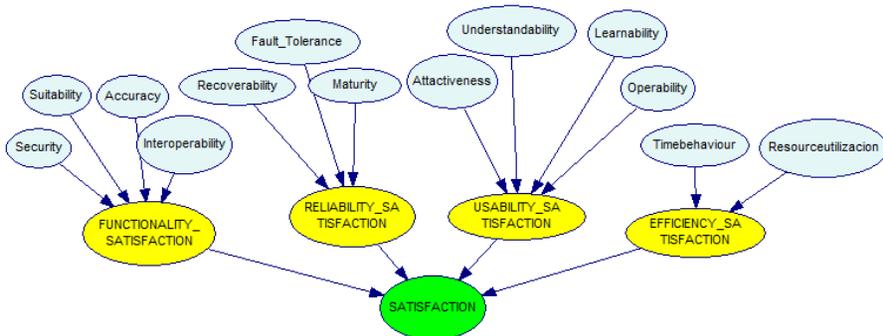


Fig. 4. Red Bayesiana para la característica de la Calidad en Uso *Satisfacción*.

Obsérvese que existen tres niveles de nodos: las subcaracterísticas de la Calidad Externa, primer nivel (nodos padres), las características de la Calidad Externa, nivel intermedio y la característica de la Calidad en Uso particular que se esté estudiando, último nivel.

Este modelo permite el estudio de cada característica de la Calidad en Uso de forma independiente y observar de forma más fácil la influencia de las subcaracterísticas de la Calidad en Externa sobre cada característica de la Calidad en

Uso. Una vez que las cuatro redes están construidas se pueden combinar para hacer un estudio global de la Calidad en Uso.

El último paso para la construcción de la red sería definir los estados o valores de cada nodo de la red y de las tablas de probabilidad asociada a cada uno de ellos. En estas tablas se reflejan las probabilidades de los estados de un nodo en función de los estados de los nodos anteriores que influyen sobre él. Un caso detallado para los componentes software se desarrolla en el siguiente apartado.

5 Uso las Redes Bayesianas para evaluar la calidad de los Componentes Software

Siguiendo los pasos indicados en el punto anterior, hemos definido y construido cuatro Redes Bayesianas, una para cada característica de la Calidad en Uso, para un dominio de aplicación específico, en nuestro caso los componentes software. Estas redes pueden ser una herramienta útil para razonar sobre la relación e influencia de la Calidad Externa sobre la Calidad en Uso en este dominio de aplicación, que es uno de los objetivos de nuestro trabajo.

5.1 Dominio de aplicación

En este punto es muy importante considerar cuál es el dominio de aplicación para el que tenemos que construir las Redes Bayesianas, ya que la mayor o menor relevancia de cada subcaracterística, así como su influencia sobre las características de la Calidad en Uso, dependerán en gran medida de dicho dominio de aplicación y del contexto de uso particular.

En nuestro estudio utilizamos los componentes software de una compañía de software que produce, vende y mantiene aplicaciones para construcción y arquitectura. El contexto de uso ha sido el desarrollo de estas aplicaciones utilizando los componentes seleccionados; es decir, los usuarios fueron los desarrolladores de sistemas cuya tarea es integrar los componentes de software para construir las aplicaciones que la compañía vende y mantiene.

Es importante señalar que es esencial adaptar el modelo de calidad ISO a las necesidades e intereses particulares del dominio de aplicación, con el fin de determinar si algunas de las características no son relevantes para los productos de ese dominio. Así, el caso particular del dominio de aplicación del CBSD, dos de las características de la Calidad Externa se han omitido: *Mantenibilidad* y *Portabilidad*. Estas dos características no eran relevantes para los usuarios finales de los componentes de software en sus contextos de uso, porque son integradores de sistemas. Esta es la razón por la que no aparecen en la red se muestra en la Fig. 5. Para el resto de las características que aparecen en la red, su grado de influencia sobre la Calidad en Uso será calculado por las Redes Bayesianas propuestas.

5.2 Tablas de probabilidad

Para definir los estados de los nodos utilizamos inicialmente las categorías definidas en el estándar internacional 1061 de IEEE[4]: Aceptable, Marginal e Inaceptable. Sin embargo, al hacer la adaptación al contexto de los componentes software,

descubrimos que trabajando con las categorías de Aceptable e Inaceptable era suficiente. De acuerdo al estándar, el valor Marginal se usa sólo cuando la característica no puede ser asignada de forma clara a los valores de Aceptable o Inaceptable. Así, un valor Marginal sólo debe representar el error permisible cuando evaluamos una característica o cuando tenemos incertidumbre sobre ella.

Las tablas de probabilidad de los nodos iniciales son probabilidades a priori derivadas de la evaluación de las subcaracterísticas de la Calidad Externa. En el caso de nuestro estudio hemos preguntado a los usuarios sobre todas las subcaracterísticas de la Calidad Externa, incluso aquellas para las que disponíamos de indicadores objetivos y validados, con el fin de usar estas respuestas como preguntas de control.

Las tablas de probabilidad de los nodos correspondientes a las características de la Calidad Externa, se han definido teniendo en cuenta el grado de influencia (mediante pesos) que cada una de ellas tiene sobre la correspondiente característica de la Calidad en Uso a la que se refiere el estudio. Los nodos del tercer nivel representan a las características de la Calidad en Uso y se construyen de forma similar a las anteriores.

Los valores de las tablas de probabilidad de los nodos del segundo y tercer nivel de la red dependen del dominio de aplicación y del objetivo de los usuarios. Para calcularlas preguntamos a los usuarios, mediante cuestionarios, su opinión subjetiva sobre un conjunto de componentes software que utilizan en la construcción de sus aplicaciones software. Por un lado, pedíamos a los usuarios que valorasen, como Aceptable o Inaceptable, todas las subcaracterísticas y las características de la Calidad Externa. Por otro lado tenían que valorar la característica de la Calidad en Uso objeto del estudio, dando un porcentaje de cómo de Aceptable era el componente respecto a dicha característica.

Con las respuestas obtenidas analizamos, en primer lugar, las posibles observaciones anómalas o respuestas inconsistentes de los encuestados que se debían tener en cuenta a la hora de obtener los parámetros de la red, ya que las tablas de probabilidad heredaban algunas de esas inconsistencias.

Esto nos proporcionó una primera versión de las tablas de probabilidad asociada a cada nodo que tuvieron que ser refinadas para evitar situaciones en las que las respuestas no eran consistentes entre sí. El juicio del experto fue fundamental para este propósito.

5.3 Validación del modelo

Después de la construcción y definición de las tablas de probabilidad de la red obtuvimos cuatro redes (cada una relativa a una de las características de la Calidad en Uso). Para validar que el modelo de decisión representado por la red es correcto, utilizamos un conjunto de datos controlados y proporcionados por los usuarios que incluía observaciones detectadas como anómalas. Evaluamos la red con todos ellos y obtuvimos el la probabilidad, expresada en porcentaje, de Aceptable del nodo final para cada usuario. Teníamos así, para cada dato, el porcentaje de Aceptable para la característica de la Calidad de Uso dado por el encuestado y el porcentaje de Aceptable para dicha característica calculada por la red. Comprobamos si dichos porcentajes estaban correlados obteniendo que el Coeficiente de Determinación, R^2 , fluctuaba entre 0.81, cuando se incluían las observaciones anómalas en el conjunto de datos controlados y 0.9 cuando el conjunto de datos no incluían los datos anómalos. La red parecía adecuada para el análisis deseado.

5.4 Resultados para los componentes software

Por motivos de espacio, vamos a detallar los resultados obtenidos en una de las redes construidas, en concreto para la Efectividad. La red de la Fig. 5 se centra en las cuatro características (y sus subcaracterísticas) de la Calidad Externa que tienen mayor relevancia en el contexto de los componentes.

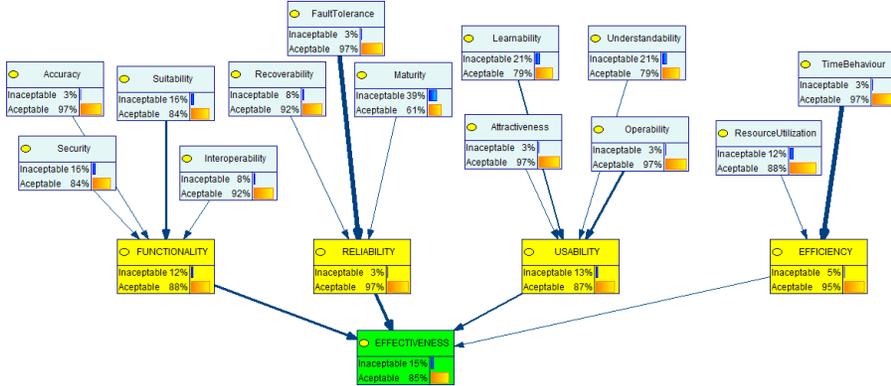


Fig. 5. Diagrama de Influencia para la Efectividad (diagrama de barras).

Instanciando el nodo de Efectividad a Aceptable se calculan las probabilidades a posteriori para los valores de Aceptable e Inaceptable de los nodos de la red y en particular para las subcaracterísticas de la Calidad Externa.

El análisis a posteriori que se realiza nos permite conocer las subcaracterísticas con mayor influencia sobre la Efectividad. Sin embargo, como la probabilidad que tiene la Efectividad de ser Aceptable inicialmente sin instanciar ningún nodo es alta (85%), los incrementos que en la Efectividad produce instanciar una única subcaracterística no sobrepasan el 2%. Por tanto, es muy difícil clasificar o determinar qué subcaracterísticas, de forma individual, tienen una influencia positiva determinante.

Lo que sí se ha observado es que, en general, si tienen una influencia positiva (aumenta la probabilidad de ser Aceptable) también la tienen negativa (disminuye la probabilidad de ser Aceptable). Las variaciones negativas sí son de mayor envergadura y más fáciles de apreciar.

En cualquier caso las variaciones tanto positivas como negativas se hacen más evidentes instanciando un conjunto de subcaracterísticas que influyan de forma significativa sobre la Efectividad.

5.4.1 Clasificación de las subcaracterísticas. Análisis hacia atrás.

Para clasificar las subcaracterísticas de la Calidad Externa hemos utilizado la red bayesiana observando las probabilidades a priori de sus nodos cuando la red no presenta evidencias, y cuando se evidencia el nodo final de Efectividad tanto como Aceptable como Inaceptable.

Hemos analizado las variaciones que experimentan los porcentajes de ser Aceptable de cada subcaracterística cuyos resultados se muestran en la Tabla 1. La columna $V_{ef}(A)$ indica la variación entre el porcentaje de ser Aceptable, inicialmente, de cada subcaracterística ($P(S_{ci}=Aceptable)$) y el porcentaje de ser Aceptable cuando la

Efectividad es Aceptable ($P(\text{Sci}=\text{Aceptable} \mid \text{Eff}=\text{Aceptable})$); la columna $V_{\text{ef}}(\text{In})$ representa la variación que experimenta el porcentaje de ser Aceptable, inicialmente, cada subcaracterística ($P(\text{Sci}=\text{Aceptable})$) y el porcentaje cuando la Efectividad es Inaceptable ($P(\text{Sci}=\text{Aceptable} \mid \text{Eff}=\text{Inaceptable})$).

Subcaracterística Sc_i	$P(\text{Sci}=\text{A})$	$P(\text{Sci}=\text{A} \mid \text{Eff}=\text{A})$	$P(\text{Sci}=\text{A} \mid \text{Eff}=\text{In})$	$V_{\text{ef}}(\text{A})$	$V_{\text{ef}}(\text{In})$
Adecuación	84%	86%	70%	2%	14%
Seguridad	84%	85%	76%	1%	8%
Aprendibilidad	79%	80%	72%	1%	7%
Tolerancia Fallos	97%	98%	91%	1%	6%
Comprensibilidad	79%	80%	74%	1%	5%
Recuperabilidad	92%	93%	89%	1%	3%
Operabilidad	97%	97%	95%	0%	2%
Corrección	97%	97%	95%	0%	2%
Interoperabilidad	92%	92%	91%	0%	1%
Madurez	61%	61%	62%	0%	1%
Comp. Temporal	97%	97%	96%	0%	1%
Atractividad	97%	97%	97%	0%	0%

Tabla 1. Resultado de evidenciar la Efectividad como Aceptable e Inaceptable.

Se puede apreciar que hay un grupo de subcaracterísticas que presentan mayores variaciones en sus porcentajes que denominaremos Esenciales. En este grupo podemos incluir las subcaracterísticas de Adecuación, Seguridad, Aprendibilidad, Tolerancia a Fallos y Comprensibilidad. También se puede observar que hay otro grupo, que denominaremos No Significativas, cuyas variaciones son poco significativas y donde se que incluye al resto de subcaracterísticas.

Así mediante el uso de la red hacia atrás, obtenemos una primera clasificación de las subcaracterísticas de la Calidad Externa en dos categorías: Esenciales y No Significativas.

5.4.2 Clasificación de las subcaracterísticas. Análisis *hacia adelante*.

Utilizar únicamente el criterio anterior para clasificar las subcaracterísticas de Calidad Externa no es suficiente. Por ello, hemos realizado un análisis de la red hacia delante, cuantificando la influencia que tienen las subcaracterísticas sobre el valor de Aceptable para la Efectividad, de tal modo que podamos tener un refinamiento sobre la clasificación propuesta.

La Tabla 2 muestra el porcentaje del valor Aceptable para la Efectividad cuando la red no muestra ninguna evidencia ($P(\text{Eff}=\text{A})$) que es el 85%; cuando se evidencia cada subcaracterística como Aceptable ($P(\text{Eff}=\text{A} \mid Sc_i=\text{A})$); cuando se evidencia cada subcaracterística como Inaceptable ($P(\text{Eff}=\text{A} \mid Sc_i=\text{In})$); y las variaciones de los porcentajes en cada caso ($V_{\text{sc}}(\text{A})$, $V_{\text{sc}}(\text{In})$), siendo:

$$V_{\text{sc}}(\text{A}) = P(\text{Eff}=\text{A} \mid Sc_i=\text{A}) - P(\text{Eff}=\text{A}) \quad \text{y}$$

$$V_{\text{sc}}(\text{In}) = P(\text{Eff}=\text{A}) - P(\text{Eff}=\text{A} \mid Sc_i=\text{In})$$

En este caso podemos observar que existen tres grupos de subcaracterísticas. El primero formado por las subcaracterísticas *Adecuación y Tolerancia a Fallos* que influyen claramente sobre la Efectividad. Estas dos subcaracterísticas aparecían también en la clasificación anterior en el grupo de **Esenciales**.

En sentido opuesto, hay claramente otro grupo con una influencia muy poco significativa sobre la Efectividad que estaría formado por las subcaracterísticas *Interoperabilidad, Madurez, Atractividad, Utilización de Recursos, Comportamiento Temporal*, que también aparecían en la clasificación anterior en el grupo de **No Significativas**.

Subcaracterística Sc_i	$P(\text{Eff}=A)$	$P(\text{Eff}=A \mid Sc_i=A)$	$P(\text{Eff}=A \mid Sc_i=In)$	$Vsc(A)$	$Vsc(In)$
Tolerancia Fallos	85%	86%	53%	1%	32%
Adecuación	85%	87%	72%	2%	13%
Seguridad	85%	86%	77%	1%	8%
Corrección	85%	85%	76%	0%	9%
Operabilidad	85%	85%	76%	0%	9%
Aprendibilidad	85%	86%	79%	1%	6%
Recuperabilidad	85%	85%	79%	0%	6%
Comprensibilidad	85%	86%	81%	1%	4%
Comp. Temp	85%	85%	80%	0%	5%
Atractividad	85%	85%	82%	0%	3%
Interoperabilidad	85%	85%	83%	0%	2%
Madurez	85%	85%	84%	0%	1%
Util. Recursos	85%	85%	84%	0%	1%

Tabla 2. Resultado de evidenciar las subcaracterísticas como Aceptable e Inaceptable.

Queda un grupo intermedio de subcaracterísticas que no podemos clasificar en ninguno de los dos grupos anteriores.

Por un lado las subcaracterísticas Seguridad, Aprendibilidad y Comprensibilidad, que aparecían como Esenciales, no parecen ser tan influyentes sobre la Efectividad cuando se hace el análisis hacia delante. El decremento para Inaceptable ($V(In)$) para ellas, es mucho menor (8%, 6% y 4% respectivamente) que el decremento para Adecuación y Tolerancia a Fallos (13% y 32%, respectivamente).

Por otro lado, las subcaracterísticas Corrección, Operabilidad y Recuperabilidad, que aparecían como No Significativas, cuando se realiza el análisis hacia delante, sí parecen tener mayor influencia que las del resto del grupo No Significativas. El decremento para Inaceptable en la Efectividad (9%, 9% y 6% respectivamente) es mayor que el que se observa para el resto de subcaracterísticas del grupo de No Significativas.

Como vemos, tenemos un grupo intermedio de subcaracterísticas que es difícil clasificar y sobre todo valorar su influencia sobre la Efectividad. Hemos decidido subdividir dicho grupo en dos categorías, según tengan una influencia media, que denominaremos Secundarias o una influencia baja, que denominaremos Auxiliares atendiendo fundamentalmente a la variación que producen en la Efectividad cuando se evidencian como Inaceptable.

Las **Secundarias**, que tienen una influencia “media”, son las subcaracterísticas *Corrección, Seguridad y Operabilidad*. Las **Auxiliares**, que tienen una influencia baja, son las subcaracterísticas: *Recuperabilidad, Comprensibilidad y Aprendibilidad*,

5.4.3 Clasificación de las subcaracterísticas utilizando los dos análisis.

Utilizando los dos análisis y el conjunto de condiciones propuestos en los puntos anteriores, las subcaracterísticas de Calidad Externa quedan clasificadas en cuatro categorías como se resume en la Tabla 3.

Categoría	Subcaracterísticas de la Calidad Externa
C1: Esenciales (Influencia Alta)	Adecuación, Tolerancia a Fallos.
C2: Secundarias (Influencia Media)	Corrección, Seguridad, Operabilidad.
C3: Auxiliares (Influencia Baja)	Recuperabilidad, Aprendibilidad, Comprensibilidad
C4: No Significativas (Influencia Muy baja o Nula)	Interoperabilidad, Madurez, Atractividad, Utilización de Recursos, Comportamiento Temporal.

Tabla 3. Clasificación de las subcaracterísticas de Calidad Externa según su influencia.

En la Tabla 4 se detallan todas las condiciones, recogidas en los dos análisis en términos de variaciones de los porcentajes. Para clasificar una subcaracterística como Esencial o No Significativa debe cumplir las cuatro condiciones correspondientes. En caso de no cumplir alguna de ellas, se clasifica como Secundaria o Auxiliar atendiendo a la valor de $V_{sc}(In)$.

Categoría	$V_{ef}(A)$	$V_{ef}(In)$	$V_{sc}(A)$	$V_{sc}(In)$
Esenciales	$\geq 1\%$	$\geq 5\%$	$\geq 1\%$	$\geq 10\%$
Secundarias	No cumplen todas condiciones de Esenciales ni de No Significativas			$\geq 8\%$
Auxiliares	No cumplen todas condiciones de Esenciales ni de No Significativas			$< 8\%$
No Significativas	$< 1\%$	$< 5\%$	$< 1\%$	$\leq 5\%$

Tabla 4. Condiciones para la clasificación de las subcaracterísticas de Calidad Externa

5.4.4 Valoración de los resultados según categorías.

Como hemos indicado anteriormente, las variaciones en los porcentajes de los valores de Aceptable o Inaceptable para la Efectividad y, por tanto, la influencia de las distintas subcaracterísticas de la Calidad Externa, quedan de forma más evidente cuando las subcaracterísticas que pertenecen a una misma categoría toman el mismo valor.

La Tabla 5 muestra la influencia de cada categoría cuando todas sus subcaracterísticas toman el mismo valor (Aceptable o a Inaceptable), recogiendo el porcentaje de Aceptable para el nodo de Efectividad. Para ilustrar la influencia de cada categoría hemos construido la Tabla 5, donde todas las subcaracterísticas de una categoría toman el mismo valor (se evidencia todas a Aceptable o a Inaceptable) recogiendo el valor para la probabilidad de Aceptable de la Efectividad.

Categoría	$P(\text{Eff} = A \mid C_i = A)$	$P(\text{Eff} = A \mid C_i = In)$
C1: Esenciales (2 subcarcat.)	88%	42%
C2: Secundarias (3 subcarcat.)	87%	60%
C3: Auxiliares (3 subcarcat.)	88%	70%
C4: No Significativas (5 subcarcat.)	85%	76%

Tabla 5. Porcentajes de Aceptable para Efectividad según se evidencian las categorías

Estos resultados confirman lo esperado: la influencia de la categoría Esenciales, con sólo dos subcaracterísticas, se hace notar en el porcentaje de Aceptable para Efectividad. Cuando se instancian las dos subcaracterísticas (Esenciales) como Inaceptable, el porcentaje de Aceptable de la Efectividad baja al 42%.

Por otro lado, la influencia del grupo No Significativas es muy baja: cuando sus cinco subcaracterísticas se instancian como Inaceptable la probabilidad de Aceptable de la Efectividad apenas disminuye al 76%, que es sustancialmente menor al caso de instanciar las dos subcaracterísticas Esenciales (42%) o las tres Secundarias (60%).

Si esta prueba la extendemos a unir e instanciar simultáneamente con el mismo valor, por un lado a las 5 subcaracterísticas Esenciales y Secundarias y por otro lado, las 9 subcaracterísticas No Significativas y Auxiliares obtenemos los resultados mostrados en la Tabla 6, que parecen confirmar la clasificación hecha.

Grupo	$P(\text{Eff} = A)$	$P(\text{Eff} = A G_i = A)$	$P(\text{Eff} = A G_i = \text{In})$
G1: {Esenciales} U {Secundarias} 5 subcaracterísticas	85%	90%	24%
G2: {No Significativas} U {Auxiliares} 8 subcaracterísticas	85%	88%	67%

Tabla 6. Valores para la Efectividad según se evidencien los grupos de subcaracterísticas

Este resultado nos indica qué conjunto de subcaracterísticas debemos vigilar en el desarrollo de nuestro producto (Esenciales y Secundarias), y qué características podemos prestar menor atención (Auxiliares y No Significativas).

6 Conclusiones

En este trabajo hemos mostrado cómo construir un Red Bayesiana para determinar la influencia de las subcaracterísticas de Calidad Externa sobre la Calidad en Uso de un producto software, y cómo esta red se puede usar para determinar las características de Calidad Externa que realmente importan a los usuarios en su particular contexto de uso. Hay varias aplicaciones interesantes de nuestra contribución. Primera, las Redes Bayesianas pueden ser una herramienta muy eficiente para el personal de marketing que trabaja en compañías que diseñan y venden productos software. Las Redes Bayesianas permiten identificar las características del producto que no son realmente relevantes para su público objetivo y cuyos costes de mantenimiento y desarrollo asociados se pueden reducir drásticamente. Segunda, una Red Bayesiana de este tipo puede ayudar a las compañías interesadas en seleccionar y utilizar productos software, a identificar cuáles son las características del producto que realmente les interesan. A partir de estas premisas, se buscarán los productos que destaquen en esas características, poniendo menos atención en otras propiedades que no son relevantes para su contexto de uso. En otras palabras, nuestro trabajo puede ser utilizado de forma efectiva por los clientes para la selección de los productos que mejor se ajusten a sus necesidades con el menor coste posible.

Por supuesto, el nivel de influencia de las subcaracterísticas de la Calidad Externa sobre las Características de la Calidad en Uso puede variar entre diferentes dominios de aplicación. Las relaciones modeladas por las Redes Bayesianas pueden necesitar

adaptarse a otros dominios. En cualquier caso, el método presentado en este documento sigue siendo válido.

Hay varias actividades que tenemos previsto abordar a corto plazo. En primer lugar, queremos refinar esta propuesta combinándola con la aplicación de técnicas bayesianas robustas (Modelo de Regresión Dicotómico Robusto) durante la definición inicial de la Red Bayesiana y de las tablas de probabilidades condicionadas, a partir del cual podremos confirmar (y mejorar) las relaciones definidas en la Fig. 5 entre subcaracterísticas de Calidad Externa y las características de Calidad en Uso.

Por último, esperamos proporcionar una aportación útil al Grupo de Trabajo de ISO que define la nueva familia de normas SQuaRe basada en nuestra investigación, no sólo confirmando la existencia de la influencia de Calidad Externa sobre la Calidad en Uso, sino además cuantificándola en algunos contextos de uso concretos; y, lo más importante, destacando el papel primordial que desempeña la Calidad en Uso en la evaluación de la calidad de cualquier producto de software como motor del resto de puntos de vista de la calidad.

Referencias

1. Bertoa, M.F., Troya, J.M., and Vallecillo, A., Measuring the usability of software components. *Journal of Systems and Software*. 79(3): 427-439 (2006)
2. Botella, P., Burgués, X., Carvallo, J.P., Franch, X., Pastor, J.A., and Quer, C., Towards a Quality Model for the Selection of ERP Systems. *Component-Based Software Quality*. Springer-Verlag: 225-245 (2003)
3. Dromey, R.G., A Model for Software Product Quality. *IEEE Transaction on Software Engineering*. 21(2): 146-162 (1995)
4. IEEE, Std. 1061-1998. IEEE Standard for a Software Quality Metrics Methodology. (1998).
5. ISO/IEC, ISO/IEC 9126. Software Engineering-Product Quality. Parts 1 to 4. International Organization for Standardization/International Electrotechnical Commission. (2001).
6. Jagdish, B., A Hierarchical Model for object-oriented Design Quality Assessment. *IEEE Transaction on Software Engineering*. 28(1): 4-17 (2002)
7. Jensen, F.V., *Bayesian Networks and Decisions Graphs*. Springer-Verlag. (2001).
8. Liu, K., Zhou, S., and Yang, H. Quality Metrics of Object Oriented Design for Software Development and Re-Development. In: *The First Asia-Pacific Conference on Quality Software (APAQS'00)* pp. 127-135. (2000)
9. Moraga, M.Á., Bertoa, M.F., Morcillo, M.C., Calero, C., and Vallecillo, A. Evaluating Quality-in-Use Using Bayesian Networks. In: *12th Workshop on Quantitative Approaches on Object Oriented Software Engineering (QAOOSE 2008)*. Paphos, Cyprus. (2008)
10. Moraga, M.Á., Calero, C., and Piattini, M., Comparing different quality models for portals. *Online Information Review*. 30(5): 555-568 (2006)
11. Neil, M., Krause, P., and Fenton, N.E., Software Quality Prediction Using Bayesian Networks, in *Software Engineering with Computational Intelligence*. Chapter 6. 2003, Kluwer