



13th Conference on Software Engineering and Databases

XIII Jornadas de Ingeniería del Software y Bases de Datos

Gijón (Spain), October 7-10 2008

EDITORS: Ana Moreira
María José Suárez-Cabal
Claudio de la Riva
Javier Tuya

13th Conference on Software Engineering and Databases

XIII Jornadas de Ingeniería del Software y Bases de Datos

Gijón (Spain), October 7-10 2008

EDITORS: Ana Moreira
María José Suárez-Cabal
Claudio de la Riva
Javier Tuya

Edita:
Ana Moreira
María José Suárez-Cabal
Claudio de la Riva
Javier Tuya

Filmación e impresión:
Gráficas Rigel

Depósito Legal:
AS - 5.236 - 08

ISBN:
978-84-612-5820-8

Volume Editors Details

Ana Moreira

Departamento de Informática
Faculdade de Ciências e tecnologia
Universidade Nova de Lisboa
2829-516 Caparica, Portugal
E-mail: amm@di.fct.unl.pt
URL: <http://ctp.di.fct.unl.pt/~amm/>

María José Suárez-Cabal

Departamento de Informática
Universidad de Oviedo
33204 Gijón, Spain
E-mail: cabal@uniovi.es

Claudio de la Riva

Departamento de Informática
Universidad de Oviedo
33204 Gijón, Spain
E-mail: claudio@uniovi.es
URL: <http://www.di.uniovi.es/~claudio/>

Javier Tuya

Departamento de Informática
Universidad de Oviedo
33204 Gijón, Spain
E-mail: tuya@uniovi.es
URL: <http://www.di.uniovi.es/~tuya/>

Preface

Celebrating 13 Years of JISBD

With the 2008 edition in Gijón (October 7 to 10), the Conference on Software Engineering and Databases (JISBD) celebrates 13 years of existence. Born as a forum where the Spanish community would publish their work, meet to discuss potential research collaborations and evaluate the progress of research projects funded by the Spanish Ministry of Science and Technology, JISBD has long since moved beyond its initial boundaries and crossed several oceans.

Presently, the conference has become an important reference for younger researchers, as well as a forum which the more experienced do not wish to miss. In recent years, JISBD has broadened its radius, accepting papers also in English and Portuguese, in addition to Spanish. This change, not only brought more conference participants, but also significantly increased the number of submissions and, principally, the quality of the submissions accepted.

The JISBD community is now self-sustained and continues to expand. The quality of work accepted is equivalent to that of other relevant international events. In recent years, it has been possible to edit a special volume of IEEE LA with extended versions of the best conference papers and this also is happening with the current edition. This special issue, together with the conference proceedings with ISBN, is a showcase of the quality of the work of JISBD.

One of the highlights of this conference has been the excellence of its keynote speakers. Many of the most admired international researchers and professionals have already been invited to address the JISBD participants.

Within this rich framework for scientific and technological interchange, the conference includes several satellite events. In addition to the presentation of high quality original papers in the main conference, the program includes tutorials, tool demonstrations and workshops for the discussion of innovative ideas and work in progress, as well as a forum to bring to a wider audience research work already published in prestigious journals or conference proceedings (with an acceptance rate below 25% and an impact factor above 0.5).

It is no exaggeration to claim that JISBD has been consolidating its position as a reference event where researchers and professionals of Software Engineering and Databases can get together to discuss results and share ideas. JISBD has become an important forum for collaboration between different strands and research groups, while continuing to offer its participants a well organized event with exceptional hospitality.

About this Edition

The increased global reach of JISBD is evident in the origin of papers received. This year, in addition to the two Iberian and ten Latin-American countries, submissions arrived also from China, France, Germany, India, Iran and Pakistan.

Of a total of 115 abstracts, 112 papers were submitted for review. Most papers were reviewed by three PC members, and several were reviewed by four. The program Committee accepted 30 full papers and selected 12 for presentation as short papers. The acceptance rate for full papers was approximately 25%.

The increasing success of the conference implies greater responsibilities in terms of guaranteeing independent judgement and ensuring compliance with international standards of ethics. For this reason, a greater effort has been made in recent years to avoid double submissions, a task made

more difficult by the fact that the conference accepts submissions in three languages. This year, three good papers were rejected due to double submission, in different languages to different events.

In addition to the accepted papers, the conference includes five workshops, one tutorial, nine tool demos, an industrial panel and also a forum to discuss important relevant work already published elsewhere.

A highlight of the conference is, without doubt, the excellence of the invited keynote speakers. This year is no exception and we are honoured indeed to receive Bashar Nuseibeh and Bran Selic.

Bashar Nuseibeh is an academic and researcher at the Open University in the UK and invited professor in various other universities, including Japan's National Institute of Informatics. Bashar chairs several international committees and is recognized also for industry work, including organizations such as the UK's National Air Traffic Services (NATS), Texas Instruments, Praxis Critical Systems, Philips Research Labs, and NASA.

Bran Selic was, for many years, a distinguished engineer and researcher at IBM, and currently heads a global consultancy based in Canada. He is internationally known for his work in large-scale industrial systems, and for his pioneering work in Model-Driven Development and Real-Time Embedded Systems.

Bashar's keynote is entitled "*The five W's (and one "H") of Security: ... Software Engineering of Secure Systems*" while Bran's is on "*Model-Based Software Engineering: Expected and Unexpected Challenges*".

Acknowledgements

A very special word of thanks is due to Bashar and Bran for having accepted my invitation and for sharing all the participants their knowledge, experience and refined wit. I sincerely hope JISBD was also for them a gratifying and unique experience.

Acknowledgements are due to a multitude of collaborators without whom the conference could not have been a success. Firstly, the paper authors for the trust placed in the quality of JISBD as a conference that merited their submissions. Secondly, to the PC members, whose diligent review work ensured that the authors' trust continues to be justified.

For managing the submission and review process, I was fortunate to have the constant help of Juan Hernández and José Javier Berrocal; they were my guardian angels, constantly alert to deadlines and ready to help as necessary. A special thanks for the contribution of my "Executive Program Committee", Antonio Vallecillo, Juan Hernández, Miguel Toro, Vicente Pelechano and Xavier Franch.

Acknowledgement is due to the main conference organizers, especially to Javier Tuya and his co-chair, Claudio de la Riva, for their efficient handling of the numerous tasks that a conference of this size and quality entails. Thanks also to those responsible for the satellite events (in alphabetical order), António Rito Silva, Antonio Vallecillo, Gustavo Rossi, João Araújo, João Falcão e Cunha, José Berrocal, José Corrales, José García-Fanjul, João Miguel Fernandes, Lidia Fuentes and María José Suárez-Cabal.

Finally, a special word of thanks to the sponsors of this conference, without whose contribution the event would have been somewhat less charming (not to mention gastronomically less satisfying).

Ana Moreira
Program Committee Chair

Prefácio

Celebrando 13 Anos de JISBD

Com a edição de 2008 em Gijón (7-10 Outubro), a Conferência em Engenharia de Software e Bases de Dados (JISBD) celebra 13 anos de existência. Apesar de ter nascido como um fórum onde a comunidade espanhola publicava os seus trabalhos e se reunia para discutir potenciais colaborações futuras de investigação, e até avaliar o estado de andamento dos projectos de investigação financiados pelo Ministério da Ciência e Tecnologia espanhola, há muito que extravasou essas fronteiras e cruzou oceanos.

Actualmente o JISBD é um marco importante na investigação dos mais jovens, mas também um fórum que os mais seniores não querem perder. Nos últimos anos a conferência abriu-se para o mundo inteiro, aceitando artigos escritos em Inglês, Espanhol e Português. Esta viragem trouxe não só mais participantes à conferência, mas também um aumento significativo do número de trabalhos submetidos e, principalmente, um aumento na qualidade desses trabalhos.

A comunidade do JISBD é agora auto-sustentada e em contínua expansão. A qualidade dos trabalhos aceites é equiparada à de muitos outros eventos internacionais de relevo. Por este motivo, nos últimos anos, foi-nos possível editar um volume especial no IEEE LA com uma versão estendida dos melhores trabalhos da conferência, o que acontecerá também nesta edição. Este volume, em conjunto com as actas formais da conferência com ISBN, é uma mostra da qualidade do trabalho que aqui se discute.

Uma das características de excelência desta conferência tem sido, desde sempre, o gabarito dos seus palestrantes convidados. É um prazer ver que muitos dos mais admirados investigadores e profissionais internacionais já foram convidados a falar para os participantes do JISBD.

Neste enquadramento fecundo para divulgação científica e tecnológica, a conferência inclui vários eventos satélite. Além dos artigos seleccionados para apresentação na conferência, o programa inclui ainda tutoriais, demonstrações de ferramentas, workshops para discussão de ideias inovadores e trabalhos em andamento, assim como um evento para a disseminação de trabalho de investigação já publicado em revistas e actas de conferências de grande prestígio (onde o índice de aceitação é inferior a 25% e o factor de impacto superior a 0.5).

Assim, não é excessivo afirmar que o JISBD se tem vindo a consolidar como um evento de referência onde investigadores e profissionais em Engenharia de Software e Bases de Dados se encontram para discutir, disseminar e trocar ideias, partilhar experiências e resultados entre diversos sectores e grupos de investigação, num contexto de excelente organização e invulgar hospitalidade.

Sobre esta Edição

A atestar o crescimento e internacionalização do JISBD está a origem dos artigos que nos chegaram. Este ano, a nacionalidade dos autores foi surpreendentemente diversificada, pois para além dois países Ibéricos e de dez países Latino-Americanos, recebemos trabalhos também da Alemanha, China, França, Índia, Irão e Paquistão.

O número total de resumos foi de 115, sendo que destes, 112 artigos foram submetidos para avaliação. Cada artigo foi avaliado por pelo menos três revisores, sendo que vários foram avaliados por quatro. O Comité de Programa aceitou 30 artigos longos e escolheu 12 para apresentação como artigos curtos. Assim, o índice de aceitação de artigos longos foi de cerca de 25%.

Este sucesso acarreta responsabilidades acrescidas em garantir a independência de julgamentos e em fazer cumprir a ética e as normas internacionais. É por este motivo que, nos últimos anos, se

tem feito um esforço muito grande para evitar submissões duplicadas, tarefa nem sempre fácil para os membros do Comitê de Programa, já que a conferência aceita três línguas de escrita. Este ano foram rejeitados três bons artigos avaliados como de submissão duplicada, em duas línguas, para eventos diferentes.

Para além dos artigos seleccionados, a conferência conta também com a organização de cinco *workshops*, um *tutorial*, nove demonstrações de ferramentas, um painel industrial e ainda um fórum onde se discutem trabalhos de relevo já publicados em revistas ou outras conferências.

Mas sem dúvida que os momentos mais altos da conferência são sempre marcados pelo admirável conjunto de palestrantes convidados. Este ano tivemos a sorte de receber Bashar Nuseibeh e de Bran Selic.

Bashar Nuseibeh é um académico e investigador da Open University, na Inglaterra, e professor convidado em várias outras universidades, incluindo o Instituto Japonês de Informática. Bashar preside vários comités internacionais e é admirado também pelo seu trabalho para a indústria, que inclui organizações como o National Air Traffic Services (NATS) do Reino Unido, Texas Instruments, Praxis Critical Systems, Philips Research Labs, e a NASA.

Bran Selic foi durante umas dezenas de anos engenheiro e investigador distinguido da IBM e actualmente preside uma empresa de consultoria internacional sediada no Canadá. É conhecido mundialmente pelos seus trabalhos em sistemas de larga escala industrial e também pelo seu pioneirismo nas áreas de desenvolvimento orientado a modelos e sistemas embutidos de tempo real.

A palestra do Bashar é intitulada “*The five W’s (and one “H”) of Security: ... Software Engineering of Secure Systems*”, enquanto que a do Bran é sobre “*Model-Based Software Engineering: Expected and Unexpected Challenges*”.

Agradecimentos

Uma palavra especial de agradecimento ao Bashar e ao Bran por terem aceite o meu convite e por brindarem todos os participantes com a sua experiência, conhecimento e refinado sentido de humor. Espero que o JISBD tenha sido também para eles uma experiência agradável e diferente.

Agradecimentos são justamente devidos ao grande número de colaboradores, sem o contributo dos quais, a conferência não poderia ter tido êxito. Aos autores, claro, por confiarem na qualidade do JISBD e submeterem, por isso, os seus trabalhos. Aos membros do Comitê de Programa cujas revisões asseguram que essa confiança continua a justificar-se.

Para gerir o sistema de submissão, contei com o apoio incondicional do Juan Hernández e do José Javier Berrocal. Eles foram os meus “anjos da guarda”, sempre atentos a todos os prazos e prontos a dar todas as explicações. Um agradecimento particular ao contributo meu “Comitê Executivo de Programa”, Antonio Vallecillo, Juan Hernández, Miguel Toro, Vicente Pelechano e Xavier Franch. Obrigada pelo vosso apoio e sugestões.

Obrigada aos organizadores principais da conferência, em especial ao Javier Tuya, e ao seu vice-presidente, Claudio de la Riva, pela gestão eficaz das inúmeras tarefas que uma conferência desta dimensão exige. Um agradecimento é ainda devido, e por ordem alfabética, aos responsáveis dos eventos satélite, António Rito Silva, Antonio Vallecillo, Gustavo Rossi, João Araújo, João Falcão e Cunha, José Berrocal, José Corrales, José García-Fanjul, João Miguel Fernandes, Lidia Fuentes e María José Suárez-Cabal.

Finalmente, um agradecimento aos patrocinadores da conferência, sem o contributo de quem o evento teria tido menos charme (e uma gastronomia muito menos requintada).

Ana Moreira
Presidente do Comitê de Programa

Prefacio

Con esta edición 2008 en Gijón (7 al 10 de Octubre), las Jornadas de Ingeniería del Software y Bases de Datos (JISBD) celebra 13 años de existencia. JISBD nació como un foro donde la comunidad española publicaba su trabajo, discutía potenciales colaboraciones en investigación y evaluaba el progreso de los proyectos de investigación financiados por el Ministerio de Ciencia y Tecnología, y en la actualidad ha traspasado fronteras y cruzado varios océanos.

Actualmente, la conferencia es una referencia importante para jóvenes investigadores, así como un foro de cita obligada para investigadores más experimentados. Durante los últimos años, JISBD se ha abierto al mundo, aceptando artículos en Inglés y Portugués, además de Castellano. Este cambio no solamente se ha traducido en más participantes, sino que ha incrementado significativamente el número de artículos enviados, y principalmente, la calidad de los artículos aceptados.

La comunidad JISBD está actualmente auto sustentada y continúa expandiéndose. La calidad de los trabajos aceptados es equivalente al de otros eventos internacionales relevantes. Durante los últimos años, ha sido posible editar un volumen especial de IEEE LA con versiones ampliadas de los mejores trabajos presentados en la conferencia, lo que sucederá también en la presente edición. Este volumen especial, junto con las actas de la conferencia con ISBN, es una muestra de la calidad de los trabajos de JISBD.

Una de las características más sobresalientes de la conferencia ha sido la calidad de los ponentes invitados. Varios investigadores y profesionales de reconocido prestigio internacional han sido invitados a participar como ponentes en JISBD.

Dentro de este marco científico y tecnológico, la conferencia incluye varios eventos relacionados. Además de la presentación de artículos originales de alta calidad en la conferencia principal, el programa incluye tutoriales, demostraciones de herramientas, talleres para la discusión de ideas innovadoras y trabajos en curso, así como la divulgación de trabajos de investigación publicados en revistas y conferencias de prestigio (con un ratio de aceptación por debajo del 25% y un factor de impacto por encima de 0,5).

No es una exageración afirmar que JISBD ha consolidado su posición como un evento de referencia donde investigadores y profesionales de la Ingeniería del Software y las Bases de Datos se reúnen para discutir resultados y compartir ideas. JISBD se ha convertido en un foro importante para la colaboración entre diferentes sectores y grupos de investigación, en un contexto de excelente organización y excepcional hospitalidad.

Sobre la presente edición

El crecimiento e internacionalización de JISBD se hace evidente analizando el origen de los artículos recibidos. En la presente edición, además de los artículos recibidos de los dos países de la Península Ibérica y los diez países Latinoamericanos, se han recibido artículos de China, Francia, Alemania, India, Irán y Pakistán.

De un total de 115 resúmenes previamente recibidos, finalmente se recibieron 112 artículos para su revisión. La mayoría de los artículos fueron revisados por tres miembros del Comité de Programa y varios por cuatro. El Comité de Programa aceptó 30 artículos largos y seleccionó 12 para su presentación como artículos cortos. El ratio de aceptación para los artículos largos fue de aproximadamente el 25%.

El éxito de la conferencia implica grandes responsabilidades en términos de garantizar la independencia de las revisiones y el cumplimiento de los estándares internacionales de ética. Por esta razón, durante los últimos años se ha realizado un mayor esfuerzo en aras de evitar envíos duplicados, una tarea especialmente dificultosa, ya que la conferencia acepta envíos en tres idiomas. En la

presente edición tres artículos fueron rechazados debido al doble envío en diferentes idiomas para diferentes eventos.

Además de los artículos aceptados, la conferencia incluye cinco talleres, un tutorial, nueve demostraciones de herramientas y foro para la discusión y divulgación de trabajos relevantes previamente publicados, así como una mesa redonda de carácter industrial.

Una característica importante de la conferencia es, sin ninguna duda, la excelencia de los ponentes invitados. La presente edición no es una excepción y estamos orgullosos de contar con la presencia de Bashar Nuseibeh y Bran Selic.

Bashar Nuseibeh es académico e investigador en la Open University del Reino Unido y profesor invitado en otras muchas universidades, incluyendo el Instituto Nacional Japonés de Informática. Bashar preside varios comités internacionales y está reconocido igualmente por su trabajo industrial, incluyendo organizaciones tales como el Servicio Nacional de Tráfico Aéreo del Reino Unido (NATS), Texas Instruments, Praxis Critical Systems, Philips Research Labs y la NASA.

Bran Selic fué durante varios años un destacado ingeniero e investigador en IBM y actualmente lidera una consultora internacional con sede en Canadá. Es internacionalmente conocido por su trabajo en sistemas industriales a gran escala y por su trabajo pionero en Desarrollo Dirigido por Modelos y Sistemas Empotrados en Tiempo Real.

La conferencia de Bashar se titula *“The five W’s (and one “H”) of Security: ... Software Engineering of Secure Systems”* y la de Bran *“Model-Based Software Engineering: Expected and Unexpected Challenges”*.

Agradecimientos

Un agradecimiento especial es para Bashar y Bran por haber aceptado mi invitación y por compartir con todos los participantes sus conocimientos, experiencia y refinado sentido del humor.

Agradecimientos también para la multitud de colaboradores sin los cuales el éxito de la conferencia no habría sido posible. En primer lugar, para los autores de los artículos por confiar en la calidad de JISBD y enviar sus trabajos. En segundo lugar, para los miembros del Comité de Programa, cuyas revisiones aseguran la calidad de los trabajos.

Para el proceso de gestión y revisión de los trabajos recibidos, fui afortunada por tener la ayuda constante de Juan Hernández y José Javier Berrocal. Ellos fueron mis ángeles guardianes, alertándome constantemente de las fechas límite y siempre preparados para ayudarme cuando lo necesitaba. Agradecimientos especiales por la contribución de mi “Comité de Programa Ejecutivo”, Antonio Vallecillo, Juan Hernández, Miguel Toro, Vicente Pelechano y Xavier Franch.

Agradecimientos también para los organizadores de la conferencia principal, especialmente al presidente del comité organizador Javier Tuya y su vicepresidente Claudio de la Riva, por su manejo eficiente de las numerosas tareas que una conferencia de este tamaño y calidad conllevan. Agradecimientos también para los responsables de los eventos relacionados (en orden alfabético) António Rito Silva, Antonio Vallecillo, Gustavo Rossi, João Araújo, João Falcão e Cunha, José Berrocal, José Corrales, José García-Fanjul, João Miguel Fernandes, Lidia Fuentes y María José Suárez-Cabal.

Finalmente, palabras especiales de agradecimiento para los patrocinadores de la conferencia, sin cuya contribución el evento habría sido menos encantador (y con una gastronomía menos refinada).

Ana Moreira
Presidenta del Comité de Programa

Conference Committee

Program Committee Chair

Ana Moreira (Univ. Nova de Lisboa, Portugal)

Organizing Chair

Javier Tuya (Univ. Oviedo, Spain)

Organizing Co-Chair

Claudio de la Riva (Univ. Oviedo, Spain)

Permanent Committee Secretary

Mario Piattini (Univ. Castilla-La Mancha, Spain)

Tutorial Chair

António Rito Silva (Univ. Técnica Lisboa, Portugal)

Workshop Chair

João Araújo (Univ. Nova de Lisboa, Portugal)

Tool Demonstrations Chair

Lidia Fuentes (Univ. Málaga, Spain)

Relevant Papers Dissemination Chairs

Antonio Vallecillo (Univ. Málaga, Spain)

João Falcão Cunha (Univ. Porto, Portugal)

Proceedings Chair

María José Suárez-Cabal (Univ. Oviedo, Spain)

Cyber Chair

Jose Javier Berrocal (Univ. Extremadura, Spain)

Web Chair

José A. Corrales (Univ. Oviedo, Spain)

Publicity Chairs

Gustavo Rossi (Univ. La Plata, Argentina)

José García-Fanjul (Univ. Oviedo, Spain)

João Miguel Fernandes (Univ. Minho, Portugal)

Organizing Committee (Univ. Oviedo, Spain)

Javier Tuya
Claudio de la Riva
José García-Fanjul
Isabel Sevilla
María José Suárez-Cabal
José Ramón de Diego
Raquel Blanco
Eugenia Díaz Fernández
José A. Corrales
Marta Fernández de Arriba

SISTEDES Executive Board

President

Miguel Toro (Univ. Sevilla, Spain)

Vice President

Juan José Moreno (Univ. Polit. Madrid, Spain)

Secretary

Nieves R. Brisaboa (Univ. Coruña, Spain)

Treasurer

Javier Tuya (Univ. Oviedo, Spain)

Members

Pere Botella (Univ. Polit. Catalunya, Spain)
Ricardo Peña (Univ. Complutense Madrid, Spain)
Coral Calero (Univ. Castilla-La Mancha, Spain)
Manuel Hermenegildo (Univ. Polit. Madrid, Spain)
Ernesto Pimentel (Univ. Málaga, Spain)
María Ribera Sancho (Univ. Polit. Catalunya, Spain)
Natalia Juristo (Univ. Polit. Madrid, Spain)
Salvador Lucas (Univ. Polit. Valencia, Spain)

Submission and Review Support System (Quercus Software Engineering Group)

Javier Berrocal (Univ. Extremadura, Spain)
Juan Hernández (Univ. Extremadura, Spain)

Secretariat

Fundación Universidad de Oviedo
C/ Principado 3, 4ª Planta
33007 Oviedo, Spain.
Tel: 34-985104927
Fax: 34-985104928

Executive Program Committee

Xavier Franch (Univ. Polit. Catalunya, Spain)
Juan Hernández (Univ. Extremadura, Spain)
Vicente Pelechano (Univ. Polit. Valencia, Spain)
Antonio Vallecillo (Univ. Málaga, Spain)
Miguel Toro (Univ. Sevilla, Spain)
Javier Tuya (Univ. Oviedo, Spain)

Program Committee

Albert Abelló (Univ. Polit. Catalunya, Spain)
Ana Paula Afonso (Univ. Lisboa, Portugal)
Ademar Aguiar (Univ. Porto, Portugal)
Jesús Aguilar (Univ. Sevilla, Spain)
José Aldana (Univ. Málaga, Spain)
Mauricio Alférez (U. Nova de Lisboa, Portugal)
Bárbara Álvarez (Univ. Polit. Cartagena, Spain)
Raquel Anaya (Univ. EAFIT, Colombia)
María José Aramburu (Univ. Jaume I, Spain)
Hernán Astudillo (U. T. Federico Santa María, Chile)
Orlando Belo (Univ. do Minho, Portugal)
Rafael Berlanga (Univ. Jaume I, Spain)
Paulo Borba (Univ. Federal Pernambuco, Brazil)
Pere Botella (Univ. Polit. Catalunya, Spain)
Rosana Braga (Univ. São Paulo, Brazil)
Nieves Brisaboa (Univ. Coruña, Spain)
Isabel Brito (Inst. Polit. Beja, Portugal)
Fernando Brito e Abreu (U. Nova de Lisboa, Portugal)
Coral Calero (Univ. Castilla-La Mancha, Spain)
Marcelo Campo (UNICEN, Argentina)
Carlos Canal (Univ. Málaga, Spain)
Valeria de Castro (Univ. Rey Juan Carlos, Spain)
Matilde Celma (Univ. Polit. Valencia, Spain)
Christina Chávez (Univ. Bahia, Brazil)
Rafael Corchuelo (Univ. Sevilla, Spain)
Dolors Costal (Univ. Polit. Catalunya, Spain)
Yania Crespo (Univ. Valladolid, Spain)
Carlos Delgado (Univ. Carlos III, Spain)
Oscar Díaz (Univ. País Vasco, Spain)
Javier Dolado (Univ. País Vasco, Spain)
Xavier Franch (Univ. Polit. Catalunya, Spain)
Pablo de la Fuente (Univ. Valladolid, Spain)
Mario Gaspar da Silva (Univ. Lisboa, Portugal)
Alessandro García (Univ. Lancaster, UK)
Marcela Genero (Univ. Castilla-La Mancha, Spain)
Cristina Gómez (Univ. Polit. Catalunya, Spain)
Jaime Gómez (Univ. Alicante, Spain)
Alfredo Goñi (Univ. País Vasco, Spain)
Silvia Gordillo (UNLP, Argentina)
Pedro Guerreiro (Univ. Algarbe, Portugal)
Juan Hernández (Univ. Extremadura, Spain)
Jon Iturrioz (Univ. País Vasco, Spain)
Elena Jurado (Univ. Extremadura, Spain)
Natalia Juristo (Univ. Polit. Madrid, Spain)
Miguel Katrib (Grupo WEBOO, Cuba)
María Lencastre (Univ. Pernambuco, Brazil)
Antonia Lopes (Univ. Lisboa, Portugal)
Adolfo Lozano (Univ. Extremadura, Spain)
Esperanza Marcos (Univ. Rey Juan Carlos, Spain)
Henrique Madeira (Univ. Coimbra, Portugal)
Eduardo Mena (Univ. Zaragoza, Spain)
Ana María Moreno (Univ. Polit. Madrid, Spain)
Juan José Moreno (Univ. Polit. Madrid, Spain)
Juan Manuel Murillo (Univ. Extremadura, Spain)
Oscar Pastor (Univ. Polit. Valencia, Spain)
Vicente Pelechano (Univ. Polit. Valencia, Spain)
Marcelo Pimenta (Univ. F. Rio Grande do Sul, Brazil)
Ernesto Pimentel (Univ. Málaga, Spain)
Mónica Pinto (Univ. Málaga, Spain)
Ángeles Places (Univ. Coruña, Spain)
Antonio Polo (Univ. Extremadura, Spain)
Claudia Pons (UNICEN, Argentina)
Tom Price (Univ. F. Rio Grande do Sul, Brazil)
Carme Quer (Univ. Polit. Catalunya, Spain)
Celia Ramos (Univ. Algarbe, Portugal)
Isabel Ramos (Univ. Sevilla, Spain)
Isidro Ramos (Univ. Polit. Valencia, Spain)
Claudio de la Riva (Univ. Oviedo, Spain)
José Riquelme (Univ. Sevilla, Spain)
José Luis Roda (Univ. La Laguna, Spain)
María José Rodríguez Fortis (Univ. Granada, Spain)
José Raúl Romero (Univ. Córdoba, Spain)
Antonio Ruiz (Univ. Sevilla, Spain)
Francisco Ruiz (Univ. Castilla-La Mancha, Spain)
José Samos (Univ. Granada, Spain)
Fernando Sánchez (Univ. Extremadura, Spain)
Juan Sánchez (Univ. Polit. Valencia, Spain)
Carla Silva (Univ. F. Pernambuco, Brazil)
Ernest Teniente (Univ. Polit. Catalunya, Spain)
Miguel Toro (Univ. Sevilla, Spain)
Ambrosio Toval (Univ. Murcia, Spain)
Juan Carlos Trujillo (Univ. Alicante, Spain)
Toni Urpi (Univ. Polit. Catalunya, Spain)
Antonio Vallecillo (Univ. Málaga, Spain)
Belén Vela (Univ. Rey Juan Carlos, Spain)

Referees

Álvaro E. Prieto Ramos
Amador Durán Toro
André L. Santos
Andrea Delgado
Ángel Herranz
Angélica Caro
Anna Grimán Padua
Antonio Jesús Roa Valverde
Antônio Oliveira Filho
Antonio Ruiz-Cortés
Arturo Zambrano
Carlos Bobed
Carlos D. Barranco González
Carlos Enrique Cuesta Quintero
Carlos Neil
Cecilia Delgado Negrete
César J. Acuña
Claudio Sant' Anna
Cristina Vicente Chicote
Daniel Rodríguez
Dante Carrizo
Diana Marcela Sánchez
Diego Alonso Cáceres
Diego Seco Naveiras
Domingo Savio Rodríguez Baena
Eduardo Rodríguez López
Elisa Yumi Nakagawa
Ellen Francine Barbosa
Encarna Sosa Sánchez
Fernando Molina Molina
Fran J. Ruiz Bertol
Francisco Javier Lucas Martínez
Francisco Luís Gutiérrez Vela
Francisco Martínez Álvarez
Ignacio García Rodríguez de Guzmán
Ismael Caballero
Ismael Navas Delgado
Ismael Sanz Blasco
Javier Pérez García
Joaquín Lasheras
Joaquín Nicolás
Jorge Gracia
Jorge Martínez Gil
José María Cavero Barca
Juan Ángel Pastor Franco
Juan M. Vara
Juan Manuel Pérez Martínez
Manuel Ángel Serrano Martín
Manuel Resinas
Márcio de Medeiros Ribeiro
Marcirio Chaves
Marcos López Sanz
Mari Carmen Otero
María Esperanza Manso Martínez
María Luisa Rodríguez Almendros
María Teresa Gómez López
María Visitación Hurtado Torres
Martin Solari
Miguel Ángel Laguna Serrano
Miguel Ángel Martínez
Miguel Rodríguez Luaces
M^a Ángeles Moraga de la Rubia
Nuno Cardoso
Orlando Avila-García
Oscar Dieste
Óscar Pedreira Fernández
Othmane Chniber
Pablo Inostroza
Pablo Trinidad
Paloma Cáceres García de Marina
Pedro Sánchez Palma
Raquel M. Crespo García
Raquel Trillo Lado
Roberto Almeida Bittencourt
Roberto Rodríguez Echeverría
Roberto Ruiz
Rui Lopes
Sascha Ossowski
Sergio Ilarri Artigas
Vicente Luque Centeno

Sponsors



Ayuntamiento de Gijón



GOBIERNO DEL
PRINCIPADO DE ASTURIAS



INTERSYSTEMS



Table of Contents¹

Keynote Address 1

The five W's (and one "H") of Security: Software Engineering of Secure Systems	1
<i>Bashar Nuseibeh</i>	

Aspects

Analysis of Modularity by an Aspect-Oriented Measurement Process.....	3
<i>José Conejero, Juan Hernández, Elena Jurado, Klaas Berg</i>	

Process Engineering

Automating the Software Process Management.....	15
<i>Javier Berrocal, José Manuel García, Juan Manuel Murillo</i>	

Software Product Lines

Generación Automática de Casos de Prueba en Líneas de Producto	27
<i>Pedro Mateo, Beatriz Lamanha, Macario Usaola</i>	
Gestión de la Variabilidad de los Requisitos de Seguridad en Líneas de Producto	39
<i>Daniel Mellado, Eduardo Fernandez-Medina, Mario Piattini</i>	

¹ The section headings below correspond to the conference program, but do not include all the presentations in each conference session (where short papers and dissemination papers on the same topic also were included). Thus, the sections here all contain fewer papers than the corresponding conference session; the short papers are listed separated in this volume, followed by a chapter with an overview of the dissemination papers.

Information Engineering

Clasificación de Imágenes en el Sistema Qatris Imanager Mediante Regresión Logística Bayesiana	51
<i>Inés Horrillo, Manuel Barrena</i>	
Efficient Retrieval of Ontology Fragments Using an Interval Labeling Écheme ...	63
<i>Victoria Romero, Rafael Llavori</i>	
Un Modelo para el Análisis y Explotación de Información Cognitiva en Repositorios Documentales	75
<i>Miguel A. Martínez-Prieto, Joaquín Adiego, Pablo de la Fuente</i>	
Un Sistema de Consulta sobre Documentos Transformados con LZCS.....	87
<i>Joaquín Adiego, Gonzalo Navarro, Pablo de la Fuente</i>	

Model Engineering

Análisis de Series Temporales Dirigido por Modelos Conceptuales sobre Datos Multidimensionales.....	99
<i>Jose Zubcoff, Jesús Pardillo, Juan Trujillo</i>	
Una Aproximación Dirigida por Modelos para el Desarrollo de Esquemas XML.....	111
<i>Verónica Bollati, Juan Vara, Belén Vela, Esperanza Marcos</i>	
Generación de Metadatos OLAP Dirigida por Modelos sobre Almacenes de Datos	123
<i>Juan Trujillo, Jesús Pardillo, Jose-Norberto Mazón</i>	

Formal Methods

Modelling Mash-up Resources	135
<i>Iván Pérez, Ángel Herranz, Susana Muñoz, Juan Moreno-Navarro</i>	
Optimizando el Funcionamiento del Algoritmo FOIL	147
<i>Pablo Palacios, José Arjona, José Álvarez, Iñaki Fernández de Viana</i>	
Towards the Correctness Verification of Business Processes Modelled with UML.....	159
<i>Luis Mendoza, Manuel Capel, Kawtar Akhlaki</i>	

Maintenance and Testing

Agil_MANTEMA: Una Metodología de Mantenimiento de Software para Pequeñas Organizaciones	171
<i>Francisco Pino, Francisco Ruiz, Jorge Triñanes, Félix García, Mario Piattini</i>	
Priorización del Valor de Artefactos Software Basada en la Frecuencia de Uso..	183
<i>Daniel Cabrero, Javier Garzas, Mario Piattini</i>	
Identificación de Fallos en Módulos Software	195
<i>José Riquelme, Roberto Ruiz, Daniel Rodríguez</i>	

Data Mining, Data Streaming and Datawarehouses

Hacia la Implementación Automática de Almacenes de Datos Seguros en Herramientas OLAP.....	205
<i>Carlos Blanco, Ignacio García-Rodríguez de Guzmán, Eduardo Fernández-Medina, Juan Trujillo, Mario Piattini</i>	
Una aproximación Basada en Diagramas de Actividades de UML para el Modelado Conceptual de Procesos ETL en Almacenes de Datos.....	217
<i>Lilia Muñoz, Jose-Norberto Mazón, Jesús Pardillo, Juan Trujillo</i>	
MeCADI*: un Marco Orientado a Objetivos para el Modelado de la Calidad en Almacenes de Datos.....	229
<i>Cristina Cachero, Jesús Pardillo, Jose-Norberto Mazón, Juan Trujillo</i>	

Reengineering and Software Modernization

Reverse Engineering of Object-Relational Database Schemas	241
<i>Jordi Cabot, Cristina Gómez, Elena Planas, M. Elena Rodríguez</i>	

Quality, Measurement & Estimation of Products & Processes

Una Metodología Basada en ISO/IEC 15939 para la Elaboración de Planes de Medición de Calidad de Datos.....	253
<i>Eugenio Verbo, Ismael Caballero, Ricardo Pérez, Coral Calero, Mario Piattini</i>	
Metodologías para Definir Programas de Medición en PyMEs: El Marco MIS-PyME.....	265
<i>María Díaz-Ley, Félix García, Mario Piattini</i>	

Visualización de la Usabilidad de Componentes Software.....	275
<i>M^a Ángeles Moraga, Sergio Susín, Virginia Arcos, Coral Calero</i>	
Aportaciones de una Visualización Metafórica al Análisis de Proyectos Software	287
<i>Amaia Aguirregoitia, J.Javier Dolado</i>	
Aplicación de las Técnicas de Modelado y Simulación en la Gestión de la Capacidad de los Servicios TI.....	299
<i>Elena Orta Cuevas, Mercedes Ruiz Carreira, Miguel Toro Bonilla</i>	
Measure Assessment for Heterogeneous XML Collections.....	311
<i>María Pérez Catalán, Ismael Sanz, Rafael Berlanga</i>	

Requirements Engineering

Revisiones Sistemáticas: Recomendaciones para un Proceso Adecuado a la Ingeniería del Software	321
<i>Oscar Dieste, Anna Grimán, Marta López</i>	
Metodologías Ágiles desde la Perspectiva de la Especificación de Requisitos Funcionales y No-Funcionales	333
<i>Pilar Rodríguez, Agustín Yagüe, Pedro Alarcón, Juan Garbajosa</i>	
Metamodelo y Perfil UML para el Modelado Orientado a Metas de Requisitos Medibles.....	345
<i>Fernando Molina, Cristina Cachero, Jesús Pardillo, Ambrosio Toval</i>	

Keynote Address 2

Model-Based Software Engineering: Expected and Unexpected Challenges.....	357
<i>Bran Selic</i>	

Short Papers

AAJ: Un Lenguaje de Descripción Arquitectónica Orientado a Aspectos.....	361
<i>María Boton, Amparo Navasa</i>	
An Ontology for IT Services	367
<i>Jorge Freitas, Anacleto Correia, Fernando Abreu</i>	

Construcción de Modelos Lógicos Multidimensionales Seguros para su Implementación en Herramientas OLAP Mediante MDA y QVT	373
<i>Carlos Blanco, Ignacio García-Rodríguez de Guzmán, Eduardo Fernández-Medina, Juan Trujillo, Mario Piattini</i>	
Desarrollo de Almacenes de Datos Espacio Temporales Dirigido por Modelos ..	379
<i>Octavio Glorio, Juan Trujillo</i>	
Generating Domain Specific Aspect Code for Navigation from Platform Specific Models in MWACSL.....	385
<i>Antonia M. Reina Quintero, Miguel Toro Bonilla, Jesús Torres Valderrama</i>	
Zentipede: Una Contribución a la Renovación de la Gestión del Proceso Software	391
<i>José Manuel García Alonso, José Javier Berrocal, Juan Manuel Murillo Rodríguez</i>	
Hacia la Definición de un Simulador para la Enseñanza de la Elicitación de Requisitos en el Contexto del Desarrollo Global del Software	417
<i>Miguel Romero, Aurora Vizcaino, Mario Piattini</i>	
Un Marco de Referencia para Comparar ESBs desde la Perspectiva de la Integración de Aplicaciones.....	403
<i>Rafael Corchuelo, Rafael Frantz, Jesús González</i>	
Refactorizaciones en la Migración del Software.....	409
<i>Rául Marticorena, Yania Crespo, Carlos López</i>	
Diseño Evolutivo de Bases de Datos XML	415
<i>Carlos Nilo, Cecilia Reyes, Jose Marti</i>	
Impacto de las Multiplicidades en la Resolución de Problemas de Sumarizabilidad para OLAP	421
<i>Jose-Norberto Mazón, Jens Lechtenbörger, Juan Trujillo</i>	

Workshops, Tutorials, Demos and Dissemination

Workshops.....	427
<i>João Araújo</i>	
Tutorials	429
<i>António Rito Silva</i>	

Tool Demonstrations	431
<i>Lidia Fuentes</i>	
ActiveRulesDBX – Ferramenta para Execução de Regras a partir da Detecção de Eventos Temporais.....	433
<i>Eugênio de O. Simonetto, Jéferson Kasper, Giovanni R. Librelotto</i>	
Deriving AO Software Architectures using the AO-ADL Tool Suite	437
<i>Mónica Pinto, Lidia Fuentes, Luis Fernández, Juan A. Valenzuela</i>	
ESFORA: a tool for the dEfinition of domain SPECific OpeRation languages.....	441
<i>David Musat, Jennifer Pérez, Pedro P. Alarcón, Agustín Yagüe</i>	
FAMA Framework	445
<i>Pablo Trinidad, David Benavides, Antonio Ruiz-Cortés, Sergio Segura</i>	
ProSÉ: A Protégé plugin for Reusing Ontologies, Safe and Économique	449
<i>Ernesto Jiménez-Ruiz, Bernardo Cuenca Grau, Ulrike Sattler Thomas Schneider, Rafael Berlanga</i>	
REMM-Studio+: Extensiones para Modelar Variabilidad y Permitir la Reutilización de Requisitos	453
<i>Begoña Moros, Cristina Vicente-Chicote, Ambrosio Toval</i>	
RUX-Tool: Una herramienta CASE para el modelado y la generación automática de Interfaces de Usuario para RIA	457
<i>Marino Linaje, Juan Carlos Preciado, Fernando Sánchez-Figueroa Rober Morales-Chaparro, David Gordillo, Fernando Sánchez-Herrera</i>	
StateML+: Diseño, Validación y Generación de Código Ada para Máquinas de Estado Jerárquicas	461
<i>Diego Alonso, Cristina Vicente-Chicote, Bárbara Álvarez</i>	
Relevant Papers Dissemination	465
<i>Antonio Vallecillo, João Falcão Cunha</i>	
Feature Oriented Model Driven Development: A Case Study for Portlets.....	467
<i>Salvador Trujillo, Don Batory, Oscar Díaz</i>	
DEX: High-Performance Exploration on Large Graphs for Information Retrieval.....	69
<i>Norbert Martínez-Bazan, Victor Muntés-Mulero, Sergio Gómez-Villamor, Jordi Nin, Mario-A. Sánchez-Martínez, Josep-L. Larriba-Pey</i>	
Determining Criteria for Selecting Software Components: Lessons Learned	471
<i>Juan Pablo Carvallo, Xavier Franch, Carme Quer</i>	

Engineering Rich Internet Application User Interfaces over Legacy Web Models	473
<i>Marino Linaje, Juan Carlos Preciado, Fernando Sánchez-Figueroa</i>	
Guideliness for Eliciting Usability Functionalities	475
<i>Natalia Juristo, Ana María Moreno, Maria-Isabel Sánchez-Segura</i>	
From Wrapping to Knowledge	477
<i>José Luis Arjona, Rafael Corchuelo, David Ruiz, Miguel Toro</i>	
Introducing Structure Management in Automatic Reference Resolution: An XML-based Approach	479
<i>M. Mercedes Martínez-González, Pablo de la Fuente</i>	
Run-time Composition and Adaptation of Mismatching Behavioural Transactions	481
<i>Javier Cámara, Gwen Salaün, Carlos Canal</i>	
Building Domain-Specific Languages for Model-Driven Development	483
<i>Jesús Sánchez Cuadrado, Jesús García Molina</i>	
Reconciling requirement-driven data warehouses with data sources via multidimensional normal forms.....	485
<i>Jose-Norberto Mazón, Juan Trujillo, Jens Lechtenbörger</i>	
Developing Secure Data Warehouses with a UML Extension.....	487
<i>Eduardo Fernández-Medina, Juan Trujillo, Rodolfo Villarroel, Mario Piattini</i>	
Author Index.....	489

Visualización de la usabilidad de componentes software

M^a Ángeles Moraga, Sergio Susín, Virginia Arcos, Coral Calero

Instituto de Sistemas y Tecnologías de la Información, Universidad de Castilla-La Mancha
Paseo de la Universidad 4 -13071 -Ciudad Real – España.

{MariaAngeles.Moraga, Sergio.Susin, Virginia.Arcos, Coral.Calero}@uclm.es

Resumen. En la actualidad el desarrollo de sistemas software comerciales utilizando tecnologías basadas en componentes está experimentando un amplio auge como consecuencia de la madurez que están adquiriendo dichas tecnologías. Ante esta situación han surgido diversas investigaciones orientadas hacia la calidad de los componentes software, las cuales proponen una gran cantidad de medidas cuyo resultado es, normalmente, un valor numérico. El objetivo de dichas medidas es obtener el nivel de calidad de un determinado componente. Sin embargo para los humanos, los conjuntos de datos que se obtienen como resultado de calcular los valores de las diferentes medidas, no son fácilmente entendibles y, por tanto, no pueden transformar los datos en información sobre la calidad del componente en estudio. Por ello y con el objetivo de facilitar la interpretación de estos datos, en este trabajo se propone el uso de la visualización como herramienta para facilitar la interpretación de dichos resultados. Gracias a la metáfora visual y a la utilización de elementos que son altamente familiares a los humanos, se puede interpretar de forma fácil y rápida las características de calidad de un componente. Como ejemplo, la metáfora se ha aplicado a dos componentes reales para conocer su nivel de usabilidad.

Palabras Clave: Calidad, Visualización, Componentes Software y Medidas.

1 Introducción

En nuestro trabajo existen tres temas principales interrelacionados: los componentes software, la calidad software y la visualización.

- Los componentes software son los elementos básicos del Diseño de Software Basado en Componentes (DSBC). El DSBC es una metodología de desarrollo propuesta por [17] que tiene como objetivo tratar de paliar la creciente complejidad de los sistemas software. Como se dijo anteriormente en el DSBC, los elementos básicos son los componentes software, los cuales pueden definirse como unidades de producción binarias independientes, adquisición y despliegue que interactúan para formar un sistema software [16]. Por tanto, desde la aparición de esta metodología, las compañías están desarrollando sus productos software mediante la adquisición e integración de componentes, en lugar de crear el producto software desde cero.

En este nuevo marco de construcción de software, el proceso principal es el de adquisición de componentes, consistente en la selección de los componentes, teniendo en cuenta que satisfagan una funcionalidad concreta. Por tanto, la mayor parte de los esfuerzos de la Comunidad del Software se centran en los aspectos funcionales del DSBC, pero esto no es suficiente en el proceso de selección, ya que es necesario tener también en cuenta los aspectos no funcionales relacionados con la calidad con el objeto de obtener no sólo aquel componente que realice la funcionalidad requerida sino el que la haga mejor (el que sea de más calidad).

- La calidad se está convirtiendo en un aspecto esencial en el desarrollo del software actual. La demanda de calidad en el software incrementa día a día, ya que la sociedad tiene una gran dependencia del software y los efectos que los errores causan pueden ser devastadores, llegando a involucrar aspectos relacionados con pérdidas económicas, con retrasos temporales, etc [12]. Sin embargo, determinar cual es el nivel de calidad de un producto software no es una tarea trivial, por lo que requiere la definición de un modelo de calidad que incluya características de calidad específicas del software que se está definiendo y midiendo [1].
- La visualización es una técnica semi-automatizada que combina el pre-procesamiento de datos y la representación de los datos con la capacidad de los humanos de analizar datos de forma gráfica. De tal forma que proporciona una herramienta muy útil para representar de forma fácil grandes volúmenes de información, a la vez que se mejora la comprensión de la arquitectura de grandes aplicaciones. Además, es posible detectar intuitivamente posibles fallos futuros en los componentes software.

Bajo nuestro punto de vista, las técnicas de visualización pueden utilizarse para representar algunas medidas o parámetros de calidad de los componentes software, y así determinar de forma sencilla cual es el nivel de calidad de los mismos. De hecho, la utilización de técnicas de visualización en los componentes software nos permite entender de forma fácil y rápida las características de calidad del componente, siempre y cuando la metáfora visual utilizada sea adecuada y esté bien definida. Teniendo todo esto en cuenta, en este trabajo se propone una metáfora visual adecuada para visualizar correctamente la calidad de componentes software.

Este artículo se estructura como sigue. En la Sección 2, se presentan las principales propuestas relacionadas con la visualización y la calidad de componentes software. En la Sección 3, se propone una metáfora visual para representar la calidad de componentes software, mientras que en la Sección 4 la metáfora se adapta para representar las sub-características de la usabilidad. En la Sección 5, a modo de ejemplo del uso de la metáfora visual, se muestra la visualización de la usabilidad de dos componentes reales. Finalmente, en la Sección 6 se presentan las conclusiones y el trabajo futuro.

2 Propuestas Relacionadas

Existen muchas propuestas relacionadas con la calidad de componentes software y la visualización. En esta sección se presentan brevemente los principales estudios en calidad de componentes software y, a continuación, se exponen los diferentes estudios sobre visualización.

2.1 Propuestas Relacionadas con Calidad de Componentes Software

En la literatura se pueden encontrar diferentes estudios orientados hacia el proceso de adquisición y selección de componentes software, los cuales utilizan los aspectos funcionales como elemento discriminatorio. Sin embargo, esto no es suficiente, ya que para realizar de forma correcta el proceso de adquisición y selección de un componente se han de tener en cuenta no sólo los aspectos funcionales sino los no funcionales, es decir, aquellos relacionados con la calidad. Por tanto para llevar a cabo el proceso de adquisición y selección, en primer lugar, se utilizarán los requisitos no funcionales obteniendo así un subconjunto de componentes que cumplan ciertos requisitos funcionales, los cuales formarán el conjunto de los componentes candidatos. A continuación, de entre los componentes candidatos se seleccionará el más adecuado en función de sus aspectos no funcionales, tales como la mantenibilidad, usabilidad, eficiencia etc. Por tanto, algunas investigaciones se están centrando en determinar cuales son los factores que contribuyen a la calidad de un componente. Algunos ejemplos de propuestas en esta dirección son:

- En [4], los autores proponen un modelo de calidad para la selección de Sistemas de Planificación de Recursos de Empresas (ERP). El modelo propuesto es una adaptación del modelo definido en el estándar ISO/IEC 9126.
- En [15], los autores identifican las características y sub-características de calidad más relevantes para componentes software. En este caso, el modelo de calidad propuesto no sólo se basa en la ISO/IEC 9126, sino que tiene en cuenta otros modelos existentes.
- En [9], los autores proponen una metodología para construir modelos de calidad estructurados basados en la ISO/IEC 9126. Además, los autores aplican la metodología al contexto de los servidores de correos electrónicos.

Una vez definidas las características de calidad será necesario acompañar a las mismas de un conjunto de medidas que, aplicadas a un componente, determinen su nivel de calidad. A pesar de que los investigadores han propuesto medidas de calidad específicas para componentes software, a la hora de utilizarlas nos encontramos con diferentes problemas que dificultan su uso. Por ejemplo, de acuerdo con [14] las medidas deben estar bien definidas, para ello deben proporcionar el nombre, la descripción de la medida, el tipo de medida, la forma de medir, la escala y la unidad de medición. Sin embargo, muchas de las medidas propuestas no proporcionan toda esta información por lo que no están bien definidas, además son incompletas, utilizan vocabulario y terminología no inconsistentes e incluso existen conflictos entre las diferentes propuestas existentes. Otro problema añadido es la dificultad en automatizar el cálculo de las medidas, lo que impide la creación de herramientas que midan automáticamente la calidad de un componente.

Como excepción, en [2] el autor presenta un conjunto de medidas para la característica de usabilidad que se encuentran bien definidas y pueden calcularse automáticamente. Además, el autor adapta la definición de las sub-características de usabilidad propuestas en la ISO/IEC 9126 [11] al contexto de los componentes software:

- **Comprensibilidad:** Capacidad del componente que permite al desarrollador del sistema comprender si el componente es adecuado y como utilizarlo en unas tareas y condiciones específicas.
- **Aprendibilidad:** Capacidad del componente que permite al desarrollador del sistema aprender a utilizarlo. Una medida de aprendibilidad tiene que ser capaz de evaluar el esfuerzo necesario de los desarrolladores de sistemas para aprender a utilizar las funciones específicas (interfaces, métodos, etc.) o para evaluar la efectividad de la documentación proporcionada.
- **Operabilidad:** Capacidad del componente que permite al desarrollador del sistema trabajar con él y controlarlo.

Con el objetivo de calcular estas tres sub-características de usabilidad, el autor define un conjunto de medidas que cumplen los siguientes requisitos: bien definidas, objetivas, reproducibles y fácilmente automatizables:

- **Número de Archivos HTML:** El número de archivos HTML incluidos en el manual del componente software.
 - **Número de Elementos Funcionales:** Es la agregación de tres tipos de elementos funcionales, las interfaces, los métodos y los parámetros configurables.
 - **Número de Palabras en los Manuales:** Calcula el tamaño del manual utilizando el número de palabras que contiene.
 - **Número de Métodos:** Los métodos ofrecidos por el componente software.
 - **Número de Parámetros Configurables:** Los parámetros configurables ofrecidos por el componente software.
 - **Número de Valores de Retorno:** El número de métodos del componente software que devuelven algún valor.
 - **Número de Argumentos:** El número total de argumentos que aparece en todos los métodos del componente software.
-

Basándose en estas medidas base (aquellas que pueden medirse directamente sobre el software), el autor define las siguientes medidas derivadas (aquellas que utilizan otras medidas en su definición):

- **Ratio de Archivos HTML por Elemento Funcional (FpEF):** Compara el tamaño de los manuales, medido en páginas HTML, con el tamaño funcional del componente, medido en Elementos Funcionales (EF), evaluando el tamaño relativo del manual.
- **Ratio de Valores de Retorno por Método (VRpO):** Calcula el número de métodos que devuelve algún valor con el número total de métodos.
- **Kilo-Palabras por Método (PpO):** Compara el tamaño del manual con el número de métodos.
- **Ratio de Argumentos por Métodos (ApO):** Calcula el ratio entre el total de argumentos en los métodos y el número de métodos. Un elevado valor de esta medida indica una elevada complejidad de los métodos ofrecidos por el componente software.
- **Kilo-Palabras por Parámetro Configurable (PPpC):** Compara el tamaño del manual con el número de parámetros configurables.

Cabe destacar que un *Parámetro Configurable* es todo elemento de un componente que permite su parametrización (esto incluye desde atributos públicos proporcionados en los objetos Java hasta las variables incluidas en los archivos de configuración de algunas aplicaciones); y un *Elemento Funcional* es el conjunto de interfaces, métodos, parámetros configurables y eventos que un componente es capaz de soportar o requerir de otros componentes para conseguir su funcionalidad, esto es, para implementar sus servicios.

La correspondencia realizada por el autor entre cada sub-característica y la medida definida para ella se muestra en la Tabla 1. Además, se especifica la relación entre las diferentes medidas [2].

Tabla 1. Definición de las Medidas Utilizadas.

Carac-terística	Sub-Característica Calculada	Medidas Derivadas		Relación
Usabilidad	Comprensibilidad (CC)	Ratio de Archivos HTML por Elemento Funcional (FpEF)	Ratio de Valores de Retorno por Método (VRpO)	$CC = 0.200 \times FpEF - 1.423 \times VRpO + 1.598$
	Aprendibilidad (AC)	Kilo-Palabras por Método (PpO)	Ratio de Argumentos por Método (ApO)	$AC = 1.789 \times PpO - 2.090 \times ApO + 1.712$
	Operabilidad (OC)	Kilo-Palabras por Parámetro Configurable (PPpC)	Ratio de Valores de Retorno por Método (VRpO)	$OC = 0.804 \times PpPC - 8.265 \times VRpO + 3.486$

2.2 Propuestas Relacionadas con la Visualización

Como se indicó anteriormente las metáforas visuales facilitan la interpretación de grandes volúmenes de datos. Aunque en la literatura se pueden encontrar diversas definiciones de metáfora visual, nosotros adoptamos la definición dada en [8] dónde una metáfora visual se define como el mecanismo utilizado para representar gráficamente la información, debido a la no inherente naturaleza física de la misma.

Por tanto, lo más importante en el área de la visualización es la metáfora de visualización utilizada para representar la información, porque es la clave para comprender los datos que se están representando.

Las ocho formas de visualización más utilizadas son: los gráficos de barras, las vistas matriciales, los paisajes, las vistas de red, los gráficos de puntos, los histogramas, las hojas de datos y las tablas de tiempo [7]:

- **Gráfico de Barras:** Los gráficos de barras son colecciones de barras verticales ordenadas en una ventana. Con esta metáfora de visualización pueden representarse únicamente dos atributos, codificados en la altura y el color de las barras. Aunque, las barras pueden ser agrupadas o apiladas para aumentar el número de atributos representables. Esta visualización es básica en cualquier aplicación que tenga como funcionalidad el manejo de datos numéricos.
- **Vistas Matriciales:** Esta metáfora de visualización muestra una o más características como función de dos índices categóricos o numéricos. La vista que obtiene es una cuadrícula bidimensional con filas que se corresponden con uno de los índices y columnas que se corresponden con el otro. La celda (i, j) contiene la descripción de uno o más atributos (normalmente numéricos) para un valor de fila i y de columna j. Dichos atributos se codifican como características visuales del gráfico, tales como el color, la textura, la forma y el tamaño.
- **Paisajes:** Esta metáfora de visualización es la versión tridimensional de las vistas matriciales. Los paisajes muestran tablas de datos en dos dimensiones usando como símbolo representativo un rascacielos, esto son torres ordenadas en una cuadrícula. Normalmente, un paisaje es observado desde un ángulo (si se mira desde arriba parece una vista matricial). La altura, el color y la forma de las torres codifican tres atributos de los datos. La codificación depende de la naturaleza de los atributos.
- **Vista de Red:** La metáfora de la red permite visualizar las características de elementos de datos individuales, así como las relaciones entre ellos. Los nodos se corresponden con los datos individuales cuyos atributos se convierten en características visuales tales como el tamaño, el color y la forma. Las relaciones entre los nodos se codifican como características visuales de los arcos (anchura, color, estampado...). Las vistas de redes son eficientes para representar gráficos con entre decenas y miles de nodos, con una fuerte dependencia de la conectividad, en el número de enlaces, y que tienen una inherente estructura de grafo.
- **Gráficos de Puntos:** Esta metáfora de visualización puede mostrar más de 100.000 puntos, dependiendo del patrón de los datos.
- **Histogramas:** Esta metáfora de visualización consiste en mostrar gráficos de barras suavizados en los que el índice de barra es ordinal o numérico. El valor de suavizado para cada píxel de una tabla de datos depende del tamaño de la ventana de visualización, dada por la función de núcleo utilizada para calcular el suavizado y los colores.
- **Hojas de Datos:** Esta metáfora de visualización consiste en visualizar texto [5, 6] con desplazamiento que permiten acceso directo a elementos de datos individuales. Una hoja de datos es simplemente una representación visual de texto organizado en varias columnas. Puede mostrar cientos de miles de filas y decenas de columnas.
- **Tablas de Tiempo:** Esta metáfora de visualización muestra, por unidad de tiempo, eventos categorizados, y puede visualizar cientos de miles de marcas organizadas en cientos de categorías.

En la Tabla 2 se resumen las principales características de estas metáforas de visualización.

Aunque existen algunas herramientas de visualización software, una de las más importante es la presentada en [13], cuya finalidad es visualizar sistemas orientados a objetos, en la que cada clase se representa con una caja tridimensional.

Tabla 2. Tabla comparativa de las diferentes metáforas de visualización.

Metáfora	Nº Atributos	Tamaño mínimo de los elementos	Resolución Monitor	Ventajas	Inconvenientes
Gráfico de Barras	2 (ancho y alto)	1 píxel/barra 1 píxel/separación	1024x768 (500 barras)	Sencillez.	Pocos atributos. Con muchos datos se hace poco legible.
Vista Matricial	1-4 (color, textura, tamaño y forma)	10x10 píxeles/celda	1280x1024 (13.000 celdas)	Permite visualizar gran cantidad de datos (13.000).	Difícil ajuste de la información a visualizar a la resolución ratio del monitor.
Paisajes	3 (alto, ancho y color)	Varios cientos de píxeles/elemento tridimensional.	1280x1024	Permite visualizar de cientos a miles de datos (13.000).	Oclusión barras pequeñas en la parte trasera. Difícil ajuste de la información a visualizar a la resolución ratio del monitor.
Vista de red	3 (tamaño y color del nodo, y anchura de enlace)	Dependiente de los datos. La anchura mínima para representar un arco es de 1 píxel.	800x600	Permite visualizar características de los datos y relaciones (con sus características) entre ellos.	Confusa visualización debido a la superposición de arcos.
Gráficos de Puntos	3 (color, abscisas y ordenadas)	3 píxeles de diámetro.	800x600	Permite representar la evolución de los datos. Predicción de tendencias.	Confusa visualización debido a la superposición de los puntos.
Histogramas	3 (color suavizado, abscisas y ordenadas)	1 píxel/punto.	800x600	Permite representar la evolución de los datos. Predicción de tendencias.	Complejidad de cálculo de la función de suavizado. Confusa visualización debido a la superposición de los puntos.
Hojas de datos	Decenas.	Mínimo legible dependiente de las características del texto.	800x600	Permite visualizar cientos de miles de datos (filas) y la ordenación de los mismos.	No es una visualización gráfica sino de texto.
Paraboxes	4 (forma, color de la forma, tipo de línea y color de línea)	Dependiente de los datos.	800x600	Permite visualizar cientos de miles de líneas y de diez a cien filas.	Confusa visualización debido a la superposición de líneas.
Tablas de tiempo	Desconocido	1 píxel/marca.	800x600	Permite visualizar cientos de miles de marcas organizadas en cientos de categorías.	Confusa visualización debido a la superposición de puntos. Sólo aplicable a muestreos temporales de datos.

3 Nuestra Propuesta

En esta sección, presentamos una metáfora de visualización genérica para visualizar la calidad de componentes software. Una vez analizadas y comparadas todas las metáforas presentadas en la sección 2.1, hemos decidido que nuestra metáfora se base en la metáfora de los paisajes ya que:

- Permite representar tres atributos, pudiéndose ampliar dependiendo de la necesidad del usuario.
- Permite representar gran cantidad de datos (13.000) con una buena legibilidad.
- El tamaño mínimo de los elementos es suficientemente grande como para que el usuario lo comprenda con facilidad.
- El factor limitante se puede evitar cambiando el punto de vista del usuario.
- La resolución de pantalla mínima permitida (1280x1024) es mayor que la normal (1024x768).
- El tiempo de representación depende de la cantidad de información mostrada por pantalla.

Así pues, en nuestra metáfora, la calidad de componentes software es representada mediante un edificio (un ortoedro o cubo rectangular) y cada característica de calidad tiene su correspondencia con alguno de los atributos de los edificios. De esta forma, la metáfora es capaz de codificar de una a siete características de calidad (aunque puede ampliarse para codificar más aún). El hecho de limitar el número de características a siete es porque, por un lado nos beneficiamos de la percepción visual de los humanos y, por otro, porque la mayoría de los modelos de calidad identifican un máximo de siete características. Una vez que tenemos claro el número de características que pretendemos representar, se establece la correspondencia entre estas características y los atributos de los edificios. En concreto, los atributos de los edificios que se van a utilizar (ver figura 1) son: altura, anchura y profundidad (mayor o menor); color (de blanco a negro); orientación (de 0° a 90°); densidad de la textura (ladrillos de mayor o menor tamaño) y la antena (mayor o menor). Obviamente, en lugar de representar mediante esta metáfora el nivel de cada una de las características de calidad que afectan a los componentes software es posible disminuir el nivel de granularidad y representar las sub-características de una determinada característica. De esta forma, se podría representar en un primer momento los niveles de calidad de las características y a continuación, en función de los resultados obtenidos el usuario podría centrarse en una de ellas con el objetivo de visualizar los resultados de cada una de sus sub-características.

Por tanto, hay dos granularidades diferentes: una de grano grueso en la que se visualizan las características de calidad y otra de grano fino en la que se representan las sub-características pertenecientes a una determinada característica. En ambos casos se utilizarán los mismos atributos del edificio aunque estos representarán medidas diferentes para cada caso.

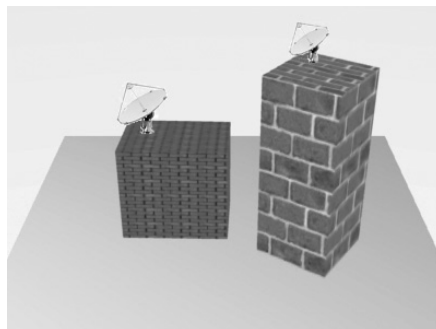


Fig. 1. Ejemplo de nuestra metáfora ampliada.

4 Adaptando la Metáfora a la Usabilidad de Componentes

Una vez definida la metáfora de visualización, en esta sección será adaptada para representar las sub-características de usabilidad de un componente software. Las medidas utilizadas para calcular las sub-características de usabilidad del componente son las presentadas en la sección 2.2. El siguiente paso es establecer la relación ente las sub-características de usabilidad y los atributos de los elementos de la metáfora de visualización. Es necesario destacar que en este caso en concreto sólo se van a utilizar tres atributos: ángulo, altura y color, para codificar las sub-características: comprensibilidad, aprendibilidad y operabilidad respectivamente:

- **Ángulo - Comprensibilidad:** El ángulo con el que aparece el cubo varía entre 0° y 90° . Con esta característica representamos la comprensibilidad, por lo que, un componente muy comprensible se representará con un ángulo de 0° . Por el contrario, si es un componente fácilmente comprensible, se visualizará con un ángulo de 90° . Esta correspondencia se ha establecido así ya que los humanos percibimos las cosas que no están alineadas como más complejas que las que sí lo están.
- **Altura - Aprendibilidad:** El cubo crece en una dimensión, la altura, que varía entre 0.5 y 10. Un componente software fácil de aprender será un edificio pequeño, un componente software difícil de aprender será un rascacielos. Asociamos la altura con la aprendibilidad porque los humanos identifican los objetos grandes como difíciles.
- **Color - Operabilidad:** El color del edificio varía entre gris y negro. Un componente software con el que es difícil trabajar se representará con un color negro, mientras que un componente fácil se representará con un color blanco. Asociamos el color con la operabilidad porque los humanos percibimos los objetos de color negro como algo malo.

Además, y con el objetivo de obtener una representación coherente, se van a normalizar los resultados de las medidas. En [2] el autor utiliza las funciones de la tabla 1 para calcular los valores de las medidas y define tres categorías para clasificar los niveles de calidad de las sub-características de usabilidad, tomando como referencia lo dispuesto por la norma IEEE 1061 [10]. Estas funciones se correlacionan con los valores asignados experimentalmente a las sub-características, que se calculan ente -3 y +3, donde un valor negativo significa un nivel inaceptable y un valor positivo indica un valor aceptable. Como consecuencia, los valores obtenidos de las sub-características deben encontrarse entorno al valor de +1 para ser aceptable. La tabla 3 muestra los valores límites para cada sub-característica de usabilidad de un componente software.

Tabla 3. Valores límites de las sub-características de usabilidad.

Sub-Característica	Aceptable	Marginal	Inaceptable
Comprensibilidad (CC)	[3-0.95]	(0.95-0.75)	[0.75-(-3)]
Aprendibilidad (AC)	[3-1.00]	(1.00-0.80)	[0.80-(-3)]
Operabilidad (OC)	[3-0.90]	(0.90-0.40)	[0.40-(-3)]

Para obtener un valor final de usabilidad proponemos una función que combina las tres sub-características (ver figura 2). En esta primera aproximación, consideramos que todas las sub-características de usabilidad tienen la misma importancia, aunque el usuario puede cambiar los pesos de acuerdo con sus necesidades o prioridades.

$$F = \frac{\sum \text{Nivel_de_las_Sub_Características}}{3}$$

Nivel de las subcaracterísticas: $\begin{cases} 1 \rightarrow \text{Nivel Inaceptable} \\ 2 \rightarrow \text{Nivel Marginal} \\ 3 \rightarrow \text{Nivel Aceptable} \end{cases}$

Fig. 2 Función de Usabilidad.

5 Aplicando la Metáfora a dos Componentes Software Reales

En esta sección, vamos a utilizar la metáfora para representar la usabilidad de dos componentes software reales. Para ello, se ha desarrollado una herramienta denominada VISUCO que calcula automáticamente los valores de las medidas de un componente y las representa visualmente haciendo uso de la metáfora de visualización.

La herramienta toma como entrada la ruta en la que se encuentra el código fuente del componente y recorre la estructura de directorios en busca de los archivos de código correspondientes. A continuación analiza los archivos encontrados para calcular las medidas necesarias y con estos valores calcula el nivel de calidad de cada una de las sub-características de usabilidad. Finalmente, visualiza los resultados. De esta forma, el usuario puede saber con exactitud, con un simple golpe de vista, el nivel de calidad de las sub-características de la usabilidad así como el nivel de usabilidad global que tiene el componente. Cabe destacar que el usuario de la herramienta será tanto el desarrollador de componentes, el cual podrá utilizarla para saber el nivel de usabilidad de los componentes que está desarrollando, como el desarrollador de sistemas basados en componentes. En este último caso la herramienta se utilizará para seleccionar el componente que presente mejor usabilidad de entre un conjunto de componentes con la misma funcionalidad.

Como se indicó al comienzo de esta sección, la herramienta se ha utilizado para evaluar el nivel de usabilidad de dos componentes reales, aunque por motivos de confidencialidad, la información relativa al componente y a la compañía que los ha suministrado no puede ser revelada.

En la tabla 4 y en la tabla 5 se presentan los valores obtenidos por los componentes para cada una de las sub-características de usabilidad. Además, en estas tablas se indican los atributos de la metáfora de visualización con los que se corresponde cada sub-característica, junto con el valor de la conversión del valor de la medida.

Tabla 4. Valores de las sub-características de usabilidad del primer componente software real.

Sub-Característica	Valor	Atributo	Valor Normalizado
Comprensibilidad	0.23961231	Ángulo [0-90°]	41.405820
Aprendibilidad	-2.19487328	Altura [0.5-10]	8.658122
Operabilidad	-3.64440000	Color [RGB=0-255, 0-255, 0-255]	RGB=[255, 0, 0]

Tabla 5. Valores de las sub-características de usabilidad del segundo componente software real.

Sub-Característica	Valor	Atributo	Valor Normalizado
Comprensibilidad	0.61697225	Ángulo [0-90°]	35.745416
Aprendibilidad	-1.25592440	Altura [0.5-10]	7.093207
Operabilidad	-2.18870186	Color [RGB=0-255, 0-255, 0-255]	RGB=[221, 0, 34]

A continuación, los resultados se visualizan haciendo uso de la metáfora visual propuesta. Para ello, los valores previos se convierten a atributos visuales de la metáfora, mediante el uso de los valores normalizados que se muestran en las tablas 4 y 5 (última columna). En la figura 3 y en la figura 4 se muestra la representación visual del primer y segundo componente software, respectivamente. La figura 3 muestra la pantalla real de la herramienta acompañada de la leyenda que mejora la entendibilidad de la metáfora. Por restricciones de espacio, la figura 4 presenta sólo la visualización en lugar de la pantalla completa.

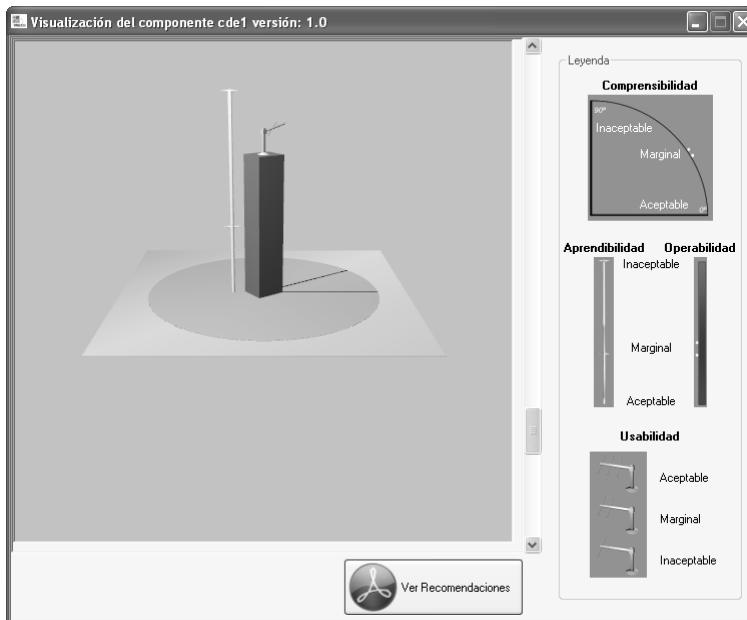


Fig. 3. Visualización del primer componente software real.

Como se puede observar en la figura 3, el primer componente software que ha sido analizado presenta unos valores de usabilidad inaceptables. La representación muestra un edificio muy elevado y de color negro, lo que se corresponde con valores muy malos de las sub-características aprendibilidad y operabilidad respectivamente. A su vez, el ángulo tiene un valor medio, por lo que la comprensibilidad no es del todo mala.

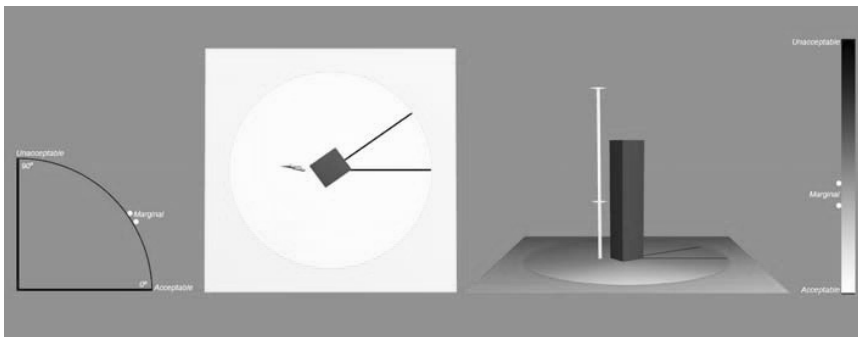


Fig. 4. Visualización del segundo componente software real.

De la misma forma, el segundo componente software real analizado (figura 4) tiene un valor de usabilidad inaceptable, pero no tan extremo con el caso anterior. La representación muestra un edificio de tamaño medio y de color gris, lo que se corresponde con valores no extremadamente malos de las sub-características de aprendibilidad y operabilidad respectivamente. A su vez, el ángulo tiene un valor medio-bajo, por lo que la comprensibilidad puede considerarse como buena.

Obviamente, los resultados coinciden con los que se obtienen aplicando la función presentada en la sección anterior. Por ejemplo, para el primer componente:

$$F(\text{Usabilidad}) = \frac{\text{nivel_comprensibilidad} + \text{nivel_aprendibilidad} + \text{nivel_operabilidad}}{3} = \frac{\text{nivel_inacceptable} + \text{nivel_inacceptable} + \text{nivel_inacceptable}}{3} = \frac{1+1+1}{3} = 1 \Rightarrow \text{Nivel_Inacceptable}$$

Mientras que para el segundo componente:

$$F(\text{Usabilidad}) = \frac{\text{nivel_comprensibilidad} + \text{nivel_aprendibilidad} + \text{nivel_operabilidad}}{3} = \frac{\text{nivel_inacceptable} + \text{nivel_inacceptable} + \text{nivel_inacceptable}}{3} = \frac{1+1+1}{3} = 1 \Rightarrow \text{Nivel_Inacceptable}$$

Sin embargo, utilizando la metáfora visual hemos obtenido la misma información sin la necesidad de aplicar la función sino, tan solo, “echando un vistazo” a la representación de las medidas.

Además, mediante la representación gráfica es muy sencillo detectar dónde se encuentran los puntos débiles del componente que se está representando. Para completar esto, la herramienta tiene la capacidad de proponer, en el caso de que el usuario quiera, un conjunto de acciones correctivas para mejorar los niveles de usabilidad. Estas acciones dan guías para obtener mejores valores en las medidas que se utilizan para determinar el nivel de usabilidad de un componente.

Por ejemplo, las acciones que la herramienta propone para mejorar la usabilidad del primer componente son las siguientes: en cuanto a la aprendibilidad, mejorar la calidad de los manuales ampliando las explicaciones de los métodos que se incluyen en el componente. O referente a la operabilidad, habría que (1) mejorar la calidad de los manuales explicando extensamente las opciones de configuración del componente, (2) aumentar las opciones de configuración y (3) reducir el número de métodos que devuelven algún valor. Finalmente, la comprensibilidad puede mejorarse extendiendo los manuales y decrementando el número de métodos con valores de retorno. Finalmente, es importante enfatizar que los resultados obtenidos se almacenan en una base de datos con la finalidad de poder comparar los valores almacenados con los nuevos valores obtenidos una vez llevadas a cabo las acciones correctivas, y así poder observar la evolución de la usabilidad del componente.

6 Conclusiones y Trabajo Futuro

En este artículo, se ha propuesto una metáfora de visualización con el objetivo de representar las características o sub-características de calidad de los componentes software. Esta metáfora de visualización utiliza elementos tan familiares para los humanos como son los edificios, y algunos de sus atributos (altura, color, giro, anchura, elementos comunes, etc.), consiguiendo así beneficiarse de la percepción visual humana y de la facilidad que éstos tienen para interpretar estos conceptos. Una vez definida la metáfora de visualización capaz de representar características o sub-características de calidad (en función de la granularidad que desee el usuario), ésta se ha adaptado para representar las sub-características de usabilidad, calculadas mediante las medidas definidas en [3].

Finalmente, la metáfora de visualización se ha aplicado a dos componentes software reales, para lo cual se ha desarrollado una herramienta que dado un componente calcula de forma automática las medidas y muestra visualmente – utilizando la metáfora de visualización – el nivel de calidad de cada una de las sub-características de usabilidad. Además la herramienta propone un conjunto de acciones correctivas orientadas a mejorar el nivel de usabilidad del componente. Dado que la herramienta almacena los resultados obtenidos en una base de datos, una vez que estas acciones hayan sido llevadas a cabo se podrá volver a examinar al componente y visualizar la evolución que este ha experimentado, en cuanto a su usabilidad.

Como trabajo futuro se propone aumentar las funcionalidades de la herramienta para que sea capaz de visualizar el nivel de calidad del resto de características de un componente software.

Además, la metáfora de visualización propuesta puede extenderse para permitir la visualización de la calidad no sólo de un componente sino de un sistema basado en componentes software, el cual estará formado por un conjunto de componentes.

Agradecimientos

Este trabajo es parte del proyecto INCOME (PET2006-0682-01) y de la red CALIPSO (TIN2005-24055-E), apoyado por el Ministerio Español de Educación y Ciencia y por el proyecto IVISCUS (PAC08-0024-5991) apoyado por la Consejería de Educación y Ciencia (JCCM).

Referencias

- [1]. April, A. and Laporte, C.Y., An Overview of Software Quality Concepts and Management Issues, in *Measuring Information Systems Delivery Quality*, E.W. Duggan and H. Reichgelt, (Eds). 2006, Idea Group: USA. pp. 28-54
- [2]. Bertoa, M., Medidas de Usabilidad para Componentes Software, in *Lenguajes y Sistemas Informáticos*. Universidad de Málaga: Málaga (2006).
- [3]. 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)
- [4]. 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: 225-245* (2003)
- [5]. Eick, S.G., Graphically displaying text. *Journal Computational and Graphical Statistics*. 3(2): 127-142 (1994)
- [6]. Eick, S.G., Graves, T.L., Karr, A. F., Mockus, A.,. Web-based text visualization. In: *SoftStat '97 Advances in Statistical Software 6*, F.F. Bandilla W., Editor pp. 3-10. (1997)
- [7]. Eick, S.G., Karr, A.F., Visual Scalability. National Institute of Statistical Sciences. (2000)
- [8]. Eick, S.G., Schuster, P., Mockus, A., and Graves, T.L., Karr, A. F., Visualizing Software Changes. National Institute of Statistical Sciences. (2000)
- [9]. Franch, X. and Carvallo, J.P., Using Quality Models in Software Package Selection. *IEEE Software*. 20(1): 34-41 (2003)
- [10]. IEEE, Std. 1061-1998. IEEE Standard for Software Quality Metrics Methodology. IEEE. (1998)
- [11]. ISO/IEC 9126, Software Engineering - Product Quality. Part 1 (2001)
- [12]. Khan, R.A., Mustafa, K., and Ahson, S.I., *Software Quality. Concepts and Practices*. Alpha Science. Oxford, U.K. (2006).
- [13]. Langelier, G., Saharaoui, H., Poulin, P., Visualization-based analysis of quality for large-scale software systems. Université de Montréal: Montreal. 214-223. (2005,)
- [14]. McGarry, J., Card, D., Jones, C., Layman, B., Clark, E., Dean, J., and Hall, F., *Practical Software Measurement*. Addison-Wesley. (2002).
- [15]. Simão, R.P. and Belchior, A. Quality Characteristics for Software Components: Hierarchy and Quality Guides. In: *Component-Based Software Quality* pp. 184-206. (2003)
- [16]. Szyperski, C., *Component Software. Beyond Object-Oriented Programming*. 1st edition ed Addison-Wesley. Boston. (1998).
- [17]. Szyperski, C., *Component Software. Beyond Object-Oriented Programming*. 2nd edition ed Addison-Wesley and ACM Press. Boston. (2002).

