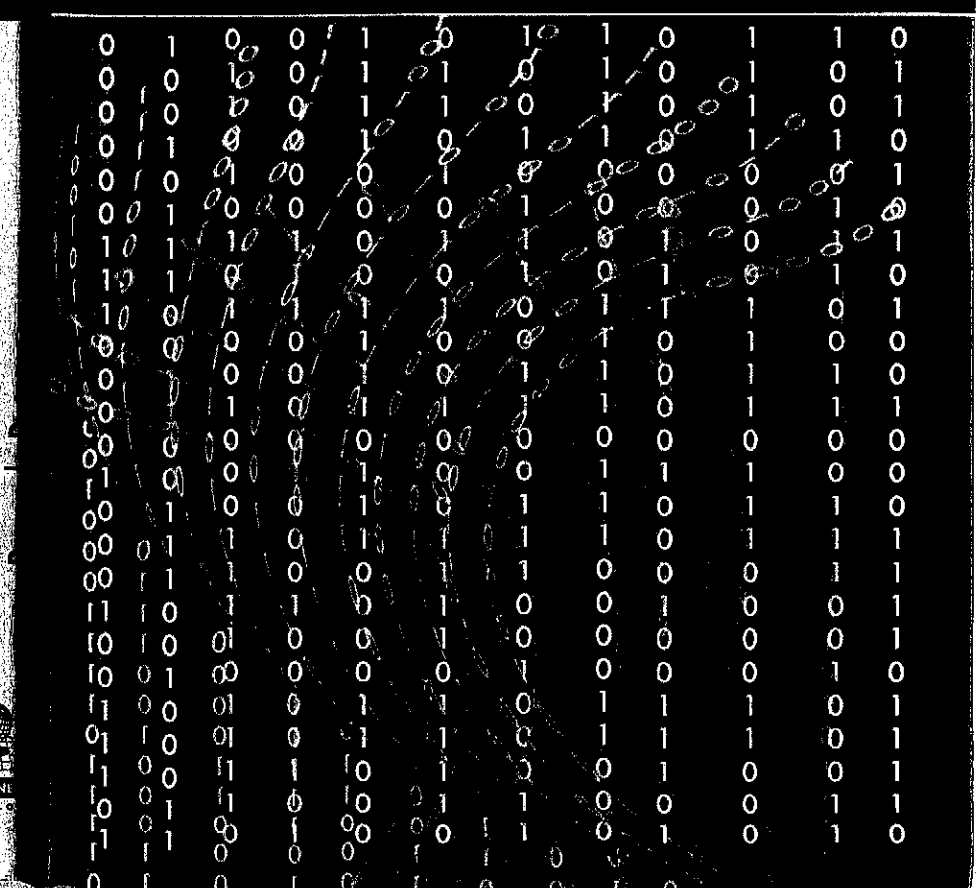
JISBD
2006
5008

Editores:
José C. Riquelme, Pere Botella

Ingeniería del Software y Bases de Datos

Editores:
José C. Riquelme, Pere Botella



UNIVERSITAT POLITÈCNICA DE CATALUNYA

FIB

gencat
www.gencat.net

INTERSYSTEMS

MINISTERIO DE EDUCACIÓN Y CIENCIA

Microsoft

CIMNE
BARCELONA

Ingeniería del Software

Ingeniería del Software y Bases de Datos

Actas de las
XI Jornadas de Ingeniería
del Software y Bases de Datos

Sitges, 3 al 6 de Octubre de 2006

Editores:

José C. Riquelme
Pere Botella

Publicado por

CIMNE
BARCELONA

Ingeniería del Software y Bases de Datos

Sitges, 3 al 6 de Octubre de 2006

Comité Ejecutivo JISBD 2006

Presidente del Comité Organizador
Pere Botella (Universitat Politècnica Catalunya)

Presidente del Comité De Programa
José C. Riquelme (Universidad de Sevilla)

Secretario Comisión Permanente
Mario Piattini (Universidad de Castilla-La Mancha)

Coordinador de Tutoriales
Xavier Franch (Universitat Politècnica deCatalunya)

Coordinador de Talleres
Antonio Ruiz (Universidad de Sevilla)

Ingeniería del Software y Bases de Datos

Primera edición, Septiembre 2006

Gran Capitán s/n, 08034 Barcelona, España
www.cimne.upc.es

Impreso por: Artes Gráficas Torres S.A., Morales17, 08029 Barcelona, España

Depósito legal: B-42461-2006

ISBN: 84-95999-99-4

Comité de Programa JISBD 2006

Jesús Aguilar (U. Sevilla)
José F. Aldana (U. Málaga)
Bárbara Álvarez (U. P. Cartagena)
María J. Aramburu (U. Jaume I)
Joao Araujo (U. Nova De Lisboa)
Orlando Belo (U. Do Minho)
Rafael Berlanga (U. Jaume I)
Pere Botella (U. P. Catalunya)
Nieves Brisaboa (U. Coruña)
Coral Calero (U. Castilla-La Mancha)
Carlos Canal (U. Málaga)
José M. Cavero (U. Rey Juan Carlos)
Matilde Celma (U. P. Valencia)
Rafael Corchuelo (U. Sevilla)
Dolors Costal (U. P. Catalunya)
Yania Crespo (U. Valladolid)
Carlos Delgado (U. Carlos III)
Oscar Díaz (U. País Vasco)
Javier Dolado (U. País Vasco)
Joao Falcão e Cunha (U. Porto)
Xavier Franch (U. P. Catalunya)
Pablo de la Fuente (U. Valladolid)
Lidia Fuentes (U. Málaga)
Mario J. Gaspar da Silva (U. Lisboa)
Marcela Genero (U. Castilla-La Mancha)
Juan Gómez (U. Alicante)
Alfredo Goñi (U. País Vasco)
Jon Iturrioz (U. País Vasco)
Elena Jurado (U. Extremadura)
Natalia Juristo (U. P. Madrid)
Antonia Lopes (U. Lisboa)
Adolfo Lozano (U. Extremadura)
Henrique Madeira (U. Coimbra)
Esperanza Marcos (U. Rey Juan Carlos)
Eduardo Mena (U. Zaragoza)
Ana Moreira (U. Nova De Lisboa)
Ana M. Moreno (U. P. Madrid)
Juan J. Moreno (U. P. Madrid)
Juan M. Murillo (U. Extremadura)
Oscar Pastor (U. P. Valencia)
Ernesto Pimentel (U. Málaga)
Ángeles Places (U. Coruña)
Antonio Polo (U. Extremadura)
Carme Quer (U. P. Catalunya)
Celia Ramos (U. Algarve)
Isidro Ramos (U. P. Valencia)
Isabel Ramos (U. Sevilla)
Antonio Rito (U. Técnica De Lisboa)
María J. Rodríguez (U. Granada)
Francisco Ruiz (Castilla-La Mancha)
Fernando Sánchez (U. Extremadura)
Juan Sánchez (U. P. Valencia)
Sofia Sousa Brito (I. P. Beja)
Ernest Teniente (U. P. Catalunya)
Miguel Toro (U. Sevilla)
Ambrosio Toval (U. Murcia)
Juan C. Trujillo (U. Alicante)
Javier Tuya (U. Oviedo)
Toni Urpi (U. P. Catalunya)
Antonio Vallecillo (U. Málaga)
Belén Vela (U. Rey Juan Carlos)

Comité Organizador
(U. P. Catalunya)

Alberto Abelló
Claudia Ayala
Xavier Burgués
Jordi Conesa
Dolors Costal
Cristina Gómez
Gemma Grau
Carme Martín
Joan Antoni Pastor
Paola Pizzi (Secretaría organización CIMNE)
Carme Quer
Angels Rius
Ruth Raventós
Toni Urpi

Revisores Adicionales

Alberto Abelló
Álvaro E. Prieto
Amparo Navasa
Ángel Herranz
Antònia Mas
Antonio Cesar Gómez
Antonio Ruiz
Arantza Illarramendi
Artur Boronat
Cesar J. Acuña
Clara Benac Earle
Cristina Vicente CHicote
Daniel Gomes
Daniel Jiménez
Dante Carrizo
Domingo S. Rodríguez-Baena
Dulce Domingos
Eduardo Pérez-Ureta
Encarna Sosa
Esperança Amengual
Fernando Molina
Fran J. Ruiz-Bertol
Francisco Gutiérrez
Francisco J. Lucas
Francisco J. García-Peñalvo
Francisco L. Gutiérrez
Herbert Kuchen
Isabel Nunes
Ismael Navas
Ismael Sanz
Javier Cámara
Javier Cubo
Javier Gutierrez
Jennifer Pérez
Jesús Arias
Joaquín Nicolás
Jordi Cabot
Jorge Martínez-Gil
José Luis Garrido
José Magno Lopes
José María Conejero
José Norberto Mazón
José Ramón Ríos
Juan Ángel Pastor
Juan Carlos Preciado
Juan Manuel Vara
Julia González
Lars-Åke Fredlund
Manuel Serrano
Marcirio Silveira Chaves
Mari Carmen Otero
María del Mar Roldán
María Esperanza Manso
María Isabel Sánchez Segura
María Teresa Gómez
María Visitación Hurtado
Marta Tabares
Martín Solari
Miguel Ángel Laguna
Miguel Á. Martínez-Aguilar
Miguel Á. Martínez-Prieto
Miguel Á. Pérez Toledano
Miguel Á. Rodríguez Luaces
Miguel Rodríguez Penabad
Mª Ángeles Moraga
Norberto Díaz-Díaz
Nuria Medina
Oscar Dieste
Paloma Cáceres
Pascal Poizat
Patricia Paderewski
Patricio Letelier
Pedro J. Muñoz
Pedro Sánchez-Palma
Pedro Valderas
Pepe Carsí
Rafael Ceballos
Raquel Trillo
Raúl Giráldez
Roberto Rodríguez-Echeverría
Santiago Melia
Sergio Ilarri Artigas
Toñi Reina
Toufik Taibi
Valeria de Castro
Vicente Luque
Vicente Pelechano
Xavier Ferré

Sistema Automático de Revisión
(Quercus Software Engineering Group)

Pablo Amaya — Universidad de Extremadura
Daniel García — Universidad de Extremadura

Entidades Patrocinadoras

FIB
Facultat d'Informàtica de Barcelona

UNIVERSITAT POLITÈCNICA DE CATALUNYA
UPC

CIMNE
BARCELONA

Microsoft

INTERSYSTEMS

gencat

MINISTERIO DE EDUCACIÓN Y CIENCIA

Prólogo

La undécima edición de las Jornadas de Ingeniería del Software y Bases de Datos se celebró en Sitges (Barcelona) entre el 3 y el 6 de Octubre de 2006. Desde aquellas primeras ediciones del año 1996 en Sevilla y La Coruña, donde las Jornadas de Ingeniería del Software y las de Bases de Datos se celebraron por separado hasta la presente edición se ha recorrido un largo camino. La unificación de las dos líneas en un solo encuentro, primero con estructuras separadas y desde hace dos ediciones con un Comité de Programa único, ha servido para consolidar a la comunidad JISBD como una de las más dinámicas en las tecnologías informáticas, como se demostró en el número de inscritos de la última edición celebrada en el seno del Primer Congreso Español de Informática (CEDI).

Como viene ocurriendo desde la edición del 2001, las JISBD han acogido la celebración, en paralelo y compartiendo algunos actos, de PROLE, las VI Jornadas de Programación y Lenguajes. Ambos eventos son organizados bajo los auspicios de SISTEDES (Sociedad de Ingeniería del Software y Tecnologías de Desarrollo de Software), sociedad constituida en Granada durante la celebración de CEDI en Septiembre del 2005. Desde la edición de 2006, todas las personas inscritas en JISBD o PROLE serán miembros de SISTEDES hasta la celebración de las siguientes jornadas.

En los diez años transcurridos, las JISBD han servido de foro de encuentro para motivar y servir de acicate al esfuerzo investigador de los participantes. Este impulso ha incrementando de manera muy importante la presencia en foros internacionales de trabajos de investigación de grupos españoles y portugueses. Así se puede consultar en el *ISI Web of Science* que el número de trabajos con las palabras claves "Software Engineering" provenientes de España o Portugal en el año 1996 fue de 3, de 15 en el 2002 y de más de 30 en el 2005. Con palabras claves referidas a Bases de Datos los resultados de crecimiento que se obtienen son similares. Es muy posible que versiones previas de esos trabajos fueron presentadas y, a su vez enriquecidas, en ediciones anteriores de las JISBD. Parece justo pensar que sin la existencia de estas Jornadas no se hubiera conseguido este significativo avance en la presencia internacional de sus participantes.

El presente libro de actas contiene los trabajos seleccionados por el Comité de Programa para la edición de este año 2006. Se recibieron un total de 123 trabajos con la siguiente distribución geográfica: 25 de Latinoamérica, 6 de Portugal, 90 de España, 1 de Francia y 1 de India. El Comité de Programa realizó una ardua tarea de revisión, mediante la cual cada trabajo fue revisado por tres o cuatro expertos, abriéndose posteriormente un debate para los trabajos que presentaban disparidad de criterios. El número final de trabajos seleccionados para publicarse completos fue de 41, considerándose además como interesantes 14 trabajos para su presentación como artículos cortos de 6 páginas.

Como es habitual de ediciones anteriores fueron dos las conferencias impartidas durante esta edición. La lección inaugural de título *Software Architecture: Past, Present, and Future* fue dictada por el profesor David Garlan de la prestigiosa *Carnegie Mellon University*. El profesor

Garlan es considerado uno de los fundadores del campo de la Arquitectura Software y, en particular, es experto en representación formal y análisis de diseño de arquitecturas. La segunda conferencia titulada *Model Independent Schema and Data Translation* fue pronunciada por el profesor Paolo Atzeni de la *Università Roma Tre*. El profesor Atzeni trabaja en tópicos relacionados con Bases de Datos, ha sido presidente de la *EDBT Association* y actualmente es secretario de la *VLDB Endowment*. Asimismo las dos conferencias de PROLE impartidas por los profesores Eelco Visser y Krzysztof Apt, han sido incluidas en el programa de JISBD.

También como en ediciones anteriores y con una importante participación e interés se desarrollaron los talleres asociados durante el primer día de las Jornadas. Un total de ocho talleres con la presentación y debate de nuevas propuestas en líneas de trabajo diversas como software orientado a aspectos, pruebas del software, sistemas hipermedias, bases de datos o servicios web. Los talleres de JISBD representan la vanguardia de la investigación y semillero de ideas, convirtiéndose en un foro de encuentro imprescindible dentro de las Jornadas. Asimismo, se ha ofrecido un interesante tutorial sobre Líneas de Producto Software por parte de los profesores Oscar Díaz y Salvador Trujillo.

La celebración de las JISBD con tan alto número de partícipes obliga a una importante labor desinteresada por parte de muchas personas. En primer lugar a los investigadores que han considerado que las JISBD eran un foro adecuado para presentar sus trabajos y a los distintos organizadores y participantes de los talleres. A los miembros del comité ejecutivo y del comité organizador que han coordinado los talleres y tutoriales, así como los detalles de la organización y celebración del encuentro. A los miembros del grupo Quercus por su, cada año mejor, sistema de revisión de trabajos. También queremos dar las gracias al personal del CIMNE, en especial a Paola Pizzi, por su eficaz soporte en la organización del evento. Finalmente, no hay palabras para agradecer y reconocer el trabajo realizado por el Comité de Programa y los revisores adicionales. Se han realizado 380 revisiones y más de 20 discusiones o debates sobre artículos con discrepancias. Gran parte del éxito de estas Jornadas se debe al tiempo que estos investigadores le han dedicado a esta tarea.

La próxima edición de las JISBD en el 2007 volverá a celebrarse en común con el CEDI en Zaragoza. Les deseamos a sus responsables un nuevo éxito de convocatoria que refleje el buen momento que goza la comunidad investigadora ibero-americana en Ingeniería del Software y Bases de Datos.

Sitges, Octubre de 2006
José C. Riquelme, Pere Botella (Editores)

INDICE

CONFERENCIAS INVITADAS

INGENIERÍA DE PROCESOS

MODELADO DE DATOS I

MANTENIMIENTO SOFTWARE

MODELADO DE DATOS II

CALIDAD

GENERACIÓN AUTOMÁTICA

MINERÍA DE DATOS

ARQUITECTURAS SOFTWARE I

INGENIERÍA DE REQUISITOS I

INGENIERÍA DE REQUISITOS II

ARQUITECTURAS SOFTWARE II

MISCELÁNEA SOFTWARE

TRABAJOS CORTOS

conceptual) utilizado en AMENITIES. Por medio de esta plantilla, hemos descrito dos ejemplos típicos de patrones conceptuales, uno de organización (*Joint Venture Pattern*) y otro de proceso (*Meeting Process Pattern*).
En estos momentos estamos construyendo un catálogo de patrones que facilite la especificación completa del modelo cooperativo [11].
En el futuro nuestra intención es establecer las posibles relaciones que pueden existir entre estos patrones (patrones conceptuales) con aquellos que pueden ser aplicados en una fase de diseño posterior (patrones de diseño).

REFERENCIAS

[1] E. Gamma, R. Helm, R. Johnson and J. Vlissides, *Design Patterns: Elements of Reusable Object-Oriented Software*, Reading, MA, Addison Wesley Professional Computing Series (1995)

[2] M. Fowler, *Analysis Patterns: Reusable Object Models*. Booch, G., Jacobson, I. and Rumbaugh, J. (eds.), Object Technology Series, Reading, MA, Addison-Wesley Publishing Company (1997)

[3] D. Martin, et al., "Finding Patterns in the Fieldwork", *proceedings of ECSCW 2001*, Bonn, Germany, Kluwer Academic Publishers, pp. 39-58, (2001)

[4] T. Schümmer, "Constructing a Groupware Pattern Language", *CSCW Workshop on Socio-Technical Pattern Languages*, New Orleans, November 16-20, (2002)

[5] J. Schümmer et al., "Collaborative Hypermedia Design Patterns in OOHDM", *2nd Workshop in Hypermedia Development: Design Patterns in Hypermedia*, (1999)

[6] A. Geyer-Schulz and M. Hahsler, "Software reuse with analysis patterns", in *Proceedings of AMCIS 2002*, Dallas, TX, (2002)

[7] J.L. Garrido, M. Gea , F.L. Gutierrez and N. Padilla, "Designing Cooperative Systems for Human Collaboration", In Dieng, R.; Giboin, A. (Eds), *Designing Cooperative Systems: The Use Of Theories and Models*, IOS press, Netherlands (2000)

[8] J.L. Garrido, *AMENITIES: Una metodología para el desarrollo de sistemas cooperativos basada en modelos de comportamiento y tareas*. PhD Thesis University of Granada (2003)

[9] J.L. Garrido P. Paderewski, M.L Rodríguez., M.Hornos, M.Noguera, "A Software Architecture Intended to Design High Quality Groupware Applications", *Proc. of The 2005 International Conference on Software Engineering Research and Practice*, CSREA Press, pp. 59-65, (2005)

[10] J.L. Isla, F.L. Gutiérrez and P. Paderewski, "Un Profile para el Modelado de Patrones de Software", *Actas de las X Jornadas en Ingeniería del Software y Bases de Datos*, Thomson Paraninfo, pp. 265-270, (2005)

[11] J.L., Isla, F.L. Gutiérrez, M. Gea, "Supporting Social Organization Modelling in Cooperative Work Using Patterns", In Shen, W. et al. (eds.), *Computer Supported Cooperative Work in Design II*, LNCS 3865, Springer, pp. 112-121, (2006),

XI Jornadas de Ingeniería del Software y Bases de Datos
JISBD 2006
José Riquelme - Pere Botella (Eds)

APLICACIÓN PRÁCTICA DE UN PROCESO DE INGENIERÍA DE REQUISITOS DE SEGURIDAD

Daniel Mellado[1]*, Eduardo Fernández-Medina[2] y Mario Piattini[2]

1: Centro Informático del Instituto Nacional de la Seguridad Social,
Ministerio de Trabajo y Asuntos Sociales. Madrid, España
e-mail: Daniel.Mellado@alu.uclm.es

2: Grupo ALARCOS
Departamento de Tecnologías y Sistemas de Información. Universidad de Castilla-La Mancha
Paseo de la Universidad 4, 13071 Ciudad Real, España
e-mail: (Eduardo.FdezMedina,Mario.Piattini)@uclm.es, web: http://alarcos.inf-cr.uclm.es/

Palabras clave: Requisitos de seguridad, Ingeniería de requisitos de seguridad, Ingeniería de requisitos, Seguridad, Desarrollo seguro, Caso de estudio.

Resumen. *Actualmente las soluciones de seguridad están principalmente enfocadas en proporcionar defensas de seguridad a posteriori, en vez de centrarse en la raíz del problema, un diseño apropiado de los Sistemas de Información desde el principio. En este artículo se presenta un caso de estudio del proceso de ingeniería de requisitos de seguridad, llamado SREP (Security Requirements Engineering Process), que hemos propuesto, demostrándose como de una forma guiada, sistemática e intuitiva se pueden tratar los requisitos de seguridad desde las primeras fases del desarrollo de software, utilizándose para ello un repositorio de recursos de seguridad e integrando los Criterios Comunes en el proceso de desarrollo de software.*

1. INTRODUCCIÓN

Hoy en día los Sistemas de Información (SI) son vulnerables a multitud de amenazas. Además, cuanto más se incrementa la complejidad de las aplicaciones y servicios, más aumenta la potencialidad de presentar brechas de seguridad [1]. Para que progrese la actual Sociedad de la Información, dependiente de un gran número de sistemas software que tienen una misión crítica, es absolutamente vital que los SI sean asegurados apropiadamente desde el principio [2, 3]. Sin embargo, el gran problema es que en la mayoría de los proyectos software la seguridad se trata una vez el sistema ha sido diseñado e implementado. Debido en muchos casos, a una gestión inapropiada de la especificación de los requisitos de seguridad del nuevo sistema, ya que la fase de especificación de requisitos suele realizarse con unas cuantas descripciones o la especificación de objetivos plasmados en unos pocos folios [4]. Además, habitualmente los requisitos de seguridad no se entienden bien. De forma que, incluso cuando se intenta especificar los requisitos de seguridad, muchos desarrolladores tienden a describir soluciones de diseño en términos de mecanismos de protección en lugar de

realizar proposiciones declarativas sobre el grado de protección requerido [5].
Una parte muy importante en el proceso de desarrollo software para conseguir sistemas software seguros es la denominada Ingeniería de Requisitos de Seguridad, que proporciona técnicas, métodos y normas para abordar esta tarea en el ciclo de desarrollo de los SI y que implica el uso de procedimientos repetibles y sistemáticos para asegurar que el conjunto de requisitos obtenidos es completo, consistente y fácilmente comprensible y analizable por parte de los diferentes actores implicados en el desarrollo del sistema [6].
Después de haber analizado diferentes propuestas técnicas relevantes relativas a los requisitos de seguridad para el desarrollo de SI seguros, tales como Yu 1997 [7], Toval et al. 2001 [8], Popp et al. 2003 [9], Firesmith 2003 [5], etc. en [10], concluimos que estas propuestas no alcanzaban el nivel deseado de integración en el desarrollo de SI, ni eran lo suficientemente específicas para el tratamiento sistemático e intuitivo de requisitos de seguridad en las primeras fases del desarrollo software. Además, hasta ahora sólo hay algunos trabajos (como el de Massacci et al [11]) que describan casos de estudio complejos que realmente son conformes a la complejidad requerida por los estándares de seguridad, como los estándares ISO/IEC 15408 e ISO/IEC 17799. Por tanto en este artículo se presenta un caso de estudio de aplicación de nuestro proceso propuesto, SREP (Security Requirements Engineering Process) [12], el cual describe cómo integrar los requisitos de seguridad en el proceso de ingeniería del software de una forma sistemática e intuitiva. Nuestra propuesta está basada en la integración de los Criterios Comunes (CC) en el ciclo de vida de desarrollo software, junto con la reutilización de requisitos de seguridad. Además, para facilitar la consecución de esta tarea proponemos utilizar varios conceptos y técnicas: como UMLSec [9], casos de mal uso [13], árboles de ataque, y casos de uso de seguridad [14]. El resto del artículo está organizado de la siguiente forma: en la sección 2, se ofrece un breve resumen de SREP. La sección 3 presenta el caso de estudio de SREP. Seguidamente, en la sección 4 se presentan las lecciones aprendidas. Finalmente, presentamos nuestras conclusiones y futuros trabajos en la sección 5.

2. SREP: PROCESO DE INGENIERÍA DE REQUISITOS DE SEGURIDAD

SREP [12] es un proceso basado en activos y dirigido por el riesgo para el establecimiento de requisitos de seguridad en el desarrollo de SI seguros. Básicamente este proceso describe como integrar los CC en el ciclo de desarrollo junto con el uso de un repositorio de recursos de seguridad para facilitar la reutilización de requisitos, activos, amenazas, tests y contramedidas. Esta metodología está centrada en la construcción de conceptos de seguridad en las primeras fases de desarrollo. Como se describe en la Fig. 1, proponemos que SREP se integre en el ciclo de vida Proceso Unificado, que como sabemos está divido en una secuencia de fases y cada fase puede implicar varias iteraciones. De esta manera, el modelo elegido por SREP es iterativo e incremental y los requisitos de seguridad y sus elementos asociados (amenazas, etc.) evolucionan a lo largo del ciclo de vida. Al mismo tiempo, los Componentes de los CC se introducen en el ciclo de vida de desarrollo software, de manera que SREP usa los diferentes componentes según la fase del ciclo de vida en que se esté y la actividad de SREP correspondiente, aunque las actividades de aseguramiento de la calidad (SQA) se realizan durante todas las fases y son en estas

actividades donde la mayoría de los requisitos de aseguramiento de los CC se incorporan.

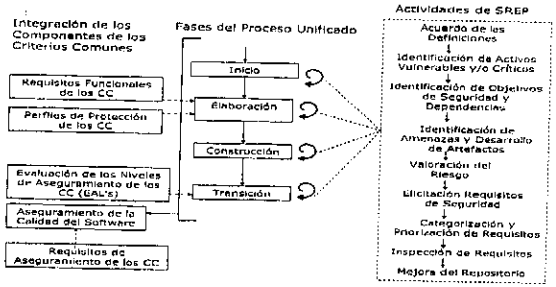


Fig. 1 Visión general de SREP

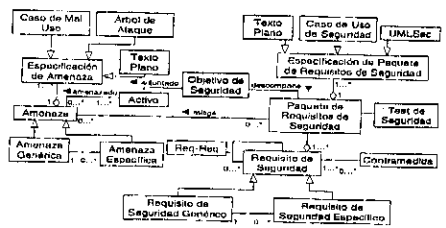


Fig. 2 Meta-modelo del repositorio de recursos de seguridad

El Repositorio de Recursos de Seguridad (RRS) facilita el desarrollo con reutilización de requisitos, lo cual incrementa su calidad, ya que las inconsistencias, errores, ambigüedades y otros problemas se pueden detectar y corregir en proyectos sucesivos [8].
Un meta-modelo de repositorio, el cual es una extensión del propuesto por Sindre et al [13], describiendo la organización del RRS se muestra en la Fig. 2. Siendo los objetos representados con fondo oscuro nuestra contribución a dicho meta-modelo. Se trata de un meta-modelo dirigido por activos así como por amenazas, porque los requisitos pueden obtenerse a través de los activos o de las amenazas. Una descripción más completa de dicho repositorio puede encontrarse en [12].
Los CC no proporcionan soporte metodológico, ni contienen criterios de evaluación de la seguridad relativos a las medidas de seguridad administrativas no directamente relacionadas con las medidas de seguridad del SI. Por tanto, y de acuerdo con la norma ISO/IEC 17799:2005, proponemos incluir el conjunto de requisitos legales, estatutarios, regulatorios, y

contractuales que deberían satisfacer la Organización, sus socios comerciales, los contratistas, y los proveedores de servicios, y su entorno socio-cultural. Con lo que después de adecuar estos requisitos al formato de los requisitos del sistema o bien requisitos software, éstos constituirán el subconjunto inicial de requisitos de seguridad del RRS para cualquier proyecto.

3. CASO DE ESTUDIO

El caso de estudio que presentamos es un caso representativo de un SI donde la seguridad es un aspecto crítico. Se analizará el caso de una unidad administrativa del INSS (Instituto Nacional de la Seguridad Social), cuyo fin es proporcionar un servicio de administración electrónica (que llamaremos PensionApp) consistente en facilitar información relativa a la pensión/es del ciudadano. Teniendo en cuenta las restricciones de espacio, este caso de estudio se ha simplificado para permitir así ilustrar fácilmente los puntos principales de SREP en este artículo.

PensionApp permitirá a los ciudadanos obtener un documento oficial donde se refleje la cuantía y/o el estatus de su/s pensión/es, así como les permitirá actualizar algunos datos personales, como su dirección y número de cuenta bancario en el que recibir la/s pensión/es. Uno de sus principales objetivos de diseño es la facilidad de uso, por ello los ciudadanos podrán acceder en línea por Internet a PensionApp o personarse en una oficina del INSS, donde un funcionario les realizará el trámite (estando esta última aplicación ya desarrollada e independiente de PensionApp). De este modo, asumimos que los requisitos funcionales iniciales ya han sido elicitados y que sólo son dos:

- Req 1: En petición-1 del Usuario, el sistema deberá presentar la información sobre su pensión/es. Esta petición deberá incluir el número de la seguridad social del Usuario.
- Req 2: En petición-2 del Usuario, el sistema deberá actualizar los datos personales del pensionista. Esta petición deberá incluir el número de seguridad social del Usuario y los datos que han sido actualizados.

Además también asumimos que la Organización ya ha introducido algunos elementos en el repositorio (como requisitos legales, etc.) tal y como se propone en SREP.

SREP define nueve actividades que se realizan en varias iteraciones, en las cuales se generan artefactos que constituyen una línea base, aunque en este artículo únicamente se describirá brevemente una iteración de SREP en las primeras fases del ciclo de desarrollo.

3.1. Actividad 1: Acuerdo de las definiciones

En esta actividad se tienen que acordar una serie de definiciones de seguridad basándose en estándares, como las propuestas en la ISO/IEC 17799:2005. Además se definirá o se recogerá la política de seguridad de la Organización y se redactará el Documento de Visión de Seguridad del SI. En este caso, se reflejará el hecho de que la información es el activo más importante y que se deberá garantizar la confidencialidad, disponibilidad e integridad de ésta y la autenticidad, no-repudio, trazabilidad de usuarios y del servicio.

3.2. Actividad 2: Identificación de activos vulnerables y/o críticos

Después de examinar los requisitos funcionales (Req 1 y Req2) (y de acuerdo con el requisito de aseguramiento ADV_FSP.3.1D de los CC) obtuvimos como único tipo de activo relevante la Información. Aunque los otros tipos de activos, los activos tangibles e intangibles, también necesitarían ser considerados en un caso de estudio real sin restricciones de espacio. Consideramos dos tipos diferentes de información, en función de su nivel de seguridad requerido (según la Ley Orgánica de Protección de Datos 15/1999):

- Información personal del pensionista: nombre, número de seguridad social, dirección.
- Información personal de la pensión: tipo de pensión (jubilación, invalidez (tipo y grado de invalidez), viudedad,...), cuantía de la pensión, número de cuenta corriente.

3.3. Actividad 3: Identificación de objetivos de seguridad y dependencias

En esta actividad el RRS se puede usar de forma que si los tipos de activos identificados en la actividad anterior están en él, podemos obtener los objetivos de seguridad asociados a éstos. Sino, se determinarán los objetivos de seguridad para cada activo, teniéndose en cuenta la política de seguridad de la Organización y las restricciones/requisitos legales de España y del INSS. Identificamos los siguientes objetivos de seguridad (OS):

- OS1: Prevenir la revelación no autorizada de información (Confidencialidad). Valoración: Alto.
- OS2: Prevenir la alteración no autorizada de información (Integridad). Valoración: Alto.
- OS3: Asegurar la disponibilidad de la información a los usuarios autorizados. Valoración: Medio.
- OS4: Asegurar la autenticidad de los usuarios. Valoración: Alto.
- OS5: Asegurar la trazabilidad. Valoración: Medio.

Esta lista no es una lista completa de objetivos de seguridad, por lo que debería ser refinada en las sucesivas iteraciones (por ejemplo estableciendo los objetivos en términos de probabilidad y las dependencias entre ellos), pero la consideramos suficiente para este caso. Estos objetivos de seguridad se plasman en el Documento de Objetivos de Seguridad, basándose en las clases de aseguramiento de los CC (clase ASE).

3.4. Actividad 4: Identificación de amenazas y desarrollo de artefactos

Si los activos identificados en la actividad anterior están en el repositorio, podemos obtener las amenazas asociados a los mismos. Sino buscaremos aquellas amenazas que pueden provocar que no se alcancen los objetivos de seguridad, mediante la instanciación de los casos de uso de negocio en casos de mal uso y/o mediante la instanciación de los árboles de ataque asociados a los patrones de negocio y aplicación. Además, analizaremos las listas de amenazas predefinidas para los tipos de activos seleccionados y siguiendo los requisitos de aseguramiento de los CC (AVA-VAN.5.2E) buscaremos también en fuentes de dominio público la existencia de vulnerabilidades potenciales para nuestro SI.

Por tanto tras analizar los casos de mal uso, identificamos varios tipos posibles de

amenazas sobre la Información:

- Amenaza Genérica 1: Revelación no autorizada de información.
- Amenaza Genérica 2: Alteración no autorizada de información.
- Amenaza Genérica 3: Indisponibilidad no autorizada a la información.
- Amenaza Genérica 4: Suplantación de identidad de usuario.

Después se desarrollarán los elementos 'Amenaza Genérica' junto con sus 'Amenazas específicas' (que son cada una de los diversos caminos de la amenaza genérica). Se muestra en la Tabla 1 un ejemplo de especificación de 'Amenaza Genérica'.

Finalmente, con la información recogida en esta actividad se redactará la primera versión del Documento de Definición del Problema de Seguridad con la ayuda de la clase de aseguramiento ASE de los CC. En este documento se plasmarán las suposiciones que hagamos, aunque no tendremos en cuenta, ya que no es el principal objetivo de este trabajo, los posibles ataques a las infraestructuras u otros elementos organizacionales.

Nombre Caso Mal Uso: Acceso al contenido de un mensaje HTTP [recibido / enviado] del Usuario		
ID: CMUG-2-2-1-1 [CMUG-Objetivo Seguridad-Amenaza Genérica- Iteración- Caso Mal Uso Genérico]		
PROBABILIDAD: [Muy Frecuente | Frecuente | Normal Frecuencia | Poca Frecuencia]		
Resumen: El atacante [tipo de atacante] accede al mensaje [nombre mensaje] intercambiado por el agente [consumidor|proveedor] [nombre agente] y el agente [consumidor|proveedor] [nombre agente] y [modifica|elimina|inserta [parte]] el mensaje a nivel de la capa de [transporte / HTTP] ubicada en la [cabecera|cuerpo|anexo] con el propósito de [objetivo]		
Precondiciones: 1) El atacante tiene acceso físico al mensaje. 2) El atacante tiene un conocimiento claro de la estructura y significado del mensaje.		
Interacciones del Usuario	**Interacciones del Mal Usuario**	**Interacciones del Sistema**
El Usuario envía el mensaje [nombre mensaje]		
	El atacante lo intercepta e identifica la parte a modificar y [elimina, reemplaza o agrega] información y reenvía el mensaje al Agente del Sistema	
		El agente del Sistema recibe el mensaje y lo procesa de forma errónea en base a la semántica alterada
Postcondiciones: 1) El sistema quedará en un estado erróneo con respecto a las intenciones originales del agente consumidor [nombre agente]. 2) En el registro del sistema sobre el que se ejecuta el agente proveedor [nombre agente] aparecerá que la petición recibida fue aquella con la semántica alterada.		

Tabla 1 Especificación de amenaza genérica usando casos de mal uso (CMUG)

3.5. Actividad 5: Valoración del riesgo

Habiendo identificado las amenazas estimaremos la probabilidad de que se materialice cada una de ellas y estimaremos su impacto y consecuentemente estableceremos su riesgo. Para realizar esta tarea usaremos una de las técnicas propuestas en la guía de técnicas de MAGERIT v.2 y que se basa en tablas para analizar el impacto y el riesgo de las amenazas.

El riesgo y el impacto se evaluará según la siguiente escala: muy bajo, bajo, medio, alto y muy alto. Y la frecuencia de las amenazas: Muy frecuente (diario), frecuente (mensual), frecuencia normal (anual), poca frecuencia (una vez cada varios años). Tendremos por tanto que generar tablas de las amenazas, los ataques (casos de mal uso: CMU) y riesgos para registrar la valoración del impacto y el riesgo según las amenazas identificadas. En la

Tabla 2 presentamos un ejemplo de análisis del riesgo de una amenaza previamente detallada en la actividad anterior. Todas estas valoraciones son capturadas en el Documento de Valoración del Riesgo que será también refinado en sucesivas iteraciones.

Tabla de Amenazas, Ataques y Riesgos - Iteración 1				
Amenaza	Impacto	Ataque	Probabilidad	Riesgo
I.2.I.1.1.1 Alteración de la información	BAJO, si no hay datos de la pensión modificados	*CMU-2-2-1-1-1*	ALTA	BAJO
	ALTO en caso contrario.	*CMU-2-2-1-1-1*	ALTA	ALTO

Tabla 2 Tabla de amenazas, ataques y riesgos

3.6. Actividad 6: Elicitación de requisitos de seguridad

A fin de obtener los requisitos de seguridad, se analiza la relevancia de cada objetivo de seguridad junto con sus amenazas que impliquen más riesgo, de manera que se seleccionen aquellos requisitos de seguridad o paquete de requisitos de seguridad que mitiguen esas amenazas a los niveles necesarios. Para ello, primeramente usaremos el conocimiento del dominio de la aplicación para transformar las entidades descritas en los objetivos de seguridad en entidades existentes en los requisitos funcionales. En este caso es sencillo, cuando en los objetivos de seguridad se hace referencia a Información sabemos que se refiere a la información acerca del pensionista y/o de su pensión en el contexto de los requisitos funcionales. Después transformaremos los objetivos de seguridad en restricciones sobre las operaciones que usan los requisitos funcionales. Además, buscaremos en el repositorio por el catálogo de requisitos de los CC, requisitos que mitiguen las amenazas identificadas que puedan provocar la no consecución de los objetivos de seguridad, y requisitos que aseguren el desarrollo seguro de PensionApp.

Los requisitos de seguridad (RS) que identificamos son los siguientes:

- RS1: Las funciones de seguridad de PensionApp deberán usar criptografía [asignación: algoritmo criptográfico y tamaño de clave] para garantizar la confidecialidad de la información relativa a las pensiones proporcionada por PensionApp. (Requisito FCO_CED.1.1 de los CC).
- RS2: Las funciones de seguridad de PensionApp deberán identificar y autenticar al usuario mediante credenciales [asignación: técnica de desafío-respuesta basada en intercambio de datos aleatorios cifrados, certificado de clave pública] antes de que el usuario se conecte a PensionApp. (Requisitos CC FIA_UID.2.1, FIA_UAU.1.1).
- RS3: Cuando PensionApp transmite información relativa al pensionista o de su pensión al usuario, las funciones de seguridad de PensionApp deberán proporcionar al usuario los medios [asignación: firma electrónica] para detectar [selección: modificación, eliminación, inserción, suplantación, otra alteración] anomalías. (Requisito FCO_IED.1.1 de los CC).
- RS4: Las funciones de seguridad de PensionApp deberán asegurar la disponibilidad de la información que se proporciona al usuario dentro de [asignación: una métrica definida de disponibilidad] dada en las siguientes condiciones [asignación: condiciones para asegurar la disponibilidad]. (Requisito FCO_AED.1.1 de los CC).
- RS5: Las funciones de seguridad de PensionApp deberán requerir evidencias de que

la información que proporciona al usuario ha sido recibida correctamente. (Requisito FCO_NRE.1.1 de los CC).

- RS6: Las funciones de seguridad de PensionApp deberán almacenar un registro de auditoría de los siguientes eventos [selección: la petición de información sobre la pensión, la respuesta de PensionApp] y cada uno de los registros deberán registrar la siguiente información: fecha y hora del evento, [selección: éxito, fallo] del evento, y la identidad del usuario. (Requisito FAU_GEN de los CC).

Después se especificamos los requisitos usando la técnica de los casos de mal uso. Por último, se redacta en esta actividad el Documento de Especificación de Requisitos de Seguridad, que será refinado en las sucesivas iteraciones, ya que se trata de evitar la innecesaria y prematura especificación de medidas arquitectónicas o de diseño. Y debido a estos requisitos de seguridad los requisitos funcionales elicitados (Req 1 y 2) tienen que actualizarse, de manera que las peticiones incluyan las credenciales del usuario.

3.7. Actividad 7: Categorización y priorización de requisitos

En función al impacto y a la probabilidad de las amenazas, es decir, según el riesgo, priorizamos los RS así: 1º- RS1; 2º- RS2; 3º-RS3; 4º- RS5 y RS 6; 5º- RS4.

3.8. Actividad 8: Inspección de requisitos

En esta actividad se generará el Informe de Validación, revisándose así la calidad del trabajo realizado hasta ahora, con la ayuda de los requisitos de aseguramiento de los CC. Estos requisitos de aseguramiento se pueden obtener según el EAL (Evaluation Assurance Level) establecido y acordado con las partes interesadas ('stakeholders') en la actividad 1, aunque dicho nivel puede modificarse en las subsiguientes iteraciones. Supondremos que para este caso se acordó EAL-1 (probado funcionalmente), con lo que los componentes de aseguramiento que se utilizarán serán los que establece dicho nivel, los cuales se asume que fueron inicialmente introducidos en el repositorio. Después escribiremos la primera versión del Documento de Fundamentación de los Requisitos de Seguridad con la ayuda de las clases de aseguramiento de los CC (clase ASE), demostrando que si se satisfacen todos los requisitos de seguridad y se alcanzan todos los objetivos de seguridad, el problema de seguridad definido previamente estará resuelto, ya que todas las amenazas son contrarestadas, las políticas de seguridad organizacionales son reforzadas y todas las suposiciones son sostenibles.

3.9. Actividad 9: Mejora del repositorio

En el repositorio se almacenarán los nuevos elementos, en el caso de esta iteración los elementos 'Generic' y 'Specific Threats' y 'Requirements' que fueron desarrollados en las actividades 4 y 6. Finalmente se escribirá el Documento de Declaración de Seguridad ('Security Target') de los CC. Además esta actividad se realizará coincidiendo con los hitos existentes al final de cada fase del Proceso Unificado.

4. LECCIONES APRENDIDAS

Podemos destacar como lecciones aprendidas más importantes en este caso de estudio presentado anteriormente las siguientes:

- La aplicación de este caso de estudio nos ha permitido mejorar y refinar las siguientes actividades de SREP: identificación de objetivos de seguridad, identificación de amenazas y elicitación de requisitos.
- Como se trata de un proceso de ingeniería de requisitos de seguridad iterativo e incremental, hemos observado que esta filosofía nos permite tener en cuenta los cambios de requisitos, facilita la reutilización y la corrección de errores en las sucesivas iteraciones, los riesgos son descubiertos y mitigados antes, y el proceso en sí mismo puede ser mejorado y refinado durante el proceso de ejecución del mismo.
- Respecto a la experiencia con los CC, hemos apreciado que en ocasiones es difícil averiguar el significado correcto de los requisitos de los CC, sería más sencillo si los CC proporcionaran ejemplos para cada uno de éstos. Sin embargo, los CC nos ofrecen una importante ayuda para el tratamiento sistemático de requisitos de seguridad, a pesar del hecho de que los requisitos de los CC tienen complejas dependencias y que los CC no proporcionan ningún método/guía para incluirlos en el proceso de desarrollo software, de manera que una modificación en un documento a veces implica la modificación de varios otros documentos.
- El soporte de una herramienta es crucial para la aplicación práctica de este proceso en sistemas software de gran magnitud debido al número de artefactos manejados y las sucesivas iteraciones que se tienen que realizar.

5. CONCLUSIONES Y TRABAJO FUTURO

En este artículo demostramos como los requisitos de seguridad para un SI donde la seguridad es crítica, pueden obtenerse de una forma guiada y sistemática mediante la aplicación de SREP, un proceso basado en estándares, que trata los requisitos de seguridad desde las primeras fases de desarrollo software de una manera sistemática e intuitiva, apoyándose para ello en la reutilización de requisitos de seguridad, proporcionando un Repositorio de Recursos de Seguridad junto con la integración de los CC en el ciclo de vida de desarrollo software. Además, SREP es conforme a la ISO/IEC 17799:2005 en lo referente a requisitos de seguridad (secciones: 0.3, 0.4, 0.6 y 12.1).

Como futuros trabajos, es necesario desarrollar una herramienta CARE (Computer-Aided Requirements Engineering) que apoye nuestro proceso facilitando el tratamiento sistemático e intuitivo de los requisitos de seguridad, así como refinar teóricamente el mismo a través de su aplicación en más casos de estudio reales.

AGRADECIMIENTOS

Este artículo es parte de los proyectos DIMENSIONS (PBC-05-012-2) de la Consejería de Ciencia y Tecnología de la Junta de Comunidades de Castilla- La Mancha y el FEDER, y los proyectos CALIPO (TIC2003-07804-CO5-03) y RETISTIC (TIC2002-12487-E) de la

Dirección General de Investigación del Ministerio de Educación y Ciencia.

REFERENCIAS

[1]. J.P. Walton, *Developing a Enterprise Information Security Policy*. 2002, ACM Press: Proceedings of the 30th annual ACM SIGUCCS conference on User services.

[2]. R. Baskeville, *The development duality of information systems security*. Journal of Management Systems, 1992. **4**(1): p. 1-12.

[3]. J. McDermott and C. Fox. *Using Abuse Case Models for Security Requirements Analysis*. in *Annual Computer Security Applications Conference*. 1999. Phoenix, Arizona.

[4]. E. Fernández-Medina, R. Moya, and M. Piattini Velthus, *Gestión de Requisitos de Seguridad*, in *Seguridad de las Tecnologías de la Información "La construcción de la confianza para una sociedad conectada"*, AENOR, Editor. 2003. p. pp 593-618.

[5]. D.G. Firesmith, *Engineering Security Requirements*. Journal of Object Technology, 2003. **2**(1): p. 53-68.

[6]. G. Kotonya and I. Sommerville, *Requirements Engineering Process and Techniques*. Hardcover ed. 1998. 294.

[7]. E. Yu, *Towards Modelling and Reasoning Support for Early-Phase Requirements Engineering*. 1997: 3rd IEEE International Symposium on Requirements Engineering (RE'97). p. 226-235.

[8]. A. Tovat, J. Nicolás, B. Moros, and F. García, *Requirements Reuse for Improving Information Systems Security: A Practitioner's Approach*. 2001: Requirements Engineering Journal. p. 205-219.

[9]. G. Popp, J. Jürjens, G. Wimmel, and R. Breu, *Security-Critical System Development with Extended Use Cases*. 2003: 10th Asia-Pacific Software Engineering Conference. p. 478-487.

[10]. D. Mellado, E. Fernández-Medina, and M. Piattini, *A Comparative Study of Proposals for Establishing Security Requirements for the Development of Secure Information Systems*. The 2006 International Conference on Computational Science and its Applications (ICCSA 2006), Springer LNCS 3982, 2006: p. 1044-1053.

[11]. F. Massacci, M. Prest, and N. Zannone, *Using a security requirements engineering methodology in practice: The compliance with the Italian data protection legislation*. Computers Standards and Interfaces, **27** (2005). p. 445-455.

[12]. D. Mellado, E. Fernández-Medina, and M. Piattini, *A Common Criteria Based Security Requirements Engineering Process for the Development of Secure Information Systems*. Computer Standards and Interfaces, (2006).

[13]. G. Sindre, D.G. Firesmith, and A.L. Opdahl. *A Reuse-Based Approach to Determining Security Requirements*. in *Proc. 9th International Workshop on Requirements Engineering: Foundation for Software Quality (REFSQ'03)*. 2003. Austria.

[14]. D.G. Firesmith, *Security Use Cases*. Journal of Object Technology, 2003: p. 53-64.

XI Jornadas de Ingeniería del Software y Bases de Datos
JISBD 2006
José Riquelme - Pere Botella (Eds)

DISENTANGLING CROSSCUTTING IN AOSD: FORMALIZATION BASED ON A CROSSCUTTING PATTERN*

José M. Conejero[1], Klaas van den Berg[2] and Juan Hernández[1]

1: Quercus Software Engineering Group
University of Extremadura
Avda. Universidad s/n C.P. 10071 Cáceres, Spain
e-mail: {chemacm,juanher}@unex.es web: http://quercusseg.unex.es

2: Software Engineering Group
University of Twente
P.O. Box 217, 7500 AE Enschede, the Netherlands
e-mail: k.g.vandenberg@ewi.utwente.nl, web: http://trese.cs.utwente.nl/

Keywords: Aspect-Oriented Software Development, Scattering, Tangling, Crosscutting

Abstract. Crosscutting is usually described in terms of scattering and tangling. However, the distinction between these concepts is vague, which could lead to ambiguous statements. Sometimes, precise definitions are required, *e.g.*, for the formal identification of crosscutting concerns. We propose a conceptual framework for formalizing these concepts based on a crosscutting pattern that shows the mapping between elements at two levels, *e.g.*, concerns and representations of concerns. The definitions of the concepts are formalized in terms of linear algebra, and visualized with matrices and matrix operations. In this way, crosscutting can be clearly distinguished from scattering and tangling. The usability of dependency matrices is illustrated in the analysis of crosscutting across several refinement levels, which can be formalized through the cascading of the crosscutting pattern.

1. INTRODUCTION

One of the key principles in Aspect-Oriented Software Development (AOSD) is Separation of Concerns (SOC). A concern can be defined very generally as a thing in an engineering process about which it cares [7]. Related with this principle is the problem of crosscutting concerns. Crosscutting is usually described in terms of scattering and tangling, *e.g.*, crosscutting is the scattering and tangling of concerns arising due to poor support for their modularization. However, the distinction between these concepts is vague, sometimes leading to ambiguous statements and confusion, as stated in [11]:

.. the term "crosscutting concerns" is often misused in two ways: To talk about a single concern, and to talk about concerns rather than representations of concerns. Consider "synchronization is a crosscutting concern": we don't know that synchronization is crosscutting unless we know what it crosscuts. And there may be representations of the

* In conjunction with AOSD-Europe Project IST-2-004349-NoE (see [1]) and MEC TIN2005-09405-C02-02