Universidad de Murcia
Departamento de Informática y Sistemas

Marco de trabajo de representación y reuso de requisitos de seguridad

Tesis Doctoral

Doctorando: Joaquín Lasheras Velasco

Director: José Ambrosio Toval Álvarez

2011

University of Murcia
Department of Informatics and Systems

A framework for representation and reuse of security requirements

PhD Thesis

PhD Candidate: Joaquín Lasheras Velasco

Advisor: José Ambrosio Toval Álvarez

2011

D. José Ambrosio Toval Álvarez, Catedrático de Universidad del Área de Lenguajes y Sistemas en el Departamento de Informática y Sistemas, AUTORIZA:

La presentación de la Tesis Doctoral titulada “MARCO DE TRABAJO DE REPRESENTACIÓN Y REUSO DE REQUISITOS DE SEGURIDAD“, realizada por D. Joaquín Lasheras Velasco, bajo mi inmediata dirección y supervisión, y que presenta para la obtención del grado de Doctor por la Universidad de Murcia.

Murcia, a 18 de Mayo de 2011

D. Jesús Joaquín García Molina, Catedrático de Escuela Universitaria del Área de Lenguajes y Sistemas Informáticos y Director del Departamento de Informática y Sistemas, INFORMA:

Que la Tesis Doctoral titulada "MARCO DE TRABAJO DE REPRESENTACIÓN Y REUSO DE REQUISITOS DE SEGURIDAD", ha sido realizada por D. Joaquín Lasheras Velasco, bajo la inmediata dirección y supervisión de D. J. Ambrosio Toval Álvarez, y que el Departamento ha dado su conformidad para que sea presentada ante la Comisión de Doctorado.

Murcia, a 18 de Mayo de 2011

Detalles Formales de Presentación

Esta tesis doctoral opta a la Mención de Doctorado Europeo y se presenta en la modalidad de compendio de publicaciones gracias a la autorización, que se adjunta, emitida en este sentido por la Comisión General de Doctorado de la Universidad de Murcia. La normativa de la modalidad de compendio de publicaciones únicamente obliga a incluir como documentación de la tesis un breve resumen de la misma y al menos tres artículos publicados, o aceptados para su publicación, en revistas indexadas de reconocido prestigio, como son las incluidas con factor de impacto en el listado ISI/JCR (*Journal Citations Report*). A continuación, se muestran las referencias a cuatro artículos presentados para la aceptación por compendio, y la de un quinto en el que también ha estado implicado el doctorando durante la elaboración de esta tesis, pudiendo ser consultados en el *Anexo I Publicaciones en JCR* de este documento:

- Ambrosio Toval, Begoña Moros, Joaquín Nicolás y Joaquín Lasheras, *Eight Key Issues for an Effective Reuse-Based Requirements Process.* International Journal of Computer Systems Science and Engineering, Vol 23 (6) p. 373-385 Noviembre 2008.
  Factor Impacto en el listado JCR 2009: 0,222 (posición 48/49, grupo Computer Sc. Hw. & Arch.) (posición 90/92, grupo Computer Sc.Theory and Methods)

- Joaquín Nicolás, Joaquín Lasheras, Ambrosio Toval, Francisco J. Ortiz y Bárbara Álvarez, *An Integrated Domain Analysis Approach for Teleoperated Systems.* Requirements Engineering Journal (REJ), Vol 14(1) p. 27-46 Enero 2009.
  Factor Impacto en el listado JCR 2009: 0,931 (posición 61/93, grupo Computer Sc. Soft. Eng.) (posición 76/116, grupo Computer Sc. Information Systems)

- Miguel A. Martínez, Joaquín Lasheras, Ambrosio Toval, Eduardo Fernandez-Medina y Mario Piattini, *A Personal Data Audit Method through Requirements Engineering.* Computer Standards and Interfaces, Vol 32(4), p 166-178 Junio 2010.
  Factor Impacto en el listado JCR 2009: 1,373 (posición 37/93, grupo Computer Sc. Soft. Eng.) (posición 18/49, grupo Computer Sc. Hw. & Arch.)

- Joaquín Lasheras, Rafael Valencia-García, Jesualdo Tomás Fernández-Breis y Ambrosio Toval, *Modelling Reusable Security Requirements based on an Ontology Framework.* Journal of Research and Practice in Information Technology (JRPIT), Vol 41(2), p. 119-133 Mayo 2009.
  Factor Impacto en el listado JCR 2009: 0,5 (posición 81/93, grupo Computer Sc. Soft. Eng.) (posición 100/116, grupo Computer Sc. Information Systems)

- Carlos Blanco, Joaquín Lasheras, Eduardo Fernandez-Medina, Rafael Valencia-García y Ambrosio Toval, *Basis for an integrated Security Ontology according to a systematic review of existing proposals.* Computer Standards and Interfaces, Vol 33 (4), p 372-388 Junio 2011
  Factor Impacto en el listado JCR 2009: 1,373 (posición 37/93, grupo Computer Sc. Soft. Eng.) (posición 18/49, grupo Computer Sc. Hw. & Arch.)

Aunque la normativa recomienda unas cinco mil palabras, se ha preferido elaborar y añadir a estos artículos una documentación más extensa en la que se incluye la introducción general donde se presentan estos artículos justificando su unidad temática, se resumen los objetivos de la investigación, la metodología utilizada, los resultados alcanzados y las conclusiones finales con el propósito de facilitar tanto a revisores, miembros del tribunal y a cualquier otro lector interesado la comprensión del trabajo realizado. La organización de esta documentación se puede consultar en la *sección 1.3.*

.

ÍNDICE

ÍNDICE DE FIGURAS

ÍNDICE DE TABLAS

Resumen

En esta tesis se presenta un **marco de trabajo de representación y reuso de requisitos de seguridad** que integra un conjunto de técnicas con el objetivo de mejorar la calidad del software desde las primeras fases del desarrollo con énfasis en la seguridad. Para ello integramos la Ingeniería del Software y la seguridad considerando a esta última como parte del proceso de desarrollo, en concreto ya desde la fase inicial con el uso de Ingeniería de Requisitos (IR), permitiendo conseguir sistemas software seguros.

Esta propuesta se construyó incrementalmente en dos fases: (1) en primer lugar se realizó una especialización en Ingeniería de Requisitos donde se ha trabajado en la definición de un **método de IR** basado en reutilización de requisitos, denominado SIREN (*SImple REuse of RequiremeNts*), en el que se han desarrollado y usado catálogos de requisitos en el ámbito de la privacidad, análisis de riesgos y sistemas teleoperados. Además se ha obtenido una **herramienta de soporte automatizado** para dicho método, que diera soporte a su vez a los **aspectos clave,** identificados por el doctorando, para realizar un proceso efectivo de reuso de requisitos. (2) Y ya finalmente, y basándose en la experiencia previa en IR, y el estudio del estado del arte, se decidió adoptar un aspecto formal en SIREN, haciendo uso de las **ontologías**, para dar soporte al marco de trabajo y representación y reuso de requisitos de seguridad basado en análisis de riesgos. En esta última fase ha sido clave considerar los beneficios de la **reutilización** dentro de la IR, y la **identificación de riesgos y amenazas** para el sistema, identificado como una de las fuentes principales de obtención de requisitos de seguridad por los estándares. Además para el aspecto formal hemos considerado el uso de **ontologías**, conociendo que su aplicación a la seguridad de los sistemas de información proporciona mejores mecanismos y conocimientos de las organizaciones para la predicción de problemas de seguridad. Esto nos ha llevado a obtener como uno de los principales resultados de la tesis **una ontología de requisitos de seguridad reutilizables basada en análisis de riesgos.**

Además, de este trabajo realizado y conociendo que para la comunidad científica es muy importante tener definidos formalmente los conceptos y relaciones que ellos comparten, concluimos la necesidad y el reto de conseguir una **ontología de seguridad general e integrada.** Para ello en esta tesis presentamos los **requisitos clave** y **primeros pasos** para obtenerla, en base a una comparación formal realizada con las propuestas más maduras en el campo de las ontologías de seguridad. Esta ontología general proporcionaría una base bien conocida en la que soportar el desarrollo de métodos, procesos y metodologías apropiadas y nos ayudaría a organizar nuestros conocimientos y a transmitirlos, a reportar incidentes de forma efectiva y a compartir datos e información a través de las organizaciones**, como soporte para capturar los elementos participantes y su semántica asociada** (las relaciones existentes entre ellos como trazabilidad, restricciones …).

Abstract

In this thesis, a **framework for representation and reuse of security requirements** is presented, which integrates a collection of techniques with the aim of improving software quality from the first stages of the development stressing in security. To achieve this goal we integrate Software Engineering and security considering the latter as part of the development process, particularly from the initial phase using Requirements Engineering (RE), in order to obtain secure software systems.

This proposal has been built incrementally in two phases: (1) first with the specialization in Requirements Engineering, through the definition of a **RE method** based on requirements reuse, called SIREN (*SImple REuse of RequiremeNts*), where requirements catalogues in the field of privacy, risk analysis and teleoperated systems have been developed and used. Moreover, an **automated tool support** for the method was developed, considering also the **key issues**, identified by the PhD candidate, to conduct an effective process of requirements reuse. (2) And finally, and based on previous experience in IR and study of the state of the art, it was decided to adopt a formal aspect in SIREN, using **ontologies** to support the framework for representation and reuse of security requirements based on risk analysis. In this last phase has been vital to consider the benefits of **reuse** within the RE, and **identifying risks and threats** to the system, considered as one of the main sources of obtaining security requirements in the standards. Moreover, to the formal aspect we have consider the use of **ontologies**, considering that their application to IT security provides us with better knowledge organization and mechanisms for the prediction of security problems. Thus obtaining as a main result of the thesis an **ontology of reusable security requirements based on risk analysis.**

Furthermore, from this previous work and knowing that for the scientific community is very important to have formally defined the concepts and relationships that are shared, we conclude with the need and the challenge of defining a **general and integrated security ontology.** Thus in this thesis the **key requirements** and **the first steps** to obtain it are presented, based on a formal comparison made for the more mature proposals in the field of security ontologies. This ontology would provide us a well known basis in order to support the development of methods, processes and methodologies appropriate and to help us to organize our knowledge and transmit them to report incidents effectively and to share data and information through organizations, **as support for capture the elements involved and their semantics** (relations between them as traceability, restrictions ...).

Agradecimientos

Cuando empecé mi carrera investigadora, o más bien, cuando tomé conciencia de cual era el objetivo primordial de la misma (la realización de la tesis), siempre me imaginé este momento en el cual redactaba los agradecimientos. Ahora que estoy aquí, se me viene a la cabeza mucha gente, muchos momentos, muchos lugares y tengo claro que no podré recogerlo todo en unas pocas palabras, pero bueno se intentará.

En primer lugar a la poquita familia que me queda, mi madre y mi hermana, puesto que han estado ahí desde el principio y conocen de buena mano el esfuerzo y los sacrificios realizados para llegar a este objetivo. Como parte ya también de la familia, a Mari luz que ha sido la que más cerca ha estado, la que ha tenido que soportarme más y de la que he recibido más apoyo, sobre todo en los últimos años con la finalización de la misma.

A Ambrosio Toval agradecerle que haya sido mi director de tesis, permitiéndome participar en el Grupo de Ingeniería del Software, y guiándome en este duro camino que ahora acaba, y con el que sólo puedo tener muestras de gratitud por sus innumerables consejos e ideas. En el DIS y en el Grupo de Ingeniería del Software, han sido unas cuantas las personas con las que he trabajado y compartido experiencias, en especial con Joaquín Nicolás y Begoña Moros, con los que empezó mi andadura, pasando por José Luís, José Sáez, Jesús García, Mercedes y tantos otros, a todos ellos muchas gracias.

Tampoco olvido a los que empezábamos en aquella sala del antiguo edificio de informática, Tono, Aurora, Jesu, Álvaro y José Antonio que nos visitaba a menudo. Después al trasladarnos al nuevo edificio conocí a nuevos compañeros con los que he compartido muchas gratas experiencias, incluso viajes, Javi, Espinazo, Jesús y Óscar, y ya en los últimos años Astrid y Javi Bermúdez. Pero sin duda, destacar a Miguel Ángel, Fernando y Fran que han sido, con diferencia, con los que he compartido más horas de viajes y trabajo, en definitiva toda mi vida investigadora, y que ya empezaron conmigo desde que estábamos en el antiguo edificio. También quiero destacar a mis compañeros del CENTIC, donde he acabado de dar los coletazos de esta tesis.

También agradecer en los momentos que he estado de estancia, tanto en Ciudad Real (UCLM) como en la Universidad de Trento, a todos aquellos con los que he colaborado, en especial, a Eduardo Fernández y Fabio Massaci que fueron mis supervisores, y a José Antonio Cruz y Nicola Zannone por ayudarme a establecerme cuando estuve allí. También agradecer a la gente con la que he participado en las publicaciones, Carlos Blanco, Pedro Sánchez, Juan Antonio Pastor, Francisco Ortiz, Bárbara Álvarez, Mario Piattini, Jesualdo Martínez y en especial a Rafa Valencia, sin cuya ayuda de todos esto tampoco habría sido posible.

No me gustaría olvidar a nadie pero seguro que lo he hecho así que sirva este párrafo final para dar las gracias a todos aquellos que alguna vez se interesaron por el estado de este trabajo o me dieron ánimos para seguir adelante. A todos ellos les doy las gracias por las experiencias vividas así como a los revisores y miembros del tribunal por el tiempo dedicado a la revisión de este trabajo.

Y mi último párrafo, sólo podía estar dedicado a una persona, a mi padre, a quien tengo presente en cada uno de los objetivos que voy cumpliendo en mi vida.

Extended Abstract

INTRODUCTION

Security has become a vital aspect nowadays, considering all its meanings: physical and operational security of Information Systems (IS) -*security*-, physical security to third parties -*safety*-, and security of personal data -*privacy*-. Also, in this regard, information security is considered a vital aspect for the development of IS [1] and the survival of enterprises [2]. Thus, as a consequence, terms as information assurance, security and privacy have moved from being considered by IS designers as narrow topics of interest to becoming critical issues of fundamental importance in our society [3]. All this concern has been reflected in the scientific community where we have noticed how in recent years the number of events and journals focused on security has increased dramatically, becoming the most popular keyword in computer science journals and proceedings identified in digital bibliography databases (such as DBLP). Furthermore, we identify the term 'security' as a keyword in the proposed research activities in the European Seventh Framework Programme (FP 2007-2013) or in European Reports as ITEA (EUREKA initiative) [4], where it is emphasized that security must be taken into account in all phases of software development lifecycle.

On the other hand, we are in a society increasingly dependent on Information Technology (IT) and software systems which mission is critical [5]. Despite ongoing advances in security technologies and software quality, new vulnerabilities continue to emerge according to the evolution of the technology. To avoid potential losses that organizations based on these IS have to confront, it is crucial to be properly secured from the beginning, integrating security in all the phases of the software development cycle [6-9]. To achieve this aim, it requires not only the implementation of security mechanisms ad-hoc, but also to ensure that the developments made from companies are secured, **in compliance with security best practices and legislation** from the early stages of the development [10].

However, traditionally, most development methods published by research groups and international standardization organizations consider the aspects related to security in the IS *a posteriori*. One reason for this is that traditional research areas of Software Engineering and Security Engineering work independently [11]. On the one hand, Software Engineering techniques and methodologies do not consider security as an important issue, (with some relevant exceptions) and, in the other hand, they often fail to provide precise enough semantics to support the analysis and design of security requirements [12].

In the early stages of software development is placed the **Requirements Engineering** (RE). In the literature we find proposals that sustain the need of consider security from this early stages, implying that security requirements are defined together with the requirements of the system [12-16]. Besides, we find some proposal to attempt it [17]. In this regard, RE has been identified as a growing area that has shown the ability to improve productivity and quality of software processes and products [18]. To combine RE with security has been identified as a burning issue in this field [19]. On the other hand, there is already a consensus for many years among software developers [20] on the many benefits of **reuse.** In fact, [21] shows empirically that the level of reuse determines the effectiveness of the improvements in productivity, quality and time-to-

market, and it concludes that greater benefits are obtained when reuse is considered during the early phases of the software development, for instance at the RE stage.

In this thesis, we present a **framework for representation and reuse of security requirements** through the integration of Software Engineering and security considering the latter as part of the development process, particularly from the initial phase using Requirements Engineering, in order to obtain secure software systems. We consider that taking into account security requirements with other functional and non-functional requirements throughout the development stages, will reduce the conflicts between functional and security requirements, avoiding or isolating them from the first phases. In this task, it is also vital to consider the benefits of **reuse** within the RE, and **identifying risks and threats** to the system, considered as one of the main sources of obtaining security requirements for the Standards.

In addition, we consider a **formal aspect** in the representation of security requirements, since a recent study of the state of the art in security requirements engineering has identified lack of scientific rigor in the current proposals [17]. To achieve this goal, the use of **ontologies** is considered, considering that their application to IT security provides us with better knowledge organization and mechanisms for the prediction of security problems [22]. We also believe that, within any scientific community, it is very important to have formally defined the concepts and relationships that are shared, where once again, the use of ontologies is considered appropriate [23]. Thus, several authors support the need to define a general ontology of security [23, 24], identifying it as an important challenge and research branch within the engineering community security [25]. This ontology would provide us a well known basis in order to support the development of methods, processes and methodologies appropriate and to help us to organize our knowledge and transmit them to report incidents effectively and to share data and information through organizations, **as support for capture the elements involved and their semantics** (relations between them as traceability, restrictions ...). As shown later in the contributions of the thesis, the design of an **ontology of reusable security requirements** is also tackled, and is considered one of the main results.

AIMS AND CONTRIBUTIONS - CONCLUSIONS

As it was previously mentioned, the global goal of this PhD is the design of a framework for representation and reuse of security requirements that integrates a set of techniques aimed at the improvement of software quality from the early stages of development through the integration of security. This general goal has followed a specific way that has concluded with the obtaining of some results by the PhD, divided mainly in two steps, and which are detailed below:

- With regard to the **specialization on RE**, we have worked firstly with the SIREN method [10, 15], a general-purpose and practical RE method based on requirements reuse, where we have developed catalogues of reusable requirements in fields as privacy, risk analysis and teleoperated systems, providing automated support too. What is more, we have gained experience through its creation and use, for instance, to audit privacy in systems o even to obtain a complete domain model of teleoperated systems for cleaning ship hulls.

- Finally, as a main result and based on this previous experience and the study of the state of art, we decided to **consider a formal aspect** to provide SIREN with more rigor. Thus, we have identified a framework for representation and reuse of security requirements based on risk analysis and the use of ontologies, concluding with the necessity of achieving a general and integrated security ontology, identifying also the first steps to obtain it.

A set of related contributions has been carried out in order to achieve these aims:

- Participation in the improvement of the **SIREN RE method** [26]. We have developed **reusable requirements catalogues** based on security standards and legislation, in particular, a privacy one related to the Data Protection Law and another one related to risk analysis, based on MAGERIT [27], the methodology of risk analysis and management of the Spanish government. Moreover, a number of **key issues** [28] were identified as **vital to conduct an effective process of requirements reuse.**

- As a consequence of working with the SIREN method, and to the identification of a lack of automated tools to give it support, we made **a prototype tool**, **SIRENTool** [29]. This tool should give support for the key issues identified and, to the best of our knowledge, was the first tool to support systematic reuse of requirements. Moreover, this tool was used and improved though an experience in a **technological transference project** where five companies were implied.

- We participated in two different experience related to the use of requirements catalogues, where also lessons learned were identified. In one of them we identified a list of **security requirements in the teleoperated systems realm** for cleaning ship hulls [30, 31]. It was the basis for identifying the elements of a **complete domain model** of teleoperated systems based on the use of features and quality attributes [32]. In the second one, we participated in an **audit method of personal data**, based on the use of the catalogue of privacy reusable requirements [33, 34]. This method was applied to audit four companies which handled personal data considered of high or special protection [35].

- A **framework for representation and reuse of security requirements** based on risk analysis has been defined, to which a risk analysis ontology and another one of requirements were defined, combining them to finally obtain one of reusable security requirements [36]. The aim of this new ontology was to improve security in information systems by detecting inconsistencies and achieving semantic processing in the requirements analysis [37].

- The state of the art in terms of security ontologies proposals [38-40] has been reviewed. This review concluded with the presentation of a formal comparison between the more mature proposals, which allowed us to identify a number of **key requirements to obtain a general and integrated security ontology**, presenting the first steps to obtain it [41], having been this general ontology identify as a need and a challenge by the scientific community.

PHD FRAME AND PUBLICATIONS

This section shows the projects in which this PhD candidate has been participated and the publications achieved.

Research group and research stays

This PhD has been mainly developed in the Software Engineering Research Group at the University of Murcia. The work has additionally been complemented with two periods of research at other universities. One of these took place in the DISI Security Group, in the Department of Information Engineering and Science at the University of Trento (Italy), under the supervision of Dr. Fabio Massacci (4 months, May-August 2006) and the other one took place in the ALARCOS Research Group, Information Systems and Technologies Department at the University of Castilla-La Mancha under the supervision of Dr Eduardo Fernández-Medina (3 months, March-May 2007).

This PhD has been financed by the Science and Innovation Ministry with a FPI grant for four years, framed in the project called PRESSURE "PREcise Software ModelS and ReqUirements Reuse", TIC2003-07804-C05-05 (developed between 2003 and 2006).

We also have participated in a project financed by the Ministry of Science and Technology, called DEDALO "Development of Quality Systems based on models and requirements", TIN2006-15175-C05-03 (developed between the years 2007 and 2009) and we nowadays participate in other project of the Spanish Ministry, PANGEA "Process for globAl requiremeNts enGinEering and quAlity", TIN2009-13718-C02-02, that will be developed in the period 2010-2012. Moreover, the PhD candidate has participated in other two projects financed by the Regional Ministry of Castilla-La Mancha: MELISA-GREIS (Global Requirements Engineering for Information Systems, PAC08-0142-335, 2008-2010) and DESERT (DEveloping Secure systEms through Requirements and Tools, PBC-05-012-3, 2005-2007).

Furthermore, we have participated in other thematic networks financed also by the Ministry of Science and Technology, which are related to the topic of security: RETISTIC (TIC2002-12487-E, 2004-2006) and RETISTRUT (TIN2006-26885-E, 2006-2008); and software quality: CALIPSO (TIN2005-24055-E, 2005-2008).

Finally, it is worth of note that we also have participated in a project of technological transference, financed by the Regional ministry of Murcia: GARTIC "Automated Management of Requirements based on Reuse for SME of the ICT sector" (2107ID0001, 2009-2010), where five companies of the Region of Murcia were implied.

Publications

Parts of the results of this work have been presented and discussed in various peer-review forums. The publications in which the author has been involved are listed below.

Journals indexed in the ISI Journal Citation Reports (JCR)

Specialization of RE

- Ambrosio Toval, Begoña Moros, Joaquín Nicolás and Joaquín Lasheras, *Eight Key Issues for an Effective Reuse-Based Requirements Process.* International Journal of Computer Systems Science and Engineering, Vol 23 (6) p. 373-385 November 2008.
  JCR Index in 2009: 0,222 (pos. 48/49, group Computer Sc. Hw. & Arch.) (pos. 90/92, group Computer Sc.Theory and Methods)

- Joaquín Nicolás, Joaquín Lasheras, Ambrosio Toval, Francisco J. Ortiz and Bárbara Álvarez, *An Integrated Domain Analysis Approach for Teleoperated Systems.* Requirements Engineering Journal (REJ), Vol 14(1) p. 27-46 January 2009.
  JCR Index in 2009: 0,931 (pos. 61/93, group Computer Sc. Soft. Eng.) (pos. 76/116, group Computer Sc. Information Systems)

- Miguel A. Martínez, Joaquín Lasheras, Ambrosio Toval, Eduardo Fernandez-Medina and Mario Piattini, *A Personal Data Audit Method through Requirements Engineering.* Computer Standards and Interfaces, Vol 32(4), p 166-178 June 2010.
  JCR Index in 2009: 1,373 (pos. 37/93, group Computer Sc. Soft. Eng.) (pos. 18/49, group Computer Sc. Hw. & Arch.)

Formal aspect

- Joaquín Lasheras, Rafael Valencia-García, Jesualdo Tomás Fernández-Breis and Ambrosio Toval, *Modelling Reusable Security Requirements based on an Ontology Framework.* Journal of Research and Practice in Information Technology (JRPIT), Vol 41(2), p. 119-133 May 2009.
  JCR Index in 2009: 0,5 (pos. 81/93, group Computer Sc. Soft. Eng.) (pos. 100/116, group Computer Sc. Information Systems)

- Carlos Blanco, Joaquín Lasheras, Eduardo Fernandez-Medina, Rafael Valencia-García and Ambrosio Toval, *Basis for an integrated Security Ontology according to a systematic review of existing proposals.* Computer Standards and Interfaces. Vol 33 (4), p 372-388 Junio 2011
  JCR Index in 2009: 1,373 (pos. 37/93, group Computer Sc. Soft. Eng.) (pos. 18/49, group Computer Sc. Hw. & Arch.)

International Conferences/Workshops

- J. Nicolás, J. Lasheras, A. Toval, F.J. Ortíz, B. Alvarez, *A Collaborative Learning Experience in Modelling the Requirements of Teleoperated Systems for Ship Hull Maintenance*. Workshop on Learning Software Organizations and Requirements Engineering (LSO + RE 2006). Hannover, Germany. March 2006.

- M.A. Martínez, J. Lasheras, A. Toval, M. Piattini, *An Audit Method of Personal Data Based on Requirements Engineering*. The 4th International Workshop on Security in Information Systems (WOSIS-2006), in the 8th International Conference on Enterprise Information Systems (ICEIS'06). Paphos, Cyprus, May 2006. ISBN: 972-8865-52-X. CORE C.

- C. Blanco, J. Lasheras, R. Valencia-García, E. Fernández-Medina, A. Toval, M. Piattini, *A Systematic Review and Comparison of Security Ontologies*. ARES International Conference on Availability, Reliability and Security, Barcelona, Spain. March 2008, IEEE computer society ISBN: 978-0-7695-3102-1. CORE B

- J. Lasheras, R. Valencia-García, J.T. Fernández-Breis, A. Toval, *An Ontology-Based Framework for Modelling Security Requirements.* The 6th International Workshop on Security in Information Systems (WOSIS-2008), in the 10th International Conference on Enterprise Information Systems (ICEIS'08). Barcelona, Spain. July 2008. ISBN: 978-989-8111-44-9. CORE C

Chapters in Spanish Books

- C. Blanco, D.G. Rosado, D. Mellado, A. Rodríguez, C. Gutierrez, J. Lasheras, E. Fernández-Medina, A. Toval, J. Trujillo y M. Piattini, *Capítulo 15: Seguridad en Ingeniería del Software* en *Calidad del producto y proceso software.* RA-MA p. 339 – 375, 2010 ISBN: 8478979611.

Technical Reports

- C. Blanco, J. Lasheras, R. Valencia-García, E. Fernández-Medina, A. Toval, M. Piattini. Security Ontologies: a sistematic review and comparison. Technical Report – UCLM-TSI-003 (July 2008), University of Castilla-La Mancha

- J. Nicolás, B Moros, J. Lasheras, A. Toval. *SIREN (SImple REuse of software requirements), a general-purpose RE method based on requirements reuse*. Technical Report – UMU-TR DIS 1-2009, University of Murcia.

Spanish and Iberoamerican Conferences

- M.A. Martínez, J. Lasheras, A. Toval, M. Piattini, *Aportaciones de la Ingeniería de Requisitos en un proceso de auditoría de datos personales*. IV Congreso Internacional de Auditoría y Seguridad de la Información (CIASI'05). Madrid, Spain. December 2005. ISBN: 84-689-5752-6.

- C. Blanco, J. Lasheras, R. Valencia-García, E. Fernández-Medina, A. Toval, M. Piattini, *Revisión sistemática y comparación de ontologías en el marco de la seguridad*. IV congreso iberoamericano de seguridad en informática CIBSI 2007. Mar de plata (Argentina). November 2007, ISBN: 978-950-623-043-2.

- J. Lasheras, A. Toval, J. Nicolás, B. Moros, *Soporte automatizado a la reutilización de requisitos*. VIII Conference on software Engineering and Databases (JISBD 2003). Alicante. November 2003. I.S.B.N: 84-688-3836-5.

- J. Lasheras, J. Nicolás, A. Toval, B. Moros, *Hacia un Modelo del Dominio de los Sistemas Teleoperados a través de una extensión de SIREN*. II Jornadas de trabajo DYNAMICA (DYNamic and Aspect-Oriented Modeling for Integrated Component-based Architectures) Málaga, November 2004.

- B. Álvarez, P. Sánchez, J.A. Pastor, A. Toval, J. Lasheras, *Experiencia, Estrategias y Retos en la Incorporación de Requisitos de Seguridad en el Sistema EFTCoR .* IX Conference on software Engineering and Databases (JISBD 2004) Málaga, November 2004. I.S.B.N: 84-688-8983-0.

- J. Nicolás, J. Lasheras, A. Toval, B. Moros, P.Sanchez, B. Alvarez, *Ingeniería de Requisitos Basada en Reutilización: una propuesta de Aplicación a los Sistemas Teleoperados para Limpieza de Cascos de Buques.* III Jornadas de trabajo DYNAMICA (DYNamic and Aspect-Oriented Modeling for Integrated Component-based Architectures) Ciudad Real, April 2005.

- M.A. Martínez, J. Lasheras, J. Nicolás, A. Toval, *Aplicación de un Proceso de Auditoría de Datos Personales Basado en el Método SIREN,*III Jornadas trabajo DYNAMICA (DYNamic and Aspect-Oriented Modeling for Integrated Component-based Architectures) Ciudad Real, April 2005.

- M.A. Martínez, J. Lasheras, J. Nicolás, A. Toval, *Un proceso de auditoría de datos personales basado en Ingeniería de Requisitos.* I Simposio sobre seguridad informática [SSI'2005], in I Congreso Español de Informática [CEDI 2005].Granada September 2005 I.S.B.N:84-9732-447-1.

- J. Nicolás, J. Lasheras, A. Toval, F.J. Ortíz, B. Alvarez, *Una experiencia de modelado de los sistemas teleoperados para limpieza de cascos de buques mediante características y casos de uso genéricos*. IV Jornadas de trabajo DYNAMICA (DYNamic and Aspect-Oriented Modeling for Integrated Component-based Architectures). Murcia, November 2005.

- C. Blanco, J. Lasheras, R. Valencia-García, E. Fernández-Medina, A. Toval, M. Piattini, *Ontologías de seguridad: revisión sistemática y comparativa,* II Simposio sobre seguridad informática [SSI'2007], in II Congreso Español de Informática [CEDI'2007], Zaragoza, September 2007. I.S.B.N: 978-84-9732-607-0.

1 CAPÍTULO 1. INTRODUCCIÓN

1.1 Introducción

La **seguridad**, en todas sus acepciones, de seguridad física y operacional de los Sistemas de Información (SI) *–security–*, seguridad física a terceros *–safety–*, y seguridad de datos personales *–privacy–*, es un tema de vital importancia en la actualidad. La seguridad de la información es considerada ahora como un aspecto clave para el desarrollo de los sistemas de información [1] y la supervivencia de las empresas [2], por lo que aspectos como la confiabilidad, seguridad y privacidad de la información han pasado de ser meros puntos de interés para los diseñadores de SI y se han convertido en cuestiones críticas y de vital importancia para la sociedad [3]. Todo esto se ha visto reflejado en la comunidad científica donde hemos comprobado cómo en los últimos años se ha incrementado drásticamente el número de eventos y revistas centrados en la seguridad, llegando hasta el punto de ser la palabra clave más popular en revistas y congresos de informática identificada en bases digitales de datos bibliográficos (como DBLP). Además, encontramos la seguridad como elemento clave en las actividades de investigación propuestas en el VII Programa Marco Europeo (I+D 2007-2013) o en informes europeos como ITEA (iniciativa EUREKA) [4], donde se remarca que debe ser tenida en cuenta en todas las fases del desarrollo de software.

Por otra parte, nos encontramos en una sociedad cada vez más dependiente de las Tecnologías de la Información (TI) y de sistemas software cuya misión es crítica [5]. Sin embargo, y a pesar de los avances en tecnologías de seguridad y calidad software, nuevas vulnerabilidades continúan surgiendo conforme dichas tecnologías evolucionan. Para evitar las potenciales pérdidas a las que se enfrentan las organizaciones que se sustentan en estos SI, resulta crucial que éstos sean asegurados apropiadamente desde el principio, integrando la seguridad durante todo el ciclo de desarrollo del software [6-9]. Para ello no sólo es necesaria la implantación de mecanismos de seguridad ad-hoc, sino también velar porque los desarrollos que se hagan desde las empresas sean seguros, **cumpliendo con la legislación y las normas de seguridad** desde las primeras etapas del desarrollo [10]. Incluso si hablamos de ahorro y eficacia en una organización, y aún sabiendo que son cuestiones relativas al depender del coste propio y la implantación inteligente de la seguridad, lo que si podemos asegurar es que este ahorro y eficacia será siempre muy superior si los requisitos y especificaciones de seguridad se incorporan en el propio desarrollo de los sistemas y los servicios de información [27].

Hasta ahora, la mayoría de los métodos de desarrollo aportados por los distintos grupos de investigación y organismos internacionales de estandarización suponen una aplicación *a posteriori* (una vez construidos) en los SI de los aspectos relacionados con la seguridad, implicando que los mecanismos de seguridad son fijados en diseños preexistentes. Una de las razones de esta situación es el hecho de que tradicionalmente las áreas de investigación de Ingeniería del Software e Ingeniería de Seguridad trabajan independientemente [11]. Por una parte, las metodologías y técnicas de Ingeniería del Software no han considerado la seguridad como un aspecto importante y normalmente éstas fallan al proporcionar una semántica suficientemente precisa para soportar el análisis y diseño de requisitos seguros [12].

Por otra parte, si queremos aplicar la seguridad desde las primeras fases de desarrollo software, nos debemos mover en el ámbito de la **Ingeniería de Requisitos** (IR). En la literatura encontramos ya trabajos que sostienen dicha necesidad, implicando que los requisitos de seguridad sean definidos junto con los requisitos del sistema [10, 12-16], incluso ya encontramos propuestas que lo consideran [17]. En este sentido, la IR se ha identificado como un área creciente, que ha demostrado la capacidad para mejorar la productividad y la calidad de los procesos y productos software [18], y su combinación con la seguridad ha sido identificada como un tema candente dentro de la propia IR [19]. Por otra parte, existe ya un consenso desde hace muchos años entre los desarrolladores de software [20] sobre los muchos beneficios de la **reutilización**, mayor incluso cuanto más sea el nivel de abstracción, es decir, teniéndola en cuenta ya desde las primeras fases de desarrollo del software. De hecho, en [21] se demuestra empíricamente que el nivel de reutilización determina la eficacia de las mejoras en la productividad, calidad y tiempo de lanzamiento al mercado, y concluyen que mayores beneficios se obtienen cuando se considera la reutilización durante las primeras fases del desarrollo de software, es decir, en la fase de IR.

En esta tesis pretendemos crear un **marco de trabajo de representación y reuso de requisitos** de seguridad, mediante la integración de la Ingeniería del Software y la seguridad considerando a esta última como parte del proceso de desarrollo, en concreto ya desde la fase inicial con el uso de Ingeniería de Requisitos, permitiendo conseguir sistemas software seguros. Consideramos que teniendo en cuenta la seguridad junto a los requisitos funcionales y otros requisitos no funcionales a lo largo de las etapas de desarrollo, ayudará a limitar los casos de conflictos entre requisitos funcionales y de seguridad, evitándolos o aislándolos desde las primeras fases. En esta tarea será clave considerar además los beneficios de la **reutilización** dentro de la IR, y la **identificación de riesgos y amenazas** para el sistema, identificado como una de las fuentes principales de obtención de requisitos de seguridad por los estándares.

En particular vamos a considerar un **aspecto formal** dentro de la representación de requisitos de seguridad, ya que según un estudio reciente del estado del arte en Ingeniería de Requisitos de seguridad se ha identificado falta de rigor científico en las propuesta actuales [17]. Para ello vamos a considerar el uso de **ontologías**, conociendo que su aplicación a la seguridad de los sistemas de información proporciona mejores mecanismos y conocimientos de la organizaciones para la predicción de problemas de seguridad [22]. Además, consideramos que dentro de cualquier comunidad científica es muy importante tener definidos formalmente los conceptos y relaciones que se comparten, siendo una vez más el uso de ontologías un elemento adecuado para ello [23]. De este modo, varios autores apoyan la necesidad de la definición de una ontología general de seguridad [23, 24], identificándola como un área de investigación importante y un reto dentro de la comunidad de la ingeniería de seguridad [25]. Esta ontología proporcionaría una base bien conocida en la que soportar el desarrollo de métodos, procesos y metodologías apropiadas y nos ayudaría a organizar nuestros conocimientos y a transmitirlos, a reportar incidentes de forma efectiva y a compartir datos e información a través de las organizaciones**, como soporte para capturar los elementos participantes y su semántica asociada** (las relaciones existentes entre ellos como trazabilidad, restricciones …). Por todo ello, en esta tesis se abordará el diseño de una **ontología de requisitos de seguridad reutilizable**, que es considerado uno de los resultados principales de la misma.

1.2 Marco de la Tesis

A continuación se detalla el entorno de trabajo en el que se ha desarrollado la presente tesis doctoral, resaltándose tanto los Grupos de Investigación en los que se ha desarrollado como los proyectos de I+D en los que se ha enmarcado.

1.2.1 Grupo de Investigación

La tesis ha sido desarrollada principalmente en el Grupo de Investigación en Ingeniería del Software de la Universidad de Murcia (www.um.es/giisw) y complementada con dos estancias de investigación. Una realizada en el departamento de Ciencias e Ingeniería Informática en la Universidad de Trento (Italia), bajo la supervisión del Doctor Fabio Massacci (cuatro meses desde mayo a agosto de 2006), y la otra realizada en el grupo de investigación ALARCOS, en el departamento de Sistemas de Información y Tecnología adscrito a la Universidad de Castilla la Mancha, bajo la supervisión del Doctor Eduardo Fernández-Medina (tres meses desde marzo a mayo de 2007).

El trabajo realizado ha sido financiado por el ministerio de Ciencia e Innovación a través de una beca FPI durante cuatro años y por diferentes proyectos de investigación que son comentados a continuación.

1.2.2 Proyectos de Investigación Aplicada y Transferencia Tecnológica

Los proyectos de investigación en los que se enmarcó este trabajo han sido los siguientes:

- PANGEA (Process for globAl requiremeNts enGinEering and quAlity, TIN2009- 13718-C02-02), integrado en el proyecto PEGASO (Processes for the Improvement of Global Software Development) en el que participan la Universidad de Castilla-La Mancha y la Universidad de Murcia. El proyecto está financiado por el Ministerio de Ciencia y Tecnología comenzando a finales de 2009 y acabando su desarrollo en el año 2012.

- DEDALO (Desarrollo de sistEmas de caliDad bAsado en modeLos y requisitOs, TIN 2006-15175-C05-03). Integrado en el proyecto META (Models, Environments, Transformations and Applications), un proyecto coordinado entre cuatro universidades españolas (Universidad Politécnica de Valencia, Universidad de Castilla-La Mancha, Universidad Politécnica de Cartagena y Universidad de Murcia) y el European Software Institute (ESI). El proyecto fue financiado por el Ministerio de Ciencia y Tecnología, comenzando en 2006 y acabando su desarrollo a finales de 2009.

- MELISA-GREIS (Metodología para el desarrollo global del Software, PAC08-0142- 335) en el que participan las Universidades de Castilla-La Mancha y Murcia. El proyecto está financiado por la Consejería de Educación y Ciencia, en el marco del Plan Regional de Investigación Científica, Desarrollo Tecnológico e Innovación de Castilla-La Mancha, siendo desarrollado entre los años 2008 y 2010.

- GARTIC (Gestión Automatizada de Requisitos basada en Reutilización para PYMES del sector TIC), un proyecto de transferencia tecnológica firmado por el Grupo de Investigación en Ingeniería del Software de la Universidad de Murcia y el Centro Tecnológico de las Tecnologías de la Información y las Comunicaciones (CENTIC) de la Región de Murcia, que contó con la participación de cinco empresas regionales a las que se formó para la utilización de técnicas de Ingeniería de Requisitos en sus proyectos de desarrollo.

- DESERT (DEveloping Secure systEms through Requirements and Tools), con número de expediente PBC-05-012-3, financiado por la Consejería de Educación y Ciencia, en el marco del Plan Regional de Investigación Científica, Desarrollo Tecnológico e Innovación de Castilla-La Mancha. En este proyecto participaron las Universidades de Castilla-La Mancha, Alicante y Murcia y tuvo una duración de 3 años (2005 hasta 2007).

- PRESSURE (PREcise Software modelS and reqUirements REuse, TIC2003-07804-C05-05), integrado en el Proyecto DYNAMICA (DYNamic and Aspect-Oriented Modeling for Integrated Component-based Architectures). En el proyecto participaron cinco universidades españolas (Universidad Politécnica de Valencia, Universidad de Castilla-La Mancha, Universidad Carlos III, Universidad Politécnica de Cartagena y Universidad de Murcia) y fue financiado por el Ministerio de Ciencia y Tecnología entre los años 2003 y 2006.

REDES TEMÁTICAS

El doctorando también ha formado parte de varias redes temáticas relacionadas con el contenido de la tesis doctoral. Los detalles de estas acciones se muestran a continuación:

- Red CALIPSO (Calidad de Producto y Proceso Software, TIN2005-24055-E), en la que participan diez universidades españolas, cinco universidades iberoamericanas y cinco empresas, con un total de más de cien investigadores y tres años de duración. El objetivo principal de esta red fue servir como punto de encuentro de todos ellos para trabajar en la mejora de la calidad de los productos y procesos software. La red, dirigida por la Dra. Coral Calero (Universidad de Castilla-La Mancha), se desarrolló entre los años 2005 y 2007 y posteriormente se extendió durante el año 2008.

- Red RETISTRUST (Red temática de Investigación en el campo de la Seguridad y confianza para los Sistemas de Información en una Sociedad Conectada, TIN2006- 26885-E), en la que participaron diferentes universidades españolas e iberoamericanas, así como la Excelentísima Diputación de Ciudad Real. Esta red mantenía los objetivos de su predecesora (la red RETISTIC, ver a continuación) y se desarrolló en el año 2008 bajo la dirección del Dr. Eduardo Fernández-Medina (Universidad de Castilla-La Mancha).

- Red RETISTIC (Red temática española de investigación en el campo de la seguridad de las tecnologías de información), como Acción Especial TIC2002-12487-E del Ministerio de Ciencia y Tecnología P. Nacional I+D+I, con duración de dos años: 2004-2005. En la red temática participaron diferentes universidades españolas e iberoamericanas, así como la Excelentísima

Diputación de Ciudad Real y la Asociación de Auditores y Auditoría y Control de Sistemas y Tecnologías de la Información y la Comunicaciones (ASIA). El objetivo principal era consolidar, unificar y divulgar los conocimientos sobre la seguridad y asentar la confianza en la tecnología. Se desarrolló bajo la dirección del Dr. Eduardo Fernández-Medina (Universidad de Castilla-La Mancha).

1.3 Organización de la Tesis

Este documento se ha organizado en los siguientes capítulos:

En el *capítulo 1* se ha mostrado una visión general acerca del marco de desarrollo de la tesis.

En el *capítulo 2* se muestra el estado del arte en aquellas disciplinas relacionadas con esta tesis doctoral, como es la Ingeniería de Requisitos y el uso dentro de la misma de reutilización e ingeniería ontológica para dar soporte a cuestiones de seguridad.

El *capítulo 3* detalla la primera fase de realización de la tesis en la cual el doctorando se especializó en la Ingeniería de Requisitos, concretamente, con sus experiencias de utilización del método SIREN.

En el *capítulo 4* se mostrará el soporte formal al método SIREN, a través del uso de una ontología de requisitos de seguridad reutilizable. Además se mostrarán los primeros pasos para obtener una ontología de seguridad general e integrada, aspecto considerado como básico y un reto por la comunidad científica.

En el *capítulo 5* se analizarán los resultados obtenidos, las conclusiones y líneas de trabajo futuro.

2 CAPÍTULO 2. ESTADO DEL ARTE

2.1 Introducción

En este capítulo se muestra el estado del arte en aquellas disciplinas relacionadas con esta tesis doctoral, como es la Ingeniería de Requisitos y el uso dentro de la misma de reutilización (*sección 2.2*) o la ingeniería ontológica para dar soporte a cuestiones de seguridad.

Descrita en el *capítulo 1* la importancia y la necesidad de la seguridad, y que su implantación es cada vez más reconocida por las organizaciones, en este capítulo mostramos los trabajos relacionados con ella y que han surgido para solucionar las distintas carencias de seguridad en el proceso software. Así, en una primera parte son presentados distintos marcos de trabajo de seguridad (*sección 2.3*), destacando los relacionados con el análisis de riesgos (*sección 2.4*), para después analizar las aproximaciones actuales de ingeniería de requisitos de seguridad (*sección 2.5*) y el uso de ingeniería ontológica dentro del ámbito de la seguridad (*sección 2.6*).

2.2 Ingeniería de Requisitos y Reutilización

Según la definición de Kotonya y Sommerville [42], el término Ingeniería de Requisitos implica el uso de procedimientos repetibles y sistemáticos para asegurar la obtención de un conjunto de requisitos relevante, completo, consistente y fácilmente comprensible y analizable por parte de los diferentes actores implicados en el desarrollo del sistema. El objetivo básico de la IR es la especificación de qué debe hacer un sistema y de las restricciones de diseño que condicionan cómo ha de ser implementado, cuyo objetivo final es el desarrollo de un software correcto y acorde con los requisitos de un cliente.

La Ingeniería de Requisitos es una de la áreas de la Ingeniería del Software que está más en auge y además es considerada como crítica para el éxito de un proyecto de desarrollo software [43]. Prueba de ello es los diferentes estudios que demuestran que la gestión inadecuada de los requisitos es la causa probada de multitud de fallos y fracasos en proyectos software [44-47]. Mientras, que por el contrario, una gestión adecuada de la misma implica una mejora en la productividad y la calidad de los procesos y productos software [18, 43].

Desde nuestro punto de vista, la Ingeniería de Requisitos debe tener en cuenta el reuso. Según [48] “la reutilización del software es la única aproximación realista para conseguir las ganancias de calidad y productividad que la industria del software necesita”. Desde mediados de los noventa la reutilización de especificaciones o de requisitos se ha venido postulando por parte de numerosos autores como una vía prometedora (y menos explorada que la reutilización de código o de diseños) para ayudar a conseguir las ganancias de calidad y productividad que el desarrollo de software necesita. Además, existe un consenso [20] sobre los muchos beneficios del reuso, mayor incluso cuanto más sea el nivel de abstracción, es decir, no limitándose a reutilizar código, sino también diseños y especificaciones [49, 50]. Además Robertson y Robertson [18] postulan que comenzar con un conjunto de requisitos que han sido especificados para otros proyectos o dominios sirve para mejorar la precisión de la especificación de requisitos y para reducir el tiempo para elaborar esta especificación. En esta línea, Rine y Nada [21], han mostrado empíricamente que el nivel de reutilización determina la efectividad de las mejoras en productividad, calidad y tiempo

de desarrollo, concluyendo que se obtienen beneficios mayores al considerar la reutilización durante los procesos iniciales del ciclo de vida de desarrollo del software. Por todo ello, cuando a finales del siglo pasado y principios de este examinaba el camino por recorrer en la investigación en IR, ya se establecía que la reutilización de modelos de requisitos constituía uno de los principales desafíos de la IR [51].

Con esta premisa y partiendo de la hipótesis de que con la introducción de un marco de reutilización de requisitos se puede mejorar la calidad del producto final y la productividad del desarrollo, en el Grupo de Investigación de Ingeniería del Software de la Universidad de Murcia, al cual el doctorando pertenece, se definió un método de reutilización de requisitos, denominado SIREN (SImple REuse of RequiremeNts) [10, 15], que describiremos en un capítulo posterior (*sección 3.2*) y del cual se obtuvieron resultados para esta tesis.

2.3 Marcos de Trabajo de Seguridad

Como ya hemos destacado previamente en el *capítulo 1*, la seguridad de los sistemas informáticos es una actividad que preocupa crecientemente a las empresas. Cada vez más un mayor número de organizaciones establecen una serie de criterios, normas y salvaguardas para protegerse de ataques externos y malfuncionamientos. Sin embargo, estas actividades suelen abordarse sólo desde una perspectiva tecnológica, sin tener en cuenta al conjunto de proceso de negocio afectados y la necesidad de volver a ponerlos en marcha después de un evento negativo. En este punto pretendemos destacar alguna de las iniciativas más destacables en cuanto a marcos de trabajo de la seguridad de la información por parte de la industria y los cuerpos de estandarización:

- **Familia de Estándares ISO/IEC 27000** [52], formada por seis estándares internacionales, que engloban la definición de los SGSI (Sistemas de Gestión de Seguridad de la Información), abarcando los requisitos de los sistemas de gestión de la seguridad, la gestión del riesgo, métricas y medidas, guías de implantación, glosario de términos y mejora continua. Merece la pena destacar el **ISO 27001** [53], equivalente a la segunda parte de la norma BS7799 del BSI (*British Standards Institution*), que especifica los requisitos necesarios para establecer, implantar, mantener y mejorar un SGSI según el "Círculo de Deming": PDCA - acrónimo de *Plan, Do, Check, Act*. Además, la norma **ISO 27002** será la adopción por parte de la ISO de la primera parte de la norma ISO 17799 2005 [54], dedicada a definir un código de buenas prácticas para la gestión de la seguridad por las organizaciones.
- **ISO/IEC 15408:2009 Common Criteria Framework (CCF)** [55], estándar que propone un marco de trabajo común para la realización de las pruebas y evaluación de la seguridad, elaborado en 1996 entre los países de Canadá, Francia, Alemania, Holanda, Reino Unido y USA. Estos criterios comunes constituyen un punto de encuentro y de consenso científico, técnico, comercial y gubernamental, para la evaluación y certificación de la seguridad de las tecnologías de la información. Más de veinte países del mundo han adoptado los criterios Comunes y trece de ellos -incluido España, a través del Consejo Superior de Informática- firmaron un Acuerdo de Reconocimiento mutuo (ARM) de los certificados expedidos por aquellos países que disponen de Esquema de Evaluación y Certificación reconocido en dicho acuerdo.

- **ISO/IEC 21827:2008 SSE-CMM** [56] (*Systems Security Engineering Capability Maturity Model*) en el cual se define un modelo de proceso de madurez para mejorar y evaluar la capacidad de la ingeniería de la seguridad de una organización.
- **IEEE P1074-2005** [57], estándar internacional que trata la gestión del software y sus ciclos de vida. Centra su aproximación en aplicar una apropiada priorización de la seguridad en los proyectos software así como durante la construcción de los controles de seguridad en los productos. Para conseguirlo, eleva la prioridad de la seguridad mediante la incorporación de un pequeño número de actividades de seguridad claves en el ciclo de vida del desarrollo del software [58].

De otras propuestas no estandarizadas, pero aceptadas por la industria software, cabe destacar:

- **COBIT** (*Control Objectives for Information and related Technology*) [59], creado por la ISACA (*Information Systems Audit and Control Association*), y el ITGI (*IT Governance Institute*) en 1992, incluye un conjunto de mejores prácticas para el manejo de información. Está enfocado para el uso de auditores de sistemas de información y profesionales de la seguridad y este año está prevista su nueva versión 5.0.
- Otras iniciativas como proyectos europeos como **RE-TRUST** http://re-trust.dit.unitn.it, (Sept 2006-2009) relacionado con la seguridad en la gestión de derechos digital o **SERENITY** http://www.serenity-forum.org (Enero 2006-2009) sobre ingeniería de sistemas para seguridad y fiabilidad.

Estos estándares y normas existentes proveen un conjunto de cláusulas que se evalúan en modo cualitativo y son altamente genéricas. Por lo tanto tienen dos carencias fundamentales: no ofrecen una guía lo suficientemente detallada para su implantación y carecen de indicaciones, de herramientas y de mecanismos que faciliten su implementación. Sólo permiten una evaluación de tipo SI/NO sin proveer un camino para la mejora.

Lo que sí se deduce tras el estudio de estos marcos de trabajo de la seguridad, es que el ámbito que abarca la seguridad en los SI es muy grande. Como se dice en [60] “es muy difícil desarrollar una metodología que satisfaga todos los criterios que comprenden las restricciones de seguridad en términos de disponibilidad, integridad y confidencialidad. Si esta metodología fuera desarrollada, su complejidad evitaría su éxito”. Por lo tanto nos obliga a buscar una solución acotada a algún entorno y un enfoque en el cual las técnicas y modelos más aceptados, así como estándares, sean usados. Considerando también que lo desarrollado sea extensible a otras técnicas, integrando los aspectos de seguridad necesarios [60]. Es por ello que en esta tesis nosotros vamos a abordar la seguridad acotada desde el punto de vista del análisis de riesgos y la IR como describiremos en las siguientes secciones (*2.4* y *2.5*).

2.4 Análisis de Riesgos en el Desarrollo de Software Seguro

Después del estudio de varias técnicas, metodologías y estándares relacionados con la seguridad y con la IR, descritas en las otras secciones del documento (*2.3, 2.5 y 2.6*), se identificó que el análisis de riesgos, con su consecuente identificación de activos del sistema, eran consideradas tareas clave en el proceso de establecimiento de la seguridad y paso necesario para su gestión [27]. Prueba de esta importancia son la existencia de estándares internacionales dedicados solamente al análisis de riesgos, como el [61] (*Information security management systems-Guidelines for information security risk management)* que es la base de la actual ISO 27005. Este estándar proporciona guía y soporte para la implementación del marco de trabajo de análisis de riesgos del estándar más global ISO 27001 [53] y es suficientemente genérico para su uso de pequeñas, medias y grandes organizaciones, aunque no es exhaustivo y podría necesitar adaptarse con otras metodologías. Por otra parte, en otras aproximaciones como [62] o [63] se identificó como necesario procesar el análisis de riesgos en las primeras fases del desarrollo de software, tal que esta identificación de riesgos ayude a generar los requisitos de seguridad, permitiendo además decidir cuanto esfuerzo se debe invertir en seguridad, indicándonos, además, el grado con el cual un recurso debe ser protegido.

En la literatura podemos encontrar varias metodologías específicas de análisis de riesgos que han sido adoptadas como estándares *de facto* en sus respectivos países:

- **CRAMM** (CCTA *Risk Analysis and Management Method*) [64], es el método de Análisis y Gestión del Riesgo del CTTA (*Carmarthenshire College of Technology and Art*) adoptado por el gobierno del Reino Unido y que es conforme con el ISO 27001.
- **MAGERIT** (Método de Análisis y GEstión del RIesgo del MinisTerio de Administraciones Públicas [27]), es el método de análisis y gestión del riesgo definido por la Administración Pública Española, actualizada a la versión 2.0 y conforme al estándar ISO 15408, *Common Criteria*.
- **OCTAVE** (*Operationally Critical Threat, Asset and Vulnerability Evaluation* [65, 66]). Método de análisis de riesgos desarrollado por el Carnegie Mellon – Software Engineering Institute- ampliamente aceptado en USA.

Estas metodologías están orientadas a tomar decisiones desde la perspectiva del negocio, permitiendo gestionar requisitos y procedimientos técnicos y operacionales, y además comparten conceptos comunes como son los de: activo, amenaza, vulnerabilidad, salvaguarda y riesgo. Sin embargo, estas metodologías dedicadas al análisis de riesgos son aplicadas una vez que el diseño arquitectónico ha sido desarrollado y por lo tanto tienen algunas limitaciones [9]. 1) Permiten solo una aproximación *a posteriori* de seguridad para TI, teniendo como consecuencia inconsistencias entre los requisitos de seguridad y las necesidades de seguridad de la empresa, implicando un desajuste entre los métodos de seguridad y el desarrollo de los Sistemas de Información [67]. 2) Los métodos de gestión de riesgos son considerados como semiformales y suelen ofrecer un “buen” proceso para identificación de riesgos, aunque sus resultados son informes en lenguaje natural que no favorecen la automatización, evolución, monitoreo y trazabilidad de la gestión del riesgo. 3) Estos métodos normalmente son complejos y requieren de un largo tiempo de aprendizaje [68], 4) Por último, no consideran aspectos de reutilización de sus resultados.

Por todo ello, en esta tesis enfocamos la obtención de un marco de trabajo con seguridad basado en identificación de activos y análisis de riesgos que considere la aplicación *a priori* de cuestiones de seguridad y que tenga un enfoque más sencillo y automatizado que las propuestas existentes, basado en el método SIREN, en consecuencia, en el uso de métodos de IR y reuso. Como elemento de partida inicial, se consideró la metodología MAGERIT, con la cual el grupo de investigación al que el doctorando pertenece ya tiene experiencia previa [10], a través de un contrato con la comunidad autónoma de Murcia[1], considerando la versión 1.0 de dicha metodología, que a su vez se basa en el estándar ISO 15408 o los *Common Criteria* [55]. Además en un futuro no descartamos que el trabajo que realicemos se pueda extender a otras fases identificadas para un SGSI (Realización del documento de la política de seguridad, monitorización constante y registro de todas las incidencias, realización de auditorias internas…) y que pueda tener en cuenta otras metodologías de análisis de riesgos.

2.5 Ingeniería de Requisitos de Seguridad

La motivación esencial de esta tesis se puede resumir por lo tanto en la incorporación explícita de la seguridad en el desarrollo de sistemas de información, desde el proceso de IR, de forma que se mejore la calidad del SI. En este sentido, Sommerville [50] afirma que un buen número de funcionamientos anómalos relacionados con la seguridad del sistema se deben a errores de especificación más que a errores de diseño. Además tenemos en cuenta que la inclusión de la seguridad está identificada como un aspecto clave y emergente dentro de la IR [19].

La Ingeniería de Requisitos de seguridad, como subsidiaria de la IR, consistirá en el proceso de elicitación, especificación y análisis de los requisitos de seguridad de un sistema de información [69]. Los requisitos de seguridad en Ingeniería del Software se suelen considerar como un tipo de requisitos no funcionales [70, 71] y se suelen definir como *restricciones o limitaciones* con las que el sistema debe operar [50]. Existen otras definiciones similares a ésta, como [72] donde los definen como *restricciones o limitaciones* en los servicios o en el comportamiento de un sistema.

La hipótesis de esta tesis será por lo tanto, que la consideración de la seguridad junto a los requisitos funcionales y otros requisitos no funcionales a lo largo de las etapas de desarrollo, ayudará a limitar los casos de conflictos entre requisitos funcionales y de seguridad, evitándolos o aislándolos desde las primeras fases de desarrollo. De esta forma se prevé que el desarrollo de software será más eficaz respecto a los costes y tendrá como resultado diseños más robustos [73], en general cobrando mayor importancia en aquellos sistemas complejos, distribuidos, evolutivos y reutilizables.

En concreto, en el estándar ISO 27002 [54] se identifica que es esencial que la Organización identifique sus requisitos de seguridad, e identifica tres fuentes principales:

- La primera fuente procede de la valoración de los riesgos de la Organización. Con ella se identifican las amenazas a los activos, se evalúa la vulnerabilidad y la probabilidad de su ocurrencia y se estima su posible impacto. En otras

[1] Contrato CARMMA 3142-UMU. CARM (Comunidad Autónoma de Murcia) y el grupo de ingeniería del software de la Universidad de Murcia. Aplicación de MAGERIT en la oficina regional de sistemas de información y telecomunicaciones. 1999.

palabras, el riesgo, entendido como la probabilidad de que una amenaza en particular ataque una determinada vulnerabilidad en la empresa [27].
- La segunda fuente es el conjunto de requisitos legales, estatutarios y regulatorios que debería satisfacer la Organización, sus socios comerciales, los contratistas y los proveedores de servicios.
- La tercera fuente está formada por los principios, objetivos y requisitos que forman parte del tratamiento de la información que la Organización ha desarrollado para apoyar sus operaciones.

Nosotros en esta tesis nos vamos a centrar en la primera opción, que una vez más muestra la importancia del análisis de riesgos para gestionar los requisitos de seguridad de un sistema, y tendremos también en cuenta la segunda fuente, al identificar requisitos de fuentes legales, en concreto procedentes de leyes de privacidad de datos personales.

En cuanto al estado del arte, aunque la seguridad de la información ha sido ampliamente estudiada y en los últimos años se ha experimentado un espectacular crecimiento de propuestas relacionadas con ella (incluyendo estándares de seguridad - *sección 2.3*), encontramos trabajos [9, 16] que muestran que los lenguajes de modelado fallan al incluir manejo especializado para los requisitos de seguridad. Además, no existe todavía una metodología completa y coherente para asegurar la seguridad en la construcción de sistemas de propósito general, si bien es cierto que sí que hay una investigación muy activa que ha conseguido resultados interesantes para objetivos particulares [17].

Los principales métodos y técnicas que han ido surgiendo con el propósito de reducir las carencias de seguridad existentes se pueden consultar en [17], donde se presenta una revisión sistemática en el ámbito de la ingeniería de requisitos de seguridad. En este punto queremos destacar que la mayoría presentan técnicas concretas para determinados ámbitos, dentro de las cuales merece la pena destacar las siguientes: técnicas de casos de uso de seguridad [74], diagramas de barrera [75], centradas en el modelado de procesos de negocio [76], modelado de amenazas [77], historias de abuso en el dominio de la ingeniería de requisitos ágil [78], o casos de mal uso conducidos por amenazas y el riesgo [79] donde se modela el comportamiento indeseable para un sistema. Existen también propuestas que modelan seguridad basándose en el uso de MDA *(Model Driven Architecture*), unas centrándose en características de confidencialidad y control de acceso [80] y otras en el uso de profiles UML: UmlSEC [81, 82] y Secure UML (utilizando OCL) [83]. Este último además aplicado a un desarrollo dirigido por requisitos en el entorno de los sistemas críticos de seguridad (requisitos fiabilidad) [84].

Otros trabajos que también merecen la pena destacar y que además proponen una base para una metodología de seguridad son:

- Metodología Secure Tropos [85-88] que esta basado en el desarrollo de software orientado a agentes y enfocado a la identificación de requisitos de seguridad relacionados con la confianza. Tiene una herramienta de soporte (ST-TOOL).
- La metodología SQUARE [62] del Carnegie Mellon, que propone utilizar casos de uso y de mal uso, y propone un proceso de 9 pasos secuenciales para elicitar, categorizar y priorizar requisitos de seguridad. Herramienta T-SQUARE.
- La metodología SREP [89-91], que propone utilizar varias técnicas UMLSec, casos de mal uso, casos de uso de seguridad y XML. Propone un proceso basado en UML, *Unified Process*, iterativo e incremental, dirigido por el riesgo y está

enfocado a líneas de producto. Tiene también una herramienta de soporte automatizado al proceso de gestión de requisitos seguridad (SREPTOOL).

Con respecto al enfoque del marco de trabajo presentado en esta tesis, basado en reutilización y análisis de riesgos, de las metodologías descritas arriba, Secure Tropos, no identifica el análisis de riesgos como elemento básico, ya que se centra más en requisitos de confianza. Además no considera específicamente cuestiones de reutilización, al igual que ocurre con SQUARE. Por último hay que decir que ninguna de ellas trata específicamente con requisitos textuales y ni considera específicamente las cuestiones de trazabilidad que una gestión de requisitos basada en reuso implica (lo veremos en detalle en el siguiente *capítulo 3, secciones 3.2 y 3.5*). Además en la propuesta descrita en esta tesis vamos a considerar un soporte formal mediante el uso de ontologías, ya que del estudio del estado del arte recogido en [17] identificamos falta de rigor en la definición de los trabajos de seguridad, que ninguna de estas 3 metodologías ofrece (Secure Tropos, SQUARE o SREP).

2.6 Ingeniería Ontológica Aplicada a la Seguridad

Una ontología es una especificación explícita de una conceptualización [92] que nos permite una formalización en la representación del conocimiento que puede dar lugar a la realización de inferencias, y que pueden utilizarse para la representación de modelos reutilizables en dominios distintos. Estos modelos nos permitirán una mejor representación, organización, razonamiento, reutilización y compartición [93-95], reduciendo de esta forma el esfuerzo necesario para desarrollar SI, permitiendo la fusión de la comprensión humana y computacional. Es por ello que existe un consenso evidente en que disponer de una ontología del dominio de aplicación de un sistema software, o de los procesos para su diseño y construcción, es una ayuda importante para evitar errores y problemas en todas las fases del ciclo de vida del producto software: desde el análisis de requisitos inicial (facilitando la interacción analista-cliente) hasta la etapa de mantenimiento (más fácil compresión de las peticiones de modificación, mejor compresión del sistema mantenido, etc.).

Por otra parte, existen beneficios de usar la tecnología ontológica referidos a la seguridad en los sistemas de información establecidos de acuerdo a tres principales propiedades [22]: organiza y reduce la diversidad de los ítems de una lista de propiedades, provee de mecanismos para prevenir los problemas de seguridad y ofrece una gran modularidad. Además, ya hay trabajos donde se muestra la utilidad del uso de ontologías de seguridad que permiten compartir y reutilizar conocimiento sobre seguridad de diversas fuentes de una forma interoperable, facilitando, además, la gestión de la seguridad separando los requisitos de seguridad de su implementación técnica [24]. Por todo ello, consideramos que la combinación de ambas era idónea para ese **soporte formal** que buscábamos para nuestro marco de trabajo presentado en esta tesis.

En cuanto al estado del arte, aquí mostramos un breve resumen de lo que se puede ver en detalle en el informe técnico [96] y en el artículo JCR adjunto (ver *Anexo I publicaciones, sección 7.5*) donde se muestra una revisión sistemática de trabajos relacionados con la ingeniería ontológica aplicada a la seguridad. Este estudio muestra que ya existen varias aproximaciones donde se combina la ingeniería ontología y la seguridad, remarcando su auge en los últimos años. En esta sección destacamos las más interesantes. Algunas de éstas están centradas en la Web semántica, en concreto en el desarrollo de una ontología enfocada en desarrollar anotaciones seguras para web [3] y

otras se centran en generar ontologías en dominios concretos: en el dominio de e-goverment [97], en el dominio de la fiabilidad [95], en el dominio de modelos de ataque [98], en dominios de confianza [99] y en otros aspectos funcionales sobre seguridad (como mecanismos, protocolos, objetivos, algoritmos y credenciales, en varios niveles de detalle, hasta siete ontologías distintas) [100, 101]. Sin embargo todas éstas no prestan importancia al análisis de riesgos. Por el contrario, merece la pena destacar los trabajos de [24, 102, 103] donde los autores presentan una ontología de análisis de riesgos basada en CRAMM, usando un marco de trabajo de gestión de seguridad para SI y en [104-107] donde se basan en análisis de riesgos de bajo coste y análisis de amenazas, permitiendo a pequeñas y medianas empresas implementar una propuesta integral de seguridad en Tecnología de la Información (TI). Por último, en [108] se propone un marco de trabajo basado en ontologías y requisitos, particularmente adaptado para el departamento de defensa, "*Information Technology Security Certification and Accreditation Process (DITSCAP)*".

Sin embargo, debemos concluir que ninguna de las ontologías descritas considera específicamente aspectos de reutilización y representación de requisitos de seguridad textuales, frente a lo ya destacado en [95], donde se muestra la utilidad de usar ontologías para representar aspectos software y de requisitos textuales. Todo esto nos llevó a diseñar una ontología de requisitos de seguridad reutilizable, donde se debían tener en cuenta los aspectos de seguridad basados en análisis de riesgos, y los aspectos sobre la identificación de requisitos textuales y su reutilización, propios de un método de IR. Además, está ontología tendrá que capturar no solo estos elementos, sino también su semántica asociada, como son las relaciones existentes entre ellos, por ejemplo las relaciones de trazabilidad entre requisitos, o las relaciones entre elementos de análisis de riesgos como las existentes entre los activos de una organización y sus amenazas. Esto se ve en detalle en el *capítulo 4, sección 4.2* donde se presenta el marco de trabajo.

Por otra parte, y como consecuencia de la realización del estudio del estado del arte, identificamos que dentro de cualquier comunidad científica es muy importante tener definidos formalmente los conceptos y relaciones que se comparten, siendo las ontologías el elemento adecuado para ello [23]. Esto es algo que podemos comprobar observando el gran número de ontologías que han surgido en el campo de la informática, quizás también por el papel que juegan en la nueva generación web y al boom que ha sufrido esta a finales del siglo XX, enfocadas la mayoría de ellas en la web semántica. De este modo varios autores apoyan la necesidad de la definición de una ontología general de seguridad [23, 24], identificándola como un área de investigación importante y un reto dentro de la comunidad de la ingeniería de seguridad [25]. Este trabajo no debe partir de cero, ya que la representación de todos los conceptos de seguridad sería inviable, sino partir de la unión de propuestas definidas anteriormente, dotando a la comunidad de las metodologías y herramientas adecuadas que permitan actualizar la ontología reflejando la evolución y aparición de los nuevos conceptos de seguridad [25]. Por todo ello en esta tesis planteamos la necesidad de conseguir una ontología de seguridad general e integrada (*capítulo 4, sección 4.3*), dando las guías para obtenerla, la cual nos permitirá tener definidos formalmente los conceptos y relaciones que se comparten. Esta ontología proporcionaría una base bien conocida en la que soportar el desarrollo de métodos, procesos y metodologías apropiadas.

3 CAPÍTULO 3. ESPECIALIZACIÓN EN IR

3.1 Introducción

En este capítulo se muestra la evolución del doctorando en su especialización en IR. Primero con su participación en la definición y mejora del método de IR SIREN (*sección 3.2*), luego explicando sus experiencias con la aplicación de catálogos de requisitos reutilizables (*sección 3.3*), para acabar con el desarrollo de la herramienta de soporte automatizado al método y su experiencia de transferencia tecnológica a cinco empresas de la región de Murcia (*sección 3.4*). Finalmente se muestran las conclusiones de dicha especialización en IR (*sección 3.5*), y como éstas implicaron abordar un soporte formal para el método SIREN.

3.2 El Método SIREN

SIREN (SImple REuse of RequiremeNts) [10] es un método práctico y sencillo, basado en reuso y estándares de IR y el uso de requisitos textuales. El objetivo de su diseño era que este método fuera fácilmente aplicable, incluso por parte de una organización de desarrollo con un proceso de IR poco maduro. El método incorpora un modelo de proceso en espiral, guías de aplicación de dicho proceso, basadas en una jerarquía de documentos estándares de requisitos, un repositorio reutilizable de requisitos organizado por catálogos y una herramienta CARE (*Computer-Aided Requirements Engineering*) de soporte, llamada *SIRENTool*. SIREN está concebido con la intención de aportar la consideración explícita de la reutilización de requisitos en cualquier método de IR, más que para ser aplicado como un método de ingeniería de requisitos general. No obstante, SIREN también puede ser utilizado como un método de IR básico en una organización de desarrollo en la que no se haya definido un método de IR, y para ello incorpora un conjunto básico de guías generales de IR. En el *informe técnico* [109], donde el doctorando trabajó, se puede ver en detalle el método.

Este método general se definió a través de la experiencia de definición de requisitos para reuso en dominios de seguridad, ya que los requisitos de seguridad y privacidad han de estar alineados con estándares, leyes y regulaciones existentes que están bien definidos, pero donde sin embargo se utiliza un lenguaje legal complejo y algunas veces ambiguo [110]. Por esto se trabajó en la elaboración de catálogos de requisitos reutilizables basados en normas y legislación de seguridad, en concreto, se trabajó en uno de **privacidad** [15] relacionado con la LOPD [111] - Ley Orgánica de protección de datos – y el RMS [112]- Reglamento de Medidas de Seguridad -, y otro sobre **análisis de riesgos** [10], basado en MAGERIT [27], la metodología de análisis y gestión de riesgos de las administraciones públicas. En SIREN podemos distinguir principalmente dos enfoques: el **desarrollo con reutilización** y el **desarrollo para reutilización** *(figura 3.1).* Durante esta primera fase de definición del método, se tuvo en cuenta sobre todo el enfoque de **desarrollo con reutilización** *(figura 3.2)*, donde partimos ya de la existencia de los catálogos de requisitos reutilizables desarrollados. En la *sección 3.4* veremos como el método evolucionó también para dar soporte al **desarrollo para reutilización,** es decir, para dar soporte a la creación de los catálogos reutilizables y ya no sólo en el dominio de la seguridad.

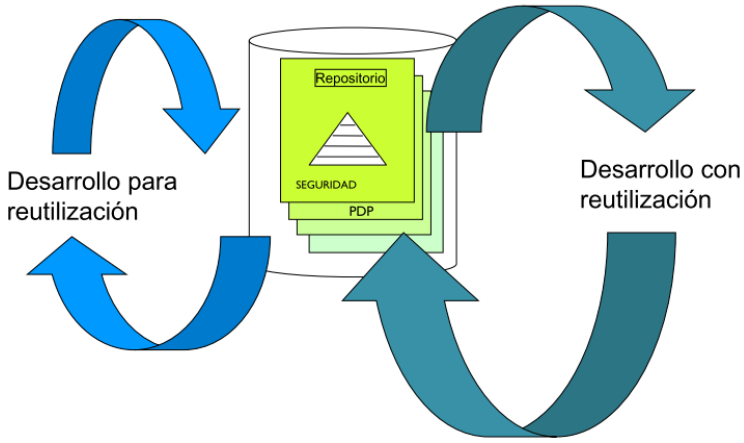

Fig. 3.1. Dos enfoques del método SIREN: desarrollo para reutilización y con reutilización

Más detalles sobre el método SIREN, en su parte de **desarrollo con reutilización,** se pueden ver en el artículo JCR adjunto en esta memoria (*sección 7.1*) y del que extraemos la *figura 3.2*, para ilustrar el modelo de proceso.

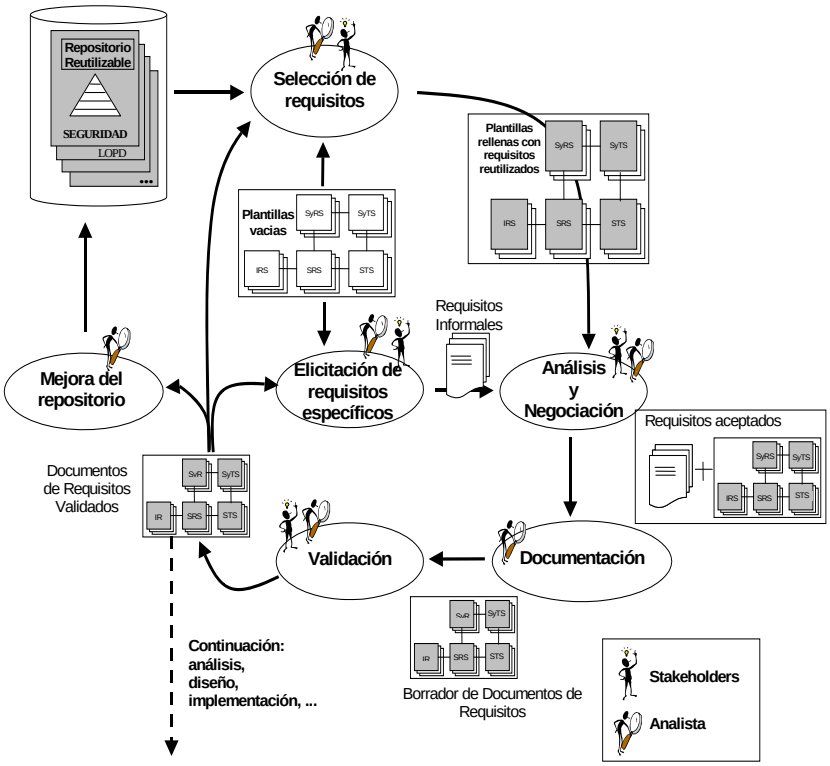

Fig. 3.2. Modelo de proceso SIREN, desarrollo con reutilización de requisitos.

Como se puede comprobar, el modelo de proceso de desarrollo con reutilización es una adaptación del modelo de proceso en espiral propuesto en [42], al que se le añaden principalmente dos nuevas actividades (*figura 3.2):*

- *Selección de Requisitos*. La aproximación al reuso implica proporcionar al ingeniero de requisitos las plantillas de los documentos de especificación rellenas con requisitos reutilizables del repositorio. El ingeniero de requisitos

junto con el resto de *stakeholders*[2] seleccionarán los requisitos del repositorio que sean adecuados para la aplicación concreta que se está desarrollando. Para el reuso de éstos se tendrá que tener en cuenta el conjunto de **atributos (metainformación)** asociado a cada requisito, las **relaciones de trazabilidad** con otros requisitos (**padre-hijo, inclusiva y exclusiva**), así como si es un **requisito parametrizado** que necesitará ser instanciado para el proyecto actual.

- *Mejora del Repositorio*. El repositorio no se debe considerar como un producto final, estático, sino como un producto en continua evolución, incluyendo nuevos requisitos reutilizables y mejorando la calidad de los existentes.

En las siguientes secciones mostramos la evolución del método SIREN y como el doctorando ha participado en dicha evolución, comenzando por el uso en experiencias de aplicación de los catálogos de requisitos reutilizables (*sección 3.3*), y acabando con la definición del soporte automatizado, tanto para el enfoque de **desarrollo con reuso** como **para reuso** *(sección 3.4)*.

3.3 Experiencias de Aplicación de Catálogos de Requisitos

3.3.1 Trabajo con los Sistemas Teleoperados

En esta experiencia se participó en la identificación de un catálogo de requisitos de seguridad en el entorno de los Sistemas Teleoperados para la limpieza de cascos de buques [30, 31]. Este catálogo sirvió de base para la identificación de los elementos de un modelo completo del domino de los Sistemas Teleoperados, basándose en el uso de características y atributos de calidad, identificando a su vez las lecciones aprendidas de la experiencia [32]. Ésta se encuentra relatada en el artículo JCR incluido en el *Anexo I, sección 7.2*. Fue de esta experiencia cuando se asumió la necesidad de contar con un paso previo para crear los catálogos de requisitos reutilizables en SIREN (**desarrollo para reutilización**, *sección 3.4*), ya que al no tener un dominio basado en normas y leyes, como sucedía en las experiencias previas con la privacidad y la seguridad, requería un tratamiento distinto.

3.3.2 Trabajo con Privacidad: Catálogo LOPD

Para esta experiencia se partió como base del catálogo de requisitos de privacidad reutilizable realizado en [15]. La experiencia consistió en aplicar un método de auditoria de datos de carácter personal, basado en dicho catálogo, el cuál previamente tuvo que ser **actualizado** con las nuevas versiones de las leyes de privacidad [33, 34]. Este método se aplicó para auditar a cuatro empresas que manejaban datos de carácter personal considerados de nivel alto o especialmente protegidos, y se identificaron las lecciones aprendidas de la experiencia [35]. Ésta se encuentra relatada en el artículo JCR incluido en el *Anexo I, sección 7.3*. Esta experiencia de aplicación a un caso real de estudio de un catálogo de requisitos reutilizable, nos mostró que en los requisitos era habitual encontrarnos **ambigüedades** propias de estar elaborados en lenguaje textual, dándonos a ver la necesidad de encontrar un **soporte formal** que las suprimiera (como podía ser el enfoque ontológico); y por otra parte, nos hizo ser conscientes de la necesidad, siempre que nos movamos en el ámbito de la seguridad, de tener en cuenta la **integración con otras normas o estándares**, como en este caso supuso tener en cuenta aspectos de COBIT o de la ISO 27001, así como tener el contenido **actualizado**.

[2] Son todas aquellas personas que son afectadas por el sistema, las cuáles tienen una influencia directa o indirecta con los requisitos del sistema.

3.4 Soporte Automatizado al método SIREN

Como consecuencia del trabajo con SIREN, y ante la identificación de falta de herramientas automatizadas que le dieran soporte, se creo un prototipo de herramienta, *SIRENTool* [29]. Esta herramienta en su momento, al menos hasta donde conocíamos, fue la primera que daba soporte sistemático a la reutilización de requisitos. Además, se identificaron una serie de aspectos clave (*key issues*) [28] señalados por el doctorando como básicos para llevar a cabo de forma eficiente la reutilización de requisitos, y que por lo tanto la herramienta debía considerar. Este trabajo queda recogido en el *artículo JCR, Anexo I sección 7.1*. Aquí extraemos, a modo de resumen y en forma de tabla (*tabla 3.1*), estos aspectos clave y cómo el método SIREN los considera.

ASPECTOS CLAVE PARA LA REUTILIZACIÓN	SOPORTE DE SIREN
1.Organización de los requisitos en estructuras reutilizables	Repositorio de Catálogos de requisitos reutilizables
2. Motor de búsquedas de requisitos reutilizables	Búsqueda normal y avanzada (*SIRENTool*)
3. Selección y reutilización de requisitos con diferentes niveles de granularidad	Reutilización por requisito, agrupados por relación de traza y catálogos completos (uno, varios o todos)
4. Reutilización de los atributos de los requisitos	Conjunto mínimo de atributos definido y modificables al reutilizarse (*SIRENTool*)
5. Reutilización de las relaciones de trazabilidad entre requisitos	Trazas padre-hijo, inclusivas y exclusivas gestionadas (*SIRENTool*)
6. Gestión de requisitos parametrizados	Posibilidad de crear y reutilizar requisitos parametrizados (*SIRENTool*)
7. Mejora del repositorio de los requisitos	Modificación de los requisitos de los catálogos, con posibilidad de seguir sus fuentes
8. Soporte automatizado al método	*SIRENTool*

Tabla 3.1. Aspectos clave para la reutilización de requisitos y soporte en SIREN

Seguidamente se trabajó en un proyecto de **transferencia tecnológica** con empresas de la Región de Murcia**,** donde además de probar la herramienta desarrollada, se trabajó en la elaboración de los pasos para desarrollar catálogos de requisitos reutilizables, lo que denominamos el **desarrollo para reuso** (*figura 3.1*) y que ya identificábamos en la experiencia de los sistemas teleoperados (*sección 3.3.1)*. Este proyecto se denominó GARTIC (*Gestión Automatizada de Requisitos basada en Reutilización para Pymes del Sector TIC*) y tuvo dos resultados principales (1) la redefinición del método SIREN de acuerdo con la experiencia obtenida durante su implantación en las cinco Pymes participantes, y (2) la evolución de la herramienta SIRENTool, para dar soporte a las cuestiones del **desarrollo para reuso** principalmente. Toda esta retroalimentación y mejora de la herramienta y el método quedaron recogidas en el *informe técnico* [109].

El objetivo general de GARTIC consistió, por tanto, en la introducción en las empresas participantes de un método de IR basado en reutilización (técnicas, proceso, guías y herramienta CARE), que ayudó a la mejora de los índices de calidad y productividad en el desarrollo de software correcto en Pymes de desarrollo de software. Se trató de dar a conocer a las empresas participantes el impacto que puede tener en el desarrollo de software la aplicación de la IR, y en particular de SIREN, para definir un método de IR adaptado a las necesidades de las empresas que esté basado en reutilización. La experiencia supuso poner en práctica **cinco proyectos piloto** donde cada empresa elaboró un catálogo de requisitos reutilizable en diversos dominios como son:

- Los gestores de contenidos.
- Aplicaciones Web de comercio electrónico.
- Aplicaciones de Terminales Punto de Venta (TPV).
- Traductores genéricos de cualquier tipo de dato fuente a dato destino.
- Gestión de fincas y de semilleros.

3.5 Conclusiones

Ya desde los primeros trabajos en SIREN [10, 15] se identificaban necesidades o mejoras de **formalizar** el método, y aunque algunas fueron resueltas con las experiencias anteriores, todavía quedaba trabajo por hacer para llegar a ese soporte formal, que permitiera cerrar el marco de trabajo general para reutilización y reuso de requisitos de seguridad.

El primer punto fue considerar, que si queríamos conseguir un marco de trabajo para requisitos de seguridad general debíamos considerar la integración de los catálogos existentes (privacidad y análisis de riesgos) en uno sólo de seguridad, lo que implicaba, por un lado **actualizar** dichos catálogos y por otro considerar las exclusividades que podían aparecer entre ellos. Así, mientras que el de privacidad era actualizado en la experiencia del caso de la auditoría (descrita en *sección 3.3.2*), el segundo de ellos basado en análisis de riesgos requería revisión. Por ello, se debía realizar la mejora y actualización del catálogo de requisitos de seguridad basado en análisis de riesgos, realizado previamente de acuerdo con la versión MAGERIT 1.0, a la nueva versión MAGERIT 2.0. En esta nueva versión los principales cambios se describían con nuevas estructuras para los documentos donde se encontraban definidos los catálogos de activos, amenazas y salvaguardas del sistema (se verá en detalle en la *sección 4.2.1*).

Además, debemos considerar, tal y como identificamos en el *capítulo 2 de estado del arte*, *sección 2.3 y* de la experiencia del trabajo con privacidad (*sección 3.3.2*), que el marco de trabajo de seguridad (al ser un ámbito muy grande el que abarca ésta) debía poder ser extensible, por ejemplo considerando otros estándares de seguridad *(sección 2.3)*, métodos de análisis de riesgos *(sección 2.4)* o técnicas/metodologías de seguridad *(sección 2.5).*

Por último, debíamos plantearnos un **soporte formal** que nos permitiera gestionar la imprecisión de manejar requisitos textuales derivados de estándares (problema ya identificado en la experiencia con privacidad - *sección 3.3.2)*. Para esto propusimos el uso de la ingeniería ontológica (cuya valía fue demostrada en la *sección 2.6*) para poder capturar la semántica asociada a cada uno de los elementos, es decir sus relaciones y restricciones que puedan aparecer. Así se tenían que tener en cuenta dos aspectos:

1) Los de modelar requisitos y considerar reutilización -> **Ontología de requisitos**
   - Relaciones de traza entre requisitos (padre e hijo, inclusiva, exclusiva).
   - Relaciones entre la metainformación asociada (conjunto mínimo de atributos).

Ambas relaciones consideradas aspectos claves para la reutilización (*tabla 3.1*).

2) Los de modelar requisitos de seguridad basados en análisis de riesgos-> **Ontología de análisis de riesgos**
   - Relaciones y restricciones entre elementos propios del análisis de riesgos: como son los activos, amenazas, y las salvaguardas. Por ejemplo: una amenaza o una salvaguarda asociada a determinados activos.

Este proceso de soporte formal se describe en detalle en el siguiente capítulo, complementado por el artículo incluido en el *anexo I publicaciones JCR, sección 7.4.*

4 CAPÍTULO 4. SOPORTE FORMAL AL MÉTODO SIREN

4.1 Introducción

En este capítulo se define el marco de trabajo para representación y reuso de requisitos de seguridad basado en análisis de riesgos, para el cual se definió una ontología de análisis de riesgos y otra de requisitos, combinándose para formar finalmente una de requisitos de seguridad reutilizables (*sección 4.2*). Este marco proporcionará el soporte formal al método SIREN. Además, como consecuencia de este trabajo realizado y tras la realización de un estudio del estado del arte para el ámbito de las ontologías de seguridad, se identificó la necesidad y el reto de abordar la definición de una ontología general y unificada de seguridad, señalando cuales son los primeros pasos para su obtención (*sección 4.3*). En la *sección 4.4* se acabarán mostrando las conclusiones de estos trabajos realizados.

4.2 Marco de Trabajo Basado en Ontologías para el Modelado de Requisitos de Seguridad Reutilizables

En este apartado se muestran los pasos seguidos para la obtención de la ontología de requisitos de seguridad reutilizables, cuya experiencia se muestra relatada en el artículo que se encuentra en el *anexo I publicaciones en JCR, sección 7.4* [37].

En primer lugar se elaboró una revisión del estado del arte en cuanto a trabajos de ontologías de seguridad [38, 39, 96] cuyas conclusiones se analizaron en la *sección 2.6*. De este estudio del estado del arte y de las necesidades descritas en la *sección 3.5* para el soporte formal de SIREN, se decidió la tarea de abordar el diseño de una ontología de requisitos de seguridad reutilizable con el objetivo de mejorar la seguridad en los sistemas de información detectando inconsistencias y consiguiendo procesamiento semántico en el análisis de requisitos. Esta ontología debía permitir una representación formal (inteligible para una máquina) de los requisitos, su metainformación asociada, sus relaciones y las restricciones y axiomas y reglas derivadas de su uso (relaciones semánticas).

Este marco de trabajo combina una ontología de análisis de riesgos (*sección 4.2.2*), diseñada a partir del catálogo de requisitos de seguridad identificado en SIREN y basado en el método de análisis de riesgos MAGERIT (*sección 4.2.1*), junto con otra de requisitos, basada también en SIREN y estándares de IR (*sección 4.2.3*), dando lugar a la ontología de requisitos de seguridad (*sección 4.2.4*). Este marco de trabajo sirvió para validar dicho catálogo de requisitos de seguridad identificado en SIREN y tiene un proceso de aplicación (*sección 4.2.5*).

Para asegurar la consistencia del trabajo realizado, las ontologías fueron desarrolladas siguiendo un método formal para comparar y diseñar ontologías [113], del cual se han seguido las recomendaciones sobre cómo los conceptos, relaciones, taxonomías y axiomas debían ser identificados y descritos. Por otra parte, de nuestra experiencia y del trabajo del estudio del estado del arte realizado [39], identificamos que la combinación de trabajo con el lenguaje para diseñar e implementar ontologías OWL (*Ontology Web Language*) y el uso de la herramienta *Protégé* (http://protege.stanford.edu/) era un entorno adecuado para la creación de las mismas, el cual nos iba a permitir

escalabilidad, conocimiento distribuido y además de capacidad de inferencia de conocimiento haciendo uso de razonadores, gracias a que OWL es un lenguaje formal y tiene implícito un modelo matemático, siendo, además, la actual recomendación del W3C para el intercambio de contenido semántico en la Web.

4.2.1 Elemento Base: Catálogo MAGERIT

El marco de trabajo partía del catálogo de requisitos de seguridad basado en MAGERIT que se había definido en la fase anterior de especialización en IR [10] y el cual tuvo que ser actualizado a la nueva versión 2.0. MAGERIT define un método para investigar los riesgos que soportan los sistemas de información, y recomendar las medidas apropiadas que deberían adoptarse para controlar estos riesgos. MAGERIT define un **catálogo de elementos** donde se especifica qué se necesita proteger (activo), de qué hay que protegerlo (amenaza) y cómo hacerlo (salvaguarda). Este catálogo de elementos tiene dos objetivos principales:

- Facilitar la labor de las personas que acometen el proyecto, ya que se les ofrece ítems estándar a los que puedan adscribirse rápidamente, centrándose en lo específico del sistema objeto del análisis.
- Homogeneizar los resultados de los análisis, promoviendo una terminología y unos criterios que permitan comparar e incluso integrar análisis realizados por diferentes equipos.

Sin embargo, como vimos en la *sección 2.4*, MAGERIT es un método que se utiliza una vez que la arquitectura ha sido diseñada permitiendo solamente una aplicación *a posteriori* de la seguridad. Además, las relaciones entre los elementos implicados en este **catálogo de elementos**, están sólo explicadas por tablas y texto en lenguaje natural. En concreto, en el catálogo aparecen cinco tipos de elementos, que serán formalizados, junto a sus relaciones, en la ontología de riesgos (*siguiente sección 4.2.2*):

- **Activos**. Todo lo que proporciona algún valor a una organización. MAGERIT distingue nueve tipos de activos:
  - *Servicios*. Función que satisface una necesidad de los usuarios.
  - *Aplicaciones (Software)*. Tareas que han sido automatizadas para su empeño por un equipo informático.
  - *Datos*. Elementos de información que, de forma singular o agrupados de alguna forma, representan el conocimiento que se tiene de algo.
  - *Redes de comunicaciones*. Medios de transporte que llevan datos de un sitio a otro.
  - *Soportes de información*. Dispositivos físicos que permiten almacenar información de forma permanente o, al menos, durante largos periodos de tiempo.
  - *Equipamiento auxiliar*. Equipos que sirven de soporte a los sistemas de información sin estar directamente relacionados con datos.
  - *Equipos informáticos (Hardware)*. Bienes materiales, físicos, destinados a soportar directa e indirectamente los servicios que presta la organización.
  - *Instalaciones*. Lugares donde se hospedan los SI y comunicaciones.
  - *Personal.* Personas relacionadas con los sistemas de información.

- **Amenazas.** Posibles amenazas asociadas a los activos de un SI:
  - *Desastres naturales*. Sucesos que pueden ocurrir sin intervención de los seres humanos como causa directa o indirecta.
  - *De origen industrial*. Sucesos que pueden ocurrir de forma accidental, derivados de la actividad humana de tipo industrial.
  - *Errores y fallos no intencionados*. Fallos no intencionales causados por las personas.
  - Ataques intencionados. Fallos deliberados producidos por las personas.

- **Dimensiones de valoración.** Son las características o atributos que hacen valioso un activo:
  - *Disponibilidad.* Asegurar que los usuarios autorizados tienen acceso cuando lo requieran a la información y sus activos asociados.
  - *Integridad de datos.* Garantía de la exactitud y completitud de la información y los métodos de su procesamiento.
  - *Confidencialidad de los datos*. Asegurar que la información es accesible sólo para aquellos autorizados a tener acceso.
  - *Autenticidad de los usuarios del servicio*. Asegurar identidad u origen de usuarios.
  - *Autenticidad del origen de los datos*. Asegurar identidad u origen de los datos.
  - *Trazabilidad del servicio*. Asegurar en todo momento quién hizo qué y en qué momento.
  - *Trazabilidad de los datos*. En todo momento se podrá determinar quién hizo qué y en qué momento con los datos.

- **Criterios de valoración.** Escala de valores para todas las dimensiones. Nos indica cómo de importante es un activo para el sistema, debiendo proteger los activos más valorados.
- **Salvaguardas.** Todo lo que permite afrontar las amenazas. Por ejemplo, control de acceso, copias de seguridad, etc.

Para modelar esta información en el catálogo de requisitos de seguridad existente, se tuvieron que añadir atributos (metainformación) a cada uno de sus requisitos, uno por cada uno de estos cinco elementos anteriormente descritos. Así en la *figura 4.1* podemos ver un extracto del catálogo requisitos de seguridad modelados en el entorno de la herramienta de RequisitePro, la cual interactúa con SIRENTool para dar soporte automatizado completo al método SIREN.

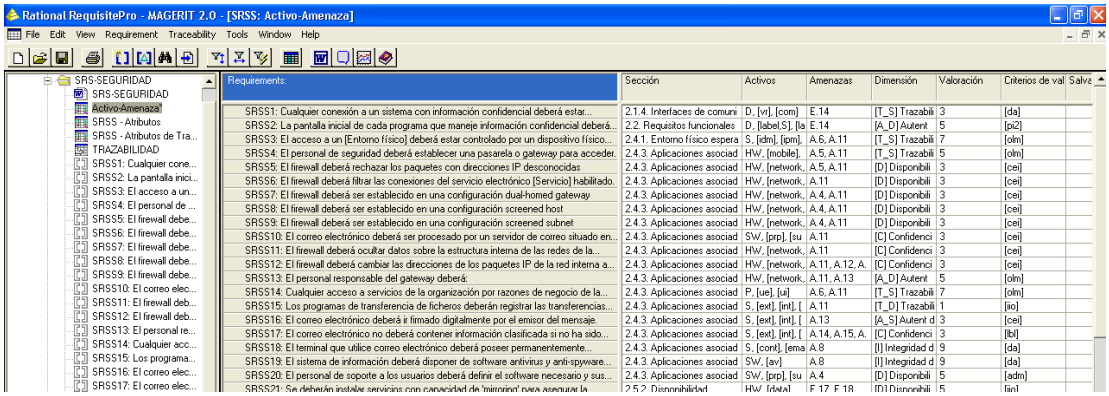

Fig. 4.1. Extracto del catálogo de requisitos de seguridad de MAGERIT en RequisitePro.

4.2.2 Ontología de Análisis de Riesgos

La ontología de análisis de riesgos desarrollada tiene como objetivo recoger los conceptos citados anteriormente (*sección 4.2.1*). Por tanto pretende ser una ontología genérica en el ámbito del análisis de riesgos en sistemas de información. La ontología desarrollada contiene relaciones, restricciones, axiomas y reglas definidos entre los elementos descritos anteriormente y cuya representación analizamos a continuación.

Amenaza o Threat. Una amenaza puede tener las siguientes relaciones y propiedades o atributos:

- *has_asset*: activo(s) sobre el(los) que actúa la amenaza. El origen de la relación es una amenaza (*Threat*) y el destino un activo (*Asset*). Esta relación ha sido restringida para cada tipo de amenaza a través de restricciones de dominio y de rango. Por ejemplo una amenaza que consista en el análisis del tráfico que circula por la red, es una amenaza intencionada que afecta solamente al activo de comunicaciones.

- *has_valuation_dimension:* indica la dimensión del activo a la que afecta la amenaza. También se encuentra restringida en rango y dominio. Para el caso de la amenaza de análisis de tráfico, afectaría al activo comunicaciones sobre la dimensión confidencialidad de los datos (*figura 4.2*).

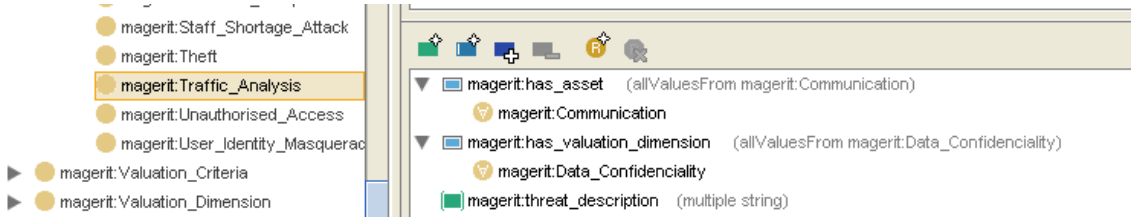

Fig. 4.2. Relación-restricciones para las amenazas en la herramienta Protégé

- *threat_description*: descripción textual de la amenaza.
- *effect:* efecto de la amenaza.
- *previous_record:* registros de ocurrencia de dicha amenaza.
- *probability_of_occurrence:* probabilidad de que ocurra dicha amenaza

Activo o asset. Un activo puede tener las siguientes relaciones y propiedades:

- *code*: código que identifica el activo
- *description:* descripción textual del activo
- *has_safeguard:* salvaguarda a aplicar en caso de daño sobre el activo. Esta relación tiene como origen un activo concreto y como destino una salvaguarda. Se encuentra modelada como la inversa de *protect_asset*, relación de la clase que representa las salvaguardas. Por ejemplo, si tiene lugar la amenaza anterior, de análisis de tráfico, que afecta al activo comunicaciones, se deberá aplicar una de las salvaguardas que protegen este activo (*figura 4.3*).

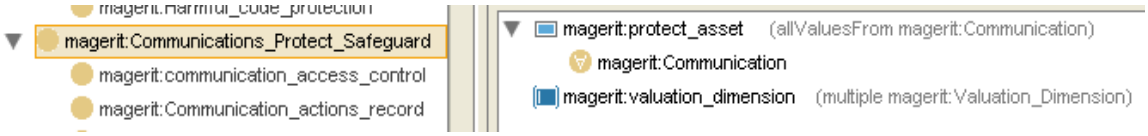

Fig. 4.3. Relación activo-salvaguarda en la herramienta Protégé

- *has_threat*: esta relación se define como la inversa de la relación *has_asset* de la clase que representa a las amenazas en la ontología. Por lo tanto sus valores vendrán restringidos por los establecidos en esa clase. Un activo de tipo comunicaciones estará amenazado por tanto, por todas las amenazas cuyo valor de la relación *has_asset* de amenaza tenga como destino el activo comunicaciones.
- *name:* nombre del activo
- *propietary*: propietario
- *responsible:* responsable
- *unit_responsible*: unidad responsable

Salvaguarda o Safeguard. En una salvaguarda se pueden distinguir las siguientes relaciones y propiedades:

- *protect_asset:* esta relación indica el activo que se protege ante una amenaza según cada una de las dimensiones del activo a la que la amenaza puede afectar. Una posible salvaguarda a la amenaza de análisis de tráfico que actúa sobre el activo comunicaciones podría ser la encriptación de los datos (*figura 4.4*).

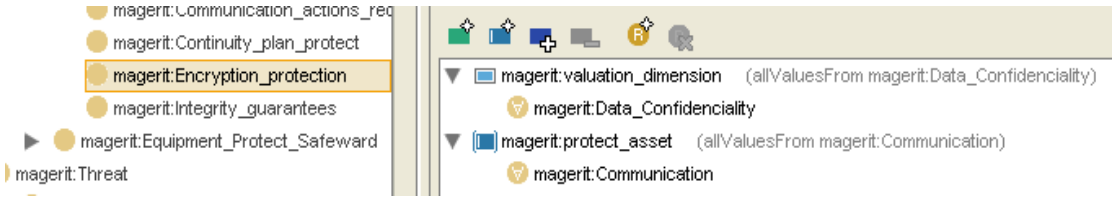

Fig. 4.4. Relación salvaguarda-activo en la herramienta Protégé

- *valuation_dimension:* dimensión del activo sobre la que actúa la salvaguarda. Como vemos en el caso de la salvaguarda que consiste en encriptación, estaríamos protegiendo la dimensión de confidencialidad del activo comunicaciones, justo la misma dimensión que se ve afectada por la amenaza de análisis de tráfico.
- *Efficacy_to_confront_a_threa*t: eficacia de la salvaguarda ante la amenaza.
- *State_of_implantation*: estado de implantación de la salvaguarda.

Dimensión de valoración*:* No posee atributos ni relaciones, establece una clasificación de las diferentes dimensiones en las que se puede valorar un activo. Una amenaza o una salvaguarda se aplicarán sobre un activo atendiendo a una o varias dimensiones de valoración del mismo.

Criterio y Criterio de Valoración: El primero establece una serie de criterios aplicables a un activo para su valoración. El segundo indica una escala de valores del 1-10, de menor a mayor importancia. Cada uno de estos valores se establecerán en base a un criterio. De ahí que dicha clase denominada "*Valuation_Criteria*" contenga un atributo denominado "*type*", cuyo rango es el de los diferentes criterios definidos y cuyo dominio se encuentra restringido según el **catálogo de elementos** de MAGERIT.

A continuación mostramos de manera gráfica algunas de las clases principales de la ontología descrita en la *figura 4.5*.

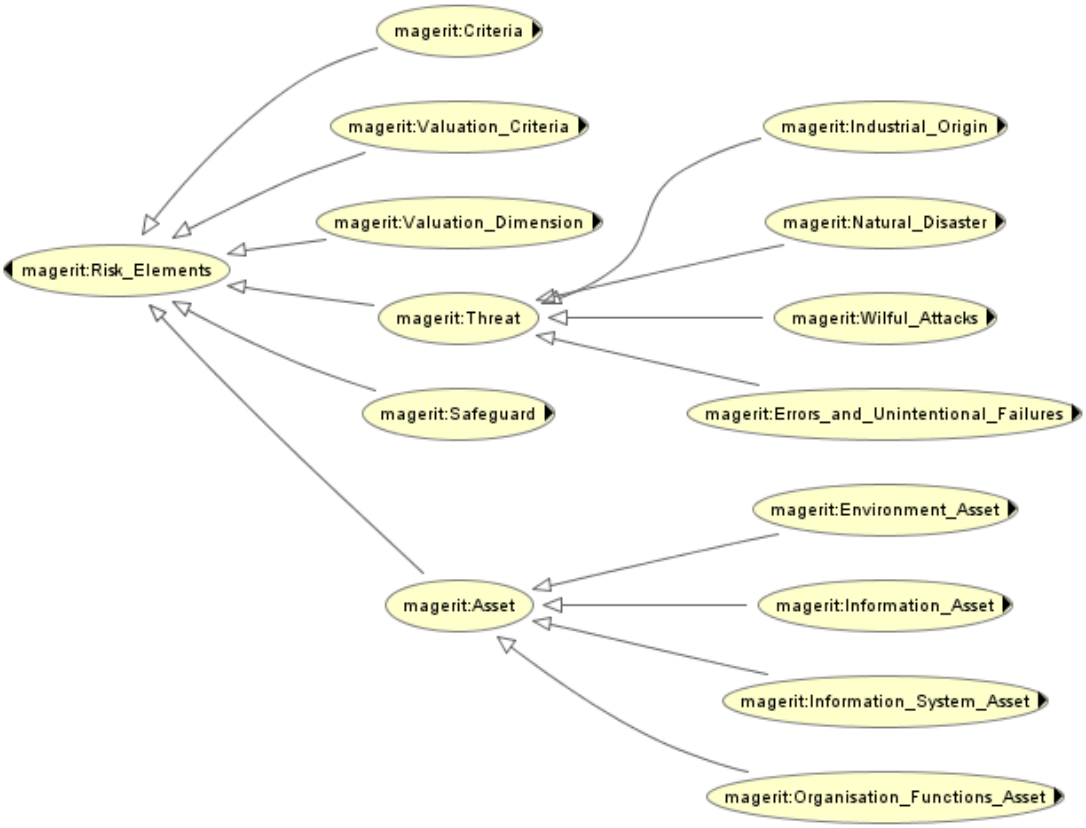

Fig. 4.5. Ontología de análisis de riesgos

En el artículo que se encuentra en el *anexo I, sección 7.4* se pueden ver más ejemplos de restricciones. Sin embargo, no todo tipo de restricción puede ser directamente formalizada con OWL. Para ello, lenguajes de reglas, como los lenguajes de reglas para Web semántica - S*emantic Web Rule Language* (SWRL), juegan un importante rol en combinación con razonadores semánticos y lenguajes de consulta de Web semántica (como SPARQL) y que como veremos en la *sección 4.2.5, aplicación del marco de trabajo*, puede ser muy útiles.

4.2.3 Ontología de Requisitos

Una vez que los conceptos referentes a la seguridad han sido modelados formalmente mediante una ontología, el siguiente paso, antes de aplicar el conocimiento modelado, fue la definición de otra ontología: una **ontología de requisitos**. Esta ontología tiene como objetivo definir una clasificación o taxonomía que permita agrupar los requisitos según su categoría. Para su elaboración se han seguido los estándares de IR que el método SIREN utilizaba para estructurar los requisitos, en concreto, el estándar IEEE 830-1998 [114] y el IEEE 1233 [115]. En estos dos estándares se pueden identificar dos tipos principales de requisitos, los requisitos software (*Software_requirements*) y requisitos del sistema (*System_requirements*), que a su vez se clasifican en (*figura 4.6*):

Requisitos de software:

- REQUISITOS DE INTERFACES EXTERNAS
  - Interfaces de usuario
  - Interfaces hardware
  - Interfaces software
  - Interfaces de comunicación
- REQUISITOS FUNCIONALES
- REQUISITOS DE PRESTACIONES
  - Requisitos de velocidad
  - Requisitos de seguridad críticos
  - Requisitos de precisión
  - Requisitos de capacidad
- RESTRICCIONES DE DISEÑO
  - Entorno físico esperado
  - Entorno tecnológico esperado
  - Aplicaciones asociadas
  - Cumplimiento de estándares
- ATRIBUTOS DEL SISTEMA SOFTWARE
  - Fiabilidad
  - Disponibilidad
  - Mantenimiento
  - Portabilidad
- OTROS REQUISITOS
  - Requisitos de apariencia
  - Requisitos de utilización
  - Requisitos políticos y culturales
  - Requisitos legales

Requisitos del sistema

- FÍSICAS
  - Construcción
  - Durabilidad
  - Adaptabilidad
  - Condiciones ambientales
- CARACTERÍSTICAS DE RENDIMIENTO DEL SISTEMA
- GESTIÓN DE LA INFORMACIÓN
- OPERACIONES DEL SISTEMA
  - Factores humanos del sistema
  - Mantenimiento del sistema
  - Fiabilidad del sistema
- POLÍTICA Y REGULACIÓN
- APOYO AL CICLO DE VIDA DEL SISTEMA

Fig. 4.6. Clasificación de los tipos de requisitos de seguridad según estándares de IR

La *figura 4.7* muestra un extracto de esta ontología de requisitos en Protégé.

Fig. 4.7. Extracto de la ontología de requisitos en Protégé

Cada una de estos tipos de requisitos modelados en la ontología (en forma de clases) deben poseer cierta metainformación asociada que permite la descripción completa del requisito, y que incluye:

- *id_req:* identificador único del requisito.
- *text_description:* descripción textual.
- *priority:* indica la prioridad con la que debe ser llevado a cabo. Su valor está restringido a {*heigh, medium,low*}.
- *rationale:* descripción textual indicando el motivo por el cual el requisito es incluido.
- *state:* indica el estado en el que puede encontrarse un requisito. Puede tomar los siguientes valores {*To be Determined, Determined, To be Revised, To Ruke out, Approved, Modelled in Analysis, Modelled in Design, Implemented or Verify*}.
- *traceability:* en general los requisitos presentes en un documento no se encuentran aislados, sino que aparecen relacionados entre ellos por medio de trazas. El modelo de trazabilidad incluye tres tipos de relaciones: inclusivas, padre-hijo, exclusivas. Estas se han modelado en la ontología mediante las propiedades *Inclusive*, *Parent-Child* y *Exclusive*.
- *fullfilment:* indica si un requisito se considera esencial para el proyecto (*mandatory*), o recomendable pero no esencial (*advisable*) u opcional (*optional*).
- *source:* indica la fuente de la que proviene el requisito, necesidades de clientes, de una solución técnica, legislación, estándares, etc.
- *verification_method:* método que se debería emplear para comprobar que un determinado requisito se cumple en el producto final. Puede tomar los siguientes valores {*inspection, analysis, demonstration, test*}.
- *validatedBy:* criterio de validación necesario para comprobar el requisito.
- *proposedBy:* el individuo que indica que se incluya el requisito.
- *responsibleFor:* el responsable de que el requisito se cumpla.

- *section:* sección del documento en el que el requisito debe situarse, en el caso de que este deba guardarse en un documento específico.
- *parametized_value:* los requisitos parametrizados son aquellos que poseen ciertos valores que pueden variar de su aplicación a un sistema u otro y deben indicarse en cada caso a través de esta propiedad.
- *author:* autor del requisito.
- *date:* fecha de creación del requisito.
- *revision_history:* información de versiones existentes para un requisito.

4.2.4 Ontología de Requisitos de Seguridad

La ontología de requisitos de seguridad combina las ontologías de análisis de riesgos y de requisitos descritas en las secciones anteriores (*4.2.2 y 4.2.3*). Para combinar ambas ontologías y poder así aplicar el conocimiento sobre seguridad modelado a través de la primera ontología (análisis de riesgos), se definieron una serie de relaciones y axiomas en la ontología de requisitos. Para estas relaciones además se han tenido en cuenta otras consideraciones reflejadas en el estándar ISO 27002 [54], como detallaremos a continuación. Estas relaciones son las siguientes:

- *has_asset:* activo(s) sobre el(los) que se define el requisito de seguridad. Su rango son los activos definidos en la ontología de análisis de riesgos (*sección 4.2.2*). Estos activos han sido restringidos, a su vez en función de los activos aplicables a los requisitos según la taxonomía definida en dicha ontología. Por ejemplo, como podemos ver a continuación, los requisitos de rendimiento afectan a los activos de *Hardware* o *Software* (*figura 4.8*). Para este elemento, y siguiendo las recomendaciones del estándar ISO 27002 [54] (*section 7-Asset management*), también vamos a tener en cuenta el propietario (*owner*), la persona (*person*) y la unidad responsable (*unit responsable for*) para cada activo.

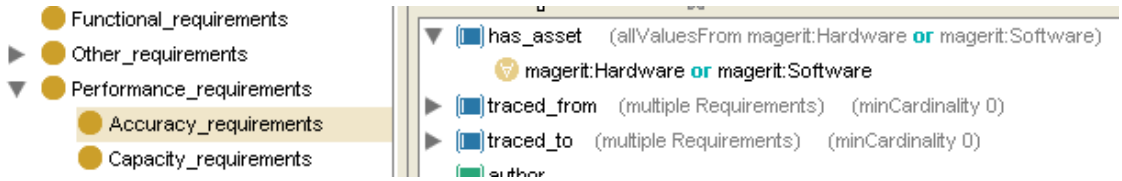

Fig. 4.8. Restricción entre requisitos y activos del sistema en Protégé

- *has_threats:* posibles amenazas que pueden ocurrir si no se cumple el requisito. Su rango son las amenazas definidas en la ontología de análisis de riesgos, en la cual se ha modelado qué amenaza afecta a qué activo. En este caso no se han restringido a nivel de la ontología de requisitos, pero si a nivel de la ontología de análisis de riesgos.
- *valuation:* Su rango es la clase *valuation_criteria* en la ontología de seguridad, establece un valor del 1-10, y el criterio utilizado para elegir dicho valor.
- *has_valuation_dimensions:* define la dimensión del activo sobre la que se define el requisito. Un activo, como hemos visto anteriormente, puede contemplarse atendiendo a diferentes dimensiones de valoración. Sobre cada una de estas dimensiones actuará una amenaza o salvaguarda determinada.
- *has_safeguards:* define la salvaguarda a utilizar. Su rango lo constituyen las salvaguardas definidas en la ontología de análisis de riesgos.

Un ejemplo de requisito modelado con estas ontologías es descrito en la *sección 3.2 del artículo incluido en el anexo I, sección 7.4.*

4.2.5 Aplicación del Marco de Trabajo

En los sistemas de información que necesitan seguridad, una línea base de protección (*baseline*) debe estar siempre implementada en el sistema, excepto para situaciones particulares. Este tipo de razonamiento es frecuentemente aplicado y nos permite el desarrollo de un mínimo de salvaguardas, establecidas por sentido común (*purely common sense*) [27]. Sabemos que existen numerosas fuentes para identificar estas salvaguardas, incluyendo estándares internacionales como los vistos en la *sección 2.3*, estándares o regulaciones nacionales – como las leyes de protección de datos – y otros estándares por sectores.

En este sentido, nuestro marco de trabajo puede ser usado como fuente para especificar esta línea base de protección, usando la ontología de requisitos de seguridad y las propiedades que modela. Esta ontología representa un catálogo de requisitos seguros que es un buen punto de comienzo para un posterior refinamiento del sistema. Además, la protección por catálogos puede ser refinado considerando valoración para los activos y cuantificando las amenazas [27]. Esto se consigue gracias a todas las propiedades y relaciones semánticas de las ontologías, permitiendo a los requisitos ser elicitados y especificados, por ejemplo, por el activo que tienen asociado o por las amenazas que afecten al sistema.

El marco de trabajo cubre un amplio rango de intereses, considerando una gran variedad situaciones en materia de seguridad. En la práctica, un usuario afronta situaciones de análisis en entornos más acotados como puede ser seguir normas respecto a datos personales o seguridad en las comunicaciones. En este caso, la metainformación asociada al requisito donde se guarda la **fuente** (*source*) asociada al requisito (*sección 4.2.3*) puede ser considerada para realizar la búsqueda e identificar requisitos.

Las ventajas de la protección mediante catálogos son: la **velocidad** para seleccionar requisitos, una vez que el catálogo está desarrollado, lo que implica a su vez **necesidad de poco esfuerzo** (frente a los inconvenientes detectados en la *sección 2.4* de los métodos de análisis de riesgos que requerían largos periodos de aprendizaje), y por último, **estandarización**, proporcionado resultados uniformes para proyectos diferentes dentro de la organización o para otras organizaciones similares. Una ventaja adicional de usar este marco es la posibilidad de identificar medidas del porcentaje de requisitos satisfechos del catálogo. Ya en nuestra experiencia con el catálogo de privacidad [34] (*sección 3.3.2*) el porcentaje de requisitos satisfecho del catálogo era utilizado como medida.

El marco de trabajo ofrece también un valor añadido a los responsables de seguridad que vayan a definir un Plan se Seguridad, ya que además de tener identificados todos los activos y amenazas a los que su organización esta expuesta, tendrían los requisitos de seguridad asociados. Además al incluir el análisis de riesgos en las primeras fases del desarrollo de software, tal que esta identificación de riesgos ayude a generar los requisitos de seguridad, permitirá a este responsable de seguridad decidir cuanto esfuerzo se debe invertir en seguridad, indicándonos, además, el grado con el cual un recurso debe ser protegido. Quedarían fuera del marco de trabajo la decisión de qué salvaguarda aplicar y la aprobación del coste que supone su implantación en la organización. Esta decisión no es tarea del responsable de seguridad sino del departamento de dirección o de los altos responsables de ésta.

APLICACIÓN AL CATÁLOGO DE REQUISITOS ACTUAL (VALIDACIÓN DE INCONSISTENCIAS EN LOS REQUISITOS)

Gracias a las relaciones modeladas en la ontología, hemos podido validar la consistencia del catálogo de requisitos de seguridad basado en MAGERIT [10] (*sección 4.2.1*). En este catálogo fueron especificados unos 350 requisitos, y mediante las propiedades (restricciones, axiomas y reglas) identificados en el marco de trabajo y descritos en la *sección 4.2.4* (*has_asset, has_threats* …), fue posible detectar hasta 27 inconsistencias, 8 relacionadas con la trazabilidad de los requisitos (comprobando que estábamos libre de ciclos en los requisitos) y 19 por restricciones de análisis de riesgos. Un ejemplo de requisito concreto que era inconsistente es mostrado en la *sección 3.1 del artículo incluido en el anexo I, sección 7.4.*

USO DE CONSULTAS SEMÁNTICAS (VERIFICAR LA CORRECCIÓN DE LOS REQUISITOS)

Otra de las ventajas de tener modelado nuestro dominio de la seguridad mediante ontologías descritas en OWL es la posibilidad de aprovechar las investigaciones en la comunidad de la Web Semántica y las herramientas desarrolladas en torno a ella. Fruto de estas investigaciones son las técnicas de integración y mapeo entre ontologías, clasificación o los lenguajes de consulta. Concretamente, ya hemos trabajado con el lenguaje de consulta SPARQL, a través del cual es posible obtener cierta información de nuestras ontologías. A continuación mostramos dos consultas realizadas y cuya intersección nos es de mucha utilidad para verificar la corrección de los requisitos:

Objetivo: verificar que las amenazas que afectan a un requisito, actúan sobre los activos que éste posee.

Para ello seleccionamos los requisitos y activos tal que los activos definidos en cada requisito sean iguales a los activos sobre los que actúa la amenaza del requisito.

```
SELECT distinct ?x ?asset
WHERE{ ?x :has_asset ?asset ;  :has_threat ?threat.
            ?threat magerit:has_asset ?b.  filter(sameTerm(?asset,?b) )
} orderby ?x ?asset
```

Además deberemos comprobar que los activos que el requisito posee se encuentran todos incluidos entre los activos a los que afecta la amenaza. Para ello primero realizamos la consulta en la que seleccionamos todos los requisitos y sus activos

```
SELECT distinct ?x ?asset
WHERE{ ?x :has_asset ?asset ;
} orderby ?x ?asset
```

Y finalmente, deberemos comprobar, implementando mediante software la intersección de estas dos consultas, que el resultado de la misma contiene los activos que el requisito posee, en caso contrario sería una inconsistencia.

Un ejemplo de inconsistencia controlada con estas consultas es mostrado en la *sección 3.3 del artículo incluido en el anexo I, sección 7.4.*

4.3 Revisión Sistemática del Estado del Arte. Hacia una Ontología General e Integrada de Seguridad

Como consecuencia de la revisión del estado del arte que nos permitió analizar el campo de la ingeniería ontológica aplicada a la seguridad [38, 39, 96], se concluyó con la necesidad y reto de obtener una ontología general de seguridad, integrada e unificada. Por ello, y tomando como base la realización de una comparativa formal entre las propuestas más maduras encontradas en el estudio, identificamos una serie de requisitos clave o necesarios para obtenerla, dando a su vez las primeras indicaciones para conseguirla [41]. Esto se encuentra en el *artículo anexo I publicaciones, sección 7.5.*

Para el análisis del estado del arte se utilizó la técnica de **revisión sistemática**, definiendo y aplicando un protocolo formal para la revisión que nos permitió centrar la pregunta de investigación y el tipo de resultados (estudios primarios) que queríamos obtener y posteriormente analizar. Del análisis de estos estudios primarios detectados concluimos que la mayoría de los trabajos identificados se centraban en dominios específicos. Por contra, había menos propuestas de ontologías generales, al ser ésta una tarea más costosa, y más en el ámbito de la seguridad donde el dominio es bastante cambiante y requiere de constante actualización. Además, observamos que la mayoría de los trabajos analizados se encontraban todavía en las primeras etapas de desarrollo.

De acuerdo con las propuestas identificadas en este estudio previo, se decidió abordar una comparación formal entre ontologías de seguridad, eligiendo las propuestas identificadas como más maduras. Para la comparación se utilizó el marco formal de comparación y construcción de ontologías OntoMetric [113], que ya habíamos tenido en cuenta para la realización de nuestras ontologías descritas en la *sección 4.2 anterior*. Este método realiza una comparación de características agrupadas en factores que a su vez se agrupan en cinco dimensiones: contenido representado, lenguaje, metodología, entorno software y coste de utilizar la ontología en nuevos sistemas. En esta comparación formal nos centramos en el estudio de la dimensión **contenido de la ontología,** al estar interesados en que los elementos que compongan la ontología sean completos y reutilizables. En esta dimensión contenido, el marco formal de comparación, presenta a su vez una serie de características descriptivas agrupadas en estos 4 factores: **conceptos, relaciones, taxonomía de conceptos** y **axiomas**. Cada propuesta fue valorada según estos factores, como se puede apreciar en la *sección 3* del artículo *anexo I publicaciones, sección 7.5.* En cuanto a la consideración de las otras dimensiones, destacar que consideramos que el lenguaje y el entorno software ideal para la construcción de las mismas era hacer uso de OWL y Protégé, tal y como describíamos en la *sección anterior (4.2)*.

Como conclusión de la comparación formal hemos identificado en las propuestas elegidas una serie de deficiencias con respecto a la definición de sus conceptos (ver más detalles en la *sección 3* del artículo *anexo I publicaciones, sección 7.5.).* Así podemos destacar, que algunas no incluyen atributos con los que definir los conceptos, o no utilizan expresiones en lenguaje natural apropiadas. Además, estás ontologías no definen particiones exhaustivas, ya que no definen todas las posibilidades del dominio estudiado, y además utilizan pocos axiomas y propiedades de relación formales entre los conceptos para inferir conocimiento, como pueden ser las propiedades de reflexividad, transitividad, simetría y asimetría. Tampoco se hace uso de la clasificación taxonómica de forma correcta, ya que no todos los conceptos esenciales están en las partes altas de

la jerarquía, pudiendo por lo tanto ser más reutilizables, y además, no todas hacen uso de las particiones disjuntas, importantes en los proceso de clasificación automática.

También del estudio, llegamos a la conclusión de que la definición de una ontología general e integrada de seguridad es considerada como una tarea primordial (ya se discutió en la *sección 2.6*), y ésta debía ser abordada de forma consensuada por parte de la comunidad científica. En definitiva, debíamos estar hablando de una ontología de seguridad flexible y fácil de actualizar con los nuevos conceptos que aparezcan en la comunidad. Sin embargo, precisamente de las conclusiones descritas anteriormente sobre las deficiencias de estas propuestas, concluíamos que las ontologías existentes no estaban debidamente preparadas para su reutilización y extensión.

Se identificó que la mejor forma de obtener esta "ontología completa e integrada" es realizar un estudio del estado actual del arte (como lo hemos hecho en este documento), y realizar una comparación de las ontologías identificadas a través de un marco formal de comparación (a través de los factores de contenido, taxonomía, relación y axiomas), con el objetivo de obtener una visión de la situación actual, para detectar deficiencias y posibles mejoras de las ontologías desarrolladas en los trabajos, con el fin de poder integrarlas y combinarlas, reduciendo así el coste de desarrollo de otras nuevas que partirían desde cero. También señalamos que la combinación del software Protégé y el lenguaje de especificación de ontologías OWL, aceptado como un estándar de la World *Wide Web Consortium* (W3C), son el instrumento más apropiado a través del cual hacerlo. Dicha ontología final deberá satisfacer las características de los factores de contenido, taxonomía, relaciones y axiomas identificados por [113].

Así, describimos los primeros pasos necesarios para obtener dicha ontología (*sección 4* del artículo *anexo I publicaciones, sección 7.5)*, e identificamos los aspectos clave (*key requirements*) que dichas ontologías debían cumplir. Como extracto de este artículo resumimos aquí estos aspectos clave (*figura 4.9*), que son analizados para las propuestas más maduras en dicho artículo, en concreto en la *tabla 10, sección 4.2.*

Conocimiento Estático: recoge que los conceptos modelados en la ontología deben estar identificados adecuadamente. Es decir, para poder integrar una ontología definida, ésta deberá describir satisfactoriamente los conceptos esenciales en lenguaje natural, además de sus propiedades asociadas (relaciones y atributos) para el dominio que modela. Además debe estar actualizada y ser conforme a estándares.

Conocimiento Dinámico: se encarga de asegurar que el conocimiento modelado en la ontología nos permita inferir conocimiento. En otras palabras, poder utilizar axiomas definidos para restringir los valores de los atributos de las instancias y las instancias de las relaciones, manteniendo la consistencia de la ontología y realizando deducciones (infiriendo conocimiento).

Reusabilidad: la ontología es desarrollada considerando aspectos que le permitan ser reutilizada y compartida. Así la clasificación taxonómica de los conceptos tendrá que ser la adecuada, y representar, de la mejor manera posible, particiones exhaustivas del dominio, aún siendo la seguridad un dominio complicado para conseguirlo. Además tendrán que considerarse el uso de lenguajes estándar para su modelado y siempre incluir comentarios en lenguaje natural de todos los conceptos modelados.

Fig 4.9. Aspectos clave para diseñar ontologías que sean integrables en una ontología general

Además, extraemos la *figura 4.10,* de la *sección 4* de este mismo artículo *anexo I publicaciones, sección 7.5*, donde se muestra gráficamente la integración de las ontologías existentes par a par (que es el primer paso para llegar a esta ontología de seguridad general), y donde se explica cómo se va integrando cada una de las ontologías maduras existentes, incluyendo su integración con nuestra ontología de requisitos de seguridad descrita en la *sección anterior (4.2).*

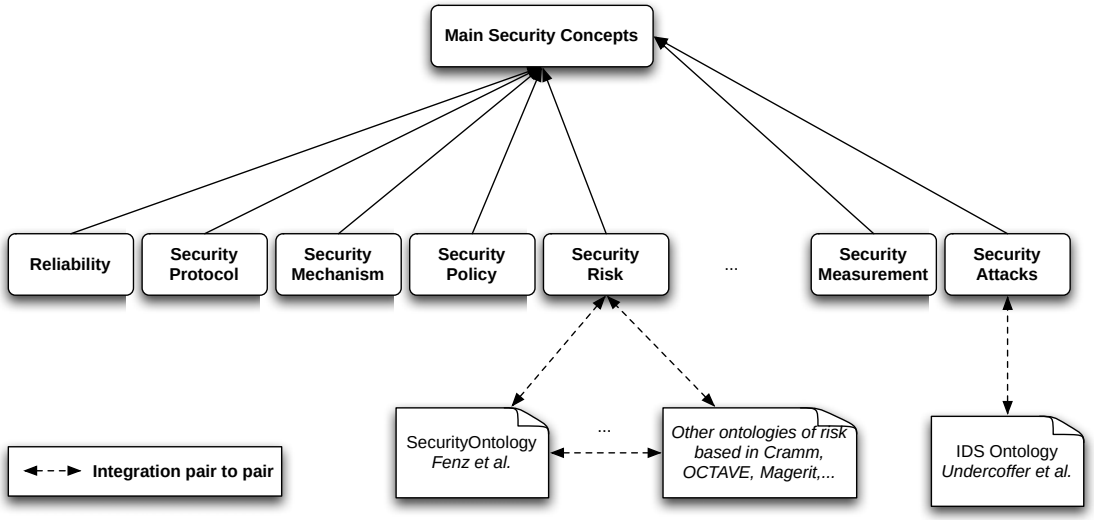

Fig. 4.10. Solapamiento entre dominios de seguridad para la ontología de seguridad integrada

4.4 Conclusiones

En este capítulo hemos visto cómo para la realización del marco se llevó a cabo la mejora y actualización de los catálogos de requisitos de seguridad actuales identificados para SIREN, implicando su integración en un único catálogo de seguridad. Después de procedió a su conversión a ontologías en el lenguaje OWL (para su soporte formal), donde se recogieron las relaciones semánticas entre los distintos elementos y se aplicaron reglas semánticas para validarlas. Esto supuso la creación de dos ontologías, una basada en análisis de riesgos y otra de requisitos, que finalmente se integraban en una única de requisitos de seguridad reutilizables. Por último, destacar que se identificó la necesidad de obtener una ontología general de seguridad (según un estudio del estado del arte realizado), de la cual, tras la realización de una comparativa formal de ontologías, se identificaron los requisitos clave y se establecieron los primeros pasos para obtenerla.

5 CAPÍTULO 5. CONTRASTE DE RESULTADOS, CONCLUSIONES Y LÍNEAS FUTURAS

5.1 Introducción

En esta sección relatamos los resultados obtenidos clasificados por tipo de contribución (*sección 5.2*), y acabamos con las conclusiones y vías futuras (*sección 5.3*).

5.2 Contraste de Resultados

Durante el desarrollo de esta tesis doctoral se han publicado y discutido varios trabajos en diversos foros científicos. La *tabla 5.1* muestra un resumen numérico de estas publicaciones, todas ellas sometidas a procesos de revisión. Además, la tabla incluye información acerca de Proyectos de transferencia tecnológica y Proyectos Fin de Carrera realizados en líneas de investigación relacionadas con esta tesis y en cuya dirección ha participado el doctorando. A continuación se muestra cada uno de estos trabajos desglosados por categorías.

Tipo de Contribución	Total
Artículos en revistas internacionales ISI/JCR	5
Capítulos en libros nacionales	1
Congresos y talleres internacionales	6
Congresos y talleres nacionales	10
Informes Técnicos	2
Proyectos Fin de Carrera	1
Proyectos de transferencia tecnológica	1
Número total de contribuciones	26

Tabla 5.1. Número de publicaciones organizadas por tipo de contribución

ARTÍCULOS EN REVISTAS INTERNACIONALES INDEXADAS EN JCR

A continuación se detallan los artículos publicados en revistas internacionales con índice de impacto:

Especialización en IR

- Ambrosio Toval, Begoña Moros, Joaquín Nicolás y Joaquín Lasheras, *Eight Key Issues for an Effective Reuse-Based Requirements Process.* International Journal of Computer Systems Science and Engineering, Vol 23 (6) p. 373-385 Noviembre 2008.
  Factor Impacto en el listado JCR 2009: 0,222 (posición 48/49, grupo Computer Sc. Hw. & Arch.) (posición 90/91, grupo Computer Sc.Theory and Methods)

- Joaquín Nicolás, Joaquín Lasheras, Ambrosio Toval, Francisco J. Ortiz y Bárbara Álvarez, *An Integrated Domain Analysis Approach for Teleoperated Systems.* Requirements Engineering Journal (REJ), Vol 14(1) p. 27-46 Enero 2009.
  Factor Impacto en el listado JCR 2009: 0,931 (posición 61/93, grupo Computer Sc. Soft. Eng.) (posición 76/116, grupo Computer Sc. Information Systems)

- Miguel A. Martínez, Joaquín Lasheras, Ambrosio Toval, Eduardo Fernandez-Medina y Mario Piattini, *A Personal Data Audit Method through Requirements Engineering.* Computer Standards and Interfaces, Vol 32(4), p 166-178 Junio 2010.
  Factor Impacto en el listado JCR 2009: 1,373 (posición 37/93, grupo Computer Sc. Soft. Eng.) (posición 18/49, grupo Computer Sc. Hw. & Arch.)

Soporte formal

- Joaquín Lasheras, Rafael Valencia-García, Jesualdo Tomás Fernández-Breis y Ambrosio Toval, *Modelling Reusable Security Requirements based on an Ontology Framework.* Journal of Research and Practice in Information Technology (JRPIT), Vol 41(2), p. 119-133 Mayo 2009.
  Factor Impacto en el listado JCR 2009: 0,5 (posición 81/93, grupo Computer Sc. Soft. Eng.) (posición 100/116, grupo Computer Sc. Information Systems)

- Carlos Blanco, Joaquín Lasheras, Eduardo Fernandez-Medina, Rafael Valencia-García y Ambrosio Toval, *Basis for an integrated Security Ontology according to a systematic review of existing proposals.* Computer Standards and Interfaces. Vol 33 (4), p 372-388 Junio 2011
  Factor Impacto en el listado JCR 2009: 1,373 (posición 37/93, grupo Computer Sc. Soft. Eng.) (posición 18/49, grupo Computer Sc. Hw. & Arch.)

A continuación se muestran otras publicaciones en las que el doctorando ha estado implicado durante la realización de la tesis doctoral, clasificadas por su tipo:

CAPÍTULOS DE LIBRO

- C. Blanco, D.G. Rosado, D. Mellado, A. Rodríguez, C. Gutierrez, J. Lasheras, E. Fernández-Medina, A. Toval, J. Trujillo y M. Piattini, *Capítulo 15: Seguridad en Ingeniería del Software* en *Calidad del producto y proceso software.* RA-MA p. 339 – 375, 2010 ISBN: 8478979611.

PUBLICACIONES EN CONGRESOS DE CARÁCTER INTERNACIONAL

- M.A. Martínez, J. Lasheras, A. Toval, M. Piattini, *Aportaciones de la Ingeniería de Requisitos en un proceso de auditoría de datos personales*. IV Congreso Internacional de Auditoría y Seguridad de la Información (CIASI'05). Madrid, España. Diciembre 2005. ISBN: 84-689-5752-6.

- J. Nicolás, J. Lasheras, A. Toval, F.J. Ortíz, B. Alvarez, *A Collaborative Learning Experience in Modelling the Requirements of Teleoperated Systems for Ship Hull Maintenance*. Workshop on Learning Software Organizations and Requirements Engineering (LSO + RE 2006). Hannover, Alemania. Marzo 2006.

- M.A. Martínez, J. Lasheras, A. Toval, M. Piattini, *An Audit Method of Personal Data Based on Requirements Engineering*. The 4th International Workshop on Security in Information Systems (WOSIS-2006), in the 8th International Conference on Enterprise Information Systems (ICEIS'06). Paphos, Chipre, Mayo 2006. ISBN: 972-8865-52-X. CORE C

- C. Blanco, J. Lasheras, R. Valencia-García, E. Fernández-Medina, A. Toval, M. Piattini, *Revisión sistemática y comparación de ontologías en el marco de la seguridad*. IV congreso iberoamericano de seguridad en informática CIBSI 2007. Mar de plata (Argentina). Noviembre 2007, ISBN: 978-950-623-043-2.

- C. Blanco, J. Lasheras, R. Valencia-García, E. Fernández-Medina, A. Toval, M. Piattini, *A Systematic Review and Comparison of Security Ontologies*. ARES International Conference on Availability, Reliability and Security, Barcelona, España. Marzo 2008, IEEE computer society ISBN: 978-0-7695-3102-1. CORE B

- J. Lasheras, R. Valencia-García, J.T. Fernández-Breis, A. Toval, *An Ontology-Based Framework for Modelling Security Requirements.* The 6th International Workshop on Security in Information Systems (WOSIS-2008), in the 10th International Conference on Enterprise Information Systems (ICEIS'08). Barcelona, España. Julio 2008. ISBN: 978-989-8111-44-9. CORE C

PUBLICACIONES EN CONGRESOS DE CARÁCTER NACIONAL

- J. Lasheras, A. Toval, J. Nicolás, B. Moros, *Soporte automatizado a la reutilización de requisitos*. VIII Jornadas de Ingeniería del Software y Bases de Datos (JISBD 2003). Alicante. Noviembre 2003. I.S.B.N: 84-688-3836-5.

- J. Lasheras, A. Toval, J. Nicolás, B Moros, Definición de Requisitos de Seguridad con Fines de Reutilización, I Taller de Seguridad en Ingeniería del Software y Bases de Datos (SISBD'2004) Málaga, Noviembre 2004

- J. Lasheras, J. Nicolás, A. Toval, B. Moros, *Hacia un Modelo del Dominio de los Sistemas Teleoperados a través de una extensión de SIREN*. II Jornadas de trabajo DYNAMICA (DYNamic and Aspect-Oriented Modeling for Integrated Component-based Architectures) Málaga, Noviembre 2004.

- B. Álvarez, P. Sánchez, J.A. Pastor, A. Toval, J. Lasheras, *Experiencia, Estrategias y Retos en la Incorporación de Requisitos de Seguridad en el Sistema EFTCoR .* IX Jornadas de Ingeniería del Software y Bases de Datos (JISBD 2004) Málaga, Noviembre 2004. I.S.B.N: 84-688-8983-0.

- J. Nicolás, J. Lasheras, A. Toval, B. Moros, P.Sanchez, B. Alvarez, *Ingeniería de Requisitos Basada en Reutilización: una propuesta de Aplicación a los Sistemas Teleoperados para Limpieza de Cascos de Buques.* III Jornadas de trabajo DYNAMICA (DYNamic and Aspect-Oriented Modeling for Integrated Component-based Architectures) Ciudad Real, Abril 2005.

- M.A. Martínez, J. Lasheras, J. Nicolás, A. Toval, *Aplicación de un Proceso de Auditoría de Datos Personales Basado en el Método SIREN,* III Jornadas trabajo DYNAMICA (DYNamic and Aspect-Oriented Modeling for Integrated Component-based Architectures) Ciudad Real, Abril 2005.

- M.A. Martínez, J. Lasheras, J. Nicolás, A. Toval, *Un proceso de auditoría de datos personales basado en Ingeniería de Requisitos.* I Simposio sobre seguridad informática [SSI'2005], dentro del I Congreso Español de Informática [CEDI'2005]. Granada, Septiembre 2005 I.S.B.N: 84-9732-447-1.

- J. Nicolás, J. Lasheras, A. Toval, F.J. Ortíz, B. Alvarez, *Una experiencia de modelado de los sistemas teleoperados para limpieza de cascos de buques mediante características y casos de uso genéricos*. IV Jornadas de trabajo DYNAMICA (DYNamic and Aspect-Oriented Modeling for Integrated Component-based Architectures). Murcia, Noviembre 2005.

- C. Blanco, J. Lasheras, R. Valencia-García, E. Fernández-Medina, A. Toval, M. Piattini, *Ontologías de seguridad: revisión sistemática y comparativa,* II Simposio sobre seguridad informática [SSI'2007], dentro del II Congreso Español de Informática [CEDI'2007], Zaragoza, Septiembre 2007 I.S.B.N: 978-84-9732-607-0.

- A. Duque, J. Lasheras, A. Toval, ECAPRIS, Metodología ágil de medición de calidad y productividad en PYMEs, XIV Jornadas de Ingeniería del Software y Bases de Datos (JISBD 2009) Actas del congreso I.S.B.N: 978-84-692-4211-7 : San Sebastian (Spain) 8-11/09/2009

INFORMES TÉCNICOS

- C. Blanco, J. Lasheras, R. Valencia-García, E. Fernández-Medina, A. Toval, M. Piattini. *Ontologías de seguridad: revisión sistemática y comparativa*. Informe Técnico – UCLM-TSI-003 (JULIO 2008)

- J. Nicolás, B Moros, J. Lasheras, A. Toval. *SIREN: Un Método Práctico de Ingeniería de Requisitos Basado en Reutilización*. Informe Técnico – UMU-TR DIS 1-2009.

PROYECTOS FIN DE CARRERA (co-dirigidos)

- "Plugin para el análisis y la negociación de requisitos en el marco de la herramienta CARE Siren Tool". Alumno: Carlos de Pro Cherenguini. Septiembre de 2007, Facultad de Informática, Universidad de Murcia.

PROYECTOS DE TRANSFERENCIA TECNOLÓGICA

- GARTIC (Gestión Automatizada de Requisitos basada en Reutilización para PYMES del sector TIC), un proyecto de transferencia tecnológica firmado por el Grupo de Investigación en Ingeniería del Software de la Universidad de Murcia y el Centro Tecnológico de las Tecnologías de la Información y las Comunicaciones (CENTIC) de la Región de Murcia, que contó con la participación de cinco empresas regionales a las que se formó para la utilización de técnicas de Ingeniería de Requisitos en sus proyectos de desarrollo

5.3 Conclusiones y Líneas de Trabajo Futuras

Se ha presentado un marco de trabajo para reuso y reutilización de requisitos, que extendía el método SIREN, dándole soporte formal mediante el uso de ontologías. Así hemos diseñado una ontología de requisitos y otra de análisis de riesgos, integrándola en una sola para crear una ontología de requisitos de seguridad. Del propio estudio del estado del arte hemos identificado que la obtención de una ontología de requisitos general, integrada y unificada de seguridad se debe plantear como un reto dentro de la comunidad científica, dando los primeros pasos para ésta e identificando los aspectos clave que una ontología debe considerar para poder ser integrada con ella. Esta ontología proporcionaría una base bien conocida en la que soportar el desarrollo de métodos, procesos y metodologías apropiadas y nos ayudaría a organizar nuestros conocimientos y a transmitirlos.

Como consecuencia de estos resultados, la principal línea futura será la integración con otros marcos de desarrollo de ontologías para ir hacia la indicada ontología general de seguridad. Empezaríamos con la integración a otros marcos de análisis de riesgos (CRAMM u Octave) o estándares de seguridad, como ISO 27001 (*figura 5.1*), incluyendo también otras técnicas de seguridad como las descritas en la *sección 2.5* o trabajos de ontologías identificados en la *sección 2.6*. Además el trabajo podría extenderse a otras fases identificadas para un SGSI, diferente al análisis de riesgos, como la realización del documento de la política de seguridad o la realización de auditorias internas.

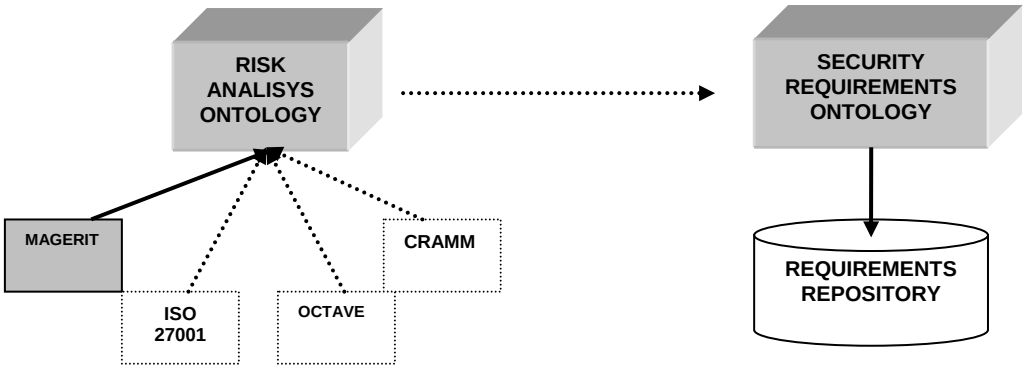

Fig. 5.1. Extensión de la ontología de riesgos con otros métodos de seguridad

Además la ontología deberá integrarse con las mejoras detectadas en los catálogos de requisitos reutilizables. En este sentido ya se ha iniciado una propuesta de mejora de los catálogos (que tendrá su implicación en el modelado con las ontologías) donde se consideró su adaptación al estándar de seguridad ISO 27001, identificando para cada requisitos de seguridad, con que control definido en el estándar se asociaba.

Por otra parte, la ontología tendrá que ser validada en casos de estudio, lo que probablemente suponga también incrementar las relaciones semánticas identificadas en los elementos de las ontologías. Esto se podría abordar gracias que al estar modeladas en OWL y descritas por un proceso formal de construcción de ontologías, nos permite aprovechar las investigaciones en la comunidad de la Web Semántica y las herramientas-técnicas desarrolladas en torno a ella, como pueden ser las técnicas de integración y mapeo entre ontologías, o los lenguajes de consulta.

6 REFERENCIAS

1. Sandhu, R., K. Ranganathan, and X. Zhang, *Secure information sharing enabled by Trusted Computing and PEI models.* ASSIACCS, 2006: p. 2-12.
2. Dhillon, G., *Information Security Management: Global challenges in the new millennium*, ed. I.G. Publishing. 2001.
3. Denker, G., L. Kagal, and T. Finin, *Security in the Semantic Web using OWL.* Information Security Technical Report, 2005. **10**(1): p. 51-58.
4. ITEA. *ITEA-Technology Roadmap for Software-Intensive Systems*. 2004 [cited 2011]; Available from: ftp://ftp.cordis.europa.eu/pub/ist/docs/directorate_d/st-ds/itea2.pdf.
5. Siponen, M.T., *Secure-System Design Methods: Evolution and Future Directions.* IT Professional, 2006. **8**(3): p. 40-44.
6. Breu, R., K. Burger, M. Hafner, J. Jürjens, G. Popp, V. Lotz, and G. Wimmel. *Key Issues of a Formally Based Process Model for Security Engineering*, in *16th Inter. Conference on Software and Systems Engineering and their Applications (ICSSEA'03)*. 2003.
7. Jürjens, J., *Secure Systems Development with UML*. 2005, Springer-Verlag.
8. Haley, C.B., J.D. Moffet, R. Laney, and B. Nuseibeh. *A Framework for Security Requirements Engineering*, in *Software Engineering for Secure Systems Workshop (SESS'06), co-located with the 28th International Conference on Software Engineering (ICSE'06)*. 2006. Shanghai, China.
9. Mouratidis, H. and P. Giorgini, *Integrating Security and Software Engineering: Advances and Future Visions*. 2007, Idea Group Publishing.
10. Toval, A., J. Nicolás, B. Moros, and F. García, *Requirements Reuse for Improving Information Systems Security: A Practicioner's Approach.* Requirements Engineering Journal (REJ), 2002. **6**(4): p. 205-219.
11. Blanco, C., D.G. Rosado, D. Mellado, A. Rodríguez, C. Gutiérrez, J. Lasheras, E. Fernández-Media, A. Toval, J. Trujillo, and M. Piattini, *Seguridad en Ingeniería del Software (capítulo 15)*, in *Calidad del producto y proceso software*, Editor C. Calero, Ra-Ma: Spain. p. 339-375. 2010
12. Crook, R., D. Ince, and B. Nuseibeh, *Modelling access policies using roles in requirements engineering* Information and Sofware Technology, 2003. **45**(14): p. 979-991.
13. Devanbu, P. and S. Stubblebine. *Software engineering for security: a roadmap*, in *ICSE, Future of Software Engineering*. 2000. Limerick, Ireland.
14. Anderson, R., *Security engineering: A guide to building dependable distributed systems*. 2001, Wiley Computer Publishing.
15. Toval, A., A. Olmos, and M. Piattini. *Legal Requirements Reuse: A Critical Success Factor for Requirements Quality and Personal Data Protection*, in *Int.Conference on Requirements Engineering (RE)*. 2002. Essen, Alemania, IEEE Computer Press.
16. Mouratidis, H., P. Giorgini, and G. Manson, *When security meets software engineering: A case of modelling secure information systems.* Information Systems, 2005. **30**(8): p. 609-629.
17. Mellado, D., C. Blanco, L.E. Sánchez, and E. Fernández-Media, *A systematic review of security requirements engineering* Computer Standards & Interfaces, 2010. **32**(4): p. 153-165.
18. Robertson, S.J., *Mastering the Requirements Process (2nd Edition)*. 2006, Addison-Wesley.
19. Cheng, B.H.C. and J.M. Atlee. *Research Directions in Requirements Engineering*, in *Future of Software Engineering 2007 (FOSE), in ICSE*. 2007. Minneapolis, Minnesota, IEEE Computer Society
20. Prieto-Díaz, R., *Status Report: Software Reusability*. IEEE Software, 1993. **10**(3): p. 61-66.

21. Rine, D.C. and N. Nada, *An empirical study of a software reuse reference model.* Inf. and Software Technology, 2000. **42**(1): p. 47-65.
22. Raskin, V., C.F. Hempelmann, K.E. Triezenberg, and S. Nirenburg. *Ontology in Information Security: A Useful Theoretical Foundation and Methodological Tool*, in *New Paradigms Security Workshop NSPW'01*. *ACM Press* 2001. Clouford, New Mexico, USA.
23. Donner, M., *Toward a Security Ontology.* IEEE Security and Privacy, 2003. **1**(3).
24. Tsoumas, B. and D. Gritzalis. *Towards an Ontology-based Security Management*, in *20th International Conference on Advanced Information Networking and Applications (AINA'06)*. 2006. Vienna, Austria, IEEE Computer Society.
25. Mouratidis, H. and P. Giorgini, *Integrating Security and Software Engineering: Future Vision and Challenges*, in *Integrating Security and Software Engineering: Advances and Future Visions*. 2007, Idea Group Publishing. p. 276-285.
26. Nicolás, J., B. Moros, J. Lasheras, and A. Toval, *SIREN (SImple REuse of software requirements), a general-purpose RE method based on requirements reuse. Technical Report - UMU-TR DIS 1-2009.* 2009.
27. MAGERIT. *Methodology for Information Systems Risk Analysis and Management. V 2.0, http://www.csi.map.es/csi/pg5m20.htm*. 2006 [cited 2011].
28. Toval, A., B. Moros, J. Nicolás, and J. Lasheras, *Automating Key Issues For Effective Requirements Reuse.* Computer Systems Science and Engineering., 2008. **23**(6): p. 373-385.
29. Lasheras, J., A. Toval, J. Nicolás, and B. Moros. *Soporte automatizado a la reutilización de requisitos*, in *VIII Jornadas de Ingeniería del Software y Bases de Datos (JISBD 2003)*. 2003. Alicante.
30. Lasheras, J., J. Nicolás, A. Toval, and B. Moros. *Hacia un Modelo del Dominio de los Sistemas Teleoperados a través de una extensión de SIREN*, in *II Jornadas de trabajo DYNAMICA*. 2004. Málaga (Spain).
31. Nicolás, J., J. Lasheras, A. Toval, F.J. Ortiz, and B. Álvarez. *A Collaborative Learning Experience in Modelling the Requirements of Teleoperated Systems for Ship Hull Maintenance* in *Workshop on Learning Software Organizations and Requirements Engineering (LSO + RE 2006)*. 2006. Hannover (Germany).
32. Nicolás, J., J. Lasheras, A. Toval, F.J. Ortiz, and B. Álvarez, *An Integrated Domain Analysis Approach for Teleoperated Systems.* Requirements Engineering Journal (REJ), 2009. **14**(1): p. 27-46.
33. Martínez, M.A., J. Lasheras, A. Toval, and M. Piattini. *Aportaciones de la Ingeniería de Requisitos en un proceso de auditoría de datos personales*, in *IV Congreso Internacional de Auditoría y Seguridad de la Información (CIASI)*. 2005. Madrid, Spain.
34. Martínez, M.A., J. Lasheras, A. Toval, and M. Piattini. *An Audit Method of Personal Data Based on Requirements Engineering*, in *the 4th International Workshop on Security In IS (WOSIS-2006)*. 2006. Paphos, Chipre.
35. Martínez, M.A., J. Lasheras, A. Toval, E. Fernández-Medina, and M. Piattini, *A Personal Data Audit Method through Requirements Engineering.* Computer Standards & Interfaces, 2010. **32**(4): p. 166-178.
36. Lasheras, J., R. Valencia-García, J.T. Fernández-Breis, and A. Toval. *An Ontology-Based Framework for Modelling Security Requirements*, in *the 6th International Workshop on Security In Information Systems (WOSIS-2008)*. 2008. Barcelona (Spain).
37. Lasheras, J., R. Valencia-García, J.T. Fernández-Breis, and A. Toval, *Modelling Reusable Security Requirements based on an Ontology Framework.* Journal of Research and Practice in Information Technology (JRPIT), 2009. **41**(2): p. 119-133.
38. Blanco, C., J. Lasheras, R. Valencia-García, E. Fernández-Media, A. Toval, and M. Piattini. *Revisión sistemática y comparación de ontologías en el marco de la seguridad*, in *IV congreso iberoamericano de seguridad en informática CIBSI 2007*. 2007. Mar de la Plata (Argentina).

39. Blanco, C., J. Lasheras, R. Valencia-García, E. Fernández-Medina, A. Toval, and M. Piattini. *A Systematic Review and Comparison of Security Ontologies*, in *International Conference on Availability, Reliability and Security (ARES)*. 2008. Barcelona, IEEE Computer Society
40. Blanco, C., J. Lasheras, R. Valencia-García, E. Fernández-Medina, A. Toval, and M. Piattini, *Security Ontologies: a sistematic review and comparison, Technical Report - UCLM-TSI-003*. 2008.
41. Blanco, C., J. Lasheras, R. Valencia-García, E. Fernández-Medina, A. Toval, and M. Piattini, *Basis for an integrated Security Ontology according to a systematic review of existing proposal*. Computer Standards & Interfaces, 2011. **33**(4): p. 372-388
42. Kotonya, G. and I. Sommerville, *Requirements Engineering. Processes and Techniques*. 1998, John Wiley & Sons.
43. Damian, D. and J. Chisan, *An Empirical Study of the Complex Relationships between Requirements Engineering Processes and Other Processes that Lead to Payoffs in Productivity, Quality, and Risk Management.* IEEE Transaction Software Engineering, 2006. **32**(7): p. 433-453.
44. Glass, R.L., *Software Runaways: Monumental Disasters*. 1998, Prentice Hall.
45. Glass, R.L., *Software Engineering: Facts and Fallacies*. 2002, Addison-Wesley.
46. Charette, R., *Why Software Fails.* IEEE Spectrum 2005: p. 36-43.
47. StandishGroup. *The Chaos Report.* http://www.standishgroup.com/ 2009 [cited 2009].
48. Mili, H., F. Mili, and A. Mili, *Reusing Software: Issues and Research Directions.* IEEE Transactions on Software Engineering, 1995. **21**(6): p. 528-562.
49. Cybulsky, J.L. and K. Reed. *Requirements Classification and Reuse: Crossing Domains Boundaries*, in *6th International Conference on Software Reuse (ICSR'2000)*. 2000. Viena, Springer, LNCS.
50. Sommerville, I., *Ingeniería del Software (7 edición)*. 2005, Madrid - Addison-Wesley.
51. Nuseibeh, B. and S. Easterbrook, *Requirements engineering: A roadmap.* ACM Press. Future of Software Engineering, 2000.
52. ISO/IEC27000. *ISO/IEC 27000 Information technology* http://www.27000.org/. 2009 [cited 2011].
53. ISO27001, *ISO/IEC 27001 Information technology - Security techniques - Information security management systems - Requirements*. 2005.
54. ISO27002, *ISO/IEC 17799-27002 Code of Practice for Information Security Management.* 2005.
55. ISO15408, *ISO/IEC 15408 (Common Criteria) Information Technology Security Techniques-Evaluation Criteria for IT Security*. 2009.
56. ISO21827, *ISO/IEC 21827:2008 Information technology - System Security Engineering - Capability Maturity Model (SSE-CMM)*. 2008.
57. IEEE, *IEEE P1704 Standard for Developing Software Project Life Cycle Processes*. 1997.
58. Biszick-Lockwood, B., *IEEE P1074-2005: Roadmap for Optimizing Security in the System and Software Life Cycle, QualityIT Redmond.* 2006.
59. COBIT. *IT_Governance_Institute. Control Objectives for Information and related Technology (COBIT 4.1),* http://www.isaca.org/Knowledge-Center/COBIT/Pages/Overview.aspx. 2005 [cited 2011].
60. Villarroel, R., F.-M. Eduardo, and M. Piattini, *A Comparison of Secure Information Systems Design Methodologies.* CAiSE 04 (The 16th Conference on Advanced Information Systems Engineering), 2004. **1**: p. 189-198.
61. BSI7799-3, *British Standard 7799-3: 2006, Information security management systems - Part 3: Guidelines for information security risk management, BSI.* 2006.
62. Mead, N.R., *Identifying Security Requirements Using the Security Quality Requirements Engineering (SQUARE) Method*, in *Integrating Security and Software Engineering: Advances and Future Visions*. 2007, Idea Group Publishing. p. 44-69.

63. Fernández, E.B., M.M. Larrondo-Petrie, T. Sorgente, and M. Vanhilst, *A Methodology to Develop Secure Systems Using Patterns.*, in *Integrating Security and Software Engineering: Advances and Future Visions*. 2007, Idea Group Publishing. p. 107-126.
64. CRAMM. *United Kingdom Central Computer and Telecommunication Agency. CCTA Risk Analysis and Management Method: User Manual, ver. 5.2. HMSO.* http://www.cramm.com/. 2005 [cited 2011].
65. OCTAVE. *Operationally Critical Threat, Asset and Vulnerability Evaluation (OCTAVE). V 2.0, Carnegie Mellon - Software Engineering Institute,* http://www.cert.org/octave/. 1999 [cited 2011].
66. Alberts, C. and A. Dorofee, *Managing information security risks: The OCTAVE (SM) approach*. 2002, Boston, Addison Wesley.
67. Mayer, N., A. Rifaut, and E. Dubois. *"Towards a Risk-Based Security Requirements Engineering Framework"*, in *11th International Workshop on Requirements Engineering: Foundation for Software Quality (REFSQ'05)*. 2005. Porto, Portugal.
68. Martínez, C., *Guía de Implantación de Sistemas de Gestión de la Seguridad de la Información. http://www.forosec.com/*, in *Dentro del proyecto: FOROSEC: FORO para la Seguridad de los Sistemas Informáticos*. 2005.
69. Haley, C.B., R. Laney, J.D. Moffett, and B. Nuseibeh, *Arguing Satisfaction of Security Requirements*, in *Integrating Security and Software Engineering: Advances and Future Visions*. 2007, Idea Group Publishing. p. 16-43.
70. Chung, L., B.A. Nixon, E. Yu, and J. Mylopoulos *Non-Functional Requirements in Software Engineering*. The Kluwer International Series in Software Engineering. 2000, Boston, Kluwer Academic Publishers.
71. Lamsweerde, A. *Elaborating Security Requirements by Construction of Intentional Anti-Models*, in *26th International Conference on Software Engineering*. 2004. Edinburgh, ACM-IEEE.
72. Mouratidis, H., P. Giorgini, and G. Manson. *Integrating Security and System Engineering: Towards the Modelling of Secure Information Systems*, in *15th Conference on Advanced Information Systems Engineering (CAiSE'03): 63-78*. 2003.
73. Kim, H.-K. *Automatic Translation Form Requirements Model into Use Cases Modeling on UML*, in *ICCSA 2005, LNCS: 769-777*. 2005.
74. Firesmith, D.G., *Security Use Cases.* Journal of Object Technology, 2003. **2(3)**: p. 53-64.
75. Jennex, M.E. *Modeling security requirements for information systems development*, in *SREIS 2005*. 2005.
76. Zuccato, A., *Holistic security managements framework applied in electronic commerce.* Computers and Security, 2007. **26**(3): p. 256-265.
77. Myagmar, S., A.J. Lee, and W. Yurcik *Threat Modeling as a Basis for Security Requirements*, in *SREIS 2005*. 2005.
78. Peeters, J. *Agile Security Requirements Engineering*, in *SREIS 2005*. 2005.
79. Opdahl, A.L. and G. Sindre, *Experimental comparison of attack trees and misuse cases for security threat identification.* Information and Software Technology, 2009. **51**(5): p. 916-932.
80. Basin, D., J. Doser, and T. Lodderstedt, *Model driven security: From UML models to access control infrastructures.* ACM Trans. Softw. Eng. Methodol., 2006. **15**(1): p. 39-91.
81. Jürjens, J. *UMLsec: extending UML for secure systems development*, in *UML 2002 - The Unified Modeling Language. Model Engineering, Languages,Concepts, and Tools. 5th International Conference. LNCS 2460: 412-425*. 2002.
82. Popp, G., J. Jürjens, G. Wimmel, and R. Breu. *Security-Critical System Development with Extended Use Cases*, in *10th Asia-Pacific Software Engineering Conference: 478-487*. 2003.
83. Jürjens, J., J. Schreck, and Y. Yu. *Automated Analysis of Permission-Based Security Using UMLsec*, in *Fundamental Approaches to Software Engineering (FASE 2008)*. 2008.

84. Jürjens, J. and S.H. Houmb. *Risk-Driven Development Of Security-Critical Systems Using UMLsec*, in *IFIP Congress Tutorials*. 2004.
85. Bresciani, P., P. Giorgini, F. Giunchiglia, J. Mylopoulos, and A. Perini, *Tropos: Agent-Oriented Software Development Methodology.* Journal of Autonomous Agents and Multi-Agent System, 2004. **8**: p. 203-236.
86. Massacci, F., 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, 2005. **27**: p. 445-455.
87. Mouratidis, H. and P. Giorgini, *Secure Tropos: a Security-Oriented Extension of the Tropos Methodology.* International Journal of Software Engineering and Knowledge Engineering, 2007. **17**(2): p. 285-309.
88. Mouratidis, H., *Secure Tropos: An Agent Oriented Software Engineering Methodology for the Development of Health and Social Care Information Systems.* International Journal of Computer Science and Security, 2009. **3**(3): p. 241-271.
89. Mellado, D., 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, 2007. **29(2)**: p. 244 - 253.
90. Mellado, D., E. Fernández-Media, and M. Piattini, *Towards security requirements management for software product lines: A security domain requirements engineering process.* Computer Standards & Interfaces, 2008. **30**(6): p. 361-371.
91. Mellado, D., E. Fernández-Media, and M. Piattini, *Security requirements engineering framework for software product lines.* Information & Software Technology, 2010. **52**(10): p. 1094-1117.
92. Gruber, T., *Towards Principles for the Design of Ontologies used for Knowledge Sharing.* International Journal of Human-Computer Studies, 1995. **43**(5/6): p. 907-928.
93. Fernández-Breis, J.T. and R. Martínez-Béjar, *A cooperative framework for integrating ontologies.* International Journal of Human-Computer Studies, 2002. **56**: p. 665-720.
94. Gruninger, M. and J. Lee, *Ontology Applications and Design.* Communications of the ACM, 2002. **45**(2): p. 39-41.
95. Dobson, G. and P. Sawyer, *Revisiting Ontology-Based Requirements Engineering in the age of the Semantic Web.* International Seminar on "Dependable Requirements Engineering of Computerised Systems at NPPs", Institute for Energy Technology (IFE), Halden, 2006.
96. Blanco, C., J. Lasheras, R. Valencia-García, E. Fernández-Medina, A. Toval, and M. Piattini, *Ontologías de Seguridad: una revisión sistemática y comparativa, Informe Técnico - UCLM-TSI-003*. 2008.
97. Karyda, M., T. Balopoulos, L. Gymnopoulos, S. Kokolakis, C. Lambrinoudakis, S. Gritzalis, and S. Dritsas, *An ontology for secure e-government applications.* First International Conference on Availability, Reliability and Security (ARES'06). IEEE Computer Society, 2006: p. 1033-1037.
98. Undercoffer, J., A. Joshi, and J. Pinkston. *Modeling Computer Attacks: An Ontology for Intrusion Detection*, in *The Sixth International Symposium on Recent Advances in Intrusion Detection*. 2003, Springer.
99. Giorgini, P., M. Haralambos, and N. Zannone, *Modelling Security and Trust with Secure Tropos.*, in *Integrating Security and Software Engineering: Advances and Future Visions*. 2007, Idea Group Publishing. p. 160-189.
100. Kim, A., J. Luo, and M. Kang. *Security Ontology for Annotating Resources*, in *4th International Conference on Ontologies, Databases, and Applications of Semantics (ODBASE'05)*. 2005. Agia Napa, Cyprus.
101. Kim, A., J. Luo, and M. Kang, *Security ontology to facilitate web services description and discovery.* Journal on data semantics IX, 2007: p. 167-195.
102. Tsoumas, B., P. Papagiannakopoulos, S. Dritsas, and D. Gritzalis, *Security-by-Ontology: A Knowledge-Centric Approach.* Security and Privacy in Dynamic Environments, IFIP International Federation for Information Processing, 2006. **201**: p. 99-110.

103. Dritsas, S., V. Dritsou, B. Tsoumas, P. Constantopoulos, and D. Gritzalis, *OntoSPIT: SPIT management through ontologies*. Computer Communications, 2009. **32**(1): p. 203-212
104. Fenz, S. and E. Weippl. *Ontology based IT-security planning*, in *12th Pacific Rim International Symposium on Dependable Computing PRDC '06*. 2006, IEEE Computer Society.
105. Fenz, S., G. Goluch, A. Ekelhart, B. Riedl, and E. Weippl. *Information Security Fortification by Ontological Mapping of the ISO/IEC 27001 Standard*, in *accepted for the Proceedings of the 13th IEEE Pacific Rim International Symposium on Dependable Computing*. 2007.
106. Ekelhart, A., S. Fenz, and T. Neubauer. *T. AURUM: A Framework for Supporting Information Security Risk Management* in *42nd Hawaii International Conference on System Sciences, HICSS2009, IEEE Computer Society*. 2009.
107. Fenz, S. and A. Ekelhart. *Formalizing information security knowledge*, in *ASIACCS '09: Proceedings of the 2009 ACM symposium on Information, computer and communications security, ACM*. 2009.
108. Lee, S.-W., R. Gandhi, D. Muthurajan, D. Yavagal, and G.-J. Ahn. *Building problem domain ontology from security requirements in regulatory documents*, in *International workshop on Software engineering for secure systems*. 2006. Shanghai, China, ACM Press.
109. Nicolás, J., B. Moros, J. Lasheras, and A. Toval, *SIREN: Un Método Práctico de Ingeniería de Requisitos Basado en Reutilización. Informe Técnico - UMU-TR DIS 1-2009*.
110. Breaux, T.D. and A.I. Antón, *Analyzing Regulatory Rules for Privacy and Security Requirements*. IEEE Transactions on Software Engineering, 2008 **34**(1): p. 5-20.
111. LOPD, *Ley Orgánica 15/1999, de 13 de diciembre, de Protección de Datos de Carácter Personal en España*. BOE 298, 1999.
112. RMS, *Real Decreto 994/1999, de 11 de junio, Reglamento de medidas de seguridad de los ficheros automatizados que contengan datos de carácter personal*. BOE 151, 1999.
113. Lozano-Tello, A. and A. Gómez-Pérez, *ONTOMETRIC: A Method to Choose the Appropriate Ontology*. Journal of Database Management. Special Issue on Ontological analysis, Evaluation, and Engineering of Business Systems Analysis Methods, 2004. **15**(2): p. 1-18.
114. IEEE, *Std 830-1998 Guide to Software Requirements Specifications* in *Volume 4: Resource and Technique Standards*. 1999, IEEE Software Engineering Standards Collection.
115. IEEE, *Std 1233-1998 Guide for Developing System Requirements Specifications*, in *Volume 1: Customer and Terminology Standards*. 1999, IEEE Software Engineering Standards Collection.

7 ANEXO I – PUBLICACIONES EN JCR

Relevancia

La normativa aplicable para la presentación de esta tesis en el formato de compendio de publicaciones implica la presentación de al menos tres artículos publicados o aceptados para su publicación en revistas indexadas en bases de datos internacionales de reconocido prestigio. Este es el caso de las revistas indexadas con índice de impacto en el listado ISI/JCR. El doctorando presentó cuatro publicaciones [32, 35, 37, 41] incluidas en este listado, habiéndose desarrollado también un quinto artículo complementario [28] durante la realización de la tesis doctoral que también aparece en dicho listado y que, igualmente, se incluye en la siguiente relación.

Las revistas *Computer Standards and Interfaces* y *Requirements Engineering Journal (REJ)* son revistas bien conocidas que llevan publicándose varios años y que en los últimos años han visto incrementado su prestigio entre la comunidad de Ingeniería del Software. Prueba de ello es su aumento en cuanto a factor de impacto en el listado ISI/JCR, apareciendo ambas en el segundo tercio, concretamente en las posiciones 37/93 (Q2 cuartil) y 61/93 (Q3 cuartil) de la categoría *Computer Science, Software Engineering*, con índices de impacto 1,373 y 0,931, respectivamente, en el listado correspondiente al año 2009. Por otra parte las revistas *Journal of Research and Practice in Information Technology (JRPIT*) y la *International Journal of Computer Systems Science and Engineering* llevan publicándose desde hace varias décadas (desde 1968 y 1986 respectivamente) y aparecen en la posición 81/93 (Q4 cuartil) de la categoría *Computer Science, Software Engineering* y la posición 90/92 (Q4 cuartil) en la categoría *Computer Science, Theory and Methods*, con índices de impacto de 0,5 y 0,222 respectivamente, en el listado correspondiente al año 2009.

A continuación, siguiendo lo indicado en la normativa, se muestran el texto completo de los cinco artículos mencionados.

7.1 Eight Key Issues for an Effective Reuse-Based Requirements Process

The experience gained through the SIREN method definition and application (Toval, Nicolás et al. 2002; Toval, Olmos et al. 2002), and research on the subject, has led us to identify eight key issues that, in our opinion, have to be taken into account for any reuse-based requirements method to succeed. These key issues are collected, described and justified homogeneously in the paper. Moreover, since these issues should be supported by a CARE (*Computer-Aided Requirements Engineering*) tool, an analysis of the state of the art of some of the most popular commercial tools with respect to requirements reuse was carried out, and the conclusions are presented in this paper. The study revealed the lack of requirements reuse support of the commercial tools analyzed and this led us to propose SirenTool as a solution. The key issues have been validated in a real case study in the context of clinical history management in the intensive care unit of a hospital.

In our view, the above contributions may be of interest to CARE tool builders (concerned with the inclusion of reuse features in their tools), researchers in the field and practitioners involved in the adoption of reuse in their requirements engineering processes.

This paper has been structured as follows: Section 2 briefly presents the SIREN method. Section 3 explains the key issues identified for a practical reuse-based requirements process. Section 4 summarizes the study of the CARE tools analyzed. Section 5 shows how the key issues have been implemented in SirenTool. Section 6 presents other approaches related to requirements reuse. Finally, Section 7 gives the conclusions.

2. THE SIREN METHOD

Like (Robertson and Robertson 2006), we believe that by starting from a set of requirements which have been specified for other projects or domains, we can improve the precision and efficiency of the requirements specification for the current project and we can also reduce the time necessary to elaborate this specification. However, systematic approaches to requirements reuse are still scarce (see Section 6). In order to explore the benefits of requirements reuse, we proposed SIREN (*Shopic Reuse of softgarE requiremeNts*) as a practical way of dealing with requirements reuse. In this section we summarize the main features of SIREN, a detailed explanation can be found in (Toval, Nicolás et al. 2002).

SIREN encompasses a spiral process model, requirement's documents templates and a reusable requirements repository which is organized by catalogs. SIREN requirements catalogs correspond to a set of generic and reusable requirements named *profiles* ("horizontal" application domains, for instance, concerning security or personal data protection regulations) and *domains* ("vertical" application domains, for instance, insurance or banking). The separation of catalogs into profiles and domains is not relevant beyond helping to identify, organize and group sets of related requirements. These catalogs are organized in a hierarchy of requirements specification documents, which are structured according to IEEE standards (IEEE 1998a; IEEE 1998b). Some examples of catalogs developed in the context of SIREN are:

The *PDP (Personal Data Protection) profile catalog* (Toval, Olmos et al. 2002), with the requirements coming from the Spanish Personal Data Protection Constitutional Law (This Law is an adaptation of the EU Directive 95/46/CE (EU 1995)).

The *Security profile catalog* (Toval, Nicolás et al. 2002), with the requirements coming from MAGERIT (MAGERIT is the information systems risk analysis and management method of the Spanish public administration. It is based on ISO/IEC 15408-1999 Evaluation Criteria for Information Technology Security Standard, also known as the Common Criteria Framework - CCF).

The *TOS (Teleoperated Systems) domain catalog* (Nicolás, Lasheras et al. 2006), modeling the requirements of the product line of teleoperated systems (basically, robotics systems) for ship hull maintenance operations, such as cleaning or painting the ship hull.

Each requirement is labeled with a *type* in the catalogs. A type denotes the catalog the requirement comes from and the requirements specification document where the requirement is included. For example, the types SYRSP and SRSP refer to requirements contained in the SyRS (*System Requirements Specification*) and SRS (*Software Requirements Specification*) documents within the PDP (P) catalog, respectively.

In SIREN, the textual information of a requirement is complemented by a set of attributes. There is a set of attributes common to all requirements (including *priority*, *rationale*, *source*, *state*, etc.), although additional attributes can be defined depending on the type of the requirement. For example, both SRSP and SYRSP types include an additional attribute called *security level*.

Besides the attributes, different traceability relationships can be defined to relate requirements. These are inclusive, exclusive and parent–child traceability relationships (see issue K5 in Section 3). In SIREN we also have *parameterized requirements*, which contain some parts that have to be adapted to each application or system and that have to be instantiated when it is reused.

The SIREN process model is an adaptation of the spiral process model proposed by (Kotonya and Sommerville 1998), but it includes the repository as a central element of the process. Consequently, new activities appear, such as *Requirements Selection* and *Repository Improvement*, and others are adapted to the new circumstances, such as the *Requirements Elicitation* (renamed as *Specific Requirements Elicitation* in SIREN) and the *Analysis and Negotiation* activities (see Fig. 1):

> *Requirements selection.* The approach for reuse consists of providing the requirements engineer with the specification documents templates, with the requirements filled in, and which are stored in the repository. The requirements engineers together with the other stakeholders (e.g. users, clients, developers, etc.) will reuse those requirements which are suitable for the specific application that is being developed by instantiating them when needed.
>
> *Specific requirements elicitation.* In the meetings with the rest of the stakeholders, the requirements engineers gather the informal and specific requirements of the current proj-

7.2 An Integrated Domain Analysis Approach for Teleoperated Systems

Requirements Eng (2009) 14:27–46
DOI 10.1007/s00766-008-0072-6

ORIGINAL ARTICLE

An integrated domain analysis approach for teleoperated systems

Joaquín Nicolás · Joaquín Lasheras · Ambrosio Toval · Francisco J. Ortiz · Bárbara Álvarez

Received: 6 August 2008 / Accepted: 3 December 2008 / Published online: 15 January 2009

Abstract Teleoperated systems for ship hull maintenance (TOS) are robotic systems for ship maintenance tasks, such as cleaning or painting a ship's hull. The product line paradigm has recently been applied to TOS, and a TOS reference architecture has thus been designed. However, TOS requirements specifications have not been developed in any rigorous way with reuse in mind. We therefore believe that an opportunity exists to increase the abstraction level at which stakeholders can reason about this product line. This paper reports an experience in which this TOS domain was analyzed, including the lessons learned in the construction and use of the TOS domain model. The experience is based on the application of extensions of well-known domain analysis techniques, together with the use of quality attribute templates traced to a feature model to deal with non-functional issues. A qualitative research method (action research) was used to carry out the experience.

Keywords Domain analysis · Product line requirements engineering · Feature modelling · Generic use cases · Teleoperated systems · Action research

J. Nicolás (✉) · J. Lasheras · A. Toval
Software Engineering Research Group,
Departamento de Informática y Sistemas,
Facultad de Informática, Universidad de Murcia,
Campus de Espinardo, 30071 Murcia, Spain
e-mail: jnr@um.es

F. J. Ortiz · B. Álvarez
Systems and Electronic Engineering Division,
Universidad Politécnica de Cartagena,
30202 Cartagena, Spain

1 Introduction

Teleoperated systems for ship hull maintenance (hereafter TOS) are robotic systems which are extremely useful in maintenance tasks such as cleaning or painting a ship's hull [1, 2]. Recent years have seen the development of a software reference architecture in the TOS domain [3]. Since TOS usually share a high number of common capabilities, this generic architecture provides a common framework for the reuse of software artefacts (*assets*). These systems can thus be considered to constitute a product line (or product family), i.e. they are a set of software-intensive systems which share a common, managed set of features that satisfy the specific needs of a particular market segment [4]. A comprehensive review of software product lines current practice is provided by van der Linden et al. [5] and Pohl et al. [6], while a recent vision of the challenges in the research in software product lines has been compiled by Käkölä and Dueñas [7].

Rine and Nada [8] have shown empirically that the level of reuse determines the effectiveness of the improvements in productivity, quality and time-to-market, and they conclude that greater benefits are obtained when reuse is considered during the early phases of the software development lifecycle. In TOS, in contrast, requirements specifications have not been developed rigorously with reuse in mind.

In this context, we propose the construction of new products in the TOS product line from a higher abstraction level—from the product line requirements instead of from its generic architecture—by developing what can be intuitively seen as a TOS *domain model* (the meaning of this and some related terms is discussed more rigorously in Sect. 4).

This paper presents an experience in analyzing the TOS domain, together with a set of lessons learned, and results from a TOS domain model that serves to (1) accelerate

7.3 A Personal Data Audit Method through Requirements Engineering.

Computer Standards & Interfaces 32 (2010) 166–178

Contents lists available at ScienceDirect

Computer Standards & Interfaces

journal homepage: www.elsevier.com/locate/csi

A Personal Data Audit Method through Requirements Engineering

Miguel A. Martínez [a,*], Joaquín Lasheras [a], Eduardo Fernández-Medina [b], Ambrosio Toval [a], Mario Piattini [b]

[a] Software Engineering Research Group, Computer and Systems Department, University of Murcia, Campus de Espinardo, 30071, Murcia, Spain
[b] ALARCOS Research Group, Information Systems and Technologies Department, UCLM-Soluziona Research and Development Institute, University of Castilla-La Mancha, Paseo de la Universidad, 4-13071, Ciudad Real, Spain

ARTICLE INFO

Article history:
Received 23 January 2008
Received in revised form 11 December 2009
Accepted 6 January 2010
Available online 18 January 2010

Keywords:
Privacy
Data protection
Audit
Requirements Engineering
Health Information Systems

ABSTRACT

Organizations using personal data in areas such as in Health Information Systems have, in recent years, shown an increasing interest in the correct protection of these data. It is not only important to define security measures for these sensitive data, but also to define strategies to audit their fulfilment. Although standardisation organisations have defined recommendations and standards related to security and audit controls, no methodological frameworks proposing the audit of these sensitive data have been described. This paper presents a methodology with which to audit personal data protection, using Requirements Engineering and based on CobiT. This methodology has been validated in four real case studies.

Contents

1. Introduction

Information Systems (IS) audit is defined as the systematic process of gathering, grouping and evaluating evidence to determine whether an IS safeguards the assets, maintains the integrity of the data, effectively carries out the aims of the organization and uses resources efficiently [1]. A special type of audit within this discipline is the software audit, whose purpose is to verify that both functional and non-functional requirements are accomplished.

According to ISO 7498-2:1989 [2], a security audit is: "an independent review and examination of system records and operations in order to test for adequacy of system controls, to ensure compliance with established policy and operational procedures, to detect breaches in security, and to recommend any indicated changes in control, policy, and procedures". A security audit may include many aspects, such as the level to which facilities or people are protected. In this paper, we focus on the security related to data and information of a personal nature (privacy), which plays a decisive role in the security

* Corresponding author.
E-mail addresses: mmart@um.es (M.A. Martínez), jplave@um.es (J. Lasheras), Eduardo.FdezMedina@uclm.es (E. Fernández-Medina), atoval@um.es (A. Toval), Mario.Piattini@uclm.es (M. Piattini).

0920-5489/$ – see front matter © 2010 Elsevier B.V. All rights reserved.
doi:10.1016/j.csi.2010.01.001

7.4 Modelling Reusable Security Requirements Based on an Ontology Framework

Modelling Reusable Security Requirements based on an Ontology Framework

Joaquín Lasheras, Rafael Valencia-García, Jesualdo Tomás Fernández-Breis and Ambrosio Toval

Department of Informatics and Systems, University of Murcia.
30071 Campus de Espinardo, Murcia, Spain.
{jolave,valencia,jfernand,atoval}@um.es

In recent years, security in Information Systems (IS) has become an important issue, and needs to be taken into account in all stages of IS development, including the early phase of Requirements Engineering (RE). Reuse of requirements improves the productivity and quality of software process and products. This can be facilitated by Semantic Web technologies. We describe an ontology-based framework for representing and reusing security requirements based on risk analysis. A risk analysis ontology and a requirement ontology have been developed and combined to represent reusable security requirements formally and to improve security in IS by detecting incompleteness and inconsistency and achieving semantic processing in requirements analysis. This extensible framework is the basis on which to elaborate a "lightweight" method to elicit and specify security requirements, based on security standards.

Keywords: Security Requirements, Requirements Reuse, Risk Analysis, Ontologies

ACM Computing Classification System: D.2.1 Requirements/Specifications, D.2.13 Reusable Software, I.2.4 Knowledge Representation Formalisms and Methods

1. INTRODUCTION

Information confidentiality, security or privacy, issues of interest for Information System designers, are critical and vital matters for today´s society (Smith and Spafford, 2004). Hence, security has to be taken into account in all stages of the software development process (Devanbu and Stubblebine, 2000). These include the early phases related to *Requirements Engineering* (RE) (Jürjens, 2005), where, security has been identified as a research hotspot (Cheng and Atlee, 2007).

Security requirements include the types and levels of protection necessary for equipment, data, information, applications, and facilities to meet security policy. Specifically, security requirements are identified by *risk analysis* – *"the systematic use of information to identify sources and to estimate the risk"* ISO 27002 (2005). Risk analysis is one of the three sources identified by the security standard ISO 27002 – *"Code of Practice for Information Security Management"* (ISO27002, 2005) – to identify security requirements. The other two sources are related to the legal, regulatory and contractual requirements of an organization and to the principles, objectives and business requirements for information processing that an organization has developed to support its operations.

Manuscript received: 23 April 2008
Communicating Editor: Eduardo Fernandez-Medina Patón

7.5 Basis for an Integrated Security Ontology According to a Systematic Review of Existing Proposals

[68] J. Zhou, E. Niemela, P. Savolainen, An Integrated QoS-Aware Service Development and Management Framework, WICSA, 2007, p. 13.
[69] M. Sabou, et al., Evaluating the semantic web: a task-based approach, Proc. of ASWC/ISWC, 2007.
[70] A. Lozano-Tello, A. Gómez-Pérez, ONTOMETRIC: a method to choose the appropriate ontology, Journal of Database Management. Special Issue on Ontological analysis, Evaluation, and Engineering of Business Systems Analysis Methods 15 (2) (2004).
[71] A. Lozano-Tello, Métrica de idoneidad de ontologías. Ph.D. thesis, in Departamento de Informática, 2002, Universidad de Extremadura.
[72] M. Sabou, et al., Ontology selection: ontology evaluation on the real semantic web, Proc. of the EON Workshop, 2006.
[73] J. Brank, M. Grobelnik, D. Mladenic, A survey of ontology evaluation techniques, Proceedings of the Conference on Data Mining and Data Warehouses (SIKDD 2005), Citeseer, Ljubljana, Slovenia, 2005.
[74] MAGERIT, MAGERIT, Methodology for Information Systems Risk Analysis and Management, Available from: http://www.csi.map.es/csi/pg5m20.htm, 2006.
[75] E.F. Hill, Jess in Action: Java Rule-Based Systems, Manning Publications co, Greenwich, CT,USA, 2003.
[76] A. Hameed, D.H. Sleeman, A. Preece, Detecting mismatches in experts' ontologies acquired through knowledge elicitation, Research and Development in Intelligent Systems XVIII, Springer, 2001, pp. 9–22.
[77] N.F. Noy, M.A. Musen, The PROMPT suite: interactive tools for ontology merging and mapping, International Journal of Human Computer Studies 59 (2003) 983–1024.
[78] U. Reimer, Knowledge integration for building organizational memories, Proceedings of the 11th Banff Knowledge Acquisition for Knowledge Based Systems Workshop, 2, KM-6.1, KM-6.20, 1998.
[79] H.S. Pinto, J.P. Martins, Ontology Integration: how to perform the process, International Joint Conference on Artificial Intelligence, 2001.
[80] D.L. McGuiness, et al., An environment for merging and testing large ontologies, in: A. Cohn, F. Giunchiglia, B. Selman (Eds.), KR2000: Principles of Knowledge Representation and Reasoning, Morgan Kaufmann, San Francisco, USA, 2000, pp. 483–493.
[81] A. Gómez-Pérez, M. Fernández-López, O. Corcho, A. Gómez-Pérez, M. Fernández-López, O. Corcho, Ontological engineering, 1st edSpringer, London, 2004.
[82] M. Fernández, A. Gómez-Pérez, J. Pazos, Building a chemical ontology using METHONTOLOGY and the ontology design environment, IEEE Intelligent Systems 14 (1) (1999) 37–46.
[83] ISO/IEC, ISO/IEC 15408-1, Information technology. security techniques. evaluation criteria for ti security. Part 1: introduction and general model, ISO/IEC, Switzerland, 1999.
[84] C. Alberts, A. Dorofee, Managing information security risks: The OCTAVE (SM) approach, Addison Wesley, Boston, 2002.
[85] T.H. Duong, et al., Complexity analysis of ontology integration methodologies: a comparative study, Journal of Universal Computer Science 15 (4) (2009) 877–897.
[86] N.T. Nguyen, Advanced Methods for Inconsistent Knowledge Management, ed. L Springer-Verlag, 2008: Springer-Verlag, London.
[87] E. Sirin, B. Parsia, Pellet: An OWL DL reasoner, Proc. of the 2004 Description Logic Workshop (DL 2004), 2004, pp. 212–213.
[88] A. Toval, et al., Requirements reuse for improving information systems security: a practitioner's approach, Requirements Engineering Journal 6 (4) (2002) 205–219, Springer.
[89] A. Toval, A. Olmos, M. Piattini, Legal requirements reuse: a critical success factor for requirements quality and personal data protection, IEEE Joint International Conference on Requirements Engineering (ICRE'02 and RE'02), Essen, Germany, 2002.
[90] ISO/IEC, ISO/IEC 15408 (Common Criteria v3.0), Information technology security techniques-evaluation criteria for IT security, 2005.
[91] J. Lasheras, et al., An ontology-based framework for modelling security requirements, The 6th International Workshop on Security in Information Systems (WOSIS-2008), INSTICC Press, Barcelona (Spain), 2008.
[92] D. Geneiatakis, C. Lambrinoudakis, An ontology description for SIP security flaws, Computer Communications 30 (6) (2007) 1367–1374.
[93] L. Kagal, T. Finin, Modeling conversation policies using permissions and obligations, AAMAS workshop on Agent communication, LNCS, Springer-Verlag, 2005.
[94] J. Kwon, C.-J. Moon, Visual modeling and formal specification of constraints of RBAC using semantic web technology, Knowledge-Based Systems 20 (4) (2007) 350–356.
[95] Z. Maamar, N.C. Narendra, S. Sattanathan, Towards an ontology-based approach for specifying and securing Web services, Information and Software Technology 48 (7) (2006) 441–455.
[96] J. McGibney, N. Schmidt, A. Patel, A service-centric model for intrusion detection in next-generation networks, Computer Standards & Interfaces 27 (5) (2005) 513–520.
[97] J.J. Tan, S. Poslad, Dynamic security reconfiguration for the semantic web, Engineering Applications of Artificial Intelligence 17 (7) (2004) 783–797.
[98] B. Thuraisingham, Security standards for the semantic web, Computer Standards & Interfaces 27 (3) (2005) 257–268.

Carlos Blanco has an MSc in Computer Science from the University of Castilla-La Mancha. He is currently a PhD student and a member of the GSyA Research Group at the School of Computer Science at the University of Castilla-La Mancha (Spain). His research activity is in the field of security in Information Systems focused on Data Warehouses, OLAP tools, MDD and Ontologies. He is the author of several papers on these topics.

Joaquín Lasheras is a PhD student at the University of Murcia, in Spain. He received a degree in computer science from the University of Murcia. He is a member of the Software Engineering research group of the Department of Informatics and System (www.um.es/giisw) whose research manager is Professor José Ambrosio Toval Álvarez. His current research interests include requirements engineering, reuse, ontologies and security. He is involved in a variety of applied research and development projects with industry and networks related to security and quality.

Dr. Eduardo Fernández-Medina holds a PhD in Computer Science from the University of Castilla-La Mancha. He leads the GSyA Research Group of the Department of Computer Science at the University of Castilla-La Mancha. His research activity is in the field of security in databases, data warehouses, web services and information systems, and also in security metrics. Fernández-Medina is a co-editor of several books and book chapters on these subjects and has presented several dozens of papers at national and international conferences (DEXA, CAISE, UML, ER, etc.). He is the author of several manuscripts in national and international journals (DSS, ACM Sigmod Record, IS, IST, CbS, ISS, etc.) and belongs to various professional and research associations (AEC, ISO, IFIP WG 11.3, etc.).

Dr. Rafael Valencia-García received his BA, MSc and PhD degrees in Computer Science from the University of Murcia. He is a Lecturer at the Department of Informatics and Systems, University of Murcia. His main research interests are Natural Language Processing and the application of Knowledge Technologies such as ontologies. He has published over 25 articles in journals, conferences and book chapters. He is the author or coauthor of several books.

Dr Ambrosio Toval Álvarez is a full professor at the University of Murcia, in Spain. He holds a BS degree in Mathematics from the University Complutense of Madrid, and received a Ph.D. in Computer Science (cum laude) from the Technical University of Valencia (both in Spain). He is involved in a variety of applied research and development projects with industry and conducts research in the design and implementation of conceptual UML model verification, requirements engineering processes and computer-aided requirement engineering tools, and security requirements. Dr. Toval is currently the Head of the Software Engineering Research Group, at the University of Murcia.