# Hacia la Armonización de Modelos de Requisitos de Seguridad

Abel E. Fornaris[1], Eduardo Fernández-Medina[1]

[1] Grupo GSyA. Dep. de Tecnologías y Sistemas de Información.
Universidad de Castilla-La Mancha. Paseo de la Universidad 4, Ciudad Real, España.
(AbelEnrique.Fornaris, Eduardo.FdezMedina)@uclm.es

**Resumen.** La ingeniería de requisitos de seguridad se ha convertido en un campo vital en el desarrollo de sistemas informáticos. En la actualidad, se utilizan lenguajes, técnicas, artefactos, diagramas; resumidos todos como *modelos*, para la representación de requisitos de seguridad desde las etapas más tempranas del desarrollo. En el presente trabajo se describen los pasos a seguir en un proceso de armonización de dichos modelos. Desde la identificación de la necesidad de armonización, hasta la estructura de una propuesta inicial para llevarla a cabo, basados en los conceptos de la Ingeniería Dirigida por Modelos (MDE). Como parte de esos pasos, se hace especial hincapié en la presentación de los resultados de una revisión sistemática de la literatura en el campo tratado; así como los de una comparación entre los modelos obtenidos, orientada a justificar la necesidad y viabilidad del proceso armonizador.

**Palabras Claves:** Requisitos de seguridad; Técnicas, modelos, lenguajes, diagramas, artefactos; Revisión Sistemática de la Literatura, MDE, MDS; Armonización.

## 1 Introducción

El uso creciente de computadoras ha significado que los activos más importantes de las empresas son cada vez en mayor medida, almacenados y manipulados a través de sistemas informáticos. La incidencia del mal uso de esos activos también ha aumentado debido al acceso mundial de Internet y la automatización de sistemas [1].
En este contexto, la información y la seguridad de los sistemas de información se han convertido en un tema fundamental y el objetivo de innumerables estudios. La comunidad de ingeniería de seguridad ha desarrollado un conjunto de estrategias para asegurar las aplicaciones. Sin embargo, las mismas han sido enfocadas muchas veces a las etapas de diseño e implementación, cuando es absolutamente imprescindible que los sistemas informáticos sean asegurados apropiadamente desde su misma concepción [2]. El costo de tratar los temas de seguridad en etapas posteriores del desarrollo crece mucho en comparación con su tratamiento en la fase de requerimientos del sistema.
En el campo de la Ingeniería Dirigida por Modelos (MDE) [3], este problema no es tratado eficientemente tampoco. La seguridad, como parte de los requisitos no

funcionales, debe ser integrada en el modelo del sistema desde el principio si se quieren aprovechar todas las ventajas que dicho campo otorga. Sin embargo, muchos estudios existentes basados en MDE, tienden a ignorar a la seguridad en la etapa de modelado, integrando sus aspectos luego de una forma no nativa. Esto puede llevar a deficiencias en la seguridad y procesos de desarrollo con bajo rendimiento costo-beneficio [4].

Aún así, se han desarrollado algunas iniciativas para integrar la seguridad dentro de MDE. Por ejemplo, la Seguridad Dirigida por Modelos (MDS) [5] aporta tres nuevos aspectos: (i) los modelos del sistema son enriquecidos con primitivas y reglas para integrar la seguridad en el proceso de desarrollo, (ii) las técnicas de transformación de modelos se extienden de forma tal que estos detalles de seguridad también son transformados, y (iii) se obtiene el sistema incluyendo propiedades de seguridad y sus correspondiente mecanismos de seguridad [6].

En los últimos años, la importancia del desarrollo seguro desde las etapas tempranas ha significado que la ingeniería de requisitos de seguridad (RS) se haya vuelto un área estudiada a través de la cual producir metodologías, procesos y marcos de trabajo que puedan integrar estos requisitos no funcionales con el resto [7].

Lenguajes, técnicas, artefactos, diagramas, que serán llamados *modelos* en general de ahora en adelante, son definidos para representar estos RS y brindar el primer paso con el cual propagarlos a través de todas las etapas del desarrollo. Sin embargo, existen problemas en ellos, lo que ha llevado a la identificación de la necesidad de lograr una propuesta integradora en este sentido. En esta línea de trabajo, en primer lugar se impone la necesidad de identificar posibles trabajos anteriores que incluyan estudios relativos al área.

A lo largo de los últimos años, un buen número de estudios se han enfocado en los RS y algunos han llevado a cabo revisiones sistemáticas de la literatura (RSL) en este tema. Sin embargo, la mayoría de estas revisiones se han enfocado en los procesos definidos para la representación de los RS y no hacen un análisis profundo de los modelos que para ese propósito aplican o definen [8]. Después de realizar búsquedas preliminares dirigidas a identificar revisiones existentes y evaluando el volumen de estudios potencialmente relevantes, podemos señalar varios trabajos que contienen resúmenes o comparaciones de modelos de seguridad, tales como [9-18]. No obstante, ninguno de ellos llevan a cabo una revisión basados en modelos de requisitos de seguridad de una forma sistemática, es decir, ninguno ha realizado una RSL de los mismos. Además, han identificado modelos de seguridad y criterios de comparación orientados a objetivos diferentes a los propuestos en este estudio. Por tanto, no son un contexto suficientemente bueno para operar en esta área.

En contraste con una revisión de la literatura más común, una RSL se lleva a cabo de forma sistemática. Esto significa que el proceso de investigación de forma sistemática sigue una secuencia de pasos metodológicos muy bien definida y estricta de acuerdo a un protocolo previamente desarrollado [19].

En el presente trabajo se presentarán en profundidad los primeros pasos de un proceso de armonización de modelos de seguridad. Dichos pasos se han definido basados en la propuesta de Pardo et al. [20], adaptada a nuestro proceso.

Se hace especial hincapié en los resultados de una RSL acerca de estos modelos de requisitos de seguridad a partir de la cual resumir las iniciativas en este campo. Además, se definen y aplican criterios de comparación entre ellos para finalmente

justificar la necesidad de la integración de los modelos. Los conceptos de MDE emergen como alternativa en este proceso por las bondades de trabajar con modelos y sus transformaciones automáticas.
En este caso, se ha llevado a cabo la RSL usando las directrices que para ellas han sido definidas por Kitchenham et al. [21], aunque de manera práctica ha sido implementada usando la plantilla descrita por Biolchini et al. [19], la cual facilita la planificación y ejecución de la RSL a través de una estricta serie de pasos definidos.
El resto del trabajo se ha organizado de la siguiente manera: en la Sección 2 se presenta un extracto del protocolo de revisión seguido para llevar a cabo la RSL y los resultados obtenidos, llámense las diferentes propuestas de modelos nuevos para la representación y obtención de los RS. La Sección 3 está compuesta de un análisis de los resultados a través de una comparación y discusión acerca del tema. En la Sección 4, a modo de resumen, se presenta la estructura de los pasos seguidos en el proceso de armonización, incluyendo además una aproximación a la definición de los elementos finales del proceso. Por último, las conclusiones del trabajo serán expuestas en la Sección 5.

## 2 Planificación y Ejecución de la Revisión.

Como se ha afirmado previamente, la RSL fue planeada y llevada a cabo siguiendo una plantilla. Un resumen de ambas fases - protocolo de revisión y resultados – se presenta a continuación.

### 2.1 Planificación de la Revisión

El *foco de la pregunta* fue identificar los lenguajes, técnicas, artefactos, diagramas, llamados *modelos* en general existentes y que son usados para la representación y obtención de requisitos de seguridad desde las etapas tempranas del desarrollo del software. Los estudios conteniendo resúmenes y/o comparaciones de estos modelos también fueron identificados y descritos.
El *problema* es que las metodologías, procesos y marcos de trabajo existentes en la ingeniería de requisitos de seguridad tienden a usar diferentes modelos para la representación de dichos requisitos. Esto conduce a la necesidad del estudio de un gran número de ellos. Además, ellos definen distintos requisitos de seguridad y formas de tratarlos dependiendo de su definición y objetivo; algunos de ellos incluso carecen de muchos RS importantes, ya que se orientan a unos en particular. Una propuesta a partir de la cual descubrir o definir un conjunto de modelos comunes o enlaces verificables entre ellos, con el objetivo de constituir un modelo estándar en esta área, podría ser una solución a este problema.
Solo fueron consideradas como *criterio de inclusión* las publicaciones que proponen y describen nuevos modelos o aproximaciones a estos, asociados a la ingeniería de requisitos de seguridad o aquellos que exponen resúmenes o comparaciones entre ellos. Por tanto, los resultados esperados a la conclusión de la RSL eran identificar,

describir y comparar los diferentes modelos, incluyendo sus resúmenes y comparaciones ya existentes en la literatura.
Las *áreas de mayor aplicación* que se beneficiarán de la RSL son el desarrollo de software seguro y la ingeniería de software, específicamente la ingeniería de requisitos de seguridad conjuntamente con expertos en seguridad e ingenieros de requisitos. También podrán obtener beneficios los investigadores y desarrolladores especializados en MDE. Se espera ofrecer una base adecuada para poder entender la extensión de modelos actualmente en uso en la ingeniería de RS así como la necesidad de su integración.
El criterio usado para seleccionar las *fuentes* de información de las cuales obtener estos estudios fue basado en la disponibilidad de artículos en Internet y/o en la biblioteca digital de la Universidad de Castilla-La Mancha. Bibliotecas digitales tales como Scopus, Springer, Science@Direct, ACM, así como IEEE Computer, fueron revisadas. Se definieron *cadenas de búsqueda* válidas para cada una, dependiendo de sus características, todas ellas en *idioma* inglés con el objetivo de obtener los mejores resultados. La base de dichas búsquedas es la siguiente cadena: "Security AND Requirement AND (Engineer OR Engineering) AND (Model OR Diagram OR Artifact OR Method OR Language OR Technique)". Se incluyeron además términos relativos a MDE luego.

## 2.2 Ejecución de la Revisión.

Después de realizadas las búsquedas en las fuentes seleccionadas, de identificaron los siguientes *estudios primarios*, cada uno de ellos alrededor de una propuesta de modelo. En la Tabla 1 y la Tabla 2 se presenta un resumen de estos resultados.

**Tabla 1.** Resumen de propuestas para la obtención directa de requisitos de seguridad

| Autor(es) | Definición y artefactos | Propuesta |
|---|---|---|
| Firesmith [22] | Casos de Uso estándar de UML que complementan los casos de mal-uso para representar los RS. | Casos de Uso de Seguridad |
| Jürjens [23] | Extensiones a varios artefactos UML con estereotipos, etiquetas y restricciones para representar los RS. | UMLsec |
| Lodderstedt, Basin et al. [24] | Diagrama de clases de UML estereotipado para representar los RS basados en roles. | SecureUML |
| Mouratidis and Giorgini [25] | Extensión a la metodología para representar RS como restricciones. Se basa en el lenguaje de modelado i*[26] y usa sus conceptos. | Tropos Seguro |
| Rosado et al. [27] | Perfil UML que usa casos de uso de UML estereotipados para seguridad en sistemas grid. | Perfil GridUCSec |
| Zannone [28], Massacci, Mylopoulos et al. [29] | Lenguaje de modelado basado en el lenguaje de modelado i* [26] para capturar RS desde el punto de vista organizacional. | SI* |
| Jackson [30], Hatebur, Heisel et al. [31] | Tipos de patrones para representar modelos de amenazas y los RS derivados de ellos. | Gráficos de Problemas de Seguridad |
| Jennex [32] | Método gráfico para identificar y modelar RS al señalar las amenazas combinadas con el concepto de defensa en profundidad. | Diagramas de análisis de barreras |
| Haley, Laney et al. [33] | Los RS se consideran en términos de restricciones que operacionalizan como objetivos de seguridad. Se deben identificar los activos involucrados en requisitos funcionales. | Restricciones |

**Tabla 2.** Resumen de propuestas para representar amenazas de seguridad y sus posibles contramedidas

| Autor(es) | Definición y artefactos | Propuesta |
|---|---|---|
| Sindre and Opdahl [34] | Casos de uso UML extendidos para representar comportamiento no deseado en el sistema. | Casos de mal-uso |
| Sindre [35] | Diagramas de actividad UML estándares para capturar actividades y actores maliciosos usando la misma semántica y sintaxis. | Diagramas de Mala-Actividad |
| McDermott and Fox [2] | Casos de uso UML estándar representando acciones perjudiciales al sistema completas. | Casos de Abuso |
| Hussein and Zulkernine [36] | Perfil UML para tratar con ataques al sistema usando varios artefactos UML estereotipados y etiquetados. | UMLintr |
| Zulkernine, Graves et al. [37] | Gráfico de estados UML para especificar ataques. Tienen la habilidad de representar ataques de múltiples pasos, lo que no es posible de obtener de otra forma. | Gráficos de Estados UML para Seguridad |
| Schneier [38] | Modelo para calcular el riesgo y costo de potenciales ataques y sus contramedidas, utilizando una estructura arbórea. | Árboles de Ataque |
| Lin et.al [1] | Permite representar amenazas de seguridad y derivar RS definiendo anti-requisitos como acciones para subvertir estos RS. | Gráficos de Abuso |
| Peeters [39] | Historias en texto plano de cómo los atacantes abusan del sistema. Son contrapartes ágiles a los casos de abuso. | Historias de Abuso |
| Graves and Zulkernine [40], Raihan and Zulkernine [41] | AsmL es un lenguaje extendido y finito de especificación software, ejecutable, basado en estados de máquina también usado para especificar escenarios de ataques en extensiones tales como AsmLSec | AsmL, AsmLSec |
| Ecckman, Vigna et al. [42] | Lenguaje independiente del dominio para la descripción de ataques que puede ser extendido de una forma bien definida para amoldarse a diferentes entornos de operación. | STATL |

Estos resultados muestran dos grandes áreas en el modelado de RS. Una que se refiere a la representación, documentación y obtención directa de los RS a través de los modelos. La otra contiene modelos para la especificación de ataques, amenazas y vulnerabilidades, con el objetivo de descubrir las contramedidas (reflejadas como RS) que serían necesarias para prevenirlos. Cada una de estas áreas se describe usando modelos como extensiones de lenguajes (mayormente UML) o definiendo nuevos modelos.

## 3 Discusión y Análisis de los Resultados.

En la Tabla 3 se presenta una comparación más profunda de las diferentes propuestas, siguiendo los criterios descritos a continuación:

- **ANS** significa "Alineación con Nivel Superior" y sus valores pueden ser "M" si el modelo es presentado como parte de una *metodología*, "P" si es parte de un *proceso* y "MT" si se define un *marco de trabajo* que lo usa.
- **DF** se refiere al grado de "Definición Formal" del modelo. Una "X" significa que no se ha encontrado ningún tipo de definición formal, "P" quiere decir que el modelo tiene una DF parcial y "*" se utiliza en los modelos que tienen definiciones formales fuertes, que pueden incluir metamodelos UML.
- **RTC** tiene que ver con la disponibilidad de especificaciones de "Restricciones" en el modelo. "*" significa Sí, mientras que "X" es No.
- **VE** quiere decir "Validación Empírica" y se refiere a las posibles encuestas, casos de estudio y/o experimentos usados con el fin de validar la propuesta. "X" significa que no se ha encontrado ninguna prueba empírica en la literatura relevante.

- **AES** (“Alineación con Estándares de Seguridad”) describe el cumplimiento o alineación del modelo con algún estándar internacional, por ejemplo ISO/IEC u otros. “X” implica que no se ha encontrado alineación con ningún estándar en la literatura relevante.
- **HA** se refiere a “Herramientas Automatizadas” encontradas para cada propuesta. “X” significa que no se ha encontrado ninguna herramienta asociada al modelo en la literatura relevante.
- **CC** (“Conciencia Científica”) mide el grado de “popularidad” de cada propuesta en la comunidad científica. Cada propuesta puede haber sido encontrada en la literatura relevante como parte de “C” una comparación o “R” un resumen, como parte de los resultados obtenidos también de la RSL.

En la Tabla 4 se realiza otra comparación en términos de las diferentes sub-características de calidad en la seguridad definidas en la ISO/IEC 25010 [43], la cual se encuentra aún en revisión. También se incluyen otros criterios que podrían ser interesantes para entender la capacidad de cada modelo de representar todos los posibles RS actualmente considerados por los estándares y la comunidad científica en general. Un “*” implica que la habilidad para representar dicho RS o las posibles contramedidas para obtenerlo (acciones de mitigación), ha sido encontrada en la literatura relevante. La “X” significa que no se ha encontrado información con respecto a dicha capacidad para el modelo en cuestión. Por último, la “P” quiere decir que el modelo potencialmente podría ser utilizado para representar ese RS, aunque esta capacidad no está claramente descrita en la literatura relevante.

Luego de realizar un análisis exhaustivo de los resultados expuestos en la Tabla 3, se pueden obtener un número de consideraciones. La columna CC nos permite identificar la popularidad del modelo en la comunidad científica a través de la cantidad de veces que es referenciado en comparativas o resúmenes. Varios modelos emergen vencedores de dicha comparación, lo que nos da una idea de dónde mirar primero a la hora de analizarlos. Esto, sin ir en detrimento de otras propuestas que por novedosas o centrarse en ciertos campos en particular, puedan no ser tan conocidas.

La columna de ANS evidencia que la mayoría de las propuestas han sido aplicadas como parte de una metodología, que puede ser considerado el nivel superior de integración del modelo en el ciclo de desarrollo del sistema. Sin embargo, muchas de estas propuestas “fuertes” en el sentido de integración carecen de una definición formal completa, lo cual puede ser un problema si se pretende integrarlas en un solo modelo o crear enlaces entre ellas. Aún así, algunas tienen definiciones completas tales como metamodelos, lo que arroja esperanza en que dicho objetivo pueda ser alcanzado a través de transformaciones entre modelos según MDE.

Por otra parte, no es posible en general especificar restricciones en muchos de los modelos, lo cual puede conspirar contra un modelado bien logrado de los RS.

Existe una carencia general de validaciones empíricas en los modelos menos populares, aunque algunos casos de estudio e incluso experimentos se han realizado en otros. Esto significa que su efectividad práctica ha sido probada en aplicaciones de la vida real. La mayoría de estas tienen herramientas automatizadas disponibles para

**Tabla 3.** Comparación de propuestas

| Propuesta | ANS | DF | RTC | VE | AES | HA | CC |
|---|---|---|---|---|---|---|---|
| Casos de Uso de Seguridad | M [44] | * | X | X | ISO 9126 | X | C [14] |
| UMLsec | P [23] | P | * | Caso de Estudio Industrial: Autenticación biométrica [23] | X | Herramienta UMLSec. | C [10, 11] [14] |
| SecureUML | M [45] | * | * | Caso de Estudio: Aplicación J2EE "Tienda de Mascotas" [45] | X | Plantilla SecureUML | C [10, 11] |
| Tropos Seguro | M [25] | * | * | Caso de Estudio: Sistema Proceso de Medición Simple (eSAP) [25] | ISO 17799 | Herramienta SI*. | C [9-11], R [18] |
| Perfil GridUCSec | M [27] | * | * | Proyecto europeo GREDIA [27] | X | X | X |
| SI* | MT [28] | * | X | X | X | Plugin Eclipse como herramienta CASE. | C [9, 10], R [18] |
| Gráficos de Problemas de Seguridad | X | P | X | Caso de Estudio: Autenticación bluetooth en PDA [31] | X | X | C [9] |
| Diagramas de análisis de barreras | M [32] | X | X | X | X | X | C [14] |
| Restricciones | MT [33] | X | * | X | X | X | X |
| Casos de Mal-uso | M [44] | * | X | Experimento: Comparación entre casos de mal-uso y árboles de ataque [16] | X | Paquete de herramientas Scenario Plus. | C [9, 11, 13, 16], R [12, 15, 17, 18] |
| Diagramas de Mala-Actividad | X | * | X | Pequeño Caso de Estudio: "El Arte del Engaño" [35] | X | X | X |
| Casos de Abuso | X | * | X | X | X | Herramienta UML de Rational Rose. | C [11], S [18] |
| UMLintr | MT [36] | P | X | X | X | Prototipo de herramienta. | C [11] |
| Gráficos de Estados UML para Seguridad | X | X | X | Experimento: AsmL y Tablas de Estado UML traducidad a reglas Snort [37] | X | Herramienta UML de Rational Rose. | C [11] |
| Árboles de Ataque | M [46] | P | X | Experimento: Comparación entre casos de mal-uso y árboles de ataque [16] | X | SecureITree | C [9, 13, 16], R [12, 17] |
| Gráficos de Abuso | X | P | X | X | X | X | C [10] |
| Historias de Abuso | X | X | X | X | X | X | C [9, 14], R [12] |
| AsmL, AsmLSec | X | X | X | Experimento: Evaluación de AsmL y AsmLSec [40] [41] | X | X | C [11] |
| STATL | X | X | X | X | X | X | C [11] |

**Tabla 4.** Comparación de las propuestas basada en requisitos de seguridad reconocidos

| Propuesta | Autenticidad | Confidencialidad | Cumplimiento | Detección de Ataques | Disponibilidad | Integridad | No-repudio | Responsabilidad | Seguridad (safety) |
|---|---|---|---|---|---|---|---|---|---|
| Casos de Uso de Seguridad | * | * | * | X | * | * | * | P | P |
| UMLsec | * | * | X | P | P | * | * | P | X |
| SecureUML | * | X | X | X | X | X | X | X | X |
| Tropos Seguro | * | * | * | X | * | * | * | P | P |
| GridUCSec-Profile | * | * | X | X | * | * | * | * | X |
| SI* | * | * | X | X | * | P | X | P | P |
| Gráficos de Problemas de Seguridad | * | * | X | * | X | * | P | X | P |
| Diagramas de análisis de barreras | * | * | X | * | X | * | P | X | * |
| Restricciones | * | P | X | X | * | * | X | * | P |
| Casos de Mal-uso | P | P | * | * | P | P | P | X | * |
| Diagramas de Mala-Actividad | P | P | X | * | P | P | P | P | P |
| Casos de Abuso | X | X | X | * | X | X | X | X | P |
| UMLintr | X | X | X | * | X | X | X | X | P |
| Gráficos de Estados UML para Seguridad | X | X | X | * | X | X | X | X | X |
| Árboles de Ataque | P | P | X | * | P | P | P | P | * |
| Gráficos de Abuso | P | P | X | * | X | P | X | X | P |
| Historias de Abuso | P | P | P | * | P | P | P | P | P |
| AsmL, AsmLSec | X | X | X | * | X | X | X | X | X |
| STATL | X | X | X | * | X | X | X | X | X |

trabajar, lo cual es un paso importante para facilitar el uso intensivo en general de todas sus capacidades.

La mayoría de los modelos no son obtenidos a partir del cumplimiento con algún estándar de RS, aunque empíricamente reflejan muchos de estos. Esta observación puede obtenerse a partir del análisis de la Tabla 4, donde se puede evidenciar cuales RS pueden ser representados por los distintos modelos y cuáles no.

La triada formada por la *confidencialidad*, *integridad* y *disponibilidad* es generalmente aceptada como la más importante de observar a la hora de desarrollar modelos de representación y obtención de RS. Como muestra la Tabla 4, la mayoría de los modelos incluyen explícitamente o tienen el potencial de incluir soporte para ellos. Además, dependiendo de su propósito, cada modelo tiene mayor, menor o ningún soporte para la *detección de ataques*. Los modelos de obtención directa de RS tienden a incluir soporte para la *autenticidad*, mientras en el resto puede variar de un modelo al otro. El *no-repudio* y la *responsabilidad* son RS que deberían ser tomados en cuenta más a menudo en los modelos, dado que no son soportados de manera global.

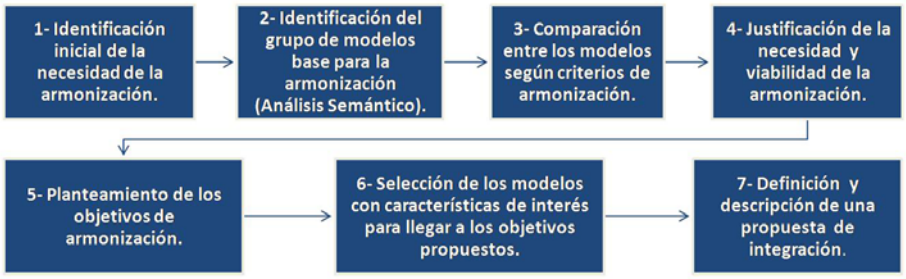
Por último, los requisitos de seguridad (*safety*) pueden ser potencialmente representados por muchos modelos, teniendo en cuenta que la diferencia entre

*seguridad* y *safety* es que estos últimos suelen estar asociados a accidentes no provocados [33].

## 4 Pasos para la Armonización de Modelos de Requisitos de Seguridad

A lo largo del presente trabajo, implícitamente se ha llevado a cabo el cumplimiento de una serie de pasos definidos para nuestro proceso de armonización de modelos de requisitos de seguridad. Como se puede observar en la Figura 1, dichos pasos han sido definidos de forma secuencial, lo cual no implica necesariamente que sus objetivos se logran secuencialmente en el tiempo. Además, pueden realizarse varias iteraciones, dependiendo del surgimiento de nuevas propuestas de modelado de requisitos de seguridad en la comunidad científica, lo que llevaría a revisitar cada uno de los pasos.

**Figura 1.** Pasos para la armonización de modelos de requisitos de seguridad




De los pasos descritos, hasta el momento se ha cumplimentado en su totalidad hasta el paso 4. Ellos son resumidos a continuación. El resto de los pasos se definen de manera preliminar y en términos teóricos.

### 4.1 Identificación inicial de la necesidad de la armonización.

La necesidad de la armonización es un tema propuesto en base a la opinión de los expertos consultados, cuyo conocimiento del estado del arte del modelado de requisitos de seguridad les brindaba una visión de los problemas existentes. Ellos proponen su justificación práctica a partir de los resultados de una RSL que arroje luz sobre esta área.

### 4.2 Identificación del grupo de modelos base para la armonización (Análisis Semántico).

Como resultado de la RSL se describen semánticamente todas las propuestas que incluyan algún tipo de modelo nuevo para la representación y obtención de requisitos de seguridad. Además se identifican y describen los trabajos donde se realizan revisiones y comparaciones previas entre los modelos identificados.

### 4.3 Comparación entre los modelos según criterios de armonización.

En este paso se han definido una serie de criterios comparativos entre los diferentes modelos con el fin de apoyar la necesidad y viabilidad de la armonización. Dichos criterios se definen en la sección anterior del presente estudio.

### 4.4 Justificación de la necesidad y viabilidad de la armonización.

También en la sección anterior se describen los resultados de la comparación de modelos de dónde se justifica la necesidad de la armonización, basados en las carencias que individualmente presentan los modelos actuales. Además, se identifican los elementos que hacen viable dicho proceso, como por ejemplo la existencia de definiciones formales en los modelos, según los objetivos que se plantean en el siguiente paso.

### 4.5 Planteamiento de los objetivos de armonización.

Identificada la necesidad y viabilidad de la armonización, se plantean los objetivos:

- Obtener una propuesta de modelado integradora y robusta, que incluya todos los RS aceptados por la comunidad científica.
- La propuesta estará orientada al paradigma contenido dentro de MDE, con el objetivo de aprovechar sus bondades, que incluyen la posibilidad de transformación automática de modelos entre las diferentes etapas del ciclo del desarrollo del sistema.
- Para ello, la propuesta debe incluir elementos de los modelos más exitosos identificados utilizando transformaciones entre ellos y/o definiendo un(os) modelo(s) integrador(es).

### 4.6 Selección de los modelos con características de interés para llegar a los objetivos propuestos.

En esta etapa, según los resultados obtenidos de la comparación entre modelos y teniendo en cuenta los objetivos de armonización, se seleccionan los modelos base que se tendrán en cuenta en la integración. Los elementos a tener en cuenta en esta selección son:

- En general, se aceptarán sobre todo modelos que posean una definición formal rigurosa (preferiblemente metamodelos), ya que esto facilitaría la obtención de nuevos modelos a partir de ellos, realizando transformaciones.
- En primer lugar, conviene empezar observando modelos “populares” en la comunidad científica, ya que esto da una garantía de éxito del mismo. Es importante además que haya sido validado empíricamente y que preferiblemente disponga de herramientas automatizadas para su operación.
- Es importante seleccionar modelos genéricos, que no se centren en un tipo de sistema específico.

- Es vital además, que logre ser capaz de especificar la mayor cantidad de RS posibles, aunque también son válidos aquellos que logren representar RS específicos con una gran robustez.
- Otro detalle importante a tener en cuenta es lograr una combinación de modelos que permitan ofrecer una visión global de la especificación de requisitos desde varios puntos de vista. Por ejemplo, desde el punto de vista de interacción, estructural, organizacional, etc.
- En el caso de los modelos basados en UML, para lograr el objetivo anterior, sería útil incluir aquellos que contengan la mayor cantidad de artefactos UML posibles.
- Un detalle que proporcionaría un valor añadido al modelo es que permita especificar restricciones, a partir de lo cual se pueden construir mecanismos de seguridad efectivos.
- Descartar modelos que no aporten elementos nuevos con respecto a otros ya seleccionados que tengan mejores prestaciones según cada criterio.

Atendiendo a esta serie de criterios, algunos ejemplos de modelos que emergen a tener en cuenta son: *Casos de mal-uso/Casos de Uso de Seguridad, UMLSec, SI*, Tropos Seguro, Árboles de Ataque*, etc.

### 4.7 Definición y descripción de una arquitectura de integración.

Utilizando los modelos seleccionados en el paso anterior se puede lograr una propuesta inicial de integración de modelos. En este sentido se consideran para investigaciones futuras tres posibles soluciones, las cuales son descritas a continuación:

1. En primer lugar, existe la posibilidad de definir formalmente un nuevo y *único modelo integrador propio*, tomando elementos de cada uno de los modelos seleccionados y que permitiera la especificación de cada uno de los requisitos de seguridad antes mencionados. La dificultad que entraña esta definición es hacer confluir todas las tendencias y líneas de trabajo en una sola. Por ejemplo, el paradigma definido según Tropos [47] (SI* y Secure Tropos), con el de UML según la OMG [48], además de incluir modelos independientes como árboles de ataque, etc.
2. Por otro lado, existe la posibilidad de *definir una única estructura de modelos y transformaciones entre ellos*. En este sentido se podrían generar reglas de transformación QVT [49] entre los distintos modelos y un modelo objetivo conteniendo una visión global del sistema. Esto es factible de lograr debido a la robustez de la definición formal de los modelos seleccionados. De esta manera, formando parte de una arquitectura MDA [48] (implementación por parte de la OMG de los principios de MDE) se logra pasar de modelos conceptuales (CIM) a modelos independientes de la plataforma (PIM). En el caso de los modelos con carencias en su definición, el proceso debe empezar planteando las equivalencias entre sus elementos y el modelo objetivo. Sin embargo, el problema de dicha solución es que comprende solamente el paradigma según OMG, teniendo que integrarla de manera no-nativa con líneas de trabajo de interés diferentes como las de Tropos.

3. Por último, existe la posibilidad de definir *varios conjuntos de modelos con distintas visiones del sistema y líneas de trabajo, así como las relaciones y transformaciones entre ellos*. En este sentido se plantea la hipótesis de brindar las pautas necesarias para la selección del grupo de modelos específico factible, según las necesidades del desarrollador. Existirían modelos comunes que pudieran ser utilizados indistintamente, así como otros específicos según los objetivos y disponibilidades. Por ejemplo, para una implementación siguiendo el paradigma UML, se podría proponer la integración de Árboles de Ataque, como herramienta robusta de especificación de ataques, integrado con Casos de Uso de Seguridad de UML, y modelos de dicho paradigma como SecureUML o UMLSec. Por otro lado, siguiendo la línea de trabajo de Tropos, integrar las potencialidades de SI* y Secure Tropos como representaciones organizacionales de la seguridad, junto con modelos de especificación de ataques. En otro sentido, si por ejemplo se necesitara un desarrollo ágil, podrían incluirse las historias de abuso en su lugar. Se utilizarían todas las potencialidades de transformaciones y relaciones entre modelos definidas en la solución 2.

De todas las posibles soluciones, se plantea continuar la investigación y trabajo en la variante número 3. Las razones para ello radican en sus bondades de reutilización de modelos existentes, de implementación de transformaciones entre modelos y sus relaciones; pero sobre todo por su flexibilidad y adaptación según el entorno y necesidades de las partes interesadas.

## 5 Conclusiones

En el presente trabajo se han descrito los pasos necesarios para la armonización de modelos de requisitos de seguridad, desde la identificación de la necesidad de armonización hasta la descripción preliminar de lo que debe contener una propuesta integradora final. Se ha hecho especial hincapié en los primeros pasos, cuyo procedimiento y resultados han sido llevados a cabo a partir de los resultados de una rigurosa RSL. Además, se definieron criterios de comparación imprescindibles para justificar la necesidad y viabilidad del proceso.

Como pasos intermedios además, se plantearon los objetivos de armonización, así como los criterios a tener en cuenta para la selección de los modelos base que formen parte de la arquitectura integradora.

Por último, quedan sentadas las bases para el refinamiento e implementación respectivamente de los dos últimos pasos del proceso, lo cual es objetivo de investigaciones posteriores.

## 6 Agradecimientos

Esta investigación es llevada a cabo en el marco de trabajo de los siguientes proyectos: QUASIMODO (PAC08-0157-0668), SISTEMAS (PII2I09-0150-3135) y SEGMENT (HITO-09-138) financiados por la "Viceconsejería de Ciencia y Tecnología de la Junta de Comunidades de Castilla-La Mancha", así como FEDER,

BUSINESS (PET2008-0136) financiados por el "Ministerio de Ciencia e Innovación" (España) y el proyecto MEDUSAS (IDI-20090557) financiado por el "Centro para el Desarrollo Tecnológico Industrial. Ministerio de Ciencia e Innovación"(CDTI).

# 7 Referencias

1. Lin, L., et al., *Introducing Abuse Frames for Analysing Security Requirements*, in *Proceedings of the 11th IEEE International Conference on Requirements Engineering*. 2003, IEEE Computer Society. p. 371.
2. McDermott, J. and C. Fox, *Using Abuse Case Models for Security Requirements Analysis*, in *Proceedings of the 15th Annual Computer Security Applications Conference*. 1999, IEEE Computer Society. p. 55.
3. Schmidt, D.C., *Model-Driven Engineering.* IEEE Computer, 2006. **39**(2): p. 25-31.
4. Xiao, L., *An adaptive security model using agent-oriented MDA.* Information and Software Technology, 2009. **51**(5): p. 933-955.
5. Basin, D., J. Doser, and T. Lodderstedt, *Model driven security for process-oriented systems*, in *Proceedings of the eighth ACM symposium on Access control models and technologies*. 2003, ACM: Como, Italy. p. 100-109.
6. Fernández-Medina, E., et al., *Model-Driven Development for secure information systems.* Information and Software Technology, 2009. **51**(5): p. 809-814.
7. 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 & Interfaces, 2007. **29**(2): p. 244-253.
8. Mellado, D., et al., *A systematic review of security requirements engineering.* Computer Standards & Interfaces, 2010. **32**(4): p. 153-165.
9. Romero-Mariona, J., H. Ziv, and D.J. Richardson, *Later stages support for security requirements*, in *The Fifth Richard Tapia Celebration of Diversity in Computing Conference: Intellect, Initiatives, Insight, and Innovations*. 2009, ACM: Portland, Oregon. p. 103-107.
10. Fabian, B., et al., *A comparison of security requirements engineering methods.* Requirements Engineering, 2009.
11. Khan, M.U.A. and M. Zulkernine, *On Selecting Appropriate Development Processes and Requirements Engineering Methods for Secure Software*, in *33rd Annual IEEE International Computer Software and Applications Conference*. 2009, Computer Software and Applications Conference, Annual International. p. pp. 353-358.
12. Tondel, I.A., M.G. Jaatun, and P.H. Meland, *Security requirements for the rest of us: A survey.* IEEE Software, 2008. **25**(1): p. 20-27.
13. Diallo, M.H., et al., *A Comparative Evaluation of Three Approaches to Specifying Security Requirements*, in *REFSQ. 12th International Working Conference on Requirements Engineering: Foundation for Software Quality*. 2006: Luxembourg.
14. Mellado, D., E. Fernández-Medina, and M. Piattini, *A comparative study of proposals for establishing security requirements for the development of secure information systems*. 2006: Glasgow. p. 1044-1053.

15. Mead, N.R., *Experiences in eliciting security requirements.* CrossTalk, 2006. **19**(12): p. 14-19.
16. 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.
17. Mead, N.R., *How To Compare the Security Quality Requirements Engineering (SQUARE) Method with Other Methods. Technical Report CMU/SEI-2007-TN-021*, C.M.U. Software Engineering Institute, Editor. 2007.
18. Hadavi, M.A., V.S. Hamishagi, and H.M. Sangchi. *Security Requirements Engineering; State of the Art and Research Challenges*. in *Proceedings of the International MultiConference of Engineers and Computer Scientists 2008 Vol I IMECS 2008*. 2008. Hong Kong.
19. Biolchini, J., et al., *Systematic review in software engineering. Rio de Janeiro Brazil, Systems Engineering and Computer Science Department, UFRJ.* 2005.
20. Pardo, C., et al., *Framework de armonización para múltiples marcos de referencia de procesos*, in *CIbSE 2010*. 2010: Cuenca, Ecuador.
21. Kitchenham, B.A. and S. Charters, *Guidelines for performing systematic literature reviews in software engineering, Tech. Rep. EBSE-2007-01, Keele University.* EBSE Technical Report, 2007.
22. Firesmith, D.G., *Security use cases.* Journal of Object Technology, 2003. **2**(3): p. 53-64.
23. Jürjens, J., *Model-Based Security Engineering with UML*. 2005. p. 42-77.
24. Lodderstedt, T., et al., *SecureUML: A UML-Based Modeling Language for Model-Driven Security*, in *Proceedings of the 5th International Conference on The Unified Modeling Language*. 2002, Springer-Verlag. p. 426-441.
25. 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.
26. Yu, E.S.K. *Towards modelling and reasoning support for early-phase requirements engineering*. 1997. Annapolis, MD, USA: IEEE.
27. Rosado, D.G., et al., *Analysis of Secure Mobile Grid Systems: A Systematic Approach.* Information and Software Technology, 2010. **52**: p. p. 517-536.
28. Zannone, N., *The si* modeling framework: Metamodel and applications.* International Journal of Software Engineering and Knowledge Engineering, 2009. **19**(5): p. 727-746.
29. Massacci, F., J. Mylopoulos, and N. Zannone, *Security Requirements Engineering: The SI* Modeling Language and the Secure Tropos Methodology*. 2010. p. 147-174.
30. Jackson, M., *Problem frames: analyzing and structuring software development problems*. 2001: Addison-Wesley Longman Publishing Co., Inc. 390.
31. Hatebur, D., M. Heisel, and H. Schmidt, *Security engineering using problem frames*. 2006. p. 238-253.
32. Jennex, M.E., *Modeling security requirements for information systems development.* SREIS 2005, 2005.

33. Haley, C.B., et al., *Security requirements engineering: A framework for representation and analysis.* IEEE Transactions on Software Engineering, 2008. **34**(1): p. 133-153.
34. Sindre, G. and A.L. Opdahl, *Eliciting security requirements with misuse cases.* Requirements Engineering, 2005. **10**(1): p. 34-44.
35. Sindre, G., *Mal-activity diagrams for capturing attacks on business processes*. 2007: Trondheim. p. 355-366.
36. Hussein, M. and M. Zulkernine *UMLintr: A UML Profile for Specifying Intrusions*, in *Engineering of Computer-Based Systems, IEEE International Conference on the 13th Annual IEEE International Symposium and Workshop on Engineering of Computer Based Systems (ECBS'06)*. 2006. p. 279-288.
37. Zulkernine, M., M. Graves, and M.U.A. Khan, *Integrating software specifications into intrusion detection.* International Journal of Information Security, 2007. **6**(5): p. 345-357.
38. Schneier, B., *Attack trees.* Secrets & Lies: Digital Security in a Networked World, 2000: p. 318-333.
39. Peeters, J., *Agile security requirements engineering.* SREIS 2005, 2005.
40. Graves, M. and M. Zulkernine, *Bridging the gap: Software specification meets intrusion detector.* Proceedings of the Fourth Annual Conference on Privacy, Security and Trust (PST), 2006: p. 265-274.
41. Raihan, M. and M. Zulkernine. *AsmLSec: An extension of abstract state machine language for attack scenario specification*. 2007. Vienna.
42. Eckmann, S.T., G. Vigna, and R.A. Kemmerer, *STATL: An attack language for state-based intrusion detection.* Journal of Computer Security, 2002. **10**(1-2): p. 71-103.
43. ISO/IEC, *ISO/IEC FCD 25010 Software engineering -- Software product Quality Requirements and Evaluation (SQuaRE) -- Quality model and guide* (DRAFT).
44. Sindre, G., D. Firesmith, and A. Opdahl, *A Reuse-Based Approach to Determining Security Requirements.* 9th International Workshop on Requirements Engineering: Fundation for Software Quality, 2003.
45. Basin, D., J. Doser, and T. Lodderstedt, *Model driven security: From UML models to access control infrastructures.* ACM Transactions on Software Engineering and Methodology, 2006. **15**(1): p. 39-91.
46. Saini, V., Q. Duan, and V. Paruchuri, *Threat Modeling using Attack Trees.* Journal of Computing Sciences in Colleges , Consortium for Computing Sciences in Colleges, USA., 2008. **vol. 23, no. 4.**
47. Bresciani, P., et al., *Tropos: An agent-oriented software development methodology.* Autonomous Agents and Multi-Agent Systems, 2004. **8**(3): p. 203-236.
48. OMG. *http://www.omg.org/.* 2010.
49. OMG, *Meta Object Facility (MOF) 2.0 Query/View/Transformation Specification version 1.0*. 2008, Object Management Group.