Primer Taller de Seguridad en Ingeniería del Software y Bases de Datos (SISBD'2004)

Málaga
9 de Noviembre de 2004

Eduardo Fernández-Medina y Mario Piattini (Eds.)

Primer Taller de Seguridad en Ingeniería del Software y Bases de Datos (SISBD'2004)

Málaga
9 de Noviembre de 2004

ACTAS

Iniciativa enmarcada en las actividades de la red RETISTIC (Red temática de investigación en el campo de la Seguridad en las Tecnologías de la Información), financiada por el Ministerio de Ciencia y Tecnología (TIC-2002-12487-E)

Presentación

La Seguridad en los sistemas de información es uno de los desafíos más importantes que están asumiendo actualmente muchas de las organizaciones. A pesar de que muchas empresas han descubierto lo crítico que resulta una correcta confidencialidad, integridad y disponibilidad de su información para el éxito de sus negocios y operaciones, muy pocas han adaptado sus sistemas para mantener la información segura, evitando accesos no autorizados, previniendo intrusos, e impidiendo el descubrimiento de información confidencial.

Actualmente, existen muchos avances tecnológicos que estimulan la utilización de sistemas de información en muchos entornos de negocio. Estos sistemas utilizan grandes cantidades de datos, que son gestionados y almacenados por bases de datos y almacenes de datos. A a menudo gestionan información que es especialmente sensible, puesto que se refieren a aspectos protegidos por las leyes de protección de datos personales (creencias, datos médicos, etc.). Por tanto, la adecuada gestión de la seguridad, así como la implantación de medidas técnicas que garanticen la seguridad de estos sistemas de información y la información que éstos gestionan resulta crucial.

Este taller se centra en analizar las aportaciones que desde la ingeniería del software y las bases de datos pueden realizarse con el fin de construir sistemas de información más seguros.

Organizadores

Eduardo Fernández-Medina (Universidad de Castilla-La Mancha)
Mario Piattini (Universidad de Castilla-La Mancha)

Grupos Participantes

- Asociación de Auditores y Auditoría y Control de Sistemas y Tecnologías de la Información y las Comunicaciones (ASIA)
- Excelentísima Diputación de Ciudad Real
- Informáticos Europeos Expertos
- Universidad Carlos III
- Universidad de Castilla La Mancha
- Universidad Católica del Maule (Chile)
- Universidad Complutense de Madrid
- Universidad Politécnica de Catalunya
- Universidad de Deusto
- Universidad de Lleida
- Universidad de Málaga
- Universidad de Murcia
- Universidad Rey Juan Carlos

Índice

Una Visión General de Metodologías para el Diseño de Sistemas de Información Seguros

Rodolfo Villarroel[1], Eduardo Fernández-Medina[2], y Mario Piattini[2]

(1) Departamento de Computación e Informática. Universidad Católica del Maule (Chile)
rvillarr@spock.ucm.cl
(2) Departamento de Informática. Universidad de Castilla-La Mancha (España)
{Eduardo.FdezMedina, Mario.Piattini}@uclm.es

Resumen. Generalmente la seguridad de los sistemas de información es considerada una vez que el sistema ya ha sido desarrollado, está en operación y los problemas de seguridad ya han surgido. Este tipo de enfoque, conocido como "*penetrate and patch*", está siendo desplazado por metodologías que incorporan seguridad en el proceso de desarrollo de los sistemas. Este artículo presenta una visión general de metodologías de diseño de sistemas de información seguros. A la vez, establece un breve análisis comparativo, lo que nos permite aclarar que los aspectos de seguridad no pueden ser completamente especificados por estas metodologías, ya sea debido a limitaciones o porque han abordado aspectos puntuales del proceso de desarrollo. A la vez, cada una de estas metodologías comprende aspectos de seguridad muy importantes que pueden ser usados como base para nuevas metodologías o extensiones que sean desarrolladas.

1. Introducción

Diversos artículos se refieren a la importancia de la seguridad en el proceso de desarrollo de software. Ghosh et al. [5] afirman que la seguridad debe influir en todos los aspectos de diseño, implementación y pruebas de software. Hall y Chapman [6] proponen ideas de cómo construir sistemas correctos que no sólo cumplan los requisitos normales sino también los de seguridad. Los problemas de seguridad de datos se han incrementado como consecuencia de los cambios tecnológicos, tales como acceso a bases de datos vía web, desarrollo del comercio electrónico, avances en almacenes de datos e incluso el uso de técnicas de minería de datos [13], lo que permite justificar el uso de metodologías que incorporen seguridad en las etapas del desarrollo de sistemas de información. Sin embargo, la seguridad de los sistemas de información es generalmente considerada una vez que el sistema se ha desarrollado. Se ha comprobado que este enfoque, conocido como "*penetrate and patch*" (penetrar y parchear) tiene malos resultados. A la vez, la mayoría de las soluciones se han enfocado en proveer defensas de seguridad en vez de resolver uno de los principales aspectos que provocan problemas de seguridad, que se refiere a un apropiado diseño de software.

Afortunadamente, hemos identificado once metodologías que incorporan seguridad en sus procesos de desarrollo [1-4, 7-9, 11, 12, 14, 15]. Cada una de estas

metodologías comprende aspectos muy importantes referentes a seguridad, que pueden ser usados como base para el mejoramiento de las actuales metodologías. Al mismo tiempo, estas metodologías tienen una serie de limitaciones que deben ser tomadas en cuenta. Este hecho, desafortunadamente, indica que aún no hay metodologías consolidadas que integran la seguridad dentro de sus procesos de desarrollo.

En la siguiente sección entregaremos una visión general y realizaremos una comparativa de las metodologías que incorporan seguridad en sus procesos de desarrollo. Finalmente, estableceremos las conclusiones y líneas futuras de trabajo a partir de este artículo.

2. Propuestas de Metodologías que Incorporan Seguridad

Existen varias propuestas que permiten modelar distintos productos (sistemas de información, bases de datos, almacenes de datos, etc.) teniendo en cuenta la seguridad. Es muy común que las características de seguridad sean construidas en una aplicación de manera ad hoc o sean sólo integradas posteriormente durante la fase de gestión del sistema. Las propuestas que se mencionarán a continuación pretender formalizar, mediante metodologías y/o técnicas de modelado, los aspectos de seguridad para que éstos estén claramente definidos antes de crear la aplicación.

Las propuestas que serán analizadas en nuestra comparativa son las siguientes:

- MOMT: *Multilevel Object Modeling Technique*, de Donald Marks, Peter Sell y Bhavani Thuraisingham [9]
- *Framework* dirigido por procesos de negocio para Ingeniería de Seguridad, de José L. Vivas, José A. Montenegro y Javier López [15]
- UMLsec: Metodología de Desarrollo de Sistemas Seguros usando UML, de Jans Jürgens [7]
- Metodología de Diseño de Bases de Datos Seguras, de Eduardo Fernández.Medina y Mario Piattini [3]
- Metodología de Análisis de Requisitos de Privacidad y Seguridad dentro de un Contexto Social, de Lin Liu, Eric Yu y John Mylopoulos [8]
- Paradigma para incorporar Seguridad en los Métodos de Desarrollo de Sistemas de Información, de Mikko Siponen [12]
- CoSMo: Un Enfoque hacia el Modelado Conceptual de la Seguridad, de Christine Artelsmair, Wolfgang Essmayr, Peter Lang, Ronald Wagner, y Edgar Weippl [1]
- Uso de Aspectos para diseñar un Sistema Seguro, de Geri Georg, Indrakshi Ray y Robert France [4]
- Metodología para el Diseño de Software Seguros, de Eduardo B. Fernández [2]
- ADAPTed UML: Un Enfoque pragmático para el Modelado Conceptual de Seguridad OLAP, de Torsten Priebe y Günter Pernul [11]
- Un *Profile* de UML para diseñar almacenes de datos seguros, de Rodolfo Villarroel, Eduardo Fernández-Medina, Juan Trujillo y Mario Piattini [14]

Hemos elegido estas once metodologías debido a que la mayoría de ellas pretenden solucionar el problema de seguridad desde las etapas tempranas del desarrollo de los sistemas de información, dándole énfasis a aspectos de modelado de la seguridad y utilizando lenguajes de modelado que facilitan el proceso de diseño de seguridad.

2.1 MOMT: Técnica de Modelado a Objetos Multinivel

Marks et al. [9] definen MOMT como una metodología para el desarrollo de bases de datos seguras, extendiendo OMT de manera de diseñar bases de datos multinivel proporcionando los elementos con un nivel de seguridad y estableciendo reglas de interacción entre los elementos del modelo. Aunque MOMT está compuesta principalmente de tres etapas, los autores sólo describen los puntos esenciales de la etapa de análisis. Las etapas definidas por MOMT son:

- Etapa de Análisis: Permite analizar los requisitos para detectar potenciales vulnerabilidades del sistema. Esta etapa consiste de tres modelos cuyo objetivo es reunir información del sistema desde varias perspectivas: modelo de objetos multinivel (para representar características estáticas), modelo dinámico multinivel (para representar características dinámicas) y modelo funcional multinivel ((para representar características de transformación del sistema).
- Etapa de Diseño del Sistema: Permite diseñar bases de datos multinivel. Para lo cual define, a alto nivel, estructuras de sistemas y bases de datos multinivel.
- Etapa de Diseño de Objetos: Permite diseñar los módulos del sistema automatizado de una manera más detallada.

2.2 *Framework* Dirigido por Procesos de Negocio para Ingeniería de Seguridad

Vivas et al. [15] proponen un método de desarrollo de sistemas dirigido por procesos de negocios, en el cual las decisiones de tecnología son dirigidas por el modelo de negocio. La motivación para expresar los requisitos de seguridad a nivel de modelo de negocios se debe al hecho que las aplicaciones, como las transacciones de comercio electrónico, son conceptualmente similares a las transacciones tradicionales de negocios no automatizadas. Nociones como no repudio, confidencialidad, integridad, control de acceso y autenticación han jugado un rol en las transacciones de negocios antes de la aparición de los sistemas automatizados. Este *framework* está basado en UML, e integra los requisitos de seguridad en un modelo de procesos de negocio del sistema. Una extensión de UML expresa las nociones de seguridad. Con el propósito de facilitar su adopción por los desarrolladores de sistemas, el *framework* intenta integrar los requisitos de seguridad en metodologías de desarrollo de sistemas que, actualmente, están basadas en UML y dirigidas por casos de uso.

Los casos de uso y los escenarios correspondientes son usados como herramientas básicas para construir modelos de amenazas y obtener requisitos de seguridad. Posteriormente, a un nivel mayor de abstracción, se realiza una representación funcional del sistema, logrando una especificación de seguridad enriquecida. Posteriormente, se genera de manera automatizada una representación XMI del sistema y se integran los requisitos de seguridad en la descripción funcional por

medio de procesos de análisis y diseño basado en patrones, generando una nueva especificación del sistema con los requisitos que han sido incorporados. La representación resultante es traducida en una notación formal para prueba, validación y verificación. Este procedimiento es iterado tantas veces como sea requerido. El resultado es usado como una entrada a las siguientes etapas del desarrollo de sistemas.

2.3 UMLsec: Metodología de Desarrollo de Sistemas Seguros usando UML

Jürgens propone una metodología [7] para especificar requisitos respecto a confidencialidad e integridad en modelos de análisis basados en UML. Los modelos de seguridad destacados son Seguridad Multinivel y Control de Acceso Obligatorio. El enfoque propuesto considera una extensión de UML, para desarrollar sistemas seguros. Para el análisis de seguridad de una especificación de un subsistema, se modela el comportamiento del potencial adversario, por lo tanto, se modelan tipos específicos de adversarios que pueden atacar diferentes partes del sistema de una manera determinada. Esta propuesta utiliza la mayoría de los diagramas de UML para modelar aspectos de seguridad, principalmente referidos a confidencialidad e integridad. Incorpora, además, la traducción de los modelos UMLsec definidos para la introducción de patrones en el proceso de diseño.

2.4 Metodología de Diseño de Bases de Datos Seguras

Fernández-Medina y Piattini proponen una metodología para diseñar bases de datos multinivel [3], incorporando la seguridad en cada una de las etapas del ciclo de vida de los sistemas. Incluye los siguientes aspectos:

- Un lenguaje de especificación de restricciones de seguridad multinivel sobre los modelos conceptuales y lógicos,
- Una técnica para la captura temprana de requisitos de seguridad multinivel,
- Una técnica para representar modelos conceptuales de bases de datos multinivel,
- Un modelo lógico para especificar las distintas relaciones multinivel, la metainformación de las bases de datos y las restricciones,
- Una metodología basada en el Proceso Unificado con diferentes etapas que permiten diseñar bases de datos multinivel,
- Una herramienta CASE que ayuda a automatizar el proceso de análisis y diseño de bases de datos multinivel.

2.5 Metodologías de Análisis de Requisitos de Privacidad y Seguridad

Liu & Yu presentan un *framework* metodológico para tratar con los requisitos de seguridad y privacidad basados en i*, un lenguaje de modelado de requisitos orientado a agentes [8].
El *framework* soporta un conjunto de técnicas de análisis:

- Análisis del atacante. Ayuda a identificar potenciales atacantes del sistema y sus intentos maliciosos.

- Análisis de la vulnerabilidad de dependencia. Ayuda a detectar vulnerabilidades en términos de relaciones organizacionales entre *stakeholders*.
- Análisis de medidas para contrarrestar ataques. Los factores necesarios para el éxito de un ataque son la motivación del atacante, las vulnerabilidades del sistema, y las capacidades de los atacantes para llevar a cabo el ataque.
- Análisis de control de acceso. Establece un nexo entre los modelos de requisitos de seguridad y los modelos de implementación de seguridad, para ello utiliza modelos de i* para refinar una solución propuesta y generar un diseño del sistema.

Los conceptos provistos por el lenguaje i* hacen posible analizar temas de seguridad dentro de su contexto social (natural), llevando a una forma sistemática para encontrar vulnerabilidades y amenazas.

2.6 Paradigma para incorporar Seguridad en los Métodos de Desarrollo de Sistemas de Información

Más que presentar otro enfoque de seguridad con sus propias características de seguridad, Siponen propone un nuevo paradigma (un metamodelo) para sistemas de información seguros que ayudará a los desarrolladores a usar y modificar sus métodos existentes de acuerdo a lo que necesiten [12]. La metanotación incluye seis dimensiones: sujetos de seguridad, objetos de seguridad, restricciones de seguridad, clasificaciones de seguridad, escenarios de abuso, y políticas de seguridad. Sujetos de seguridad denotan las diferentes entidades relevantes de seguridad, es decir, entidades que tienen una conexión de seguridad relevante a los activos de la organización (objetos de seguridad). El término objetos de seguridad se refiere a los activos de la organización que son relevantes en términos de seguridad de la información. Estos activos pueden variar desde cosas físicas tales como papel hasta entidades electrónicas tales como ficheros. Las restricciones de seguridad pueden incluir acceso a escritura, acceso de lectura., etc., y pueden ser obtenidas al analizar los requisitos de seguridad (confidencialidad, integridad, disponibilidad y no-repudio) por cada objeto de seguridad. La clasificación de seguridad proviene de la necesidad de clasificar objetos y sujetos de seguridad de acuerdo a la sensibilidad de seguridad de la información. Los sujetos de abuso forman una clase especial de sujetos de seguridad y se refieren a aquellos sujetos que pueden llevar a cabo una violación de seguridad. Los Sujetos de abuso y escenarios de abuso pueden ser necesarios para dos tipos de situaciones. En primer lugar, cuando existe una necesidad de explorar e identificar que potenciales escenarios de amenaza existen. En segundo lugar, esta clase es relevante para propósitos de prueba ya que ayuda a chequear que el sistema y software bajo diseño pueda sobrellevar escenarios no deseados o ataques de personas o procesos no autorizados.

2.7 CoSMo: Un Enfoque hacia el Modelado Conceptual de la Seguridad

En [1] se identifica la necesidad de integrar consideraciones de seguridad en el proceso de modelado de software. El modelado conceptual debería abordar requisitos

de seguridad y mecanismos de seguridad de alto nivel. Los autores trabajaron en el desarrollo de un método de modelado conceptual de la seguridad al cual se refieren como CoSMo (*Conceptual Security Modeling*).

Antes de proporcionar una visión general de los mecanismos de seguridad que permiten hacer cumplir determinados requerimientos, es necesario abordar temas fundamentales asociados a políticas de seguridad. Una política de seguridad consiste de un conjunto de leyes, reglas y prácticas que regulan cómo una organización gestiona, protege y distribuye información sensible. Cada requisito de seguridad puede hacerse cumplir por uno o más mecanismos de seguridad, resultando en una matriz de requisitos/mecanismos. Los requisitos de seguridad y mecanismos son genéricamente definidos, ya que son usados para el modelado de la seguridad a nivel conceptual.

En primer lugar, los autores muestran cómo las consideraciones de seguridad pueden ser integradas en el proceso de modelado conceptual. A continuación, sistemáticamente enumeran frecuentemente los requisitos de seguridad encontrados y claramente indican que mecanismos son usados para hacerlos cumplir. Generalmente, un requisito de seguridad no es parte del diagrama de casos de uso, pero es parte de la descripción del caso de uso. En CoSMo, será posible modelar este requisito de seguridad a nivel conceptual, aún en un diagrama de casos de uso.

2.8 Uso de Aspectos para Diseñar un Sistema Seguro

En [4] los autores se enfocan en el uso de aspectos para el modelado en problemas de seguridad. Proponen una técnica de diseño orientada a aspectos (AOD) para el diseño de un sistema seguro. Un diseño orientado a aspectos consiste de un modelo primario y uno o más aspectos que capturan problemas de diseño que cruzan-cortan las unidades de diseño del modelo primario. La incorporación de estos aspectos en el modelo primario es llamada *weaving* (tejido). En esta metodología los aspectos son tratados como patrones de diseño.

Un aspecto es definido en términos de estructuras de roles llamados Modelos de Rol. Los Modelos de Rol son usados debido a que permiten expresar los aspectos como patrones. La manera en que los múltiples aspectos son tejidos dentro de un modelo primario es determinada por estrategias de tejido. Una estrategia de tejido identifica aspectos de seguridad teniendo en consideración los tipos de ataques que son posibles en un sistema y los mecanismos que pueden ser usados para detectar, prevenir y recuperarse de tales ataques. Un aspecto de diseño (por ejemplo, un problema de seguridad) puede ser modelado desde varias perspectivas. Los autores se enfocan en dos vistas de aspectos: vistas estáticas y de interacción. Una vista estática de un aspecto define las propiedades estructurales del aspecto. La vista de interacción especifica los patrones de interacción asociados con el aspecto.

2.9 Metodología para el Diseño de Software Seguro

La principal idea de la metodología propuesta en [2] es que los principios de seguridad deberían ser aplicados a cada etapa de desarrollo y que cada etapa debe ser

probada para el cumplimiento de aquellos principios. El ciclo de vida del software seguro es el siguiente:

- Etapa de requisitos: A partir de los casos de uso, se pueden determinar los derechos necesitados por cada actor y de esta manera, aplicar la política necesaria. El conjunto de todos los casos de uso define los usos del sistema y de todos los casos de uso se determinan todos los derechos para cada rol. Se pueden considerar posibles ataques en el contexto de estos casos de uso.
- Etapa de Análisis: Se construye un modelo conceptual en el cual las aplicaciones repetidas de los patrones de autorización realizan los derechos determinados por los casos de uso. Los patrones de análisis pueden ser construidos con autorizaciones predefinidas de acuerdo a los roles en sus casos de uso.
- Etapa de Diseño: Las interfaces pueden ser aseguradas aplicando nuevamente el patrón de autorización. Las interfaces de usuario debería corresponder a casos de uso. Las interfaces seguras hacen cumplir autorizaciones cuando los usuarios interactúan con el sistema. Los Diagramas de Despliegue pueden definir configuraciones seguras para ser usadas por los administradores de seguridad. Una arquitectura multicapa es necesaria para hacer cumplir las restricciones de seguridad definidas a nivel de aplicación. En cada nivel, los patrones son usados para representar los mecanismos de seguridad apropiados.
- Etapa de Implementación: Esta etapa requiere reflejar en el código las restricciones de seguridad definidas por la aplicación. Ya que estas restricciones son expresadas como cláusulas y asociaciones, pueden ser implementadas como clases funcionales.

Al final de cada etapa, se pueden realizar auditorías para verificar que se siguen las políticas institucionales. Si es necesario, las restricciones de seguridad pueden ser más precisas usando un lenguaje de restricción de objetos (Object Constraint Language, OCL) en vez de restricciones textuales. Usando capas, se pueden definir patrones en todos los niveles que en conjunto implementan un sistema seguro o confiable. La idea principal del patrón de capas es la descomposición de un sistema en capas jerárquicas de abstracción, donde los niveles mayores usan los servicios de los niveles inferiores.

2.10 ADAPTed UML: Un Enfoque Pragmático para el Modelado Conceptual de Seguridad OLAP

Una metodología y un lenguaje para el modelado conceptual de la seguridad OLAP se presenta en [11] creando una notación basada en UML llamada "ADAPTed UML" (que usa símbolos de ADAPT como estereotipos). Los autores eligieron ADAPT debido a que éste ha sido mencionado en diversas conferencias de grupos de usuarios ORACLE Express, y esta notación es una de las pocas excepciones que no usa una intuitiva notación como ER en el modelado conceptual. El modelo conceptual que ha sido desarrollado para modelos conceptuales OLAP introduce los elementos cubo, medida, nivel de dimensión, y atributo de dimensión.

Como parte del diseño lógico, el modelo conceptual independiente del sistema es transformado en un "modelo de implementación" lógico (dependiente del sistema). Las restricciones del sistema escogido (es decir, software de OLAP o sistemas de gestión de bases de datos) deben ser tomadas en cuenta. Los autores usan un modelo

lógico OLAP que se basa en los Servicios de Análisis de Microsoft (que viene con SQL Server 2000).

El modelo de seguridad para OLAP está basado en el supuesto de de una política se seguridad central (basada en un administrador). Las restricciones de acceso son definidas como restricciones de autorización realizando la identificación de los objetos y sujetos de seguridad necesarios. En este punto, ellos asumen la noción de roles (no solapados, no jerárquicos) como sujetos de seguridad. Por tanto, el elemento rol es introducido en forma adicional a los elementos anteriormente mencionados (cubos, dimensiones, etc.). Las restricciones de autorización pueden ser positivas (autorizaciones explícitas) o negativas (denegación explícita). El modelo de seguridad está basado en una política abierta (es decir, el acceso a los datos es permitido a menos que sea explícitamente denegado) con restricciones de autorización negativas. Los autores plantean un lenguaje de restricciones de seguridad multidimensional (MDSCL) que está basado en la representación MDX del modelo lógico OLAP usado por Microsoft.

2.11 Un *Profile* de UML para Diseñar Almacenes de Datos Seguros

En [14] se presenta una extensión de UML que permite representar la información de seguridad de los datos y sus restricciones en el modelado multidimensional (MD) a nivel conceptual. La extensión propuesta es un *profile* de UML que permite considerar las principales propiedades de modelado MD. Los autores de esta metodología consideran el modelo de seguridad multinivel, pero se enfocan en tener en consideración aspectos respecto a operaciones *read* debido a que ésta es la operación más común para aplicaciones de usuario final. Este modelo permite clasificar tanto la información y el usuario en clases de seguridad, y hacer cumplir el control de acceso obligatorio. Usando este enfoque, es posible implementar modelos MD seguros con cualquiera de los sistemas de gestión de bases de datos que son capaces de implementar bases de datos multinivel, tales como *Oracle Label Security* y *DB2 Universal Database* (UDB). Un *profile* de UML comienza con una breve descripción y luego lista y describe todos los estereotipos, valores etiquetados, y restricciones del *profile*. Además de estos elementos, un *profile* contiene un conjunto de reglas bien-formadas. Estas reglas son usadas para determinar si un modelo es semánticamente consistente con sí mismo. De acuerdo a lo anterior, los autores definen un *profile* de UML para el modelado conceptual multidimensional seguro siguiendo un esquema compuesto de los siguientes elementos: *description* (una pequeña descripción del *profile* en lenguaje natural), *prerequisite extensions* (indica si la actual extensión necesita la existencia de extensiones previas), stereotypes / tagged values (la definición de los esteretipos y/o los valores etiquetados), *well-formedness rules* (la semántica estática de las metaclases son definidas tanto en lenguaje natural y un conjunto de invariantes expresadas por medio de expresiones OCL), y *comments* (cualquier comentario adicional, decisión o ejemplo, generalmente escrito en lenguaje natural).

En la Tabla 1, presentamos una síntesis de las contribuciones, en términos de seguridad, de cada una de las metodologías analizadas.

Tabla 1. Resumen de las contribuciones, en términos de seguridad, de cada metodología

	Estándar de Modelado /Desarrollo	Tecnología	Tipo de Control de Acceso	Especificación de restricciones	Soporte de herramienta CASE
MOMT	OMT	Bases de Datos	MAC	NO	NO
Vivas	UML	Sistemas de Información	-------	NO	SI (Golog)
UMLSec	UML patrones	Sistemas de Información	MAC (Multinivel)	-------	NO
Fdez-Medina & Piattini	UML Proceso Unificado	Bases de Datos	MAC, DAC, RBAC	OSCL (basado en OCL)	SI (add-in para Rational Rose)
Liu & Yu	Lenguaje Orientado a Agentes (i*)	Sistemas de Información (sólo requisitos)	RBAC	Notación de modelado para objetos Alloy	SI (Alloy)
Siponen	---------	Sistemas de Información (metametodología)	-------	NO	NO
CoSMo	UML	Sistemas de Información (sólo requisitos)	-------	-------	NO
George et al.	UML Patrones	Sistemas de Información (sólo diseño)	RBAC	Usando una plantilla	NO
Fernández	UML Patrones	Sistemas de Información	Access Matrix RBAC	Sólo se refiere a OCL como una buena solución	NO
ADAPTed UML	ADAPT UML	OLAP	RBAC	MDSCL (basado en MDX)	NO
Villarroel et al.	UML	Almacenes de Datos	MAC DAC RBAC	OSCL (basado en OCL)	NO

Todas las anteriores propuestas son muy interesantes y nos proveen de importantes contribuciones para resolver el problema de seguridad en una forma metodológica. Podemos concluir que, a nivel general, todas cumplen con criterios asociados a aspectos formales, son propuestas serias y basadas y soportadas por un lenguaje de modelado. Sin embargo, se observa una deficiencia en el soporte automatizado que necesita cada una de estas metodologías.

La metodología para diseñar bases de datos multinivel llamada MOMT sólo explica la etapa de análisis; no propone soluciones válidas para las situaciones actuales, en las que han cambiado las tecnologías utilizadas y las necesidades de seguridad.

La propuesta de Vivas et al. es un trabajo que establece un *framework* de desarrollo de software basado en UML, así como también para integrar requisitos de seguridad en un modelo de procesos de negocio del sistema. La propuesta es exploratoria, y está

enfocada en una discusión e identificación de los problemas más que en proveer soluciones.

La propuesta de Liu et al. está asociada principalmente al proceso de análisis de requisitos de seguridad desde una perspectiva top down o bottom up. La debilidad de esta metodología es que no menciona el tratamiento de las bases de datos, no considera herramientas que soporten el tipo de razonamiento respecto a seguridad, y la metodología está pensada principalmente en contrarrestar los ataques de intrusos.

La propuesta de seguridad UMLsec considera requisitos de seguridad relacionados con aspectos de confidencialidad e integridad. No abarca aspectos asociados al diseño de seguridad en bases de datos según los aspectos conceptual, lógico y físico, lo cual es esencial es la seguridad de los sistemas.

La propuesta de Fernández-Medina y Piattini utiliza sólo diagramas de casos de uso, diagramas de clases y OCL para el modelado de la seguridad en bases de datos. Por lo tanto, no es adecuada para desarrollar sistemas de información seguros

La propuesta de Siponen permite que los desarrolladores puedan seguir usando sus métodos de diseño seguro de sistemas de información favoritos. Nuestra crítica está asociada a la deficiencia de los modelos gráficos y lenguajes usados para soportar esta propuesta o para proveerla de una mayor formalidad. La propuesta será satisfactoria dependiendo de la habilidad del usuario, que debe incorporar elementos de seguridad en las metodologías que utiliza en un momento dado.

La propuesta CoSMo nos provee de un método para el modelado conceptual de sistemas seguros, basado en UML, pero éste ha sido desarrollado parcialmente; no existe una definición de una notación formal basada en UML o la integración de éste en las herramientas existentes.

La propuesta de Georg et al. difiere de las otras, debido a que el foco no está en las extensión de notaciones de modelado como UML, sino que en cómo los problemas de seguridad pueden ser primero encapsulados y luego "tejidos" (woven) dentro de los modelos de diseño primarios. Los autores no han desarrollado una herramienta prototipo que soportará tejidos flexibles, proporcionando a los usuarios un lenguaje para la descripción reusable de estrategias y procedimientos de tejido.

La propuesta de Fernández afirma que la combinación de arquitecturas multicapa con patrones nos provee de un *framework* para desarrollar un enfoque sistemático y reusable para construir sistemas que satisfacen requisitos no funcionales específicos, pero es necesario trabajar en este tema para agregar más patrones en cada nivel y para reunir y unificar estos patrones.

La propuesta de Priebe y Pernul es interesante, pero su modelos están limitados a sujetos de seguridad muy simples (roles no jerárquicos, no solapados) Los modelos de control de acceso basado en roles generalmente nos proveen con la posibilidad de jerarquías de roles. Adicionalmente, debido a la dificultad de mantener de manera coherente el conjunto de restricciones, han limitado su modelo de seguridad a autorizaciones negativas.

La propuesta de Villarroel et al., a pesar de ser muy interesante y sólida en términos de modelado conceptual, tiene deficiencias asociadas a las restantes etapas del proceso de desarrollo. Además, no tiene, hasta el momento, un soporte automatizado con el cual trabajar, por ejemplo, la implementación de una herramienta CASE basada en UML incorporada en el modelado multidimensional.

3. Conclusiones y Trabajo Futuro

Existen propuestas metodológicas interesantes, pero varias proponen diferentes notaciones para los aspectos de modelado de los sistemas de información, bases de datos, y/o almacenes de datos. Proponemos que se desarrolle un enfoque metodológico estandarizado que nos permite construir sistemas tomando en cuenta los aspectos de seguridad desde las etapas más tempranas hasta el final del proceso de desarrollo. Este enfoque metodológico debería ser una extensión de las metodologías de modelado existentes, de otro modo, las organizaciones que estén realmente interesadas en seguridad de almacenes de datos tendrían que hacer un gran esfuerzo para adaptarse a la nueva tecnología.

El estándar de modelado más ampliamente usado es UML [10]. Por lo tanto, consideramos interesante obtener un consenso de estandarización de las diferentes metodologías para especificar un perfil (*profile*) de seguridad usando UML. UML ha sido ampliamente aceptado como un lenguaje estándar de modelado orientado a objetos que permite modelar diferentes aspectos de sistemas software. Por lo tanto, cualquier enfoque que utilice UML minimizará el esfuerzo de los desarrolladores en aprender nuevas notaciones o metodologías para cada subsistema a ser modelado. A la vez, es un lenguaje extensible, ya que provee mecanismos (estereotipos, valores etiquetados y restricciones) en dominios específicos, si es necesario, tales como aplicaciones web, modelado de negocios, etc. Consideramos apropiado el uso de una metodología de diseño que utilice un *profile* de UML para agregar aspectos de seguridad. Además, pensamos que es esencial el uso de un lenguaje basado en OCL que nos permita especificar restricciones de seguridad en los diagramas UML. También es importante el uso de una herramienta CASE (integrada, por ejemplo, a *Rational Rose*) que permita soportar el proceso de diseño de sistemas de una manera segura, para una validación posterior de la propuesta aplicada a situaciones reales. El mayor costo de sistemas software es el mantenimiento, lo cual es una consecuencia de documentación imprecisa, incompleta y arbitraria. Con un perfil de UML que nos permita modelar los requerimientos de seguridad de los sistemas de información, se logrará una especificación más robusta.

Nuestro trabajo futuro se dedicará a la construcción de una metodología completa basada en UML y en el Proceso Unificado, apoyada con una herramienta CASE, para desarrollar almacenes de datos seguros que garanticen la seguridad de la información y que ayuden al cumplimiento de las leyes existentes sobre protección de datos de carácter general.

Agradecimientos

Esta investigación es parte de los proyectos CALIPO (TIC2003-07804-C05-03) y RETISTIC (TIC2002-12487-E), soportados por la Dirección General de Investigación del Ministerio de Ciencia y Tecnología, y la red VII-J.RITOS2 financiada por CYTED.

Referencias

1. Artelsmair, C., Essmayr, W., Lang, P., Wagner, R., y Weippl, E. *CoSMo: An Approach Towards Conceptual Security Modeling.* in *Database and Expert Systems Applications: 13th International Conference (DEXA 2002).* Air-en-Provence, France: Springer-Verlag (2002)
2. Fernández, E.B. *A Methodology for Secure Software Design.* in *The 2004 International Conference on Software Engineering Research and Practice (SERP'04).* Las Vegas, Nevada, USA (2004)
3. Fernández-Medina, E. y Piattini, M. *Designing Secure Database for OLS.* in *Database and Expert Systems Applications: 14th International Conference (DEXA 2003).* Prague, Czech Republic: Springer (2003)
4. Georg, G., Ray, I., y France, R. *Using Aspects to Design a Secure System.* in *Eighth IEEE International Conference on Engineering of Complex Computer Systems (ICECCS'02).* Greenbelt, Maryland, USA (2002)
5. Ghosh, A., Howell, C., y Whittaker, J., *Building Software Securely from the Ground Up.* IEEE Software. 19(1) (2002) 14-16
6. Hall, A. y Chapman, R., *Correctness by Construction: Developing a Commercial Secure System.* IEEE Software. 19(1) (2002) 18-25
7. Jürjens, J., *UMLsec: Extending UML for secure systems development*, in *UML 2002 - The Unified Modeling Language, Model engineering, concepts and tools*, Jézéquel, J., Hussmann, H., y Cook, S., Editors, Springer: Dresden, Germany (2002) 412-425
8. Liu, L., Yu, E., y Mylopoulos, J. *Security and Privacy Requirements Analysis within a Social Setting.* in *11th International Requirements Engineering Conference:* IEEE Computer Society (2003)
9. Marks, D., Sell, P., y Thuraisingham, B., *MOMT: A multi-level object modeling technique for designing secure database applications.* Journal of Object-Oriented Programming. 9(4) (1996) 22-29
10. OMG, *Object Management Group: Unified Modeling Language Specification 1.5.* (2004)
11. Priebe, T. y Pernul, G. *A Pragmatic Approach to Conceptual Modeling of OLAP Security.* in *20th International Conference on Conceptual Modeling (ER 2001).* Yokohama, Japan: Springer-Verlag (2001)
12. Siponen, M., *Designing secure information systems and software (Academic Dissertation)*, in *Department of Information Processing Science* University of Oulo: OuLu, Finland (2002)
13. Thuraisingham, B., Schlipper, L., Samarati, L., Lin, T.Y., Jajodia, S., y Clifton, C., *Security Issues in data warehousing and data mining: panel discussion.* Database Security XI: Status and prospects, (1998) 3-16
14. Villarroel, R., Fernandez-Medina, E., Trujillo, J., y Piattini, M. *Un profile de UML para diseñar almacenes de datos seguros.* in *IX Jornadas de Ingeniería del Software y Bases de Datos (JISBD 04).* Málaga, España (2004)
15. Vivas, J.L., Montenegro, J., y López, J. *Towards a Business Process-Driven Framework for Security Engineering with the UML.* in *Information Security, 6th International Conference (ISC 2003).* Bristol, UK: Springer-Verlag (2003)