Toward a Pattern Language for Privacy-enhancing Design Techniques

Y.S. Martín, J. M. del Alamo, and J. C. Yelmo

Abstract— **Despite that privacy requirements are becoming mandatory in many circumstances, and privacy-enhancing technologies are more and more readily available, system designers are facing a lack of guidance to choose the most appropriate solutions to protect the privacy of the system users in each situation. This paper introduces a pattern language —that is, a structured collection of interrelated, reusable solutions— for privacy-enhancing design. We present the structural model of our patterns (techniques) and introduce their use within usual analysis and design activities, with a special emphasis on the relation between patterns and the privacy requirements they fulfil.**

Keywords— **Design patterns, Pattern language, Privacy by Design, Privacy-enhancing design, Privacy requirements operationalization.**

I. INTRODUCCIÓN

LA privacidad de los usuarios sólo se ha empezado a considerar como una cuestión que tener en cuenta a lo largo del proceso completo de desarrollo de sistemas hace relativamente poco tiempo [1]; sin embargo, su importancia está creciendo rápidamente. Por ejemplo, el nuevo marco legislativo de protección de datos de la U.E. (en elaboración), exigirá aplicar la Privacidad desde el Diseño (*Privacy by Design*) en todos los sistemas que traten con datos personales [2], esto es, los creadores de sistemas deberán llevar a cabo las acciones necesarias para garantizar la privacidad de sus usuarios, desde la concepción inicial del producto o servicio.

Para que un sistema cumpla con los requisitos de privacidad que establecen esta u otras legislaciones, hace falta definir un diseño que los satisfaga. Sin embargo, la traducción de los requisitos de privacidad en el diseño no es obvia, ya que implica explorar el espacio de posibles soluciones de diseño y valorar, para cada una de ellas: el cumplimiento de los distintos requisitos (la propia funcionalidad del sistema junto con los requisitos de privacidad y de otras categorías), la adecuación del diseño al entorno (organización, arquitectura, etc.), y el análisis del impacto sobre otras limitaciones del dominio (tecnología empleada, disponibilidad de la solución, experiencia requerida, etc.)

Pero en el ámbito del diseño software, los desarrolladores en general no necesitan dar con la mejor solución de diseño por sí mismos cada vez que crean un nuevo sistema. Si así fuera, o bien se dispararían los costos de desarrollo, o bien simplemente acabarían descartando y haciendo caso omiso de muchos de estos requisitos. En lugar de ello, se recurre a **soluciones de diseño reutilizables** que guían el diseño con recetas probadas, basadas en la experiencia y en los conocimientos previos, reduciendo la incertidumbre y el costo del proceso de diseño. Aunque cada sistema siempre tiene requisitos y restricciones específicas que requerirán particularizar las soluciones para cada caso individual, siempre hay aspectos comunes que se podrán reutilizar, a modo de plantilla, en todos aquellos escenarios que compartan una serie de características. Frecuentemente, los diseñadores recurren a compilaciones o catálogos creados por distintas fuentes (foros de estandarización, trabajos académicos, etc.) que recogen y estructuran las mejores prácticas de la comunidad, para usarlos a modo de recetario donde encontrar las soluciones más adecuadas para cada necesidad y escenario.

Aunque existen abundantes soluciones técnicas para satisfacer distintos requisitos de privacidad, los diseñadores carecen de una herramienta como las mencionadas que les permita seleccionar sistemáticamente las soluciones más adecuadas para el desarrollo de sistemas respetuosos con la privacidad, en función del problema al que se estén enfrentando, los requisitos exigibles y el contexto de aplicación. La presente ponencia viene a rellenar precisamente este hueco, definiendo un lenguaje de patrones de diseño garantes de la privacidad, es decir, un catálogo de técnicas interrelacionadas y ligadas a los requisitos que resuelven, aplicables en el diseño de un sistema respetuoso con la privacidad. En primer lugar, la sección II introduce los conceptos de patrones de diseño y lenguajes de patrones y la sección III repasa el estado del arte de su uso en el ámbito de la Ingeniería de Privacidad. Después, la sección IV enmarca la utilización de los patrones en el proceso de desarrollo de sistemas y la sección V presenta el modelo estructural para definir los patrones. Concluimos con la valoración actual del estado y las perspectivas de nuestro trabajo (sección VI).

II. ACERCA DE LOS PATRONES DE DISEÑO

Un **patrón de diseño** representa una solución general y reutilizable para un problema frecuente, que se puede aplicar cada vez que el problema aparece en un contexto de aplicación determinado, sin necesidad de reinventarla una y otra vez a partir de la nada, aunque adaptándola a las peculiaridades de dicho contexto. Un patrón recoge el conocimiento demostrado a partir de la experiencia de su aplicación previa, y lo pone a la disposición de una comunidad de práctica. Se habla también de antipatrones para describir los intentos típicos de crear una solución que resultan ineficaces o incluso contraproducentes. En el ámbito de la Ingeniería del Software, los patrones de diseño describen de manera prototípica un conjunto de componentes software y sus relaciones (habitualmente designados como participantes y colaboraciones), que actúan

Y.S. Martín, Universidad Politécnica de Madrid, Madrid, España, samuelm@dit.upm.es
J.M. del Álamo, Universidad Politécnica de Madrid, Madrid, España, jmdela@dit.upm.es
J.C. Yelmo, Universidad Politécnica de Madrid, Madrid, España, jcyelmo@dit.upm.es

como plantillas para que los desarrolladores creen el código fuente específico que implementará el patrón en su producto. En el contexto de la Interacción Persona-Ordenador, se habla también de patrones de interacción y de interfaz de usuario, e incluso se aplican a modelos de negocio y organización.

El diseñador querrá escoger de manera sencilla el patrón (es decir, la solución) más adecuado en cada momento. Sin embargo, para asegurarse de ello, no le basta con disponer de un inventario de los patrones existentes. Al contrario, la mera acumulación de patrones en número excesivo le abrumará si no va acompañada de mecanismos eficientes de selección y descubrimiento. Para ello se crean los lenguajes de patrones. Un **lenguaje de patrones** [3] no se limita a describir una colección de soluciones, sino que también presenta la motivación para escoger cada solución como el mejor compromiso posible entre distintas fuerzas en conflicto, indica en qué circunstancias se aplican, las ejemplifica y, lo más importante, describe relaciones entre las soluciones de distintos tipos (p. ej. jerarquía, escala, secuencia, refinamiento, composición, incompatibilidad, competencia, compleción, combinación, etc.). Estas relaciones permiten navegar a través de la colección y le facilitan al diseñador la aplicación de varias soluciones en consonancia. Por ello, un lenguaje de patrones es, más que un catálogo, una *red de patrones*.

III. ESTADO DEL ARTE DE LOS PATRONES DE DISEÑO EN LA INGENIERÍA DE PRIVACIDAD

Hace tiempo que se llevan publicando obras que recopilan aquellos patrones de diseño más útiles [4]–[6] para los diseñadores de sistemas y, más recientemente, estos catálogos se están recogiendo en sitios web cuyo contenido se actualiza dinámicamente, en muchas ocasiones con la contribución de los propios visitantes. De manera similar, con frecuencia se crean recopilaciones de patrones orientados a satisfacer requisitos de una categoría específica. Por ejemplo, existen múltiples trabajos académicos que tratan de organizar los patrones de diseño en el dominio de la seguridad, clasificándolos según: la capa de servicio a la que se dirigen [7], la etapa del proceso de desarrollo en la que se aplican [8] y su utilidad, las propiedades de seguridad relacionadas [9], u otras características [10], [11]. Incluso existen trabajos que recopilan patrones de seguridad dirigidos a un tipo concreto de sistemas como son las aplicaciones de voz sobre IP [12] o las aplicaciones web [13]. Otros trabajos van más allá del diseño del producto e incluyen patrones orientados a introducir la seguridad en el proceso de desarrollo ("patrones de diseño seguro" [14] o "patrones procedimentales" [15]).

Precisamente, dada la escasez de repositorios similares de patrones en el ámbito de la privacidad, nos hemos inspirado en varias ideas aplicadas al dominio de la seguridad por su relativa cercanía. En este sentido, caben destacar las contribuciones de Schumacher [9] y Fernandez-Buglioni [16], quienes han definido un lenguaje de patrones a partir del estudio de estándares de seguridad; Kienzle, quien define cuatro tipos de patrones según su generalidad (conceptos, clases, patrones y ejemplos) [15], [17]; y el proyecto TERESA (teresa-project.org) cuyo repositorio de patrones considera la trazabilidad de cada patrón a propiedades específicas de seguridad y fiabilidad, las dependencias de interfaces externas requeridas, y las limitaciones respecto a otras categorías de requisitos. Estos aspectos resultan relevantes a la hora de definir interrelaciones entre patrones, imprescindibles para construir un lenguaje de patrones, según se ha indicado. La cercanía entre los dominios de la seguridad y la privacidad queda asimismo patente en algunos de estos patrones que entran en terrenos más propios de la privacidad [18].

Ya en el ámbito estricto de la privacidad, distintos autores proporcionan listados parciales de soluciones reutilizables, aunque en general sus descripciones no se han creado siguiendo un enfoque sistemático, por lo que no alcanzan la exhaustividad de un patrón; ni se organizan en un sistema estructurado como un lenguaje de patrones. Por ejemplo, Jiang *et al.* definen un espacio de soluciones de diseño para la privacidad, que permite clasificarlas según varias dimensiones (espacios de información y sus límites, ciclo de vida de la información personal y asimetrías en la transparencia del uso de datos personales) [19]. Por su parte, Hoepman agrupa distintas técnicas para la protección de la privacidad (PETs) en patrones, y a su vez estos en lo que denomina estrategias de diseño [20], derivadas del estudio de las regulaciones de protección de datos, que se implementan mediante técnicas de privacidad concretas. Por otro lado, el método PriS presenta un conjunto de patrones de comportamiento, que satisfacen una serie de metas de privacidad sucesivamente refinadas, y que se pueden particularizar de acuerdo con las metas de la organización [21]. Quizás los resultados más avanzados sean los de Hafiz [22], [23], quien presenta un conjunto organizado de patrones de privacidad, pero necesariamente parcial por las dimensiones del trabajo. Doty *et al.* también han propuesto una pequeña lista de antipatrones [24] y creado una comunidad en línea para publicar patrones de privacidad (privacypatterns.org), categorizados y mapeados a principios; sin embargo, apenas se ha conseguido poblar y las funcionalidades de descubrimiento no están activas. Nótese que muchas de estas clasificaciones no se circunscriben a ofrecer patrones de diseño, sino que también entran en otros ámbitos, como los patrones de ataques y amenazas.

En el ámbito institucional, encontramos la Guía de Medidas para la Privacidad de la autoridad de protección de datos francesa, la Comisión Nacional de Informática y Libertades (CNIL) [25], que adolece de lo contrario: si bien abarca un amplio espectro de medidas organizadas en una jerarquía, apenas entra en detalle de cada una ni establece relaciones, por lo que se trata más bien de una guía de buenas prácticas. Desde otra perspectiva, OASIS PMRM [26], ISO/IEC 29101 [27] y los controles de privacidad del NIST estadounidense [28] definen un conjunto de unos pocos servicios de privacidad que deberían incluir todos los sistemas, como parte de su arquitectura de privacidad, pero no ofrecen una contextualización de las soluciones, sus consecuencias, la relación con otros requisitos y entre los distintos servicios, etc., en definitiva, no se trata de un lenguaje de patrones.

Una razón que complica la elección de las técnicas que utilizar para proteger la privacidad radica en la falta de una

metodología sistemática para entender cuáles sirven mejor para cumplir un requisito de privacidad determinado y en un contexto específico. Además, no hay un consenso sobre la forma en la que se deben describir los patrones de privacidad, el nivel de abstracción que deben proporcionar, o los criterios que utilizar para escoger uno u otro.

IV. EL DISEÑO PARA PROTEGER LA PRIVACIDAD EN EL PROCESO DE INGENIERÍA DE SISTEMAS

Las metodologías actuales para el desarrollo de sistemas y servicios software no suelen considerar los aspectos de privacidad o, si acaso, los tratan como un problema que se resolverá una vez que el sistema esté construido. Sin embargo, la máxima de la Privacidad desde el Diseño prescribe que estos aspectos se deben tratar desde la concepción inicial del producto o servicio y a lo largo de todo el ciclo de vida del sistema. A pesar de ello, el estado del arte apenas cuenta con metodologías que permitan llevar a cabo esta máxima de forma efectiva. Es por ello que el proyecto PRIPARE (PReparing Industry to Privacy-by-design by supporting its Application in REsearch) del 7º Programa Marco de la UE está desarrollando una metodología de proceso de desarrollo iterativa, que se pueda integrar con las prácticas actuales de gestión de proyectos e ingeniería de sistemas. La metodología PRIPARE describe 24 procesos que cubren todas las etapas del ciclo de vida del desarrollo y operación de sistemas [29]: gestión del entorno y de la infraestructura, análisis, diseño, implementación, verificación, entrega, mantenimiento, y retirada. En esta ponencia nos centramos en algunos procesos concretos de las **etapas de análisis y diseño**, que libran un rol principal en la definición de la solución del futuro sistema.

El objetivo último de un proceso de ingeniería de sistemas es construir sistemas que respondan a las expectativas de todos los interesados o *stakeholders*. Para ello, el primer paso (etapa de análisis) consiste en conocer cuáles son esas expectativas para definir *qué debe hacer* el sistema, en definitiva, disponer de una especificación detallada que determine los requisitos que debe cumplir. Más adelante (etapa de diseño), se procede a definir *cómo serán* la estructura y el comportamiento del sistema para que cumpla con los requisitos, esto es, se proporcionará la representación abstracta de una solución que ofrezca las funcionalidades requeridas y satisfaga las demás categorías de requisitos (incluyendo los de privacidad). Se hace notar que muchos modelos de proceso de desarrollo software siguen una estructura iterativa donde el diseño y el análisis no se suceden de manera estrictamente secuencial: nuestros resultados se pueden aplicar igualmente también a los proyectos que sigan esos modelos de proceso, aunque por simplicidad sigamos aquí una explicación basada en un modelo secuencial.

A. *De los principios de privacidad a requisitos operativos.*

Durante la etapa de análisis, se deben identificar los aspectos de privacidad relevantes y traducirlos en requisitos del sistema. En general, los requisitos de privacidad se basan en un marco normativo recogido en diversas directrices y regulaciones, de origen legal o industrial, ya sean genéricas o específicas de un dominio, con el apoyo adicional de las políticas corporativas. Este marco establece una serie de principios de privacidad, a modo de atributos de calidad que deben cumplir los sistemas que se atengan a tal normativa. Un **principio de privacidad** es tanto una consecuencia esencial de la privacidad, que define sus fundamentos, como una característica que el sistema debe ofrecer para respetarla y garantizarla. Esto es, un sistema es respetuoso con la privacidad si y sólo si se atiene a los principios de privacidad designados como tales. Algunos principios de privacidad son: la calidad de los datos personales, la minimización y proporcionalidad de los datos, la especificación y limitación de la finalidad del uso a un propósito legítimo, el derecho de acceso por el interesado a sus datos personales, la privacidad y protección de datos desde el diseño (*Privacy by Design*), etc.

Normalmente, varios principios de privacidad aparecen definidos a la vez dentro de una familia de principios: una partición del concepto de privacidad en un conjunto de principios disjuntos dos a dos y que cubren la privacidad exhaustivamente entre todos. Existen varias definiciones en disputa sobre la privacidad [30]–[32], y en consecuencia, existen varias familias de principios desarrolladas a partir de diferentes orígenes legales, industriales y académicos (cuya comparación se puede encontrar en [33]–[35]). En nuestro caso, utilizamos una familia de catorce principios de privacidad [36] que reflejan la mayoría de los existentes en otras familias, y que tienen asimismo en cuenta la normativa europea en este ámbito y su evolución prevista. No obstante, se pueden aplicar los mismos procesos fácilmente a cualquier otra familia de principios a los que el regulador o el contexto corporativo pudieran obligar al desarrollador a atenerse.

Los principios de privacidad son conceptos abstractos, a menudo expresados en términos lejanos al ámbito de la ingeniería, y por tanto, les suelen resultar difíciles de comprender a los ingenieros, que necesitan una metodología sistemática y bien definida para especificar los requisitos de privacidad más adecuados en cada caso. Por ello, una idea básica en el proceso de elicitación de requisitos de privacidad es la **operativización de requisitos:** como los principios son demasiado vagos y abstractos para medir su cumplimiento, hace falta transformarlos en requisitos objetivables, que realmente se puedan incorporar en un proceso de desarrollo y para cuya medición exista un método operativo. La operativización de requisitos reemplaza así estas definiciones abstractas de los principios de privacidad por los efectos que se derivan de su aplicación, que deben ser observables empíricamente, definidos específicamente, y medibles objetivamente. El resultado es un conjunto de requisitos técnicos que los ingenieros son capaces de entender e introducir más adelante en sus diseños [37]

Nuestro proceso de operativización se basa en la aplicación de un proceso de elicitación de requisitos orientado a metas. Este es un paradigma frecuente (aunque no el único) entre las múltiples propuestas existentes en el ámbito de la Ingeniería de Requisitos de privacidad [38], [39], inspirado en enfoques que ya se venían aplicando en el ámbito de la seguridad. Los procesos de elicitación de requisitos orientados a metas se

centran fundamentalmente en la definición y negociación de los objetivos de cada una de las partes interesadas en el sistema. Estas metodologías reconocen que las **metas** de cada parte pueden ser distintas, incluso contradictorias, y es necesaria una etapa de elaboración, especificación y priorización de forma que se puedan ir identificando y resolviendo las interdependencias entre los objetivos de las distintas partes interesadas, hasta llegar a un grafo de objetivos interdependientes. Así, el grafo de objetivos de alto nivel se puede ir refinando en requisitos técnicos más específicos, vinculados con los requisitos funcionales del sistema.

Una parte interesante de algunas de estas metodologías (p. ej. NFR) es que proponen que la construcción del grafo de objetivos pueda estar soportada por la experiencia previa de otros ingenieros, registrada en **catálogos** de clasificación (para identificar y refinar objetivos), correlación (para identificar y resolver interdependencias) y operacionalización (para refinar los objetivos de alto nivel a otros más técnicos). Estos catálogos permitirían sistematizar el proceso de ingeniería de requisitos proponiendo una serie de etapas concretas que el ingeniero debe seguir, ofreciendo unas guías basadas en la experiencia y el conocimiento previo. Aprovecharemos esta idea de catálogos de requisitos, que consideramos imprescindible para sistematizar el proceso de operativización, en una disciplina donde aún no hay una visión unificada.[38]

En nuestro proceso de operativización, partimos de cada principio, tomado como una meta abstracta, que se va descomponiendo sucesivamente en metas más y más concretas: primero en pautas generales, necesarias para cumplir con el principio, y luego en criterios operativos. Estos **criterios de privacidad** definen los requisitos técnicos y organizativos que deben cumplir los sistemas y las organizaciones con el fin de abordar las cuestiones de privacidad, expuestos siguiendo la forma de un enunciado neutro respecto a la tecnología, observable por el usuario del sistema, y comprobable de manera objetiva y/o medible. El proceso parte de un **catálogo predefinido de criterios** de privacidad [29], que son neutrales respecto a las distintas partes interesadas, basados en la experiencia, estructurados, jerarquizados, y priorizados. La criticidad de cada uno de los criterios de conformidad viene reflejada en la asignación de diferentes niveles de prioridad a los distintos criterios (y, en consecuencia, el nivel de conformidad con los principios o el grado de protección de la privacidad puede variar entre distintos sistemas, en función de la prioridad de los criterios que cumpla). El analista del sistema utiliza este catálogo de requisitos junto con la especificación del sistema, su arquitectura y los resultados del análisis de impacto sobre la privacidad (teniendo en cuenta los flujos de datos de carácter personal involucrados); para llegar a un conjunto específico de requisitos aplicables al sistema en desarrollo. Como se verá después, esta organización del catálogo de criterios resultará fundamental en la estructura del lenguaje de patrones.

No todos los criterios de privacidad serán exigibles en todos los casos, sino que dependerán de: (1) las funciones ofrecidas por el sistema o subsistema objeto de análisis (ya que habrá criterios no exigibles a un sistema, por referirse a funciones no ofrecidas por este), (2) el nivel de conformidad deseado y/o necesario, (3) los principios de privacidad impuestos por el marco normativo y (4) otras restricciones de la organización (por ejemplo, en términos de presupuesto, o prioridad de otros requisitos en contienda). Existen otros catálogos de criterios de privacidad (a veces bajo otros nombres), que podríanse usar igualmente con nuestro proceso[1]

B. Reutilizando soluciones de diseño para la privacidad.

Una vez que se dispone de una especificación del sistema, en forma de requisitos técnicos comprobables de manera operativa, la etapa de diseño establece medidas técnicas y organizativas que satisfarán estos requisitos. El diseño define la estructura y el comportamiento del sistema, lo que incluye:

- los componentes del sistema, su organización, sus interfaces internas y externas, y las relaciones entre ellos;
- la estructura, la representación y la semántica de los datos procesados por el sistema completo (incluidos los datos personales) y por cada uno de sus componentes; y
- los flujos de datos a través de los diferentes componentes y de los procesos que intervienen en el sistema (p. ej. entrada, validación, almacenamiento, movimiento, transformación, procesamiento y salida de los datos) y durante la interacción con los usuarios humanos (adquisición y presentación).

Se trata de un proceso costoso, en tanto que la propuesta y evaluación de las múltiples soluciones potenciales supondría una tarea inabordable en el tiempo normalmente disponible. Sin embargo, la existencia de **soluciones reutilizables** puede aliviar el esfuerzo exigido al ingeniero, ya que este dispondrá de un recetario al que recurrir ante determinados requisitos y donde podrá encontrar soluciones conocidas y comprobadas que los satisfacen. Nuestro proceso proporciona y hace uso de este tipo de soluciones reutilizables, que definen cualquiera de las tres facetas de diseño arriba mencionadas (componentes, datos y flujos), a fin de satisfacer los requisitos de privacidad.

En concreto, el proceso se basa en un lenguaje de patrones que satisfacen (en virtud de la experiencia) los requisitos de privacidad (principios, pautas y criterios operativos), a los que estos a su vez se corresponden. En nuestro proceso, estos patrones reciben el nombre de técnicas. Una **técnica de diseño para la privacidad** es una manera confiable e implementable de satisfacer las definiciones de uno o más requisitos operativos de privacidad. Las técnicas proporcionan una orientación para satisfacer los requisitos de privacidad siguiendo buenas prácticas, aunque no son necesariamente la única ni la mejor manera de cumplir con esos requisitos. Tan sólo resultan ser una forma bien conocida y heurísticamente demostrada de lograr su cumplimiento, que han demostrado ser muy útiles como recursos reutilizables, listos para usar, y disponibles para los desarrolladores, sin que tengan que reinventarlas para cada sistema. Se pueden crear una cantidad innumerable de técnicas, y distintas fuentes han definido cientos de ellas (a veces bajo los nombres de mecanismos, heurísticas, etc.), que cubren una amplia variedad de casos.

[1] Se remite al lector a https://cloudsecurityalliance.org/research/ccm/ para consultar la correspondencia de los criterios establecidos por distintas iniciativas y normas, abarcando tanto seguridad como privacidad.

Se considera que las técnicas son meramente informativas (opcionales), en lugar de normativas (obligatorias). Esto se debe a que puede haber diferentes maneras (diferentes técnicas) de cumplir con el mismo criterio; así que, en general, no hace falta una técnica concreta para satisfacer un requisito o un criterio de privacidad dado. Además, en el futuro pueden aparecer nuevas técnicas que todavía no existen y que podrían perfectamente ayudar a satisfacer un requisito determinado. De ahí que las técnicas deban mantenerse actualizadas, para que aplicarlas a las nuevas tecnologías o simplemente para mantenerse al día de a los desarrollos más recientes.

El proceso global de diseño para la protección de la privacidad selecciona y crea instancias de las técnicas de diseño que se deben implantar en el sistema con el fin de cumplir con los requisitos de privacidad que se han especificado de manera operativa (como criterios de privacidad) en la anterior etapa de análisis, teniendo en cuenta las características específicas del sistema objeto de diseño.

En primer lugar, se tienen que **seleccionar** las técnicas más adecuadas del catálogo, en función de las especificaciones del sistema y las propiedades del contexto. Sucede que algunas técnicas no resultarán aplicables, al referirse a criterios de privacidad que no atañen al sistema. Más aún, su adecuación se ve también limitada por las características específicas de cada sistema: sus funciones, los actores involucrados (sus roles y sus ámbitos de control de datos personales), la arquitectura, las tecnologías empleadas para su ejecución, e incluso la experiencia del equipo de desarrollo. Por ejemplo, una técnica para minimizar la información de ubicación no tiene sentido en un sistema que no se ocupa de ese tipo de información. O igualmente, una técnica para preservar la privacidad específicamente en sistemas RFID no se aplicará a un sistema que utilice una tecnología alternativa. Es decir, cada técnica tiene un alcance o contexto de aplicación específico. A veces, el diseñador se encuentra con que puede aplicar más de una técnica para implementar un control de privacidad en su sistema: entonces deberá elegir una u otra, en función de otras restricciones (experiencia, costo, impacto sobre otros requisitos como usabilidad, rendimiento, etc.).

En segundo lugar, se necesita **individualizar o instanciar** las técnicas seleccionadas. Es decir, se particularizan según las características específicas del sistema y su arquitectura. Por ejemplo, una técnica puede establecer que las coordenadas de ubicación del usuario se deben reemplazar por el nombre genérico de la región donde se encuentra. La individualización de esta técnica implica analizar cuándo se está utilizando la información de ubicación específicamente en el sistema, y llevar a cabo la sustitución en esos puntos. La correcta aplicación de este paso dependerá de cómo se haya llevado a cabo previamente análisis de los flujos de datos personales.

V. MODELO ESTRUCTURAL DE TÉCNICAS DE DISEÑO PARA LA PRIVACIDAD

Todos los patrones de diseño de un lenguaje de patrones deben responder a una estructura dada. Siguiendo la analogía lingüística, la estructura sería equivalente a la morfología del lenguaje: no sólo nos muestra las reglas para generar un patrón, sino que además lo sitúa en relación con otros. Existen distintas maneras de describir un patrón de diseño, aunque todas ellas incluyen una serie de elementos más o menos comunes. Algunas de las más tradicionales son [40]: la forma de Alexander, la de la Banda de los Cuatro (Gang-of-Four), la de Portland y la de Coplien. Nuestas técnicas siguen una plantilla inspirada en estas formas, con modificaciones para enfatizar las propiedades más relevantes para nuestro dominio. Así, tradicionalmente los patrones incluyen una sección que presenta las distintas fuerzas en conflicto que intervienen en el contexto de aplicación: en nuestro caso aparecen repartidas en varias secciones, destacando la inclusión de los requisitos de privacidad a los que la técnica ofrece respuestas de diversas índoles. Asimismo, para facilitar la indexación y la navegación a través de las técnicas, incidimos en los varios tipos posibles de relaciones entre ellas.

Las técnicas representan un nivel más refinado tras el descenso desde los principios genéricos a los requisitos operativos, por ello, contienen una gran cantidad de detalles (Fig. 1) que permiten determinar cuándo (ámbito de aplicación o contexto), cómo (descripción detallada del proceso, más opcionalmente ejemplos de aplicación y pruebas), para qué (correspondencia con los requisitos de nivel superior), y por quién (rol responsable a quien se dirige la técnica) se pueden aplicar, y hasta qué punto es esto posible (madurez y apoyos de terceros). Así, cada técnica se puede escribir como un documento que incluye las siguientes secciones y campos:

Título. Una frase que ayuda a identificar la técnica.

Descripción detallada. El **objetivo** explica qué pretende lograr esta técnica, cuál es el problema que pretende resolver. El objetivo muestra a la vez qué pretende conseguir y qué consigue esta solución. No debe ocupar más de un párrafo, preferentemente encabezado por una introducción de una sola línea. Las **palabras clave** resultan útiles con finalidad de indexación. Las **instrucciones** describen un procedimiento de diseño o unas reglas de diseño. Siempre deben estar escritas en un tono descriptivo, no prescriptivo (aunque esté demostrado que la técnica sirve para satisfacer determinados requisitos de privacidad, no representa el único enfoque que puede funcionar). En función del patrón del que se trate, las instrucciones pueden incluir desde la descripción general de una estrategia a la descripción de una estructura de componentes *software* y su comportamiento, o incluso hasta el pseudocódigo de un algoritmo, el esbozo de una interfaz gráfica, etc. Dentro de las instrucciones, se pueden incluir diagramas explicativos (de clases, de secuencia, de disposición de la interfaz, etc.) Por último, se incluyen los **resultados** esperados, producidos al aplicar esta técnica y la **explicación** de las razones que llevan a que funcione.

Contexto (Ámbito): algunas técnicas pueden ser bastante genéricas, mientras que otras sólo se aplican en contextos muy determinados; en cualquier caso, una técnica debe especificar cuándo puede aplicarse. El ámbito de aplicación de una técnica puede verse limitado por varios factores. En cuanto a la **tecnología**, hay *técnicas genéricas* que se pueden aplicar a cualquier sistema, independientemente de su tecnología; mientras que las *técnicas específicas para una tecnología* sólo

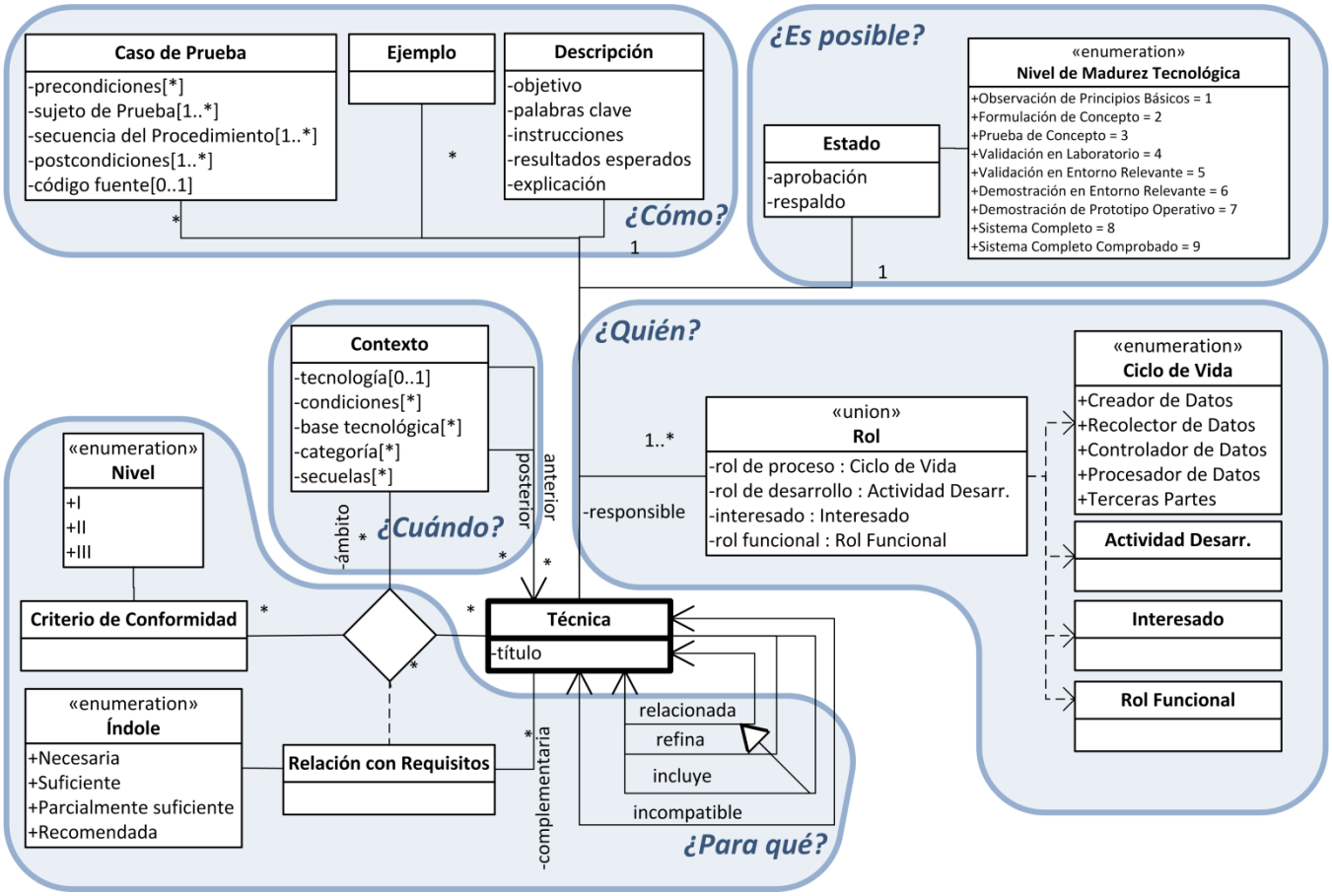

Figura 1. Modelo estructural de las técnicas de diseño orientadas a la privacidad.

se pueden aplicar a sistemas desarrollados utilizando esa tecnología respectiva (por ejemplo RFID). En cuanto a las **condiciones** de aplicabilidad, algunas técnicas sólo se podrán aplicar en algunos contextos (por ejemplo, seguimiento de ubicación del usuario) o bajo restricciones específicas. La **base tecnológica de referencia** define la plataforma tecnológica necesaria para que se pueda aplicar esta técnica (por ejemplo, esto permite descartar plataformas heredadas). La **categoría** define el área de conocimiento o de ingeniería, o área de la organización donde se aplica la técnica (por ejemplo, autenticación, computación en nube, etc.) Por último, las **secuelas** representan efectos secundarios resultantes de aplicar de esta técnica. A veces puede que no sean deseables o pueden afectar a otras categorías de requisitos (por ejemplo, rendimiento, usabilidad, etc.) Aquí se presentan como una limitación a priori del ámbito de aplicación (si la técnica implica consecuencias no deseadas, puede no ser aplicable).

Relación con requisitos. Cada técnica puede ayudar a cumplir con uno o más principios, pautas y criterios de privacidad. Esta relación entre técnicas y requisitos representa una prolongación del concepto de operativización al ámbito del diseño y justifica la introducción de dicha actividad en la fase de análisis. Se puede entender que las técnicas continúan el proceso de guiar la construcción del sistema mediante el refinamiento sucesivo de los principios.

La relación entre una técnica y un criterio de privacidad puede ser de **distinta índole**. Una técnica es *necesaria* si es obligatoria para cumplir un criterio; es decir, este sólo puede quedar satisfecho cuando se aplica la técnica (suponiendo que se cumplan sus condiciones de aplicabilidad). O, dicho a la inversa, una técnica necesaria se puede presentar en forma negativa como una mala práctica, si la aplicación de dicha mala técnica lleva inexcusablemente a incumplir un criterio. Una técnica es *suficiente* si su aplicación lleva a satisfacer un criterio, bajo las condiciones de aplicabilidad y descripción exactas; sin embargo, no es imprescindible para asegurar la conformidad con el criterio, ya que puede haber otras técnicas que lo logren. Una técnica es *parcialmente suficiente* si lleva a satisfacer un control sólo cuando se aplica en combinación con otras técnicas específicas y complementarias. Una técnica está *recomendada* si no ofrece directamente la conformidad con un criterio; pero ayuda a alcanzarlo o a avanzar más allá.

Nótese que una técnica puede estar relacionada con dos o más criterios, o con un mismo criterio en dos o más contextos, y el tipo de relación puede variar en cada caso. Asimismo, se debe tener en cuenta que el nivel de conformidad está unido directamente a los criterios de privacidad, no a cada técnica. Se recuerda que cada criterio tiene asociado un **nivel** de prioridad dado, por lo que en función de las técnicas utilizadas se alcanzará un nivel de protección de la privacidad. Normalmente, los distintos criterios de privacidad son independientes entre sí, pero también puede darse el caso de que el contenido de dos criterios de privacidad sean similares en su contenido, pero uno de ellos sea más restrictivo que el otro, y por tanto tenga asociado un nivel más estricto.

Relación con otras técnicas. Una técnica puede estar

relacionada con otras de varias maneras, algunas de las cuales se reflejan explícitamente en el diagrama, mientras que otras están mediadas por la relación mutua con un criterio. La aplicación simultánea de varias técnicas *complementarias* satisface un criterio específico (en relación con lo recién indicado sobre técnicas parcialmente suficientes para el cumplimiento de requisitos de nivel superior). Las técnicas *relacionadas* guardan cierta similitud entre sí en su objetivo, su procedimiento o su ámbito, por lo que les remite los diseñadores a estas por si les pudieran resultar útiles. Algunos casos particulares de estas relaciones pueden consistir en mecanismos de extensión como el *refinamiento* (una técnica se aplica a un contexto más restringido que otra) y la *inclusión* (una técnica añade nuevos pasos o reglas a otra, posiblemente para satisfacer más criterios o el mismo en en más contextos). Dos técnicas *incompatibles* no se pueden aplicar al mismo tiempo, mientras que dos técnicas *competidoras* representan soluciones alternativas, aunque no mutuamente excluyentes, para un mismo criterio en un contexto dado. Por último, la aplicación de una técnica en un contexto puede implicar la aplicación de otra técnica *anterior* o *posterior*.

Estado. Las técnicas pueden tener distintos grados de madurez, que puede venir refrendada por varias propiedades. La **aprobación** se refiere a aquellas normas o catálogos en los que aparece esta técnica, o a fuentes externas donde se propone. El **respaldo** se refiere a casos de la vida real donde se ha aplicado y validado esta técnica. Y el ***Nivel de Madurez Tecnológica* (TRL)** refleja el estado de desarrollo, expresado mediante un vocabulario controlado y graduado en una escala de nueve niveles (desde la existencia de información sobre los principios básicos hasta los despliegues funcionando de manera exitosa en entornos operativos reales).

Rol(es) responsable(s). La comunidad de práctica de la ingeniería de privacidad ha identificado diferentes roles que intervienen en las interacciones que afectan a la privacidad del interesado. En consecuencia, las técnicas pueden requerir la participación de diferentes roles que se encargarán de aplicarlas. Estos roles pueden venir especificados lo largo de diferentes dimensiones coexistentes. En relación con las **etapas del ciclo de vida** por las que pasan los datos personales, estos se crean, recogen, conservan, procesan y comparten. Por lo tanto, se pueden definir diferentes roles (en línea con, p. ej. OASIS PMRM [26], la Ley de Protección de Datos del Reino Unido [41] y la ISO/IEC 29100 [42]) para los creadores, recolectores, controladores y procesadores de datos, así como los terceros implicados. También cabe hablar de **roles funcionales**, que se corresponden con los sistemas que cumplen una función específica (recoger, comunicar, procesar, almacenar o desechar datos personales) dentro del ámbito de un dominio de privacidad [26]. En cuanto a las **actividades de desarrollo**, en el proceso de desarrollo de un sistema, intervienen varios trabajadores (cada uno de los cuales asume la responsabilidad de diferentes tareas): jefes de proyecto, analistas, arquitectos, diseñadores, desarrolladores, etc. Por último, en relación con los distintos **interesados** cabe hablar de las distintas personas físicas o jurídicas (ej.: sujeto de los datos, autor de la información, consumidor, redifusor, etc.), autoridades públicas, agencias o cualesquier otros organismos que pueden afectar, verse afectados, o sentirse afectados por una decisión o actividad relacionada con el procesamiento de datos personales. Las autoridades de protección de datos también quedan clasificadas en esta dimensión.

Casos de prueba (opcionales). Procedimientos que se podrán llevar a cabo para determinar si la técnica se ha aplicado de manera efectiva. Un caso de prueba es atómico y define una prueba parcial de un requisito, por lo que se puede entender como la capa de más bajo nivel en el descenso que comenzamos con los principios y seguimos con la operativización. Así, se podrían interpretar los casos de prueba como requisitos, en tanto que definen un contrato declarativo que especifica y limita el comportamiento que debe cumplir un sistema (es decir, que pase la prueba con éxito). El procedimiento de un caso de prueba puede describirse tanto mediante lenguaje natural, como mediante extractos de código fuente vinculados a entornos y herramientas de pruebas específicos. Esto último puede permitir automatizar la ejecución de los casos de prueba. Un caso de prueba contiene: **precondiciones y prerrequisitos,** incluyendo la tecnología de referencia necesaria para aplicar la prueba; definición del **sujeto de la prueba**, es decir los elementos del sistema donde se aplicará; el **procedimiento de prueba**, detallado como un conjunto de pasos; las **post-condiciones y resultados esperados** en caso de éxito; y el **código de pruebas** para plataformas y/o entornos específicos, en su caso.

Ejemplos (opcionales). Por ejemplo, se pueden indicar mejores prácticas que resulten útiles sin ser obligatorias; fragmentos de código fuente; ejemplos descriptivos; capturas de pantalla, maquetas o esqueletos de la interfaz de usuario; enlaces a ejemplos operativos y a aplicaciones para dominios específicos, etc. Se pueden incluir también aquí escenarios motivadores que presenten el problema y la solución aplicados a un caso concreto.

VII. CONCLUSIÓN Y LÍNEAS FUTURAS

Hemos descrito el proceso en el que se enmarca el uso de técnicas de diseño (patrones) para la privacidad y el modelo estructural empleado para definirlas. Este modelo es directamente aplicable a un conjunto inicial de técnicas, recopiladas a partir del estado del arte presentado al principio y que han servido para abstraer el modelo presentado. Además, se ha contribuido dentro de PRIPARE con la definición de patrones según una estructura simplificada, en el portal colaborativo https://privacypatterns.eu.

Se espera que las organizaciones extiendan y completen este lenguaje de patrones a partir de fuentes externas, adaptándolo a dominios específicos y enriqueciéndolo con su propia experiencia. Cabe señalar que el concepto de técnicas está muy alineado con otros enfoques similares para la ingeniería de privacidad (como las heurísticas, las estrategias o las tácticas), por lo que los resultados de estos enfoques son reaprovechables para definir nuevas técnicas.

Por nuestra parte, para favorecer la adopción por la comunidad, estamos trabajando activamente en foros que pretenden definir estos catálogos de soluciones interoperables,

como la Red de Ingeniería de Privacidad de Internet (IPEN), auspiciada por el Supervisor Europeo de Protección de Datos.

AGRADECIMIENTOS

Este trabajo se apoya en el proyecto PRIPARE, financiado por el Séptimo Programa Marco de la Unión Europea, bajo el acuerdo de financiación número ICT-610613 (http://pripareproject.eu/). Los puntos de vista expresados corresponden solamente a los autores, y no pretenden reflejar de ningún modo los de la Comisión Europea.

REFERENCIAS

[1] A. Cavoukian, J. Stoddart, A. Dix, I. Němec, V. Peep, and M. Shroff, "Privacy by Design Resolution," *32nd International Conference of Data Protection and Privacy Commissioners, Jerusalem (Israel)*. .

[2] 2014, *Resolución legislativa del Parlamento Europeo, de 12 de marzo de 2014, sobre la propuesta de Reglamento del Parlamento Europeo y del Consejo relativo a la protección de las personas físicas en lo que respecta al tratamiento de datos personales y a la libr*. .

[3] F. Buschmann, K. Henney, and D. C. Schmidt, "Pattern-Oriented Software Architecture: On Patterns and Pattern Languages. Table of contents," in *Pattern-Oriented Software Architecture: On Patterns and Pattern Languages, Volume 5*, vol. 26, no. 5, 2007.

[4] E. Gamma, R. Helm, R. E. Johnson, and J. Vlissides, *Design patterns: elements of reusable object-oriented software*, vol. 206. 1995.

[5] E. Freeman, B. Bates, K. Sierra, and E. Robson, *Head First Design Patterns*. O'Reilly Media, Inc., 2004.

[6] M. Fowler, *Patterns of Enterprise Application Architecture*, vol. 48, no. 2. Addison-Wesley Professional, 2002.

[7] C. Steel, R. Nagappan, and R. Lai, *Core Security Patterns: Best Practices and Strategies for J2EE, Web Services, and Identity Management*. Prentice Hall, 2005.

[8] N. Yoshioka, H. Washizaki, and K. Maruyama, "A survey on security patterns," *Progress in Informatics*, no. 5. pp. 35–48, 2008.

[9] M. Schumacher, E. Fernandez-Buglioni, D. Hybertson, F. Buschmann, Sommerlad, and Peter, *Security Patterns: Integrating Security and Systems Engineering*. Wiley, 2013.

[10] M. Hafiz, P. Adamczyk, and R. E. Johnson, "Organizing security patterns," *IEEE Softw.*, vol. 24, no. 4, pp. 52–60, 2007.

[11] S. Konrad, B. H. C. Cheng, L. A. Campbell, and R. Wassermann, "Using Security Patterns to Model and Analyze Security Requirements," *2nd Int. Work. Requir. Eng. High Assur. Syst.*, pp. 13–22, 2003.

[12] E. B. Fernandez, J. C. Pelaez, and M. M. Larrondo-Petrie, "Security Patterns for Voice over IP Networks," in *2007 International Multi-Conference on Computing in the Global Information Technology (ICCGI'07)*, 2007, pp. 33–33.

[13] D. M. Kienzle and M. C. Elder, "Security patterns for web application development," *Univ. Virginia Tech. Rep.*, 2002.

[14] C. R. Dougherty, K. Sayre, R. Seacord, D. Svoboda, and K. Togashi, "Secure design patterns," 2009.

[15] P. D. M. C. E. P. D. D. S. T. Darrell M. Kienzle, "Introduction Security Patterns Template and Tutorial."

[16] E. Fernandez-Buglioni, *Security Patterns in Practice: Designing Secure Architectures Using Software Patterns*. Wiley, 2013.

[17] M. C. E. P. D. P. D. D. T. J. E. Darrell M. Kienzle, "Security patterns repository, version 1.0."

[18] M. Schumacher, "Security Patterns and Security Standards - With Selected Security Patterns for Anonymity and Privacy."

[19] X. Jiang, J. I. Hong, and J. A. Landay, "Approximate Information Flows : Socially-Based Modeling of Privacy in Ubiquitous Computing," *Proc. Int. Conf. Ubiquitous Comput. (UbiComp '02)*, no. January 2001, pp. 176–193, 2002.

[20] J. H. Hoepman, "Privacy design strategies," *arXiv Prepr. arXiv1210.6621*, p. 12, 2012.

[21] C. Kalloniatis, E. Kavakli, and S. Gritzalis, "Addressing privacy requirements in system design: the PriS method," *Requir. Eng.*, vol. 13, no. 3, pp. 241–255, Aug. 2008.

[22] M. Hafiz, "A collection of privacy design patterns," in *Proceedings of the 2006 conference on Pattern languages of programs - PLoP '06*, 2006, p. 1.

[23] M. Hafiz, "A pattern language for developing privacy enhancing technologies," *Softw. - Pract. Exp.*, vol. 43, no. 7, pp. 769–787, 2013.

[24] M. G. Nick Doty, "Privacy Design Patterns and Anti-Patterns Patterns Misapplied and Unintended Consequences."

[25] "Measures For the Privacy Risk Treatment," 2012.

[26] J. Sabo, M. Willett, P. F. Brown, G. Janssen, and D. N. Jutla, "Privacy Management Reference Model and Methodology (PMRM) Version 1.0," 2013.

[27] "ISO/IEC 29101:2013 - Information technology -- Security techniques -- Privacy architecture framework," 2013.

[28] R. M. Blank and P. D. Gallagher, "NIST Special Publication 800-53, Security and Privacy Controls for Federal Information Systems and Organizations, Revision 4.," Washington, DC, 2013.

[29] A. Crespo García, N. Notario McDonnell, C. Troncoso, D. Le Métayer, I. Kroener, D. Wright, J. M. del Álamo, and Y. S. Martín, "Deliverable D1.2 - Privacy and Security-by-design Methodology," 2014.

[30] S. Gürses, "PETs and their users: a critical review of the potentials and limitations of the privacy as confidentiality paradigm," *Identity in the Information Society*, vol. 3, no. 3. pp. 539–563, 2010.

[31] D. Solove and P. M. Schwartz, "Privacy, information, and technology," *Elect. Ser.*, vol. 18, p. xxiii, 530 p., 2009.

[32] D. J. Solove, "Conceptualizing privacy," *California Law Review*, vol. 90, no. 4. pp. 1087–1155, 2002.

[33] Y. S. Martin, J. M. del Alamo, and J. C. Yelmo, "Engineering Privacy Requirements: Valuable Lessons from Another Realm," in *1st International Workshop on Evolving Security and Privacy Requirements Engineering - ESPRE2014*, 2014.

[34] "Privacy Management Reference Model," 2009.

[35] "AICPA - Comparison of International Privacy Concepts." [Online]. Available: http://www.aicpa.org/InterestAreas/InformationTechnology/Resources/Privacy/GenerallyAcceptedPrivacyPrinciples/Pages/InternationalPrivacyConcepts.aspx. [Accessed: 20-Apr-2015].

[36] A. Kung, A. Crespo Garcia, N. Notario McDonnell, I. Kroener, D. Le Métayer, C. Troncoso, J. M. del Álamo, and Y. S. Martín, "Deliverable D1.1 - Privacy and Security, Concepts and Principles Report," 2014.

[37] A. Cavoukian, "Operationalizing Privacy by Design: A Guide to Implementing Strong Privacy Practices," Toronto, Ontario (Canada), 2012.

[38] K. Beckers, "Comparing privacy requirements engineering approaches," in *Proceedings - 2012 7th International Conference on Availability, Reliability and Security, ARES 2012*, 2012, pp. 574–581.

[39] C. Kalloniatis, E. Kavakli, and S. Gritzalis, "Methods for designing privacy aware information systems: A review," in *PCI 2009 - 13th Panhellenic Conference on Informatics*, 2009, pp. 185–194.

[40] J. O. Coplien, *Software Patterns*. SIGS Books & Multimedia, 1996.

[41] UK Legislation, "Data Protection Act," *UK Legislation*, 1998. [Online]. Available: http://www.legislation.gov.uk/ukpga/1998/29/contents.

[42] "Information technology -- Security techniques -- Privacy framework ISO/IEC 29100:2011," Geneva (CH), 2011.

Yod Samuel Martín (S'14) es Ingeniero de Telecomunicación (2004) por la Universidad Politécnica de Madrid (UPM), donde ha trabajado como investigador en el Departamento de Ingeniería de Sistemas Telemáticos (DIT) y en el Center for Open Middleware desde 2004. En la actualidad, sus líneas de investigación se centran en la introducción de requisitos no funcionales en los servicios telemáticos, con especial énfasis en la accesibilidad y la privacidad.

José M. del Álamo es Doctor Ingeniero de Telecomunicación (2009) y profesor (2011) en DIT - UPM. Sus líneas de investigación incluyen la gestión de datos personales, incluyendo gestión de identidad y privacidad, y su introducción en las metodologías de ingeniería de software y sistemas.

Juan C. Yelmo es Doctor Ingeniero de Telecomunicación (1996) y Profesor Titular de Universidad (1998) en DIT - UPM. El profesor Yelmo ha desarrollado una intensa y continuada actividad investigadora caracterizada por la transferencia y publicación de resultados de investigación mediante artículos en revistas de primer nivel mundial en su sector y ponencias en congresos de relevancia internacional y patentes internacionales. Sus líneas de investigación actuales incluyen la ingeniería de servicios, la gestión de la identidad y la privacidad y el modelado de usuario en servicios y redes sociales.