Caracterización de los Contextos de Uso de la Ingeniería Inversa

Martín E. Monroy⁽¹⁾, José L. Arciniegas⁽²⁾ y Julio C. Rodríguez⁽¹⁾

(1) Universidad de Cartagena, Facultad de Ingeniería, Programa de Ingeniería de Sistemas, Piedra de Bolívar, Bolívar – Colombia (e-mail: mmonroyr@unicartagena.edu.co, jrodriguezz@unicartagena.edu.co)

(2) Universidad del Cauca, Departamento de Telemática, Sector Tulcán, Cauca – Colombia (e-mail: jlarci@unicauca.edu.co)

Recibido Dic. 21, 2016; Aceptado Mar. 1, 2017; Versión final Mar. 22, 2017, Publicado Ago. 2017

Resumen

El objetivo de este trabajo fue definir un modelo de caracterización de los contextos de uso de la ingeniería inversa. Para lograrlo se utilizó la técnica de coincidencia de patrones para hacer análisis sobre los resultados de la revisión de la literatura. El modelo definido se aplicó para caracterizar los contextos de producción de software, seguridad informática, computación forense y educación. La utilidad del modelo se evaluó en dos escenarios aplicando la técnica del juicio de expertos, estableciendo la disponibilidad de los artefactos software y el nivel de relevancia de las vistas arquitectónicas recuperadas. Se concluye que el modelo de caracterización definido sirve como referente para establecer estrategias de ingeniería inversa, orientadas a obtener resultados pertinentes al contexto en el cual se presenta la necesidad de realizar procesos de ingeniería inversa.

Palabras clave: caracterización; contextos de uso; ingeniería inversa

Characterization of the contexts of use of reverse engineering

Abstract

The aim of this study was to define a model for characterizing the contexts of use of reverse engineering. To achieve this, the pattern matching technique was used to analyze the results of a literature review. The defined model was applied to characterize the contexts of software production, computer security, computer forensics and education. The utility of the model was evaluated in two scenarios applying the technique of expert judgment, establishing the availability of software artifacts and the level of relevance of the recovered architectural views. It is concluded that the characterization model serves as a reference for defining reverse engineering strategies, which are aimed at obtaining pertinent results to the context in which the need to perform reverse engineering process is presented.

Keywords: characterization; contexts of use; reverse engineering

INTRODUCCIÓN

Desde su génesis la ingeniería inversa ha sido utilizada como técnica de mantenimiento de software (Bourque y Fairley, 2014). No obstante, a lo largo de su evolución ha servido para resolver varios problemas que afronta la ingeniería de software, tales como: la recuperación de arquitecturas y patrones de diseño, la re-documentación de programas y bases de datos, la identificación de activos reutilizables, la definición de la trazabilidad entre artefactos de software, la identificación del impacto de los cambios del producto software, la reestructuración de sistemas existentes, la renovación de interfaces de usuario, la migración hacia nuevas arquitecturas y plataformas, entre otros (Canfora et al., 2011).

A partir de su extensa experiencia Stoermer et al. (2003) identificaron varios contextos en los que se puede aplicar la ingeniería inversa, que incluyen: mejorar el entendimiento de la arquitectura de sistemas de software, mejorar la arquitectura como tal, evaluar los atributos de calidad del software, mejorar la documentación de la arquitectura, entre otros. Sin embargo, además del campo del desarrollo de software, la ingeniería inversa también se ha utilizado en otros contextos, como la educación (Klimek et al., 2011; Cipresso, 2009; Ali, 2006; Ali, 2005), la seguridad informática (Treude et al., 2011; Ligh et al., 2010, Sharif et al., 2009) e incluso en la computación forense (Nelson et al., 2014; Mueller, 2010; Ligh et al., 2010).

Si bien, la utilidad real de un proceso de ingeniería inversa se evidencia en el uso que se le dé a los artefactos recuperados (Canfora et al., 2011), la revisión de la literatura (Monroy et al., 2016b) reveló que hasta ahora, todas las propuestas metodológicas para realizar procesos de ingeniería inversa están orientadas exclusivamente al ámbito del desarrollo del software. Esto implica que para los contextos diferentes al de la producción de software, se desconocen los propósitos y circunstancias que originan la necesidad de realizar el proceso de ingeniería inversa, porque no se analiza la situación dentro del ámbito del contexto particular en el que se presenta. En consecuencia se limita la posibilidad de obtener resultados pertinentes que lleven a mayores y mejores beneficios. En este orden de ideas, el principal aporte del presente trabajo es el modelo de caracterización de los contextos de uso de la ingeniería inversa, que sirve como instrumento para hacer análisis sobre la forma cómo se debe de realizar el proceso de ingeniería inversa. La revisión de la literatura que se hizo no reveló ninguna propuesta similar. A continuación se presenta la metodología usada en la investigación, luego se explican los resultados en función del modelo de caracterización obtenido y su aplicación en los contextos de uso identificados. Posteriormente se hace un análisis de los resultados, enfocado en los recursos disponibles y las vistas arquitectónicas recuperadas, teniendo en cuenta las distintas situaciones establecidas en cada uno de los contextos de uso identificados. Al final se presentan las conclusiones del trabajo.

METODOLOGÍA

La investigación se realizó en dos fases: la primera fase se centró en la definición del modelo de caracterización de los contextos de uso de la ingeniería inversa y la segunda en la evaluación del modelo definido para determinar su utilidad. La primera fase inicio con la revisión de la literatura aplicando la propuesta metodológica definida por Kitchenham et al. (2011) y Petersen et al. (2008). Como resultado se identificaron los contextos de uso de la ingeniería inversa y se establecieron las particulares que los caracterizan. Con base en el análisis de la literatura encontrada y aplicando técnicas de coincidencia de patrones (Ying, 2013) se definió el modelo de caracterización. Posteriormente, en esta fase se describieron los contextos de uso identificados aplicando el modelo de caracterización definido. En la segunda fase de la investigación se evaluó la utilidad del modelo de caracterización en dos escenarios, aplicando la técnica del juicio de expertos (Cabero y Llorente, 2013). En el primer escenario se usó para determinar la disponibilidad de los artefactos software, en cada una de las situaciones que se presentan en los distintos contextos de uso identificados. De la misma manera, en el segundo escenario se aplicó el modelo de caracterización definido para establecer el nivel de relevancia de las vistas arquitectónicas recuperadas, para cada situación de los contextos de uso identificados.

RESULTADOS

Como resultado de la primera fase de la investigación, la revisión de la literatura permitió identificar que actualmente la ingeniería inversa se utiliza en procesos de desarrollo de software, en seguridad informática, en computación forense y en el campo de la educación. Además, el análisis de la literatura encontrada, aplicando la técnica de coincidencia de patrones, hizo posible la definición del modelo de caracterización de los contextos de uso de la ingeniería inversa, que se explica a continuación, así como también la descripción de cada uno de estos contextos con base en el modelo definido.

Según la real academia de la lengua española, el contexto es el entorno físico o de situación, político, histórico, cultural o de cualquier otra índole, en el que se considera un hecho. El contexto puede ser

generalmente descrito como un conjunto de estados físicos y conceptuales de interés a una entidad particular (Pascoe, 1998) y tiene tres aspectos importantes (Schilit et al., 1994): ¿dónde está?, ¿con quién está? y ¿qué recursos hay cerca? Basado en esto Dey (2001) define el contexto como: "cualquier información que puede ser utilizada para caracterizar la situación de una entidad".

El entorno de un sistema o contexto, está determinado por el conjunto de circunstancias que influyen en el sistema, incluyendo aspectos tecnológicos, organizacionales, de negocio, operacionales, políticos, económicos, legales, económicos e influencias sociales (ISO/IEC/IEEE 42010:2011). El entorno físico o de situación, que responda a las preguntas planteadas por Schilit et al. (1994), implica la existencia de un espacio comprendido dentro de los límites determinados por el contexto, es decir un *ámbito*, que para este caso representa el espacio ideal configurado por las cuestiones y los problemas de una o varias actividades o disciplinas relacionadas entre sí, por ejemplo la seguridad informática, la producción de software, la computación forense y la educación.

La *situación* implica un estado o constitución de las cosas o personas, determinado por el conjunto de factores o circunstancias que afectan a alguien o algo en un determinado momento, por lo tanto existen uno o varios interesados que manifiestan sus propósitos sobre el contexto. Un *interesado* es un individuo, grupo, organización (o clases de ellos) que tienen un interés sobre lo que sucede en el contexto (ISO/IEC/IEEE 42010:2011). El *propósito* hace referencia al conjunto de intenciones que tienen los interesados con respecto al contexto. Para el cumplimiento de los propósitos, atendiendo a la tercera pregunta planteada por Schilit et al. (1994), dentro del contexto existen *recursos*, entendidos como un conjunto de elementos disponibles para resolver una necesidad o cumplir con un propósito. En consecuencia, un contexto está determinado por circunstancias que representan las situaciones que se manifiestan en el contexto y por un conjunto de elementos representados por el ámbito, los recursos, los propósitos y los interesados, como se muestra en la figura 1, que simboliza el modelo de caracterización de los contextos de uso de la ingeniería inversa.

Fig. 1: Modelo de caracterización del contexto

En este orden de ideas, cada contexto de uso tiene un ámbito específico determinado por sus propósitos y situaciones, en el que se generan problemas que exigen a los interesados realizar procesos de recuperación y análisis de la arquitectura de productos software, teniendo en cuenta las características del contexto. Por otra parte, la evolución de la ingeniería inversa durante las últimas dos décadas ha permitido ampliar su campo de aplicación, que inicialmente sólo se limitaba al proceso de mantenimiento de software, a distintos contextos de uso como: la producción de software, la seguridad informática, la computación forense y la educación. Para cada uno de estos contextos a continuación se describe la forma como es utilizada la ingeniería inversa y se aplica el modelo de caracterización propuesto.

Producción de software

Comprende todos los procesos que definen el ciclo de vida del software, establecidos en el estándar ISO/IEC 12207:2008. La ingeniería inversa se utiliza concretamente en los siguientes procesos:

i) Mantenimiento de software: El estándar para el mantenimiento de software IEEE – 1219 recomienda la ingeniería inversa como la principal estrategia para tratar sistemas cuya única representación fiable es el código fuente (Canfora et al., 2011), debido a que la comprensión del software consume una parte sustancial del esfuerzo de mantenimiento. Adicionalmente se utiliza para la identificación de errores, al permitir examinar el sistema con el fin de localizar posibles defectos que presente, facilitando su modificación y actualización al implementar cambios para mejorarlo o ajustarlo a las nuevas necesidades, convirtiéndose en un paso obligatorio de los procesos de reingeniería (Chikofsky y Cross, 1990).

ii) Documentación: La ingeniería inversa se convierte en la principal estrategia para reconstruir la documentación de un sistema software, al lograr una representación de más alto nivel que permita su comprensión al recuperar las vistas de diseño y arquitectura, los requerimientos e incluso los modelos del negocio (Van Geet et al., 2010). También se puede utilizar para mantener el control de versiones en sistemas legados a partir de sus implementaciones, porque facilita la recuperación de artefactos en sistemas que han sido desarrollados hace mucho tiempo y que son difíciles de actualizar, haciendo posible establecer diferencias entre las distintas versiones que ha presentado el producto software (Boussaidi et al., 2012; Van Geet et al., 2010).

iii) Verificación de software: La ingeniería inversa es una excelente herramienta de control de calidad al momento de verificar la coherencia y trazabilidad entre el código fuente y: los modelos de diseño, la arquitectura, el modelo de negocio y los requisitos establecidos para el producto software. Desde la perspectiva del enfoque conocido como Arquitectura Dirigida por Modelos (MDA), se plantea la necesidad de mantener los modelos en tiempo de ejecución, lo cual consiste en sincronizar los modelos y el código fuente de tal forma que al modificarse uno, el otro se actualice respondiendo a dicho cambio, lo cual es posible gracias a la ingeniería inversa (Jouault et al., 2009).

iv) Reutilización de activos: Este proceso consiste en un conjunto de actividades que soportan la habilidad que tiene la organización para adquirir, desarrollar y gestionar activos software reutilizables. La ingeniería inversa juega un papel importante al momento de identificar piezas de software que puedan ser utilizadas en nuevas soluciones a partir de desarrollos previos, evitando así la necesidad de volver a implementarlas (Vasconcelos y Werner, 2011). Por otra parte, cada vez es mayor la importancia que toma la ingeniería inversa en contextos de líneas de producción de software, sobre todo al momento de mantener la consistencia de la arquitectura de la familia de productos (Wu et al., 2011; Stoermer y O'Brien, 2001).

En consecuencia, al aplicar el modelo de caracterización propuesto al contexto de uso de la producción de software, basados en los resultados del análisis de la revisión de la literatura, se obtiene la caracterización presentada en la tabla 1, donde se muestra cada uno de los elementos del modelo de caracterización con sus valores específicos para éste contexto.

Tabla 1: Caracterización del contexto Producción de software

Contexto:	Producción de software		
Ámbito:	Ingeniería de software		
Propósito general:	Construir software de calidad		
Posibles situaciones	Propósitos específico	Interesados	Recursos
Mantenimiento de software	Entender el producto software	Responsables de mantenimiento, arquitecto	El producto software, documentación, IDE
Documentación	Recuperar la documentación del software	Responsables de documentación, arquitecto	El producto software
Verificación	Validar la conformidad entre la arquitectura conceptual y la arquitectura concreta	Responsables de calidad, arquitecto	El producto software, documentación
Reutilización de activos	Identificar activos reutilizables	Arquitecto	El producto software

Seguridad informática

Como disciplina se encarga de definir normas, procedimientos, métodos y técnicas destinados a garantizar que un sistema de información sea seguro y confiable. En este campo se utiliza la ingeniería inversa con el fin de analizar y entender el código malicioso que pueda estar presente en el sistema, para facilitar la definición de estrategias a seguir en la solución del riesgo e inconvenientes que genera (Treude et al., 2011; Ligh et al., 2010, Sharif et al., 2009). También se puede utilizar para identificar posibles fallas de vulnerabilidad en la seguridad de productos de software, como lo demuestran Abi-Antoun y Barnes (2010). Al aplicar el modelo de caracterización propuesto al contexto de uso de la seguridad informática, basados en los resultados del análisis de la revisión de la literatura, se obtiene la caracterización presentada en la tabla 2, donde se detallan para las posibles situaciones que se presentan, sus propósitos específicos, sus interesados y los recursos disponibles.

Tabla 2: Caracterización del contexto Seguridad informática

Contexto:	Seguridad informática		
Ámbito:	Seguridad		
Propósito general:	Definir normas, procedimientos, métodos y técnicas destinados a garantizar que un producto software sea seguro y confiable		
Posibles situaciones	Propósitos específico	Interesados	Recursos
Análisis de software malicioso	Comprender la estructura y el comportamiento del software malicioso para solucionar el riesgo que representa	Experto en seguridad	Software malicioso
Problemas de seguridad en productos software	Identificar posibles problemas de seguridad en el producto software	Productor y usuario del software	El producto software

Computación forense

Noblett y Pollitt (2000) la definen como la ciencia de adquirir, preservar, recuperar y presentar datos que han sido procesados electrónicamente y guardados en un medio computacional, con el fin de interpretarlos y construir evidencia que permita establecer los hechos y formular hipótesis relacionadas con el caso de estudio. Para lograr este propósito es necesario desarrollar múltiples habilidades en áreas como el análisis de código malicioso, la ingeniería inversa de software y análisis forense de dispositivos móviles (Nelson et al., 2014, ; Ligh et al., 2010), como lo evidencia Robert S. Mueller (2010) director de la oficina federal de Investigación del FBI, en la conferencia sobre seguridad cibernética en San Francisco, California, donde mencionó experiencias de uso de la ingeniería inversa para atender casos de terrorismo cibernético. Con base en los resultados del análisis de la revisión de la literatura, se aplicó a este contexto el modelo de caracterización propuesto, obteniendo los resultados presentados en la tabla 3.

Tabla 3: Caracterización del contexto Computación forense

Contexto:	Computación forense		
Ámbito:	Jurídico		
Propósito general:	Interpretar y construir evidencia que permita establecer los hechos y formular hipótesis relacionadas con el caso de estudio		
Posibles situaciones	Propósitos específico	Interesados	Recursos
Recuperación de evidencia	Reconstruir evidencias a partir del producto software	Investigador, afectado por el hecho	El producto software

Educación

Para el Constructivismo Filosófico de Kant, el principal postulado indica que “el conocimiento humano no se recibe de forma pasiva sino que, más bien, es procesado y construido de una forma activa por el individuo que realiza la actividad de conocer y que, gracias a su aparato cognitivo, puede ir adaptando y modificando el objeto de estudio sobre el cual actúa, lo que permite al conocedor, (alumno o aprendiz, hablando en términos de aprendizaje) organizar su mundo, interactuar con él y registrar sus experiencias desde una perspectiva individual y vivencial” (Flórez, 1994, p.235, citado por Heredia 2012).

En este sentido, la ingeniería inversa se convierte en una herramienta didáctica que posibilita (Klimek et al., 2011; Cipresso, 2009; Ali, 2006; Ali, 2005): (i) El aprendizaje a partir de éxitos y fallas reales, porque hace

posible el uso de productos software para mostrarle a los estudiantes el diseño y la arquitectura de los casos de éxito con el fin de replicarlos, así como también los fracasos para evitarlos; (ii) El desarrollo de la curiosidad, estimulando el interés de los estudiantes al evidenciar la importancia de los modelos en los procesos de desarrollo de software, tomando como referentes implementaciones reales; (iii) La recuperación del conocimiento representado en productos software desarrollados, para facilitar la comprensión de conceptos en el campo de la ingeniería de software, como es el caso de la identificación de patrones aplicando técnicas de ingeniería inversa (Tonella et al., 2007); (iv) El desarrollo de habilidades para modelar, programar y hacer mantenimiento a productos software a partir implementaciones reales; y (v) La corrección de errores en el proceso de aprendizaje, mediante la utilización de herramientas de ingeniería inversa que permitan identificar posibles errores de diseño, o identificar inconsistencias entre los modelos de diseño y la implementación realizada.

Siguiendo la misma estrategia utilizada para los contextos de uso de la ingeniería inversa explicados previamente, en la tabla 4 se presenta la descripción del contexto de uso relacionado con el campo de la educación, obtenida al aplicar el modelo de caracterización propuesto con base en el análisis de los resultados de la revisión de la literatura.

Tabla 4: Caracterización del contexto Educación

Contexto:	Educación		
Ámbito:	Aprendizaje de la ingeniería de software		
Propósito general:	Aprender y desarrollar habilidades		
Posibles situaciones	Propósitos específico	Interesados	Recursos
Herramienta didáctica	Facilitar la comprensión de un tema y desarrollar habilidades específicas	Estudiantes y profesores	Documentación sobre el tema, código fuente
Herramienta de evaluación	Verificar la conformidad entre lo solicitado y lo entregado por el estudiante		Material entregado por el estudiante

ANÁLISIS DE LOS RESULTADOS

El análisis de los resultados se hace a partir de la evaluación de la utilidad del modelo de caracterización propuesto, realizada en la segunda fase de la investigación en dos escenarios. En el primer escenario se determinó la disponibilidad de los artefactos de software, en calidad de recursos, en cada una de las situaciones que se presentan en los distintos contextos de uso identificados; mientras que en el segundo escenario se aplicó el modelo de caracterización definido para establecer el nivel de relevancia de las vistas arquitectónicas recuperadas, para cada situación de los contextos de uso identificados, como se explica a continuación.

Para que el proceso de ingeniería inversa del producto software sea posible, es necesario contar con los artefactos que lo representan: archivos ejecutables, código fuente, archivos de configuración, estructuras de datos y documentación (Canfora et al., 2011); siendo obligatorio tener por lo menos el ejecutable o el código fuente. Sin embargo, la revisión de la literatura y la aplicación de la técnica del juicio de expertos (Cabero y Llorente, 2013), revelaron que la disponibilidad de estos recursos depende del contexto en el que se presente el problema objeto de estudio. En la tabla 5 se relacionan las situaciones que se pueden presentar en los contextos de uso y la disponibilidad de los artefactos software, según los resultados obtenidos al aplicar la técnica del juicio de expertos. La letra D se utiliza para indicar que el artefacto está disponible, la letra N significa no disponible, mientras que la letra P quiere decir que existe la posibilidad de que el artefacto esté disponible, dependiendo de las circunstancias que presente la situación objeto de estudio. Como se observa en la tabla 5, en el proceso de desarrollo de software el único artefacto que posiblemente no esté disponible es la documentación, mientras que en el contexto de la seguridad informática y la computación forense, sólo se tiene certeza de la disponibilidad del ejecutable. En el caso del contexto de la educación, el código fuente es el único artefacto que debe estar disponible. Esta situación afecta la forma como se debe realizar el proceso de ingeniería inversa, obligando a aplicar diferentes estrategias para cada contexto y situación.

Por otra parte, aunque en un proceso de ingeniería inversa se pueden recuperar distintas vistas arquitectónicas, la revisión de la literatura (Monroy et al., 2016b) y la aplicación de la técnica del juicio de expertos (Cabero y Llorente, 2013), permitieron identificar que la relevancia de la vista recuperada depende también del contexto en el que se realice el proceso. Algunas vistas son de carácter obligatorio (O), otras complementarias (C), otras no son relevantes (--), y otras son relativas (R) al problema concreto que se esté tratando: esto quiere decir, que en algunas circunstancias las vistas arquitectónicas pueden ser obligatorias

y en otras pueden ser complementarias para la misma situación, como se muestra en la tabla 6. Se toman como referentes las vistas arquitectónicas que corresponden a la fusión de las propuestas hechas por Bass et al. (2013) y Callo Arias et al. (2011), porque al unir las comprenden en forma completa todos los aspectos que describen el producto software, pertinentes al proceso de recuperación y análisis. Además, están documentadas según los lineamientos establecidos por el estándar ISO/IEC/IEEE 42010:2011.

Tabla 5: Disponibilidad de artefactos software en contextos de uso

Contexto	Situación	Recursos: artefactos de software				
		Ejecutable	Código fuente	Archivos de configuración	Estructuras de datos	Documen-tación
Producción de software	Mantenimiento de software	D	D	D	D	P
	Recuperación de documentación	D	D	D	D	P
	Verificación de software	D	D	D	D	D
	Reutilización de activos	D	D	D	D	P
Seguridad informática	Análisis de software malicioso	D	N	N	N	N
	Identificación de problemas de seguridad en productos software	D	P	P	P	P
Computación forense	Recuperación de evidencia	D	P	P	P	P
Educación	Herramienta didáctica	P	D	P	P	P
	Herramienta de evaluación	P	D	P	P	P

Tabla 6: Vistas arquitectónicas según contexto de uso

Contexto	Situación	Punto de vista												
		Estructuras modulares					Componentes y conectores				Estructuras de asignación			
		Descomposición	Usos	Capas	Clases	Modelo de datos	Servicios/Procesos	Concurrencia	Ejecución de la concurrencia	Perfil de la ejecución	Implementación	Asignación de trabajo	Despliegue	Uso de recursos
Producción de software	Mantenimiento de software	O	O	O	O	R	R	R	R	R	O	C	R	R
	Recuperación de documentación	O	O	O	R	R	R	R	C	C	R	C	R	C
	Verificación de software	O	O	O	O	O	R	R	R	R	O	--	R	R
	Reutilización de activos	O	O	C	C	R	C	C	C	C	C	--	C	C
Seguridad informática	Análisis de software malicioso	O	O	C	C	C	--	C	C	O	--	--	R	R
	Identificación de problemas de seguridad en productos software	O	O	C	C	C	R	C	C	O	C	--	R	R
Computación forense	Recuperación de evidencia	O	O	R	R	R	R	R	R	R	R	--	R	R
Educación	Herramienta didáctica	O	O	R	R	R	R	R	R	R	R	--	R	R
	Herramienta de evaluación	O	O	R	R	R	R	R	R	R	O	--	R	R

De igual manera, estos resultados demuestran la utilidad del modelo de caracterización definido, porque al utilizarlo se pudo establecer la relevancia de las vistas a reconstruir en un proceso de recuperación de arquitecturas, teniendo en cuenta las características del contexto en el que se presente el problema objeto de estudio. Adicionalmente, el modelo de caracterización se aplicó para definir un modelo ontológico para contextos de uso de herramientas de ingeniería inversa (Monroy et al., 2016a), que se integró a la caracterización de herramientas hecha por el grupo de investigación (Monroy et al., 2012).

CONCLUSIONES

Se definió un modelo de caracterización de los contextos de uso de la ingeniería inversa, que se utilizó para describir los contextos de uso identificados en la investigación: la producción de software, la seguridad informática, la computación forense y la educación. Se comprobó la utilidad del modelo de caracterización definido, al usarlo para determinar la disponibilidad de los artefactos software, en calidad de recursos, en cada una de las situaciones que se presentan en los distintos contextos de uso identificados; así como también para establecer la relevancia de las vistas arquitectónicas recuperadas, en función de las situaciones que se presentan en el problema objeto de estudio.

Por lo anterior, se puede inferir que el proceso de ingeniería inversa depende del contexto en el que se presente la situación que se desea resolver. Mientras que en un contexto como el desarrollo de software se puede aplicar un enfoque orientado por los atributos de calidad (Stoermer et al., 2003), en contextos como la computación forense y la seguridad informática, cuando se hace análisis de software malicioso, este enfoque no es pertinente, porque no se está revisando el cumplimiento de un requisito específico del sistema (Bass et al., 2013), sino una condición particular del contexto de uso. Por consiguiente, un enfoque de recuperación de arquitecturas dirigido por las características del contexto en el que se presenta el problema a resolver, permite el logro de resultados pertinentes a los propósitos de los interesados involucrados en el problema objeto de estudio.

Entre las limitaciones del estudio se debe mencionar que el modelo de caracterización propuesto sólo aplica para procesos de ingeniería inversa, porque las variables que se tuvieron en cuenta al aplicar la técnica de coincidencia de patrones se tomaron de los contextos de uso identificados en la revisión de la literatura (Monroy et al., 2016b). También se resalta la posibilidad de haber identificado en forma incompleta las variables al aplicar la técnica de coincidencia de patrones, por lo que se propone como trabajo futuro evaluar el modelo de caracterización en otros escenarios para complementarlo. De igual forma se plantea la necesidad de definir de una metodología para el proceso de ingeniería inversa, que aplique el modelo de caracterización propuesto, para determinar las actividades a realizar teniendo en cuenta las características particulares de los contextos de uso identificados en este estudio.

AGRADECIMIENTOS

Este trabajo es parte de los resultados parciales de la tesis doctoral titulada Marco de referencia para la recuperación y análisis de vistas arquitectónicas de comportamiento, desarrollada por Martín Monroy Ríos en el programa de Doctorado de la Universidad del Cauca. Agradecemos a la Red Latinoamericana de Coordinación y Cooperación para unificar buenas prácticas de desarrollo de Software - Proyecto SPU CoCoNet-LA por su apoyo.

REFERENCIAS

- Abi-Antoun, M. y J. M. Barnes, Analyzing security architectures. ASE'10 - Proceedings of the IEEE/ACM International Conference on Automated Software Engineering, 3-12 (2010)
- Ali, M.R., Why teach reverse engineering? ACM SIGSOFT Software Engineering Notes, 30(4), 1-4 (2005)
- Ali, M.R., Imparting effective software engineering education, ACM SIGSOFT Software Engineering Notes, 31(4), 1-3 (2006)
- Bass, L., Clements, P. y R. Kazman, Software Architecture in Practice, Third edition, Addison-Wesley (2013)
- Bourque, P., y R. E. Fairley, Guide to the software engineering body of knowledge (SWEBOK (R)): Version 3.0. IEEE Computer Society Press (2014)
- Boussaidi, G. E., Belle, A. B., Vaucher, S., y H. Mili, Reconstructing architectural views from legacy systems, Reverse Engineering (WCRE), 19th Working Conference IEEE, 345-354 (2012)
- Cabero, J. y M. C. Llorente, La aplicación del juicio de experto como técnica de evaluación de las tecnologías de la información (TIC). ISSN: 1856-7576, Revista de Tecnología de Información y Comunicación en Educación, 7 (2), 11-22 (2013)

- Callo Arias, T. B., Avgeriou, P., America, P., Blom, K., y S. Bachynskyy, A top-down strategy to reverse architecting execution views for a large and complex software-intensive system: An experience report, *Science of Computer Programming*, 76(12), 1098-1112 (2011)
- Canfora, G., Di Penta, M., y L. Cerulo, Achievements and challenges in software reverse engineering, *Communications of the ACM*, 54(4), 142-151 (2011)
- Chikofsky, E. J., y J. H. Cross, Reverse engineering and design recovery: A taxonomy, *Software, IEEE*, 7(1), 13-17 (1990)
- Cipresso, T., *Software reverse engineering education* (2009)
- Dey, A. K., Understanding and using context, *Personal and ubiquitous computing*, 5(1), 4-7 (2001)
- Heredia, Y. y Sánchez, A., *Teorías del Aprendizaje en el Contexto Educativo*, Editorial Digital, Tecnológico de Monterrey, México (2012)
- Jouault, F., Bézivin, J., y M. Barbero, Towards an advanced model-driven engineering toolbox, *Innovations in Systems and Software Engineering*, 5(1), 5-12 (2009)
- Kitchenham, B.A., D. Budgen, y O.P. Brereton, Using mapping studies as the basis for further research. A participant-observer case study, *Information and Software Technology*, 53(6), 638–651 (2011)
- Klimek, I., Keltika, M., y F. Jakab, Reverse engineering as an education tool in computer science. *Emerging eLearning Technologies and Applications (ICETA)*, 9th International Conference IEEE, 123-126 (2011)
- Ligh, M., Adair, S., Hartstein, B., y M. Richard, *Malware analyst's cookbook and DVD: tools and techniques for fighting malicious code*, Wiley Publishing (2010)
- Monroy, M. E., Arciniegas, J. L., y J. C. Rodríguez, Caracterización de Herramientas de Ingeniería Inversa, *Información Tecnológica*, 23(6), 31-42 (2012)
- Monroy, M. E., Arciniegas, J. L., y J. C. Rodríguez, Modelo Ontológico para Contextos de uso de Herramientas de Ingeniería Inversa, *Información Tecnológica*, 27(4), 165-174 (2016a)
- Monroy, M. E., Arciniegas, J. L., y J. C. Rodríguez, Recuperación de Arquitecturas de Software: Un Mapeo Sistemático de la Literatura, *Información Tecnológica*, 27(5), 201-220 (2016b)
- Mueller, R.S., RSA Cyber Security Conference San Francisco, California, FBI. March 04, 2010. Disponible en: <https://goo.gl/OKMXVu>. Acceso: febrero (2016)
- Nelson, B., Phillips, A., y C. Steuart, *Guide to computer forensics and investigations*. Cengage Learning, (2014)
- Noblett, M.G y M. Pollitt, Recovering and Examining Computer Forensic Evidence, *Forensic Science Communication*, 2(4), 1-13 (2000)
- Pascoe, J., Adding generic contextual capabilities to wearable computers. *Wearable Computers*, 1998, Digest of Papers. Second International Symposium IEEE, 92-99 (1998)
- Petersen, K., Feldt R., Mujtaba S. y M. Mattsson, Systematic Mapping Studies in Software Engineering, 12th International Conference on Evaluation and Assessment in Software Engineering (EASE), 1-10, Bari, Italy junio (2008)
- Schilit, B., Adams, N., y R. Want, Context-aware computing applications. *Mobile Computing Systems and Applications*, 1994, WMCSA 1994, First Workshop IEEE, 85-90 (1994)
- Sharif, M., Lanzi, A., Giffin, J., y W. Lee, Automatic reverse engineering of malware emulators, *Security and Privacy*, 2009, 30th IEEE Symposium, 94-109 (2009)
- Stoermer, C., y L. O'Brien, MAP-mining architectures for product line evaluations. In *Software Architecture*, 2001. Proceedings, Working IEEE/IFIP Conference, 35-44 (2001)

- Stoermer, C., O'Brien, L., y C. Verhoef, Moving Towards Quality Attribute Driven Software Architecture Reconstruction, WCRE, 3, 46-56 (2003)
- Tonella, P., Torchiano, M., Du Bois, B., y T. Systä, Empirical studies in reverse engineering: state of the art and future trends, Empirical Software Engineering, 12(5), 551-571 (2007)
- Treude C., Figueira F. y M. Storey, An Exploratory Study of Software Reverse Engineering in a Security Context. 18th Working Conference on Reverse Engineering, Limerick, Ireland, 184 – 188 (2011)
- Van Geet, J., Ebraert, P., y S. Demeyer, Redocumentation of a legacy banking system: an experience report. Proceedings of the Joint ERCIM Workshop on Software Evolution (EVOL) and International Workshop on Principles of Software Evolution (IWPSE), ACM, 33-41 (2010)
- Vasconcelos, A., y C. Werner, Evaluating reuse and program understanding in ArchMine architecture recovery approach. Information Sciences, 181(13), 2761-2786 (2011)
- Wu, Y., Yang, Y., Peng, X., Qiu, C., y W. Zhao, Recovering object-oriented framework for software product line reengineering, Top Productivity through Software Reuse, Springer Berlin Heidelberg, 119-134 (2011)
- Yin, R. K. Case study research: Design and methods, Sage publications (2013)