

Caracterización de Herramientas de 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, jrodriguezr@unicartagena.edu.co)

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

Recibido Abr. 27, 2012; Aceptado Jun. 20, 2012; Versión final recibida Jun. 28, 2012

Resumen

Se propone un instrumento para la caracterización de herramientas de ingeniería inversa. Se define una nueva estructura de caracterización basada en dos criterios: el aspecto estructural y las propiedades comunes entre ellas. Para el primero la estructura corresponde a un elemento, una función, un aspecto y una característica, mientras que para el segundo se tiene un factor y una propiedad. El instrumento es validado bajo tres situaciones que demuestran su utilidad al momento de clasificar y evaluar este tipo de herramientas. Se concluye indicando que el instrumento propuesto es un complemento de los trabajos presentados por otros autores, manteniendo su esencia y facilitando su interpretación a través de la nueva estructura definida. La propuesta permite la valoración independiente y conjunta de cada uno de los componentes que conforman las herramientas de ingeniería inversa.

Palabras clave: ingeniería inversa, caracterización de herramientas, estructura de caracterización, herramientas de ingeniería

Characterization of Reverse Engineering Tools

Abstract

An instrument for the characterization of reverse engineering tools is proposed. A new characterization structure based on two criteria, the structural aspect and the common properties between them, is proposed. For the first one the structure corresponds to an element, a function, an aspect and a characteristic, while the second one has a factor and a property. The instrument is validated in three situations that prove his usefulness for classifying and evaluating such tools. It is concluded that the proposed instrument is a complement to the work presented by other authors, maintaining its essence and facilitating their interpretation through the new defined structure. The proposal allows the independent and joint evaluation of each of the components that form reverse engineering tools.

Keywords: reverse engineering, tool characterization, characterization structure, engineering tools

INTRODUCCIÓN

En las dos últimas décadas la ingeniería inversa ha evolucionado representativamente, posibilitando un continuo mejoramiento de los procesos de construcción de software, permitiendo a arquitectos y desarrolladores tener una clara imagen del sistema que están construyendo, disminuyendo la posibilidad de error al facilitar la verificación de la coherencia entre el código desarrollado y la arquitectura propuesta, facilitando a su vez el mantenimiento y la adquisición de conocimiento de sistemas heredados (Canfora y Di Penta, 2007). Esto se evidencia en el surgimiento de un número cada vez más amplio de herramientas de ingeniería inversa que ofrecen múltiples funcionalidades.

Esta situación afecta a usuarios y productores de este tipo de herramientas, así como también a investigadores interesados en la ingeniería inversa. Mientras los usuarios se ven obligados a seleccionar la que mejor se ajusta a sus necesidades, en una lista que se amplía y diversifica con frecuencia, sin contar con un instrumento que facilite la toma de decisión; los productores requieren identificar nuevas funcionalidades y características que representen valor agregado a su producto, lo que les exige conocer con claridad qué características brindan las herramientas actuales. Por su parte los investigadores desean proponer nuevos métodos, técnicas y herramientas que representen mejoras a los procesos de desarrollo de software y recuperación de conocimiento de sistemas heredados, para lo cual requieren identificar en forma precisa las características de las herramientas disponibles.

Una revisión detallada del estado del arte ha permitido identificar que son múltiples los trabajos que se ha realizado al respecto, como se indica a continuación. Canfora presenta en 1992 uno de los primeros estudios tendiente a analizar las dificultades que surgen en el uso de documentos producidos por herramientas de Ingeniería Inversa, identificando tres problemas: 1. El nivel de detalle (son demasiado detalladas o son demasiado generales); 2. No producen información para los encargados del mantenimiento; 3. La inflexibilidad de las herramientas (Canfora y Cimitile, 1992).

Cuatro años más tarde Storey documenta un experimento para evaluar dos enfoques opuestos disponibles para la visualización de las estructuras de software en el editor gráfico de la herramienta Rigi, resaltando la importancia que tiene la facilidad de navegación visual entre los documentos generados por las herramientas de ingeniería inversa (Storey et al, 1996).

En el mismo año Brown y Wallnau definen un marco de referencia para evaluar software basado en dos objetivos: Entender las diferencias existentes entre tecnologías y entender cómo esta diferencia se orienta a las necesidades específicas de un contexto determinado. Para lograr dichos objetivos establecen un proceso de 3 fases: La primera es la descripción del modelo, la cual se puede hacer a través de la genealogía de la tecnología o del hábitat del dominio del problema; la segunda es el diseño del modelo, que comprende las actividades de análisis comparativo de características, formulación de la hipótesis y diseño del experimento. Finalmente la tercera fase corresponde a la evaluación del experimento, a través de la cual se confirma o refuta la hipótesis (Brown y Wallnau, 1996). Este marco de referencia es utilizado por Gannod y Cheng para definir un marco de referencia basado en cualidades semánticas para clasificar y comparar software de ingeniería inversa y técnicas de recuperación de diseño, donde la cualidad semántica mide la habilidad por producto de transmitir certeramente información de comportamiento (Gannod y Cheng, 1999).

Más concretamente, para el caso de la recuperación de arquitecturas y patrones de diseño, Jing Dong et al (2007) realizan un estudio comparativo entre distintos enfoques, teniendo en cuenta los siguientes aspectos: lo que describe (estructura, comportamiento, semántica), las representaciones intermedias, el grado de coincidencia con el patrón, la representación del resultado final, el grado de automatización (automático, semiautomático) y los patrones que soporta. Por otra parte, Ghulam Rasool et al (2010) recomiendan directrices basadas en la observación y en la evaluación de diferentes herramientas, que pueden ser utilizadas para

comparar las características de los instrumentos existentes y para desarrollar nuevas herramientas de recuperación de patrones de diseño.

Ninguno de los trabajos anteriormente mencionados, analiza de manera integral las características de las herramientas de ingeniería inversa, sólo se limitan a un aspecto determinado como la salida que generan, las formas de visualización, las diferencias entre tecnologías y la recuperación de arquitecturas y patrones.

Desde el punto de vista de caracterización, uno de los trabajos más detallados es el realizado por Bellay y Gall, quienes basados en su experiencia definen cuatro categorías funcionales para evaluar herramientas de ingeniería inversa: Análisis, representación, edición/navegación y capacidades generales, cada una de las cuales a su vez está conformada por un conjunto de criterios específicos (Bellay y Gall, 1997). Casi una década más tarde, en la Décima Conferencia Europea sobre Mantenimiento de Software y Reingeniería, Guéhéneuc et al (2006) plantean la necesidad de contar con un marco de referencia comparativo común bien definido y fácil de comprender, que haga menos complejo el trabajo de evaluación de herramientas de recuperación de diseño. Los autores proponen un marco de referencia comparativo común, construido a partir de su experiencia en calidad de extensión de otros marcos comparativos existente, analizando las siguientes características: contexto en el cual se aplica, intención, tipos de usuarios, tipos de entradas, técnicas utilizadas, tipos de salidas que provee, como está implementada y grado de madurez de la herramienta. Aunque estos trabajos sirven como referentes para caracterizar herramientas, no contemplan nuevos requerimientos que han surgido en los últimos años.

Luego, en la Décimo Quinta Conferencia de Trabajo sobre ingeniería inversa, Jenő Fülöp et al (2008) presentan su trabajo sobre la aplicación de un enfoque comparativo llamado BEFRIEND (BEenchmark For Reverse engInEering tools workiNg on source coDe), que posteriormente sustenta en calidad de tesis doctoral (Fülöp, 2011), con el que se puede evaluar y comparar con facilidad y eficiencia las salidas de las herramientas de ingeniería inversa, tales como, herramientas de minería de patrones de diseño, detectores de duplicación de código y verificadores de violación de las reglas. BEFRIEND soporta diferentes tipos de familias de herramientas, lenguajes de programación y sistemas de software, permitiendo a los usuarios definir sus propios criterios de evaluación. Este trabajo se limita a presentar la herramienta, pero no definen un instrumento de caracterización.

En consecuencia, no se ha encontrado en la literatura un instrumento que permita a usuarios, productores e investigadores caracterizar las herramientas de ingeniería inversa, centrando su atención exclusivamente en aquellos criterios particulares que le interesan, con el propósito de facilitar el análisis, la comprensión y la selección de la herramienta adecuada para la toma de decisiones al momento de adquirir, construir o mejorar un producto de este tipo, ocasionando pérdida de tiempo, esfuerzos y recursos. En este orden de ideas, surge la necesidad de un instrumento, bien definido y fácil de comprender que sirva a usuarios, productores e investigadores para caracterizarlas en forma clara y completa (Guéhéneuc et al, 2006; Fülöp et al, 2008; Canfora et al, 2011) y que permita a los diseñadores, ingenieros y administradores de proyecto, analizar las tecnologías necesarias que posibiliten ejecutar proyectos de diseño y desarrollo de una manera más avanzada, eficiente y cooperativa (Riba y Molina, 2006; Muñoz y Muñoz, 2010). Es por eso que en este artículo se propone un instrumento para la caracterización de herramientas de ingeniería inversa, construido a partir de un análisis imparcial y fundamentado en los trabajos previos relacionados anteriormente.

METODOLOGÍA

Inicialmente se hizo una revisión detallada del estado del arte que llevó a identificar los trabajos previos relacionados anteriormente, para cada uno de ellos se estableció la estructura de caracterización utilizada con el propósito de determinar aspectos comunes y complementarios. A partir de éste primer resultado se identifican las características que no se habían incluido, tomando como referente las últimas revisiones del estado del arte hechas por Canfora et al

(2007;2011), Tonella et al (2007) y Müller (2006), por ser autores que han realizado aportes significativos a la ingeniería inversa.

Luego se define una nueva estructura de caracterización fundamentada en la arquitectura genérica de las herramientas de ingeniería inversa propuesta por Chikofsky y Cross (1990). Posteriormente se procede a organizar dentro de la nueva estructura el listado general de características obtenidas hasta el momento. Finalmente se complementa la caracterización con nuevos criterios identificados por los autores a partir de su experiencia.

RESULTADOS

La caracterización de las herramientas de ingeniería inversa se hace teniendo en cuenta dos criterios: el primero corresponde al aspecto estructural, centrandó la atención en los elementos básicos que conforman la arquitectura genérica de este tipo de herramientas: Analizador e Interfaz (Chikofsky y Cross, 1990), mientras que el segundo se centra en las propiedades comunes entre ellas.

Aspecto estructural

Para caracterizar las herramientas de ingeniería inversa, tomando como principal referente su arquitectura genérica, se establece la siguiente estructura: elemento, función, aspecto y característica, como se observa en la tabla 1. Se han definido dos tipos de *elementos*: *Analizador e Interfaz*. El primero es el componente responsable de la transformación del software objeto de análisis a un mayor nivel de abstracción, mientras que la interfaz se encarga de permitir la interacción entre el usuario y la herramienta. Una *función* corresponde a las tareas que realiza cada elemento de la arquitectura, por ejemplo el analizador cumple con tareas de entrada, proceso y salida. El *aspecto* indica un matiz o rasgo diferenciador para cada función que cumplen los elementos, tal es el caso de la función entrada que cumple el analizador, para la cual se definen cinco aspectos: los supuestos de la entrada, la fuente, la precisión, la automatización y el manejo de versiones. Por último se encuentra la *característica*, entendida como una cualidad que sirve para distinguir a cada uno de los aspectos identificados para las funciones que cumplen los elementos.

En la tabla 1 se presenta la caracterización de las herramientas de ingeniería inversa tomando como referente principal los aspectos estructurales. Debido a que la mayoría de las características se han tomado a partir de propuestas previas y que el principal aporte de este trabajo se centra en la organización de dichas características para facilitar el análisis, la comprensión y selección de las herramientas, de tal manera que se puedan identificar sus ventajas y desventajas, a continuación sólo se explican aquellas que se han incluido, las demás se resaltan con un asterisco (*) para indicar que ha sido definidas por Bellay y Gall (1997) y con un signo más (+) las establecidas por Guéhéneuc et al (2006).

Para la función de entrada del analizador sólo se incluyen dos características nuevas asociadas a la fuente. La primera hace referencia al tipo de fuente, asumiendo que en la actualidad el proceso de ingeniería inversa incluye, además del código fuente, colecciones de datos y documentación. Adicionalmente, se ha definido la relación entre las características: tipo, modelo y representación de la fuente, teniendo en cuenta que para cada tipo de fuente existe un modelo que la describe y una forma de representación, como se muestra en la tabla 2.

La segunda característica corresponde al tipo de aplicación y se utiliza para definir la naturaleza de la tecnología visual según la experiencia de usuario y la tecnología con la que está implementada la aplicación objeto de análisis para la herramienta de ingeniería inversa, con el ánimo de facilitar la identificación del uso de patrones arquitectónicos. Los posibles tipos son: aplicación de escritorio, aplicación web, aplicación RIA (Aplicación Rica de Internet), servicio, aplicación móvil y aplicación para televisión digital.

Para la función proceso se ha definido el aspecto denominado tipo de analizador, teniendo en cuenta la propuesta hecha por Canfora et al (2011), quien distingue cuatro tipos de analizadores: Estático: extrae la estructura del sistema; dinámico: extrae el comportamiento del sistema; híbrido: extrae tanto la estructura como el comportamiento del sistema. Adicionalmente está el analizador histórico que extrae información que permite establecer la evolución de un producto software.

Tabla 1: Caracterización de Herramientas de Ingeniería Inversa según su arquitectura

Elemento	Función	Aspecto	Características
Analizador	Entrada	Supuestos (+)	Estructurales, Semánticos
		Fuente	Tipo, Modelo (+), Representación (+), Origen de datos, Tipos de datos(+), Tipos de aplicación
		Precisión (+)	Semántica, Sintáctica
		Automatización (+)	Automática, Semiautomática, Manual
		Versiones (+)	Múltiples, Singular
	Proceso	Técnica (+)	Adaptabilidad, Automatización, Complejidad, Determinismo, Explicación, Tratamiento de aspectos indefinidos, Nivel de detalle, Incremental, Iterativo, Independiente del lenguaje, Madurez, Método, Modelo, Escalabilidad, Semántica
		Tipo de analizador	Dinámicos, Estáticos, Híbridos, Históricos
		Aspectos generales	Reconocimiento de patrones, Artefactos que recupera, Análisis incremental (*), Reparsing (*), Preprocesador de comandos configurable (*), Soporte para conmutadores de compilador(*), Capacidad para analizar Funciones/Variables externas (*), Capacidad para trabajar con información incompleta, Capacidad para trabajar con caja negra, Identificación de clonación, Ingeniería inversa de Documentos WSDL en UML
	Salida	Tipo	Modelo, Representación
		Reporte (+)	Legibilidad, Nivel de detalle, Modelo, Calidad, Representación, Tipo de datos
Interfaz	Visualización	Trazabilidad	Horizontal, Vertical
		Tipo (*)	Estática, dinámica
		Tipo de Navegación (*)	Capas, Ojo de pescado, SHriMP: Simple Hierarchical Multi Perspective view
		Calidad (+)	Sintáctica, Semántica
		Niveles de abstracción	Código, Diseño, Arquitectura, Requerimientos, Modelo del negocio
		Análisis de los resultados	Capacidades de hipertexto (*), Capacidad para analizar diferencias, Mecanismos de consulta, Capacidad para agrupar cambios relacionados, Capacidad para clasificar y combinar información desde repositorios de software diferentes y heterogéneos
	Edición	Adaptabilidad del reporte generado	Anotaciones de ayuda (*), Permite organizar el informe (*), Capacidad para mejorar el diseño del reporte de salida (*), Facilidades de edición
		Capacidad de salida (*)	Generación automática de documentación, Formatos de exportación, Impresión de informes
		Generales	Definición de proyectos, Función de búsqueda, Capacidad para seleccionar entradas representativas

Por otra parte, en los aspectos generales se han incluido las siguientes características que permiten calificar de manera más detallada al analizador de una herramienta de ingeniería inversa: *Reconocimiento de patrones*: indica si la herramienta reconoce o no la existencia del uso de patrones a partir de la entrada que se le ha suministrado, en caso afirmativo se debe especificar el tipo de patrón reconocido (de diseño, arquitectónico, etc.) y los nombres de los patrones concretos que identifica según el tipo (singleton, Abstract factory, etc).

Tabla 2: Relación entre el tipo, el modelo y la representación de la fuente

Fuente		
Tipo	Modelo	Representación
Código Fuente	Paradigma de programación	Lenguajes de programación
Colección de datos	Relacional, jerárquico, etc.	SQL, XML, etc.
Documentación	UML, ADL, Lenguaje natural	Gráficos, texto

Artefactos recuperados: corresponde a las piezas de información que recupera el analizador y está directamente relacionado con su tipo, el modelo y la representación de la salida. Para el caso de un analizador estático que utilice como modelo UML, la representación de la salida podría ser a través de diagramas de clase, estructuras compuestas, de paquetes, de componentes o de despliegue; mientras que para un analizador dinámico que utilice el mismo modelo, la representación de la salida podría ser diagramas de casos de uso, de secuencia, comunicación, actividades, máquinas de estado, general de interacción, procesos, estado o tiempos.

Adicionalmente se han incluido características que corresponden a los retos actuales que deben atender las herramientas de ingeniería inversa, tales como la capacidad para trabajar con información incompleta, la capacidad para trabajar con caja negra para el caso concreto de los sistemas orientados a servicios, la capacidad para identificar clonación de código facilitando el trabajo de reutilización y la comprobación del manejo de derechos de autor; así como también la capacidad para realizar tareas de ingeniería inversa a partir de documentos WSDL.

Con respecto a la salida del analizador se ha incluido el tipo de salida, que es determinado por el tipo de analizador, por ejemplo el tipo de salida puede corresponder a una vista estática, cuyo modelo en UML está representado a por medio de un diagrama de clases.

El segundo componente representativo de cualquier herramienta de ingeniería inversa es la interfaz de usuario, la cual permite realizar funciones de visualización y edición. En el caso de la visualización, se presenta con frecuencia la necesidad de rastrear los modelos obtenidos como resultado de la ingeniería inversa desde el código fuente hasta el modelo del dominio del problema, pasando por el diseño y los requerimientos, lo que se conoce como trazabilidad horizontal. Del mismo modo, es necesario rastrear los distintos artefactos obtenidos para cada una de estas actividades (trazabilidad vertical), por ejemplo cuando a nivel de diseño se hace el seguimiento de la estructura de un componente por medio de su diagrama de clases y su comportamiento a través de un diagrama de secuencia (Pfleeger, 2010).

Adicionalmente, se incluye un nuevo aspecto denominado nivel de abstracción, que hace referencia a la capacidad que tiene la herramienta para permitir visualizar los resultados que arroja, desde el código fuente hasta el modelo del negocio, pasando por el diseño, las vistas arquitectónicas y los requerimientos del sistema. Además, se incorporan nuevas características que permiten establecer la capacidad de análisis de resultados de la herramienta, a través de mecanismos de consulta, de análisis de diferencias, de agrupación de cambios relacionados y de clasificación y combinación de información desde repositorios de software diferentes y heterogéneos.

Con respecto a las capacidades de edición de la herramienta se incluyeron características como la facilidad de edición, que implica acciones como cortar, copiar, pegar y deshacer, entre otras que permiten la adaptabilidad del reporte. También se agregaron aspectos generales de edición, tales

como: la posibilidad de definir proyectos para agrupar resultados según la necesidad del usuario; la capacidad para realizar búsquedas en los artefactos recuperados y la capacidad para seleccionar entradas representativas, dándole mayor libertad al usuario y haciendo más eficiente su trabajo.

Propiedades comunes

Para el segundo criterio que se utilizó en la caracterización se establece un factor y una propiedad, como se observa en la tabla 3. El factor corresponde a un matiz o rasgo diferenciador de la herramienta, mientras que la propiedad hace referencia a una cualidad que distingue a cada uno de los factores identificados para las herramientas. Todas las propiedades identificadas han sido definidas previamente por Guéhéneuc et al (2006) y aquellas marcadas con un asterisco (*) en la tabla 3 han sido descritas por Bellay y Gall (1997). En este trabajo se distinguen cuatro aspectos: Contexto, intención, perfil del usuario y madurez de la herramienta.

El *contexto* define el entorno y las restricciones externas de la herramienta, la *intención* establece los objetivos para los cuales ha sido desarrollada la herramienta, el *perfil de usuario* indica las habilidades y características que debe cumplir la persona que utilice la herramienta, mientras que la *madurez de la herramienta* hace referencia al conjunto propiedades que determinan el grado con el cual la herramienta cumple con los requerimientos técnicos del usuario.

Tabla 3: Propiedades Comunes

Factor	Propiedades
Contexto	Tiempo de vida, Metodología, Campo de acción, Escenario, Universo del discurso
Intención	Objetivos a largo plazo, Objetivos a corto plazo
Perfil del Usuario	Conocimiento, Experiencia, Prerrequisitos, Usuario objeto, Tipo de usuario
Madurez de la Herramienta	Calidad, Tipo de licencia, Independencia del lenguaje, Documentación, Plataformas soportadas (*), Editor Integrado o externo (*), Navegador Integrado o externo (*), Capacidad de almacenamiento (*), Base de usuarios, Soporte a múltiples usuarios

ANÁLISIS DE LOS RESULTADOS

El instrumento propuesto además de facilitar el análisis y entendimiento de las herramientas de ingeniería inversa, permite la valoración independiente y conjunta de cada uno de sus componentes. No pretende establecer un modelo terminado de caracterización, sino un instrumento dinámico que debe ser permanentemente actualizado para que atienda a las necesidades del momento. Se recomienda utilizarlo cuando se requiera caracterizar herramientas de ingeniería inversa, teniendo en cuenta que no es indispensable valorar todas las características identificadas, más bien, se sugiere establecer claramente cuáles son pertinentes a partir del objetivo propuesto para la valoración

Es de utilidad para usuarios de herramientas de ingeniería inversa, porque ofrece una plantilla de características que brindan este tipo de herramientas, de tal manera que puede seleccionar aquellas de su particular interés para establecer un comparativo entre las herramientas existentes, sirviéndole de referente y como instrumento de apoyo para la toma de decisión al momento de escoger la más apropiada. Del mismo modo, este instrumento puede ser utilizado por los productores, para caracterizar la herramienta que desarrollan, identificando sus fortalezas para mostrarlas como estrategia de mercadeo y las debilidades para establecer los aspectos a mejorar y futuros desarrollos.

Por otra parte, los investigadores en el campo de la ingeniería inversa pueden utilizarlo para hacer estudios tendientes a clasificar las herramientas o a identificar aspectos pendientes por resolver,

como se demuestra a continuación, a partir de tres situaciones que sirven para validar el instrumento propuesto. La primera situación corresponde a la clasificación que se hizo tomando como principal referente los tipos de entrada y de salida de las herramientas, como se observa en la tabla 4 y en la figura 1. Esto hace posible centrar la atención sólo en aquellas herramientas que reciban un tipo de entrada y salida específica de acuerdo a la necesidad del interesado.

Tabla 4: Clasificación de las herramientas de ingeniería inversa

Tipo de Herramienta	Entrada	Salida
Desensamblador	Archivo ejecutable	Lenguaje ensamblado
Decompilador	Archivo ejecutable	Lenguaje de alto nivel
Depurador	Archivo ejecutable	Simulación de ejecución
Analizador de colección de datos	Colección de datos	Modelo de datos
Analizador de código fuente	Código fuente	Modelo de la aplicación

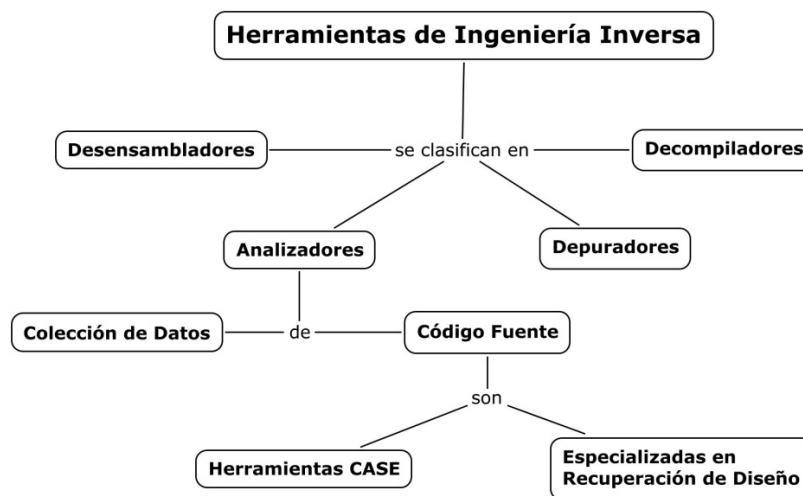


Fig. 1: Clasificación de las herramientas de ingeniería inversa

La segunda situación plantea como propósito identificar qué artefactos de las vistas estáticas y dinámicas son recuperados actualmente por las herramientas de ingeniería inversa. Para lograr el objetivo se caracterizaron 38 herramientas (analizadores de código fuente) relacionadas en la tabla 5, centrandó la atención en la característica denominada: artefactos recuperados, que corresponde a los aspectos generales de la función proceso realizada por el analizador, según el instrumento propuesto. Los resultados obtenidos se presentan en la tabla 6 y evidencian que la mayoría de las herramientas se concentran en la vista estática, sobre todo en el diagrama de clases, pero ninguna recupera diagramas de componentes, diagramas de estructuras compuestas ni diagramas de despliegue.

También se identificó que sólo una tercera parte recupera el comportamiento del sistema a través de diagramas de secuencia, mientras que apenas una lo hace por medio de diagramas de casos de uso y otra por medio de diagramas de actividades, pero ninguna lo hace usando diagramas de comunicación, de procesos, de estado, ni de tiempos.

La tercera situación plantea como objetivo identificar para qué tipos de aplicación realizan las herramientas de ingeniería inversa los procesos de recuperación de las vistas del sistema. Esto se logró caracterizando las mismas 38 herramientas, valorando para la función entrada del analizador la característica tipo de aplicación del aspecto denominado: fuente. Los resultados obtenidos se presentan en la tabla 7 y evidencian que casi la totalidad lo hacen para aplicaciones de escritorio, sólo ocho lo hacen para aplicaciones web y cuatro para servicios, mientras que ninguna lo hacen para aplicaciones ricas de internet, para aplicaciones móviles ni para aplicaciones para televisión digital.

Es pertinente aclarar que al momento de hacer una evaluación completa de las herramientas de ingeniería inversa, se debe tener en cuenta que son sistemas software y por lo tanto se requiere utilizar marcos de referencia generales para su evaluar, además son herramientas computacionales para el apoyo a la ingeniería de software, motivo por el cual, la evaluación también se debe ajustar al estándar ISO/IEC 14102: guía para la selección y evaluación de herramientas CASE (ISO/EC, 2008).

Tabla 5: Listado de herramientas caracterizadas

Herramientas CASE		Especializadas en recuperación de diseño				
Visual Paradigm	Together	Fujaba–Archimatrix	Bauhaus	Imagix 4D	SrcML	Sniff+
Rational Rose	Altova	Fujaba–Reclipse	REgoLive	MyEclipse	SolidFX	Rigi
Magic Draw	Metamil 5.0	Q-ImPRESS SISSy	MARPLE	Green UML	Refine/C	XDRE
Enterprise Archect	Argo UML	MacTranslator	Columbus	CodeLogic	CPP2XMI	
Plataforma Moose	UML Studio	WinTranslator	MoDisco	ESS-Model	PRECISO	
PowerDesigner		WSAD extractor	Jar2UML	SWAG KIT	Jinsight	

Tabla 6: Caracterización de herramientas según artefactos recuperados

Vista	Artefacto recuperado	Cantidad	Porcentaje
Estática	Diagramas de clase	24	63%
	Diagramas de paquetes	6	16%
	Diagramas de componentes	0	0%
	Diagramas de estructuras compuestas	0	0%
	Diagramas de Despliegue	0	0%
	Grafo de relaciones	13	34%
Dinámica	Diagramas de secuencia	12	32%
	Diagramas de comunicación	0	0%
	Diagramas de casos de uso	1	3%
	Diagramas de actividades	1	3%
	Diagramas de procesos	0	0%
	Diagramas de estado	0	0%
	Diagramas de tiempos	0	0%
	Diagramas de Flujo	1	3%

Tabla 7: Caracterización de herramientas según tipo de aplicación

Tipo de aplicación	Cantidad	Porcentaje
Aplicación de escritorio	35	92%
Aplicación Web	8	21%
Aplicaciones ricas de internet	0	0%
Aplicación móvil	0	0%
Servicio	4	11%
Aplicación para televisión digital	0	0%

CONCLUSIONES

El instrumento propuesto es un complemento a trabajos anteriores, manteniendo su esencia y facilitando su interpretación, organiza 103 características a través de una nueva estructura, que permite la valoración independiente y conjunta de cada uno de los componentes que conforman las herramientas de ingeniería inversa, incluyendo los nuevos requerimientos y desafíos que actualmente deben atender este tipo de herramientas, con el propósito de facilitar la toma de decisiones a usuarios, productores e investigadores al momento de seleccionarlas, evaluarlas y perfeccionarlas.

La validación del instrumento permitió demostrar su uso en distintas situaciones, a partir de las cuales se puede concluir que:

i) Se reconfirma que la mayoría de herramientas de ingeniería inversa se dedican a recuperar los aspectos estáticos del sistema, concretamente a través de diagramas de clase y grafos de relaciones, sin embargo, todavía queda mucho por hacer con respecto a representaciones con un mayor nivel de abstracción a través de diagramas de paquetes, de componentes y de estructuras compuestas.

ii) Las pocas herramientas que recuperan los aspectos dinámicos del sistema lo hace a través de diagramas de secuencia, ninguna aborda aspectos específicos como el seguimiento del estado de los componentes del sistema, la sincronización de procesos ni la concurrencia.

iii) Enterprise Architect, Visual Paradigm y Rational Rose son las herramientas que cumplen con la mayor parte de los parámetros deseables de la ingeniería inversa.

iv) Se deben desarrollar herramientas de ingeniería inversa para los nuevos tipos de aplicación emergentes.

En el futuro el grupo de investigadores realizará una evaluación de la calidad de los artefactos recuperados por las herramientas de ingeniería inversa, identificando los niveles de precisión y confiabilidad de la información recuperada, según lo establecido por la norma ISO/IEC 25010, en los aspectos relacionados con la característica denominada adecuación funcional.

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.

REFERENCIAS

Bellay B. y G. Harald, A Comparison of four Reverse Engineering Tools, 4th Working Conference on Reverse Engineering, Amsterdam, Holanda, 2 – 11, 6 al 8 de octubre (1997).

Brown A. y K. Wallnau, A Framework for Evaluating Software Technology. IEEE Software, 13(5), 39–49 (1996).

Canfora G. y A. Cimitile, A Logic-Based Approach to Reverse Engineering Tools Production, IEEE Transactions On Software Engineering, 18(12), 1053 – 1064 (1992).

Canfora G. y M. Di Penta, New Frontiers of Reverse Engineering, IEEE Future of Software Engineering, 326 – 341 (2007).

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, IEEE Software, 7(1), 13 – 17 (1990).
- Dong J., Zhao Y. y Tu Peng, Architecture and Design Pattern Discovery Techniques – A Review, 29th International Conference on Software Engineering, 621 – 627, Minneapolis, Minnesota USA, 19 al 27 de mayo (2007).
- Fülöp, L.J., Hegedus, P., Ferenc, R. y T. Gyimothy, Towards a Benchmark for Evaluating Reverse Engineering Tools. 15th Working Conference on Reverse Engineering. 335 – 336, Antwerp, Bélgica, 15 al 18 de octubre (2008).
- Fülöp, L.J., Evaluating and Improving Reverse Engineering Tools. Doctoral thesis, University of Szeged, Department of Software Engineering, June (2011).
- Gannod G. y B. Cheng, A Framework for Classifying and Comparing Software Reverse Engineering and Design Recovery Techniques, 6th Working Conference on Reverse Engineering, 77 – 88, Atlanta, Georgia USA, 6 al 8 de octubre (1999)
- Guéhéneuc Y., Mens K. y R. Wuyts, A Comparative Framework for Design Recovery Tools, Proceedings of the 10th European Conference on Software Maintenance and Reengineering, 134 – 143, Bari, Italia, 22 al 24 de marzo (2006).
- Müller H. A., Bits of History, Challenges for the Future and Autonomic Computing Technology, 13th Working Conference on Reverse Engineering, 9 – 18, Benvenuto, Italia, 23 al 27 de octubre (2006).
- Muñoz D. F y D. Muñoz, Planeación y Control de Proyectos con Diferentes Tipos de Precedencias Utilizando Simulación Estocástica. Información Tecnológica. 21(4), 25-33 (2010).
- Pfleeger S. L, Ingeniería de Software: Teoría y Práctica, Edición en español, 568, Pearson Education S.A. Buenos Aires, Argentina (2002).
- Rasool G., Maeder P., y I. Philippow, Evaluation of design pattern recovery tools, ScienceDirect Procedia Computer Science, 3, 813 – 819 (2010).
- Riba C. y A. Molina, Ingeniería concurrente: una metodología integradora. Primera edición, 306, Edicions UPC, Barcelona, España (2006).
- Storey M., y otros 5 autores, On Designing an Experiment to Evaluate a Reverse Engineering Tool, Third Working Conference on Reverse Engineering (WCRE '96), 31 – 40, Monterey, California USA, 10 y 11 de noviembre (1996).
- Tonella P., y otros 3 autores, Empirical studies in reverse engineering: state of the art and future trends, Springer Science + Business Media, 12(5), 551-571 (2007).

