

CAPITULO VII

Conclusión

1. Análisis del Enfoque Propuesto

Se define a un concern como cualquier cuestión de interés en un sistema de software. Involucra todo lo que un stakeholder quiera considerar como una unidad conceptual, incluyendo características, requerimientos no funcionales y decisiones de diseño e inclusive *programming idioms* [2]. Los concerns que atraviesan los módulos correspondientes a la descomposición principal del sistema son llamados crosscutting concerns.

El enfoque propuesto tiene como objetivo asistir en la búsqueda de crosscutting concerns que pueden convertirse en potenciales aspectos en un código fuente. De esta manera, se puede realizar una futura refactorización con el fin de mejorar el entendimiento y mantenimiento del sistema.

AMT (Aspect Mining Tool) es una herramienta desarrollada como un plugin para la plataforma Eclipse. La misma analiza el código fuente de un sistema desarrollado en Java utilizando técnicas de aspect mining estáticas con el fin de identificar los crosscutting concerns presentes en el sistema. Se implementaron 5 técnicas, cada una constituyendo un sistema experto. Las técnicas implementadas son: análisis de Fan-in [10], análisis de Métodos Únicos [7], análisis de Relaciones de Ejecución [20], análisis de Métodos Redirectores [22] y un 5to enfoque que converge a los 3 primeros.

La herramienta fue probada con el sistema Health Watcher (HW) [1]. HW es la típica aplicación web de información, y el propósito principal del mismo es permitir a ciudadanos

registrar sus quejas referidas a temas de salud. El mismo es seleccionado debido a que es un sistema real y suficientemente complejo. El sistema involucra un gran número de concerns clásicos como por ejemplo sistemas usables, concurrentes, persistentes y distribuidos. Adicionalmente, la aplicación utiliza tecnologías utilizadas en contextos industriales como RMI (Java Remote Method Invocations) [1.18], Servlets [1.22] y JDBC (Java Database Connectivity).

Se realizaron 3 análisis sobre dicho sistema, dos correspondiente al enfoque denominado sinergia (variando sus parámetros de entrada) y un tercer análisis utilizando métodos redireccionadores. Los concerns identificados se compararon con los concerns identificados en trabajos previos [1]. El enfoque Sinergia identificó el 50% y 66.66% de los concerns. El análisis de métodos redirectores identificó el 16.66% de los crosscutting y agregó al conjunto la identificación de patrones adapters diseminados en la capa de negocio.

Las técnicas implementadas por AMT obtienen sus resultados a partir de la información estática del código fuente de un sistema. Una ventaja importante que presenta esta herramienta es la flexibilidad con que se implementan los análisis, debido a que se utiliza un sistema experto para cada uno, el cual mapea el conocimiento en reglas. Este tipo de implementación permite combinar más de una técnica con facilidad (por ejemplo Sinergia). Otra ventaja importante es la rapidez con la que se ejecutan los análisis sobre todo el código fuente de una aplicación una vez que se cuenta con la información representada como hechos en la base de datos. Esto se debe a la utilización de un motor de inferencia para la ejecución de las reglas de cada algoritmo.

Una desventaja de esta herramienta reside en la naturaleza de las técnicas implementadas. Dichas técnicas identifican dos tipos de concerns: aquellos que fueron implementados como llamadas entre métodos y aquellos que fueron implementados utilizando patrones del estilo adapter o decorator. Debido a esto, los concerns que pueden ser identificados por la herramienta se reducen al grupo listado previamente.

2. Trabajos Futuros

A continuación se listan posibles proyectos que puedan extender o continuar el trabajo realizado:

- Creación de nuevos sistemas expertos con el fin de implementar nuevos algoritmos de aspect mining.
- Creación de nuevos sistemas expertos con el objetivo de combinar más de una técnica de aspect mining realizando un estudio detallado de los parámetros y las variaciones de los mismos.
- Implementación de recomendación de refactorings utilizando las salidas obtenidas y nuevas reglas de inferencia.
- Implementación de refactorings automatizados utilizando las salidas obtenidas y nuevas reglas de inferencia.

Referencias

[2] Kiczales, G., Hilsdale, E., Hugunin, J., Kersten, M., Palm, J., Griswold, W. G.: An Overview of Aspectj. In: Knudsen, J. L., Knudsen, J. L. (eds.) ECOOP 2001. LNCS, vol.2072, pp. 327--353, Springer (2001)

[24] F. Buschmann, R. Meunier, H. Rohnert, P. Sommerlad, M. Stal, P. Sommerlad, and M. Stal, Pattern-Oriented Software Architecture, Volume 1: A System of Patterns. John Wiley & Sons, August 1996.

[1] Soares, S.; Borba, P.; Laureano, E.: Distribution and Persistence as Aspects. In: Software Practice and Experience, Wiley, vol. 36 (7), (2006) 711-759.

[10] Tonella, P., Ceccato, M. Aspect Mining through the Formal Concept Analysis of Execution Traces. In: 11th Working Conference on Reverse Engineering, pp. 112--121. IEEE Computer Society, Washington DC, USA (2004)

[7] Gybels, K. and Kellens, A. "Experiences with Identifying Aspects in Smalltalk Using Unique Methods," in: International Conference on Aspect Oriented Software Development. Amsterdam, The Netherlands 2005.

[22] Marius Marin, Leon Moonen, Arie van Deursen, "A common framework for aspect mining based on crosscutting concern sorts," wcre, pp.29-38, 13th Working Conference on Reverse Engineering (WCRE 2006), 2006

[20] J. Krinke. Mining Control Flow Graphs for Crosscutting Concerns. In Proceedings of the 13th Working Conference on Reverse Engineering (WCRE 2006), pages 334–342, Washington, DC, USA, 2006. IEEE Computer Society.

[1.18] S. Microsystems. Java Remote Method Invocation (RMI). At <http://java.sun.com/products/jdk/1.2/docs/guide/rmi>, 2001.

[1.22] Hunter, J., Crawford, W.: Java Servlet Programming. O'Reilly and Associates Inc. 1998