

Herramienta para la generación de código Android a partir de modelos conceptuales

William J. Giraldo, Manuel A. Pineda, Robinson Arias , María L. Villegas y
Fáber D. Giraldo¹

Ingeniería de Sistemas y Computación, Universidad del Quindío, Colombia
email: {wjgiraldo, mapineda, rarias, mlvillegas,
fdgiraldo}@uniquindio.edu.co

Abstract. En este trabajo se presenta un prototipo de herramienta para generación de aplicaciones Android, a partir de un conjunto de modelos conceptuales y diagramas. Esta herramienta está concebida para generar el 100% de la estructura funcional de una aplicación Android.

Keywords: Modelos conceptuales, Android, generación de código.

1 Descripción de la propuesta

En el marco del proyecto de investigación titulado *Framework para la ejecución de modelos ejecutables específicos del dominio*, avalado por COLCIENCIAS en Colombia, y derivado de [1], se ha desarrollado una herramienta bajo los principios del paradigma de la ingeniería dirigida por modelos, con el objetivo de generar aplicaciones Android a partir de los modelos conceptuales a continuación relacionados:

1. *Diagrama de interacción:* este diagrama modela las tareas que realizará la aplicación. También modela el comportamiento y comunicación entre éstas. El modelo de interacción se expresa mediante el árbol de tareas concurrentes (Concurrent Task-Tree o CTT [3] por sus siglas en inglés).
2. *Interfaces concretas de usuario (Concrete User Interfaces - CUI):* se seleccionan los posibles widgets que se mostrarán en la interfaz, basado en las restricciones de la tecnología (en este caso Android) y el *diagrama de interacción*.
3. *Diagrama de navegación:* este diagrama modela la navegación entre cada una de las interfaces. El diagrama está basado en máquinas de estado.
4. *Modelo de dominio:* este diagrama modela los datos de la aplicación para permitir la persistencia e integridad de la información. Para representar este diagrama se usar un modelo de clases.
5. *Diagrama de diálogo:* este diagrama modela el *diálogo interno* de las interfaces. Para esto se hace una propuesta basada en el diagrama de CTT. Dentro de las tareas se encuentran los *objetos de interacción*, los cuales estarán compuestos de *widgets* y *tipos de interacción*, esto con el fin de indicar

que el objeto de interacción tiene widgets que actúan cada vez que se realiza el tipo de interacción.

La herramienta recorre cada uno de los modelos con el fin de detectar los elementos deben a generar. La herramienta recopila información de cada uno de los diagramas, por ejemplo:

- Desde el *diagrama de CUI* se recopila la información de los widgets para crear las variables necesarias en su manipulación. Adicionalmente se crea la interfaz de usuario.
- Desde el *diagrama de dominio* se recopila la información de las variables anteriormente mencionadas, requeridas para la lógica del aplicativo
- Desde el *diagrama CTT* se recopila la información de las tareas que realizará la aplicación.
- Desde el *diagrama de diálogo* se recopila la información sobre los eventos que se realizaran sobre los widgets, el orden en que se deben mostrar en pantalla éstos elementos y las propiedades que estos puedan tener.
- Desde el *diagrama de navegación* se recopila la información necesaria para crear los métodos de invocación a otras actividades

La aplicación generada cuenta con un planificador de tareas basado en el algoritmo de Luyten [2] con el fin de dar ejecución a las tareas de la aplicación basada en el orden en el que fue estipulado desde el diagrama de interacción CTT. El planificador detecta las tareas que se deben ejecutar en cada momento y desplanificará las tareas que han terminado su ejecución.

La herramienta genera completamente la estructura funcional de una aplicación Android, a excepción de la operaciones de la lógica de negocio, la cual deberá ser introducida por el desarrollador. Aunque la herramienta no captura ni describe por completo la lógica de negocio para una gran cantidad de aplicaciones, esto se podría realizar mediante la combinación de diagramas de navegación y de diálogo (trabajo futuro).

La herramienta fue desarrollada en Eclipse EMF y GMF, y se basa en la metodología TD-MBUID (*Task & Data Model Based User Interface Development*) formulada en el grupo de investigación. Las transformaciones modelo-texto se realizan directamente en Java (es decir, sin usar plantillas ni plug-ins como Aceleo o EGL), creando para ello un plug-in Eclipse propio. En el CIbSE 2015 se presentará el uso de la herramienta a través de un Lab demo.

References

1. William Joseph Giraldo. *Marco de Desarrollo de Sistemas Groupware Interactivos Basado en la Integración de Procesos y Notaciones*. Thesis, 2010.
2. Kris Luyten, Promotor Prof, and Karin Coninx. *Dynamic User Interface Generation for Mobile and Embedded Systems with Model-Based User Interface Development*. Thesis, 2004.
3. F. Paternò. *The Handbook of Task Analysis for Human-Computer Interaction*, chapter ConcurTaskTrees: An engineered notation for task models, pages 483–503. Lawrence Erlbaum Associates, 2003.