

# Análisis del estado del arte en la depuración de sistemas embebidos

M. Giura, M. Prieto, N. González, L. Sugezky, M. Trujillo, J.M. Cruz

Universidad Tecnológica Nacional, Facultad Regional Buenos Aires

## Abstract

El proceso de depuración en sistemas embebidos permite detectar errores en la lógica de la programación. En los últimos años han surgido técnicas para sistematizar este proceso. En particular, el uso de modelos en sistemas embebidos se ha vuelto cada vez más frecuente y necesario para describir su comportamiento, por lo que contar con herramientas que permitan realizar la depuración del sistema resulta imprescindible. En el marco del proyecto de investigación "Desarrollo de software de simulación para la integración con uModelFactory" (EIUTNBA0002436) se lleva adelante el presente trabajo orientado a relevar y caracterizar las herramientas y técnicas existentes para la depuración de sistemas embebidos con el objetivo de implementarlas en el software uModel Factory.

## Introducción

En la actualidad existen diferentes enfoques orientados a la depuración de sistemas embebidos, principalmente pueden clasificarse en intrusivos o no intrusivos. A su vez, podemos evaluar los mismos como de tiempo real o de tiempo diferido (Figura 1). En el presente trabajo se analizan diferentes propuestas y herramientas orientadas al proceso de depuración

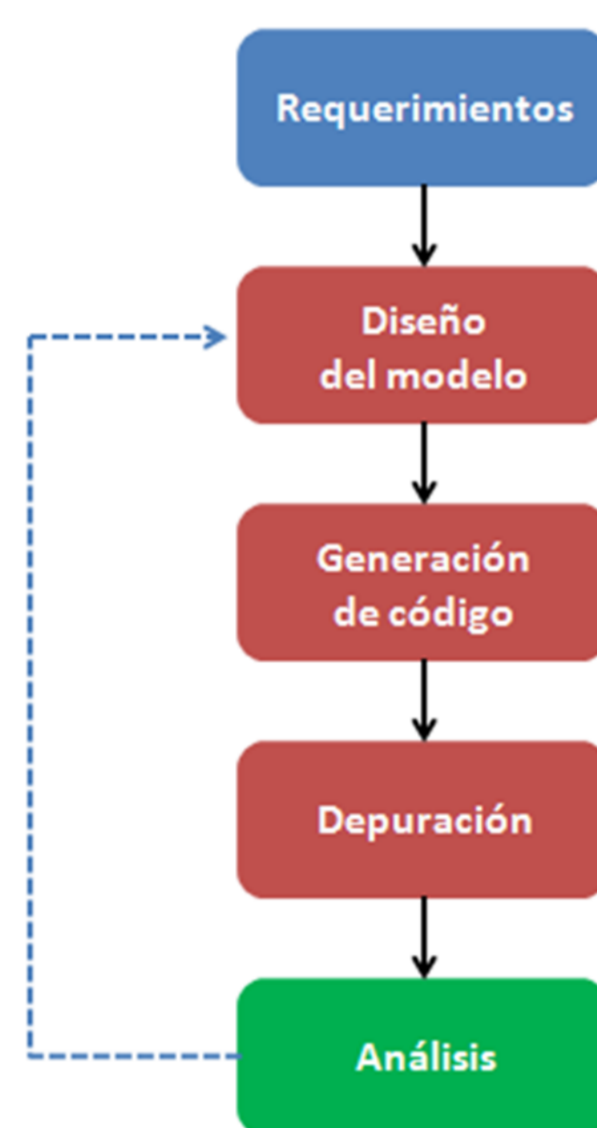


Figura 1

## Técnicas Actuales

### Evaluación y análisis de los datos provenientes del proceso de depuración

Los autores que abordan el tema centran su trabajo en el rastreo y la gestión de rastreo (tracing and trace management), donde analizan los problemas actuales del proceso de depuración mediante la recopilación de información sobre la ejecución del programa embebido.

Partiendo de las ventajas y desventajas de las técnicas actuales de rastreo (datos "crudos" almacenados en archivos y archivos formateados con convenientes estructuras de datos), realizan una propuesta con un nuevo enfoque: utilizar una infraestructura de gestión de trazas para los sistemas embebidos que aborde los problemas de las cuatro cuestiones ya señaladas.

### El proceso de depuración en tiempo real. Técnicas intrusivas y no intrusivas.

La técnica de trazado en tiempo real no intrusiva, cada vez con más soporte en los nuevos procesadores, cuenta con las siguientes ventajas:

- No es necesario detener la aplicación a depurar.
- Se cuenta con la facilidad de capturar y almacenar detalles de la temporización (timing) en tiempo real de ejecución.
- Son capaces de capturar y almacenar eventos específicos de interés, descartando el resto, pudiendo observar hacia adelante o hacia atrás, en tiempo real, desde ese punto de interés (trigger condition), qué es lo que está pasando.

### El proceso de depuración bajo un sistema operativo embebido

La propuesta consiste en registrar una marca de tiempo (time-stamp) de los eventos elegidos a almacenar y que en principio estos eventos solo sean el cambio de tareas, pudiendo también tomar como posibles eventos a almacenar las IPCs y System Calls llamadas por las tareas

### La depuración en embebidos utilizando herramientas de bajo nivel

Se presenta el CoreSight el cual es una tecnología propuesta por ARM la cual proporciona funcionalidades de seguimiento y depuración con el objetivo de tener información del SoC entero. CoreSight es un conjunto de componentes de hardware que pueden ser elegidos e implementados apropiadamente por el diseñador del sistema para extender la depuración

## Soluciones Actuales

### Embedded UML Target Debugger

La propuesta de esta solución consiste en una técnica compuesta por un módulo para depuración del lado de la interfaz más un módulo de monitoreo del lado del microprocesador. El módulo de monitoreo está incluido en el sistema operativo de tiempo real (RTOS) que se ejecuta en el microprocesador. El módulo de depuración que se encuentra del lado de la interfaz gráfica recibe los datos de seguimiento recopilados por el módulo de monitoreo en tiempo real y reconstruye e interpreta dicha información por medio del uso de diagramas UML como son diagramas de tiempo o secuencia.

### LieberLieber Embedded Engineer for Enterprise Architect

Esta solución propone dentro de un mismo software tanto la generación del código en ANSI-C o C++ a partir de las estructuras UML, las máquinas de estados y las actividades del modelo como un módulo de depuración del modelo (UML).

Este desarrollo por medio del módulo de depuración UML ha resuelto un problema que venía influyendo a las herramientas de generación de código ya que el desarrollo y la depuración se daban en lugares diferentes, mientras el desarrollo se realizaba con el modelo UML la depuración se llevaba a cabo en el microprocesador en C o C++ sin conexión. Con la herramienta de depuración que contiene este software se permite la conexión entre la depuración del modelo desarrollado y el código generado que se ejecuta en el hardware.