

USO DE LENGUAJE UNIFICADO EN LA MODELIZACIÓN DE SISTEMAS OPERATIVOS (WIP)

Nicanor Casas¹, Inés Casanovas², Graciela de Lucha³

Abstract — *In the current operating systems, Object Oriented modeling tools are being partially used, because they stand on traditional concepts of process, memory location and file. At the moment, the system integrative architecture is based on an abstract machine which offers a minimal functionality and an operating system that has to extend this functionality, thus colliding with the starting functionalities of the system. The aim of the study is to confirm that the modeling of an operating system by means of Unified Language techniques has some inconveniences in defining the required procedures, specially those issues related with low level interaction, being necessary the creation or redefinition of tools which allow the modeling of a generic operative system having also the vision of a Object Oriented comprehensive system as grounding part of a distributed operating system.*

Index Terms — *Operating systems, modeling, Unified Language inconveniences.*

DESCRIPCIÓN DEL PROBLEMA

Es sabido que los sistemas operativos debido a su tamaño y complejidad, han sufrido históricamente de problemas de modelización. La utilización de tecnologías orientadas a objetos se propone como solución a estos problemas de ingeniería. Las características del modelado de objetos de un sistema operativo no debieran diferir de las que son consideradas de manera tácita como las más representativas de la OO (Orientación a Objetos), y que fundamentalmente se encuentran recogidas en UML.

Nos planteamos si es posible mantener los beneficios de la encapsulación, herencia y polimorfismo en un sistema distribuido modelado con herramientas OO. Los sistemas de software de tiempo real, en particular los sistemas operativos, representan un desafío de modelización porque tienen comportamientos orientados a eventos y contienen limitaciones temporales y de recursos. El análisis basado en escenarios proporciona soluciones a las problemáticas de alto nivel, mientras que la programación orientada a objetos proporciona las adecuadas a las de bajo nivel. Queda pendiente de resolver este gap. En la mayor parte de los sistemas operativos actuales, la aplicación de las herramientas de modelado de OO se realiza de una manera parcial, dado que se sustenta en los conceptos tradicionales de proceso, posición de memoria y archivo, y los

mecanismos y políticas están codificados de manera rígida reduciendo la posibilidad de adaptarlos al entorno en el que va a prestar servicios.

Consideramos que una aproximación interesante a la integración de ambos aspectos, la del modelado y la programación, es la creación de un modelo que permita la generación de un sistema operativo general que tenga la visión de un sistema integral OO como parte fundante de un sistema operativo distribuido. La integración de la máquina abstracta y el sistema operativo se conseguiría a través de una arquitectura reflectiva de la primera, que permita definir la máquina como una estructura de objetos perfectamente cuantificables. Una visión panorámica de los enfoques actualmente discutidos en el dominio de investigación de los Sistemas Operativos muestra los siguientes métodos: a)UML Use Cases (una aproximación basada en casos [1]), b)Diagramas de transición de estados (vistas a nivel de detalle de eventos y sus transiciones)[2], c)Notación jerárquica gráfica (escenario-grafo de episodios conectados [3]), y d)Mapas de Casos de Uso (descripción visual de relaciones causales entre responsabilidades impuestas en estructuras de componentes abstractos [4]). Estas técnicas de modelado, en función de la hipótesis de este estudio, resultarían inadecuadas para representar claramente todos los atributos presentes en tiempo real.

El equipo del proyecto, de dos años de duración, se propone constatar que el modelado de un sistema operativo a través de técnicas de UML, presenta inconvenientes para definir los procedimientos a bajo nivel que requiere un sistema operativo, en especial lo relacionado a la interacción entre lenguajes cercanos a la máquina, o lenguajes de bajo nivel y las causas, efectos y responsabilidades a nivel de comportamiento de sistemas donde los escenarios pueden cambiar en tiempo de ejecución, y se requiere capacidad de refinamiento durante la variación de estos escenarios

REFERENCIAS

- [1] Jacobson J. et al., "OO Software Engineering", Addison Wesley, 1992.
- [2] Glintz M. et al., "Improving the Quality of Requirements with Scenarios", Proceedings 2nd World Congress for Software Quality, Japón, 2000.
- [3] Regnell B. et al., "A hierarchical Use Case Model with Graphical Representation", IEEE Symposium on Engineering of CBS, 1996.
- [4] Buhr R., "Use Case Maps as Architectural Entities for Complex Systems, IEEE Transactions on Software Engineering, 24-12, 1998, p.1131.

1 Nicanor Casas, Universidad Tecnológica Nacional – FRBA, Buenos Aires, Argentina, pecas@interar.com.ar

2 Inés Casanovas, Universidad Tecnológica Nacional – FRBA, Buenos Aires, Argentina, inescasanovas@gmail.com

3 Graciela De Luca, Universidad Tecnológica Nacional – FRBA, Buenos Aires, Argentina, nicari@ciudad.com.ar