An efficient component model for the construction of adaptive middleware

Michael Clarke, Gordon S. Blair, Geoff Coulson and Nikos Parlavantzas

Presentado por Carlos E. Agüero

caguero@gsyc.escet.urjc.es

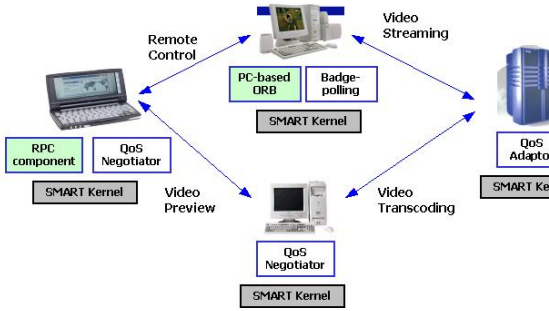
Universidad Rey Juan Carlos

El documento original está disponible en
`http://www.lancs.ac.uk/postgrad/parlavan/publications/middleware2001.pdf`

Índice

Introducción

- Middleware
  - Alto nivel
  - Independencia de plataforma
  - Enmascaramiento de recursos distribuidos
  - CORBA, RMI, JINI, ...
- Modelos basados en componentes
  - Configuración mínima en sistemas empotrados
  - Configuración máxima en multimedia
- Modelos basados en reflexión
- Revisado de OpenORB usando OpenCOM



Diseño de OpenCOM (I)

- Basado en Microsoft COM:
  - Standard
  - Muy extendido y usado
  - Independiente del lenguaje
- Uso del núcleo de COM y construcción propia del widdleware

Funcionalidad

- Interfaces: Ofrece algún servicio
- Receptáculos: Requieren algún servicio
- Conexiones: Conectan receptáculos con instancias de interfaces
- *run-time* OpenCOM: Singleton, gestiona repositorio de componentes

Diseño de OpenCOM (II)

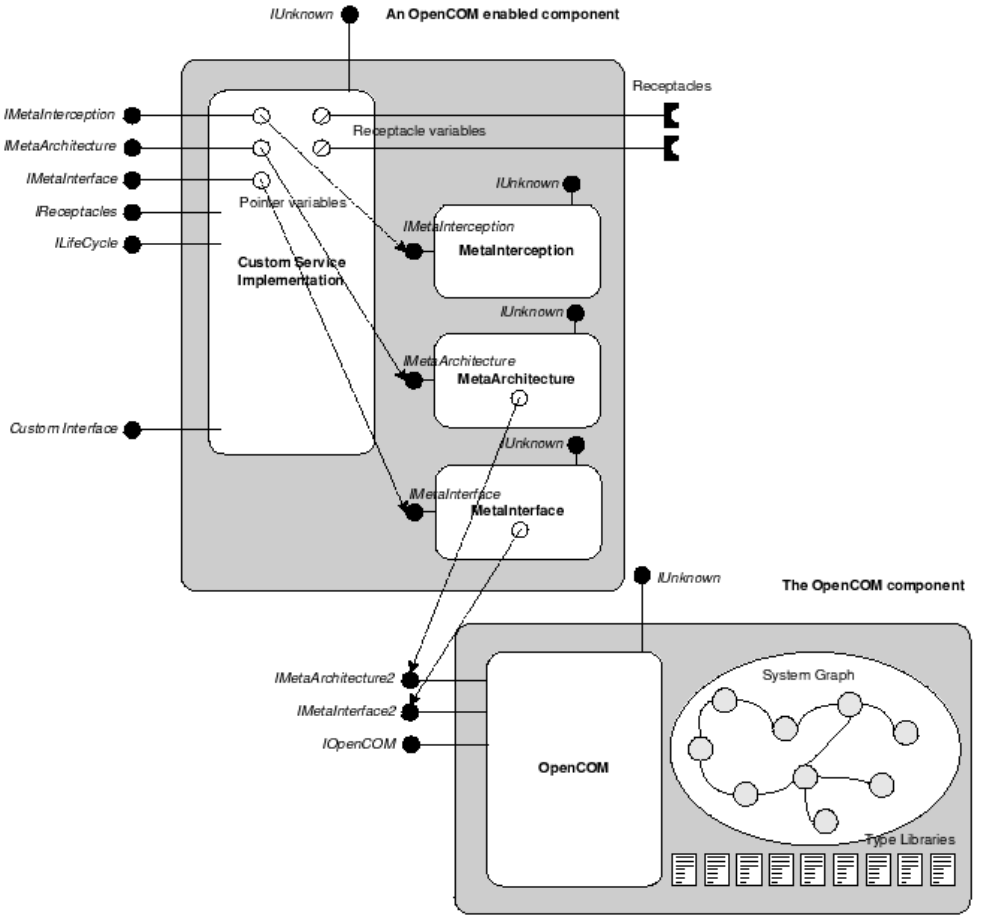
Interfaces IReceptacles e ILyfeCycle

- Usados por el run-time para el creado y borrado de componentes

Reflexión

- Componentes deben heredar de 3 interfaces:
  - IMetaIterception: Permite al programador asociar componentes con interfaces
  - IMetaArchitecture: Permite obtener Id's de receptáculos e interfaces conectados
  - IMetaInterface: Retro-inspección de los tipos declarados

Diseño de OpenCOM (III)



Implementación de OpenCOM

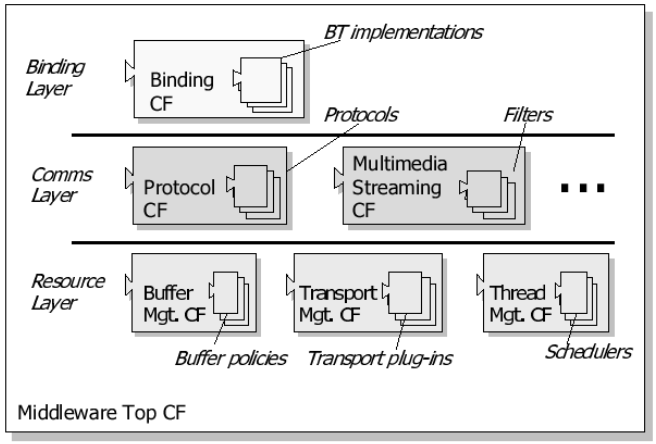
- Lenguaje utilizado: C++
- **Receptáculos**
  - Plantillas con la implementación del componente
  - Puntero a la interfaz conectada
  - Punteros simples, multipunteros, ...
- **MetaArquiquitectura**
  - Arquitectura del run-time
  - Grafo del sistema con piezas conectadas
  - MetaArquitectura envuelve este grafo y sus conexiones
- **MetaInterfaz**
  - Recubrimiento de la biblioteca de tipos (objeto COM)

OpenORB v2

Diseño

- OpenCOM para construir Middleware adaptativo
- Usada la noción de *Component Framework*
  - Reglas e interfaces para regular la interacción de componentes

Estructura

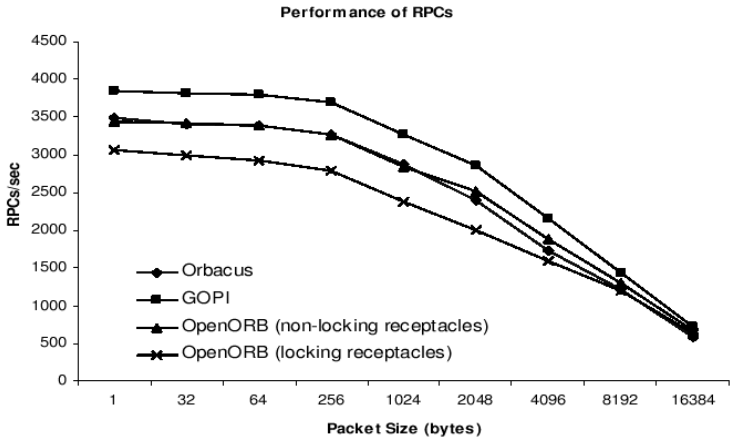


Pruebas de rendimiento

Rendimiento de OpenCOM

- Principales cuellos de botella:
  - Invocaciones a receptáculos e intercepciones
- Marcada diferencia en el proceso necesario para invocar un método

Rendimiento de OpenORB v2



Conclusiones

- Modelo de componentes ligero y eficiente
- Usa reflexión y útil para el desarrollo de Middleware
- Middleware resultante es muy flexible
  - Gracias a los diferentes tipos de receptáculos
  - Gracias a la introspección
  - Compatibilidad hacia atrás con COM
  - Implementación realizada (OpenORB v2)

An efficient component model for the construction of adaptive middleware

Michael Clarke **Gordon S. Blair** **Geoff Coulson** **Nikos Parlavantzas**

Distributed Multimedia Research Group

Lancaster University

Presentado por Carlos E. Agüero

¿Preguntas?