

Middleware Models and Architectures

Claudia Salado Méndez

1 Resumen

Este documento nos introduce en el mundo de la tecnología moderna al presentar las características del estándar de la plataforma CORBA, que se ha convertido en el punto de referencia esencial y el soporte para varias aplicaciones abiertas y es punto de referencia para la intercomunicación entre componentes de software heterogéneo. El componente "CORBA" es el proyecto de middleware más grande y ambicioso de la industria hasta la fecha.

1.1 Introducción y Objetivos

CORBA fue el primero de los productos lanzados por OMG (Object Management Group, que es un consorcio que se formó en 1989 que se dedica a cuidar y establecer diversos estándares de tecnologías orientadas a objetos. A parte de CORBA también son los creadores de UML, XMI y BPMN. Es una organización sin ánimo de lucro que promueve el uso de estas tecnologías mediante guías y especificaciones.

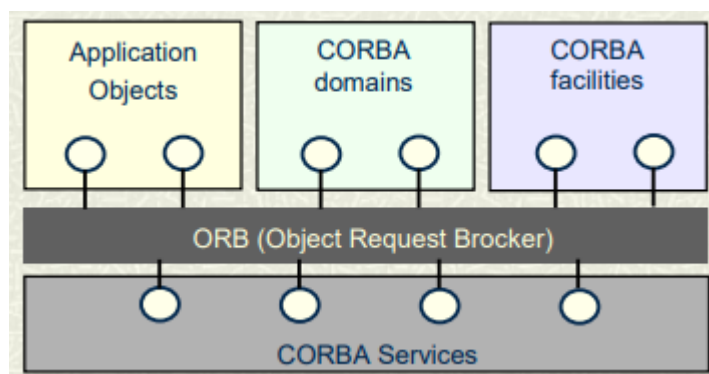
El objetivo de este producto es ayudar a disminuir la complejidad, los costes y hacer que la introducción de nuevas aplicaciones informáticas sea más veloz, para ello promueven la teoría y la práctica de la tecnología de objetos en los sistemas distribuidos.

Es una tecnología cuyo propósito es ocultar la programación a bajo nivel de aplicaciones distribuidas. Por lo tanto también permite al programador utilizar una tecnología orientada a objetos, hay que recalcar que las funciones objetos y estos pueden estar en diferentes computadoras, pero el programador puede acceder a ellos a través de funciones propias dentro de su programa.

CORBA es más que una especificación multiplataforma, también define servicios habitualmente necesarios como seguridad y transacciones. Y así este no es un sistema operativo en sí, en realidad es un middleware.

1.2 Diseño

Esta imagen representa la arquitectura de CORBA, vamos a explicar por encima sus partes, el ORB se explicara en la parte de mecanismos:



- **CORBA services:** Provee los servicios básicos a nivel de sistema, tales como:
 - Servicios de nombres.
 - Servicio de persistencia

- Servicios de notificación de eventos.
- Concurrencia
- Ciclo de vida
- Externalización
- Transacciones.
- Seguridad.
- **CORBA Facilities:** Proporcionan un conjunto de funciones de alto nivel que facilitan que cubren aspectos generales como interfaces de usuario, gestión de información, etc.
 - Administración de las interfaces de usuario.
 - Administración de la información
 - Administración de sistemas
 - Administración de tareas
- **CORBA Domain:** Son especificaciones y definiciones semánticas que son comunes a ciertos dominios de aplicación en los que CORBA está implantada.
 - Finanzas
 - Internet
 - Telecomunicaciones
 - Empresas
 - Cuidado de la salud

1.3 Características

- **Independencia:** fue diseñada para liberar a los programadores de las limitaciones en cuanto al diseño del software. Por eso en lenguaje de programación y sistema operativo son independientes. Puede admitir los siguientes lenguajes: Ada, C, C++, C++11, Lisp, Ruby, Smalltalk, Java, COBOL, PL/I y Python.
- **Posibilidad de interacción entre diferentes tecnologías:** uno de los principales beneficios de la utilización de CORBA es que posibilita normalizar las interfaces entre las diversas tecnologías que se utilizan y así poder combinarlas sin que nos den errores de incompatibilidad.
- **Transparencia de distribución:** el sistema se encarga de saber y controlar si la aplicación está distribuida o centralizada, por ello tanto el cliente como el servicio no necesitan saber si esta distribuida o centralizada.
- **Transparencia de localización:** el cliente no tiene que tener en cuenta donde se está ejecutando el servicio y este a su vez no tiene la necesidad de saber dónde se está ejecutando el cliente.
- **Integración de software existente:** se amortiza la inversión previa reutilizando el software con el que se trabaja, se pueden utilizar sistemas heredados (legacy systems).
- **Activación de objetos:** los objetos remotos no tienen porqué estar en memoria permanentemente, esto se hace de manera invisible para el cliente.
- **Comunicación flexible:** CORBA tiene múltiples modos de comunicación
 - Síncrono: tipo invocación de métodos/RPC
 - Asíncrono: sin respuesta.

También tiene varios modelos de comunicación:

- Invocación estática
- Invocación dinámica

También tiene muchas otras características tales como: el tipado fuerte de datos, la alta capacidad de configuración, libertad de elección de los detalles de transferencia de datos, o la compresión de los datos.[1][4]

1.4 Mecanismos

En CORBA los objetos poseen muchas características de otros sistemas orientados a objetos, como utiliza la herencia en cuanto a las interfaces y el polimorfismo. Esto hace que haya la posibilidad de proporcionar estas características a lenguajes no orientados a objeto como C o COBOL, aunque CORBA trabaja mejor con lenguajes orientados a objeto como son Java o C++.

Vamos a hablar un poco de los mecanismos más importantes que utiliza CORBA

- **Object Request Broker (ORB):** Este es un componente esencial de la estructura de CORBA, su objetivo es facilitar la comunicación entre objetos. Envía las peticiones a los objetos y devuelve las respuestas a los clientes que las invocan por el proceso de serialización.

ORB es el que le da a CORBA la característica de transparencia, ya que facilita la comunicación entre el cliente y el servidor ocultando la localización de los objetos, el cliente no tiene que saber dónde se encuentra el objeto destino, este puede estar en su máquina o en una remota. También oculta la implementación de los objetos, el cliente ignora el lenguaje en el que se ha programado el objeto remoto, también su implementación y el sistema operativo en el que está. Pasa lo mismo con el estado de ejecución del objeto, cuando se envía una petición sobre un objeto remoto el ORB inicializa el objeto si es necesario antes de enviar la petición. Por último los mecanismos de comunicación de objetos, el cliente no conoce qué mecanismo está utilizando ORB para enviar dichas peticiones y devolver el resultado.[1][3]

- **El Adaptador de Objetos (OA):** es el módulo que permite a las implementaciones de los objetos acceder a los servicios que nos da el ORB, este genera las referencias a los objetos. El OA exporta una interfaz pública por la implementación del objetos y una privada para que la use el esqueleto del objeto que depende de la implementación del OA.[3]

Las funciones que lleva a cabo el adaptador de objetos son las siguientes:

- Genera e interpreta las referencias a objetos
 - Invoca métodos
 - Proporciona seguridad en las interacciones
 - Activa y desactiva objetos e implementaciones
 - Traduce las referencias objetos con sus correspondientes implementaciones.
 - Registra implementaciones, debido a que las implementaciones de objetos dependen del OA se deben definir la menor cantidad posible.
- **IDL (Interface Definition Language):** Para especificar los servicios proporcionados por los objetos que forman parte de un sistema distribuido abierto, necesita un lenguaje bien definido y preciso, independientemente de la representación o estructura de datos definida, así como la futura implementación de los objetos que especifica.[1][3]

El estándar ISO / IEC 14750 (ITUT X.920) define un idioma llamado ODP IDL después del idioma de definición de interfaz ODP u ODP IDL su abreviatura en inglés.

El objetivo principal es describir la signature del objeto especificado, la estructura de los datos a procesar y el registro del trabajo que define el servicio. De esta forma se consigue la ofuscación necesaria para el desarrollo de aplicaciones abiertas.

Las interfaces en IDL describen un conjunto de posibles operaciones que un cliente puede solicitar en un objeto. Los objetos que pueden satisfacer los requisitos de otros objetos pueblan la interfaz. Las interfaces proporcionan un mecanismo complejo para permitir que estos objetos admitan múltiples interfaces.

Las acciones realizadas representan servicios que se pueden combinar y ejecutar para cambiar valores y obtener valores. Una transacción se identifica mediante un identificador de transacción. Las actividades no valen la pena.

Los tipos de datos que manipula CORBA en IDL son:

- Tipos básicos : long, short, ushort, ulong, float, double, char, boolean, enum, string, octet, any
- Tipos compuestos: struct, union, array
- Tipos derivados: sequence <tipo>
- Tipos de objeto: interface, referencia a objetos

1.5 Propiedades

- **Heterogeneidad:** es un sistema con un conjunto de elementos interconectados de hardware y software de diferentes fabricantes y que puede integrar aplicaciones con diferentes tecnologías.

Lo que hace es integrar antiguas tecnologías con modernas para así obtener un completo beneficio ya que la infraestructura de los sistemas de información antiguos son irremplazables debido al coste de su desarrollo y al tiempo para implantarlo.

- **Movilidad:** la migración de procesos en los sistemas distribuidos tradicionales funciona muy bien para mejorar el reparto de la carga en las distintas máquinas. Su objetivo es garantizar el rendimiento global y ciertas restricciones de administración o seguridad.
- **Eficiencia:** es eficiente ya que la red lleva menos mensajes, el servidor es el que realiza la mayor parte del trabajo, y se evita la latencia e inestabilidad de la red en los procesos.
- **Adaptación al cliente:** el cliente tiene la posibilidad de extender la funcionalidad del servidor, la instalación resulta fácil para el usuario, no es necesaria la instalación del servidor, no deben acordarse los procedimientos entre los clientes y los servidores, la instalación de los procedimientos del cliente en el servidor es dinámica
- **Tiempo de desempeño:** la ejecución asíncrona permite que los procesos puedan controlar la gestión y terminación de la tarea, y que el cliente pueda terminar o continuar haciendo otras cosas mientras en su sistema, por otra parte se reduce el tráfico en la red y la capacidad de cómputo del cliente.
- **Robusto:** se reduce la dependencia de la disponibilidad de la red y del cliente-servidor, los procesos migrados al servidor no se ven afectados por fallos del cliente o de la red, estos procesos se ejecutan realizando tareas específicas en lugares diferentes, se automatizan las tareas distribuidas.[4][3]

1.6 Protocolos y Servicios para la Comunicación y Coordinación

CORBA define tres protocolos por capas para enviar mensajes entre clientes y objetos:

- **Common Data Representation (CDR):** se utiliza para representar tipos de datos estructurados o primitivos que se pasan como argumentos o resultados durante invocaciones remotas en objetos distribuidos de Common Object Request Broker Architecture (CORBA).

Permite que los clientes y servidores escritos en diferentes lenguajes de programación trabajen juntos. Por ejemplo, traduce little-endian a big-endian. Se asume un acuerdo previo sobre el tipo, por lo que no se proporciona información con representación de datos en mensajes.[5]

- **General Inter-ORB Protocol (GIOP):** es un protocolo abstracto por el cual los ORBs se comunican. Además la arquitectura GIOP proporciona a su vez otra serie de protocolos que son:[1][4]
 - **Internet InterORB Protocol (IIOP):** implementación de GIOP para su uso por internet, proporcionando una interfaz entre los mensajes GIOP y la capa TCP/IP. (Ahora lo veremos con más detalle)
 - **SSL InterORB Protocol (SSLIOP):** protocolo IIOP sobre SSL, proporcionando encriptación y autenticación.
 - **HyperText InterORB Protocol (HTIOP):** protocolo IIOP sobre HTTP, proporcionando transparencia remota.
 - **Zipped IOP (ZIOP):** una versión comprimida de GIOP que reduce el uso de ancho de banda.
- **IIOP (Internet Inter-ORB Protocol):** es un protocolo que hace posible que los programas distribuidos escritos en distintos lenguajes de programación se comuniquen a través de Internet.

IIOP es una parte fundamental del CORBA, utilizándolo junto con otros protocolos relacionados, una empresa podría escribir programas que podrán comunicarse con los programas existentes o futuros de su propia empresa o de otra empresa donde quier que se encuentren y sin tener que entender nada sobre el programas, mas que su servicio y un nombre.

CORBA e IIOP compiten con una estrategia similar de Microsoft llamada Modelo de objetos componentes distribuidos (DCOM). Aunque ambos han acordado desarrollar puentes software entre los dos modelos para que los programas diseñados para CORBA puedan comunicarse con los programas diseñados en DCOM.[6]

1.7 Análisis Crítico

Ya que hemos visto todas las propiedades y características de CORBA y cual es su objetivo, y cómo esto ayuda a los programadores, tenemos que hablar también de los problemas que surgen o que se generan. El problema fundamental de los sistemas de integración como CORBA es el software, todavía no existe mucha experiencia en el diseño, implantación y uso de software.

Actualmente, este es un campo de investigación. Las redes son indispensables para la comunicación entre las máquinas pero estas pueden, pueden generarnos problemas de saturación, embotellamiento, interrupción o pérdidas de mensajes.

Además el constante acceso al sistema por parte de los usuarios también plantea varios problemas de seguridad, esto hace que sea necesario un sistema de seguridad que se adapte a las necesidades de los clientes CORBA ya maneja la seguridad pero es necesario que lo haga más.

1.8 Conclusiones

CORBA como hemos podido observar es un middleware que nos proporciona una infraestructura y un modelo comunes que pueden integrar requisitos expresados en diferentes lenguajes (diferentes metodologías de desarrollo) para formar un sistema globalmente cohesivo.

CORBA proporciona un conjunto de mecanismos, los que hemos explicado anteriormente, muy conveniente para desarrollar aplicaciones distribuidas, lo que le permite crear aplicaciones potentes, eficientes y competitivas e integrarlas con otros sistemas que cumplan con estos criterios. Los sistemas desarrollados con tecnologías más antiguas se pueden integrar con sistemas más nuevos utilizando CORBA. Es decir, resuelve el problema gradualmente construyendo una interfaz que permite intercambiar información local o remota a través de la red.

Varias tecnologías tienen interfaces integradas para intercambiar información vía CORBA y desarrolladas para facilitar la integración de servidor y cliente de la filosofía CORBA. Java como motor incorpora interoperabilidad con CORBA siempre que el objeto se ajuste a la especificación y utilice un ORB que admita IIOP como protocolo de comunicación.

Finalmente, el protocolo de comunicación IIOP establecido por la especificación CORBA para la interoperabilidad entre diferentes plataformas se ha convertido en el protocolo preferido utilizado por defecto para asegurar la interoperabilidad entre todas las plataformas del sistema.

1.9 Referencias

1. <https://es.wikipedia.org/wiki/CORBA>
2. https://www.ctr.unican.es/asignaturas/procodis_3_II/Doc/Procodis_8_01.pdf
3. <http://nomadas.ucentral.edu.co/index.php/inicio/32-investigacion-y-transformaciones-sociales-nomadas-17/479-tecnologia-corba-common-object-request-broker-architecture>
4. http://www.it.uc3m.es/mcftp/docencia/si/material/6_orb_mcftp.pdf
5. https://en.wikipedia.org/wiki/Common_Data_Representation
6. <https://whatis.techtarget.com/definition/IIOP-Internet-Inter-ORB-Protocol>

Autoevaluación

1. ¿La explicación es clara y el contenido está estructurado?

Si

2. ¿Queda reflejado que se han estudiado y entendido las cuestiones de diseño, mecanismos, protocolos y servicios para la comunicación y coordinación relacionados en la bibliografía estudiada?

Si

3. ¿Se incluyen ejemplos que ayudan a explicar la descripción de la tecnología middleware seleccionada?

No

4. ¿Queda reflejado en el trabajo que se ha buscado y analizado suficiente bibliografía adicional y se han incluido las referencias encontradas?

Si

5. ¿Se incluye un análisis crítico positivo y/o negativo de la tecnología middleware escogida y unas conclusiones como resultado del estudio y explicación realizadas?

Si

6. ¿El informe realizado me ha llevado tiempo y servido para comprender la solución al problema abordado?

Si

7. ¿Se ha realizado alguna aportación adicional más allá de lo que en esencia pedía este trabajo?

No

En base a los anteriores aspectos, el estudiante debe incluir al final de esta autoevaluación un valor numérico de calificación (de 0 a 10 puntos con un decimal) para el trabajo entregado.

8.5