Corba
An Overview

Joseph Kehoe[1]

[1]Department of Computing and Networking
Institute of Technology Carlow

CDD101, 2018

Introduction

- Common Object Request Broker Architecture
- Open Standard Managed by the Object Management Group (OMG)
- Ratified by ISO
- Set up to create an open standard for distributed object interaction
- CORBA enables collaboration between systems on different operating systems, programming languages, and computing hardware
- CORBA uses an object-oriented model although the systems that use the CORBA do not have to be object-oriented

Corba

Corba allows communication between:

- Different languages
- Different operating systems
- Distributed devices

in a transparent manner!
So everything appears local and as if implemented in my favourite programming language (ish)

Corba Issues

Location transparency Objects residing in the same address space and accessible with a simple function call are treated the same as objects residing elsewhere. This makes all object access as complex as the most complex case

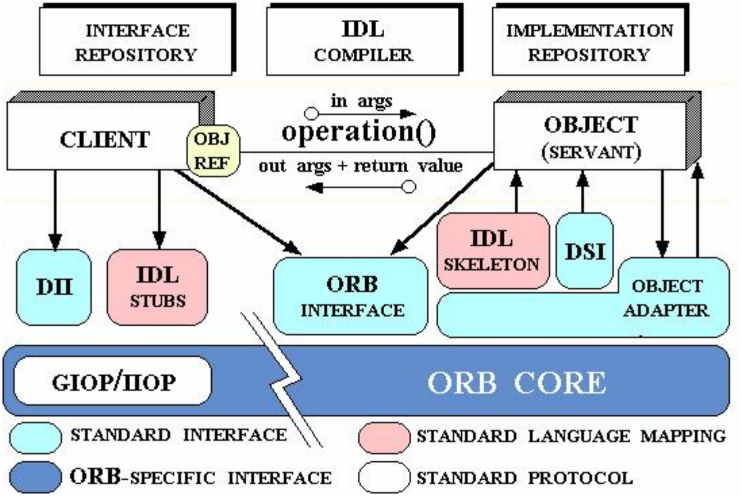
Design and process deficiencies The creation of the CORBA standard is also often cited for its process of design by committee. There was no process to arbitrate between conflicting proposals or to decide on the hierarchy of problems to tackle. This made the specification complex, expensive to implement entirely, and often ambiguous.

Corba Issues

Problems with implementations Through its history, CORBA has been plagued by shortcomings in poor ORB implementations.

Firewalls CORBA (more precisely, GIOP) is not tied to any particular communications transport. If the client is behind a very restrictive firewall or transparent proxy server environment that only allows HTTP connections to the outside through port 80, communication may be impossible. Recent CORBA implementations, though, support SSL and can be easily configured to work on a single port.

Basic Architecture



Components

- **DII** Dynamic Invocation Interface allows programs to interogate object interfaces at Run-Time and generate calls
- **DSI** Dynamic Server Interface. Server side version of DII where all Dynamic Invocations are received and dealth with
- **Intereface Repository** Database of all known defined interfaces. Used by DII
- **Implementation Repository** Database of all server implementations of interfaces

Components

ORB	Object Request Broker through which all communication is channeled. Different for each implementation BUT...
ORB Interface	Standard interface that must be implemented by all ORBs
GIOP	General Inter Object Protocol which defines how objects interact with ORB
IIOP	Internet Inter Object Protocol (nuff said)

Corba Services Part I

Security
: includes security models and interfaces for application development, security administration, and the implementation of security services themselves

Naming
: defines how CORBA objects can have friendly symbolic names

Events
: decouples the communication between distributed objects (e.g. asynchronous communication)

Relationships
: provides arbitrary typed n-ary relationships between CORBA objects

Externalization
: coordinates the transformation of CORBA objects to and from external media

Corba Services Part II

Transactions	Allows use of transactions (e.g. as seen in DBMS)
Concurrency Control	provides a locking service for CORBA objects in order to ensure serializable access
Property	supports the association of name-value pairs with CORBA objects
Trader	supports the finding of CORBA objects based on properties describing the service offered by the object
Query	supports queries on objects

IDL Compiler

- Interface Definition Langauge defines object interfaces in language agnostic way
- There exist mappings that translate IDL into Java, C, C++14, Python, ADA, and so on
- Compilers exist that will translate IDL file into client and server Corba enabled code in any language
- generates IDL stubs for client wishing to use object
- produces IDL Skeleton for server where object is implemented