The CORBA Component Model

Nanbor Wang
Center for Distributed Object Computing
Department of Computer Science
Washington University, St. Louis
nanbor@cs.wustl.edu

February 25, 2000

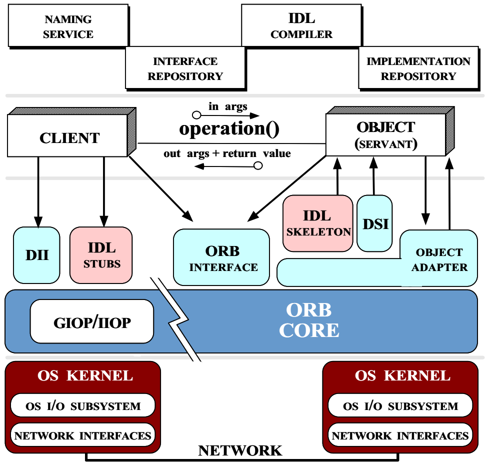
Research Supported by NASA

Contents

What Are Components?

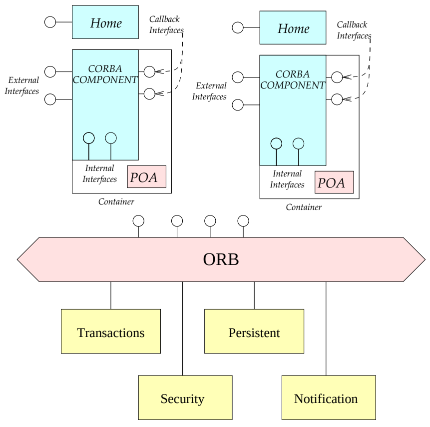
- A software component is a unit of composition with contractually specified interfaces and explicit context dependencies only. A software component can be deployed independently and is subject to composition by third parties
- Components are units of independent deployment
- Components encourage composition

Overview of CORBA



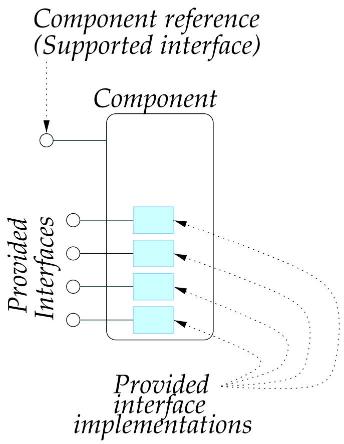
- Provides the communication infrastructure
- Contracts between clients and servers
- Shortfalls
  - Standardized servant interaction model
  - Standardized service intergration model
  - Standardized deployment mechanism

The CORBA Component Model



- **Features**
  - Navigation among interfaces supported by components
  - Standardized system-component interaction
  - Standardized component life-cycle menagement
  - Component interconnections
  - Standardized component configuration
  - Standardized ORB services interfaces

Components and Component Homes



- Component Homes
  - Component lifecycle management interface
  - Keyed and keyless operations
- Components
  - Support multiple provided (user) interfaces
  - Navigation interfaces
  - Receptacle interfaces for object interconnections
  - Event source/sink interfaces
  - Attribute interfaces for dynamic configuration

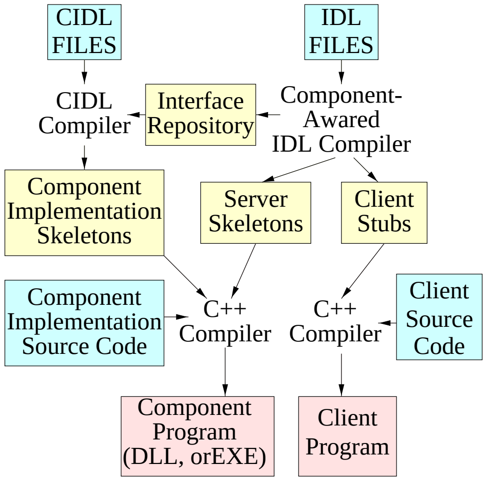
Containers

- Define the component-server interfaces
- Bridge the managed component with common ORB services
- Customize the run-time environment of a component
- Manage exactly one component
- Dedicated POA for all interfaces in the component

Containers Programming Models

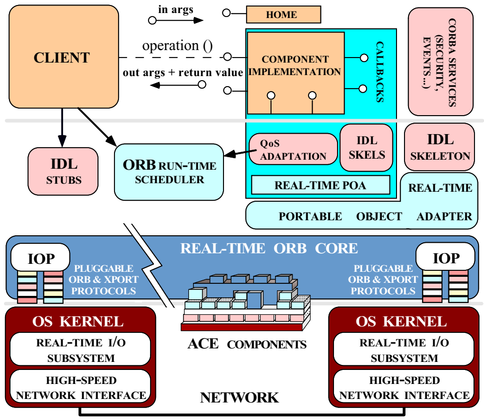
CORBA Usage Model	Object Reference	Container API Type	Primary Key	Component Categories
stateless	TRANSIENT	session	No	Service
conversational	TRANSIENT	session	No	Session
durable	PERSISTENT	entity	No	Process
durable	PERSISTENT	entity	Yes	Entity

The Component Implementation Framework



- Component Implementation Definition Language (CIDL) generates common code (component implementation skeletons)
- Users provide component implementation

Proposed CCM optimizations in TAO



- Collocation optimizations
- Dynamic QoS and real-time support using containers
- Dynamic activation/eviction of component elements