OpenCCM

MdC Philippe Merle
LIFL - INRIA
Philippe.Merle@lifl.fr

http://corbaweb.lifl.fr/OpenCCM/
http://www.objectweb.org/OpenCCM/ (soon)

OpenCCM

- **The Open CORBA Component Model Platform**
- **1st public available implementation of the OMG CCM**
  - Few partial Java & C++ Open Source implementations already available
  - No commercial products planned
- **An Open Source project**
  - Started since 2000
  - 1st releases since Q1 2001
  - Done by LIFL
- **Available at http://corbaweb.lifl.fr/OpenCCM/**

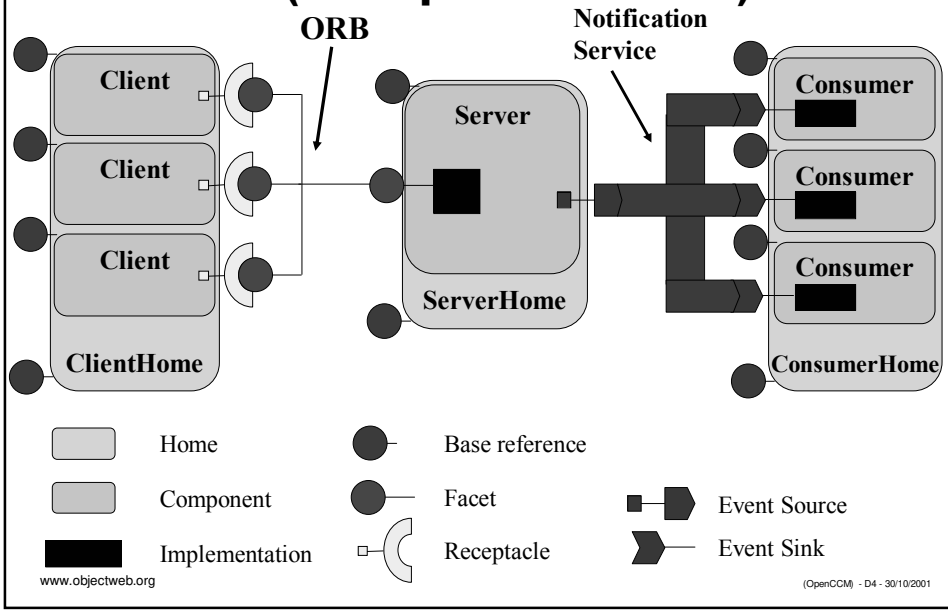
- **Dedicated to research & experimental activities around next Distributed Component Computing and Middleware technologies**

The CORBA Component Model

- **1st open standard for Distributed Component Computing**
  - Multi-languages, multi-OSs, multi-ORBs, multi-vendors, etc.
- **Like Entreprise Java Beans (EJB)**
  - Components created and managed by homes
  - Running into containers managing system services transparently
  - Hosted by application component servers
- **But one component could**
  - Have several input and output interfaces (operations & events)
    - via the extended OMG Interface Definition Language (OMG IDL3)
  - Be implemented by several classes
    - via the Component Implementation Definition Language (CIDL)
- **One CCM application could be deployed and run on several distributed servers simultaneously!**

www.objectweb.org

A Distributed CCM Application (The OpenCCM demo3)



www.objectweb.org

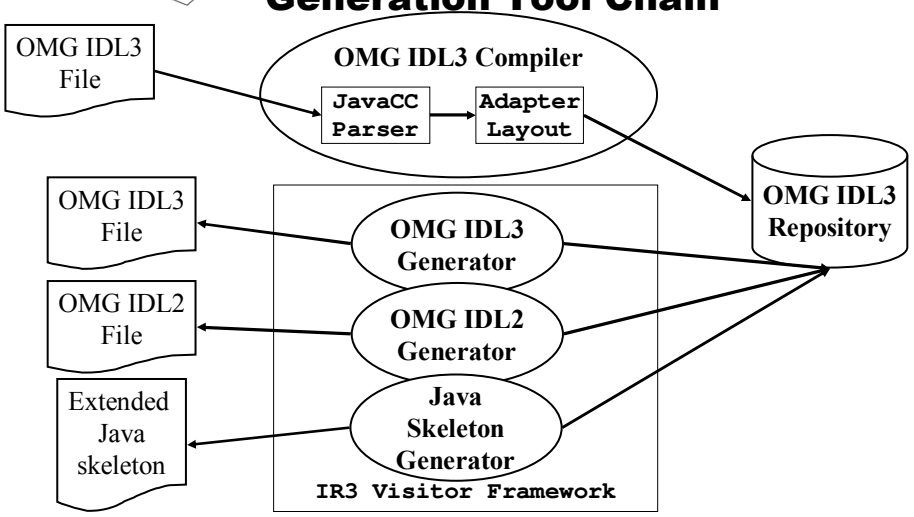
An OMG Component IDL Example

```
interface Service { ... business operations ... };
eventtype Event { ... business data ... };
component Client {
  uses Service to_server;
};
component Server {
  provides Service the_service;
  publishes Event to_consumers;
};
component Consumer {
  consumes Event from_servers;
};
valuetype Key { . . . };
home ServerHome manages Server primarykey Key {
  factory create_server( . . . );
  . . . other business home operations . . .
};
```


Current OpenCCM Features

- **Fully written in Java**
  - portability, maintenance and support
  - Linux, Solaris, and Windows supported
  - ORBacus 4.x, OpenORB 1.x, and VisiBroker 4.5 supported
- **An open compilation & generation tool chain**
  - An OMG IDL3 Compiler
  - An OMG IDL3 Repository
  - A generator for equivalent OMG IDL2
  - A generator for extended Java skeleton classes
    - extended skeletons ~ JOnAS interposition classes
- **A flexible distributed deployment & execution middleware infrastructure**
  - A generic Java server to host Java components
  - API to start component servers and download component archives
  - Deployment driven by OMG IDLscript

The OpenCCM Compilation & Generation Tool Chain



➔ **A component framework for OMG IDL3 compilation and code generation**

Distributed Deployment with OpenCCM

➔ **Could be done by any CCM program at any time**

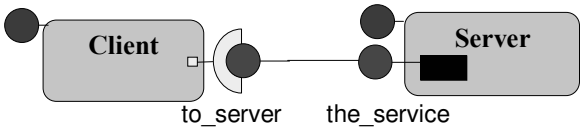
```
➢serverActivator = NS.resolve_str(«AnHost»)
➢componentServer = serverActivator.
       create_component_server(«language=Java and
                          ORB=ORBacus and ...»)
➢container = componentServer.create_container(...)
➢home = container.install_home
     («http://www.lifl.fr/~merle/server.jar», ...)
➢server = home.create_server(...)
```

➔ **Dynamic control of distributed deployment!**

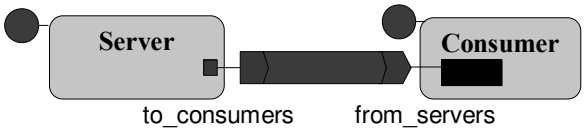
➔ **Deployment does not imply code generation!**

Distributed Deployment with OpenCCM

➔ **Could be done by any CCM program at any time**



- `service = server.get_the_service()`
- `client.connect_to_server(service)`



- `sink = consumer.get_from_servers()`
- `server.subscribe_to_consumers(sink)`

➔ **Dynamic control of component interconnections!**

➔ **Build distributed application architectures!**

Some Main Current Uses

➔ **French RNRT CESURE project**
- Distributed component based application deployment
- Driven by Gemplus smart cards (i.e. Java Card + RMI)
- Component replication, load balancing, and trading
- Also running inside Compact iPAQ & SmartLife Gemplus PDA
  - IBM J9 + ORBacus 4.1 + OpenCCM server ~ 5 Mo
- See http://www.gemplus.fr/cesure

➔ **Thales ATM**
- OMG IDL3 -> XMI -> Objecteering UML tool

➔ **Parallel components for GRID metacomputing**
- PARIS team at IRISA Rennes

➔ **Around 500 downloads**

Future Work: Related to CCM

- ➔ **Component Implementation Definition Language (CIDL)**
- ➔ **Component Implementation Framework (CIF)**
- ➔ **Containers for system services**
  - ➢ security, persistence, transactions, and notification services
- ➔ **Persistent State Definition Language (PSDL)**

- ➔ **ZIP archives and XML DTDs for packaging and deployment**
- ➔ **Lastest CCM deployment APIs**

- ➔ **JacORB 1.4 and Orbix 2000 support**
- ➔ **Generator and runtime for C++ components**

Future Work: Related to ObjectWeb

- ➔ **Integration with other ObjectWeb projects**
  - ➢ Jonathan for its CORBA personality
  - ➢ JOnAS for its distributed transaction service
  - ➢ JORM for its persistence service
  - ➢ JORAM for its reliable asynchronous middleware
  - ➢ . . .
- ➔ **Reusing most of basic common component frameworks like MonoLog, etc.**
- ➔ **Providing input for next common component frameworks**
  - ➢ open containers and application servers
  - ➢ distributed deployment infrastructure
  - ➢ component repositories
  - ➢ . . .
- ➔ **Available soon at http://www.objectweb.org/OpenCCM/**

Future Work: Advanced features

- **Open, adaptive, and extensible container framework**
  - Applied to fault-tolerance, load balancing, real-time services
  - In cooperation with Thales
- **Composite design and execution (CODeX)**
- **Trading-based deployment and supervision (TOSCA)**
- **Component scripting engine inside**
- **Various Graphical User Interfaces (GUI)**
  - for designing, assembling, deployment, and management

Other LIFL's Projects

- **CorbaScript**
  - The OMG CORBA Scripting Language Specification
  - Running on top of major C++ ORB products
  - http://corbaweb.lifl.fr/CorbaScript/
- **Java IDLscript**
  - An OMG IDLscript + Java scripting engine
  - http://www.lifl.fr/~roos/jidlscript/
- **TORBA**
  - Trader Oriented Request Broker Architecture
  - Generation and runtime for safe trading components
  - http://www.lifl.fr/~leblanc/TORBA/
- **Perhaps, some future ObjectWeb projects?**