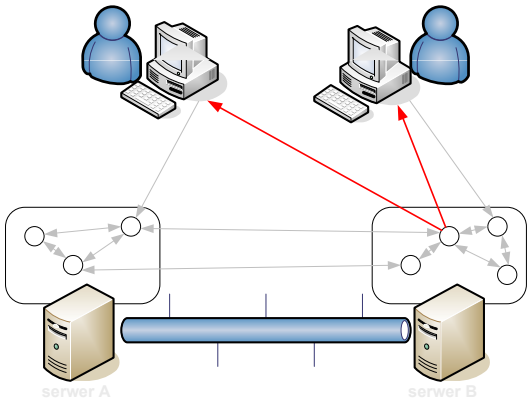
CORBA

Plan wykładu

- Wprowadzenie
- Architektura CORBA
- IDL – język definicji interfejsów
- ORB – Object Request Broker
- Usługi i POA
- Aplikacje CORBA
  - tworzenie serwera
  - tworzenie klienta

Aplikacje rozproszone

- Rozproszenie danych
- Rozproszenie obliczeń
- Rozproszenie użytkowników

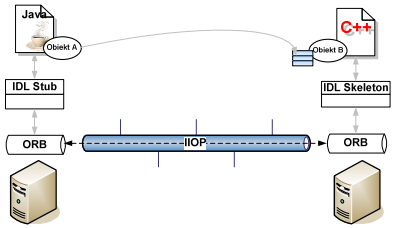


Cechy aplikacji rozproszonych

	System lokalny	System rozproszony
Komunikacja	szybka	wolna
Awarie	obiekty ulegają awarii wszystkie na raz	obiekty ulegają awarii niezależnie, możliwy podział sieci
Współbieżny dostęp	wielowątkowo	tak
Bezpieczeństwo	wysokie	niskie

Co to jest CORBA

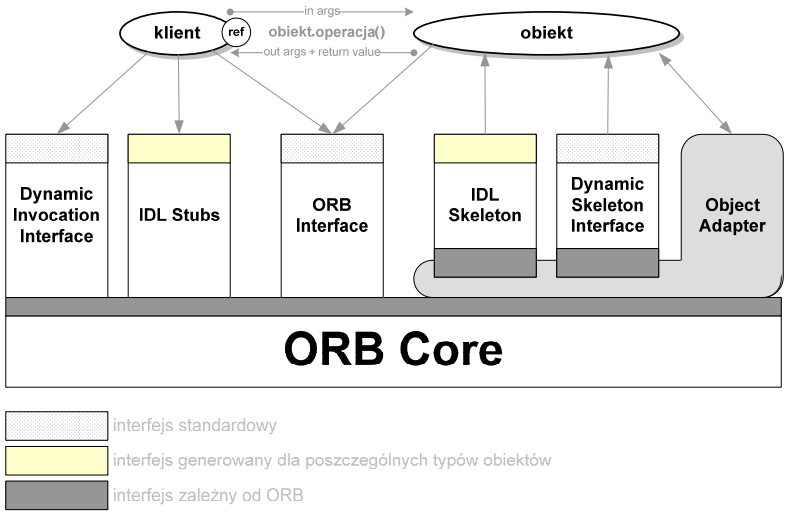
- **C**ommon **O**bject **R**equest **B**roker **A**rchitecture
  - architektura obiektowych systemów rozproszonych umożliwiająca współpracę między heterogenicznymi, rozproszonymi kolekcjami obiektów
- Paradygmat: żądanie usługi od zdalnego obiektu



OMG i historia standardu

- Object Management Group
  - Konsorcjum 800 firm informatycznych
  - Standardy: OMA, CORBA, UML, XMI, MDA
- Historia standardu CORBA
  - CORBA 1.0, grudzień 1990
  - CORBA 1.1, 1.2 (wprowadzenie IDL), grudzień 1993
  - CORBA 2.0 (wprowadzenie IIOP), grudzień 1994
  - CORBA 2.1, 2.2, …, 2.6, grudzień 2001
  - CORBA 3.0, lipiec 2003

Architektura CORBA



Cechy architektury CORBA

- Model pojęciowy OMA
- Język IDL (abstrakcyjny język obiektowy)
- Wiązania do wielu języków programowania
- Automatyczne generowanie kodu pieńków i szkieletów
- Transparentność lokalizacji obiektów i usług
- Wspólne usługi i udogodnienia CORBA
- Niezależność od sprzedawców

CORBA a RPC, Web Services

- Podobieństwa
  - Zdalne wywoływanie metod
  - Niezależność od języka implementacji
- Różnice
  - Abstrakcyjny język obiektowy (IDL)
  - Pełna definicja protokołów i zachowania
  - Wiązania do wielu języków programowania
  - Udogodnienia do zarządzania obiektami
  - Standardowe usługi
  - Duży narzut na zasoby (szybkość, pamięć)

IDL – wprowadzenie

- Cechy IDL
  - Deklaratywny język definicji interfejsów
  - Niezależność od języka implementacji
  - Składnia podobna do C++
  - Brak odniesień do implementacji operacji
- Pojęcia
  - **Moduł**: zbiór interfejsów (pakiet, przestrzeń nazw)
  - **Interfejs**: specyfikacja cech i operacji obiektu (klasa)
  - **Atrybut**: cecha obiektu (składowa)
  - **Operacja**: opis zachowania się obiektu
  - **Wyjątek**: opis nietypowej sytuacji

IDL – przykład

messageBox.idl

```
module MessageModule {
    typedef sequence<string> MessageSeq;
    interface MessageBox {
        exception boxFull {};
        attribute string reply;
            string leaveMessage(in string msg) raises (boxFull);
        MessageSeq getMessages();
    };
};
```

IDL – odwzorowania językowe

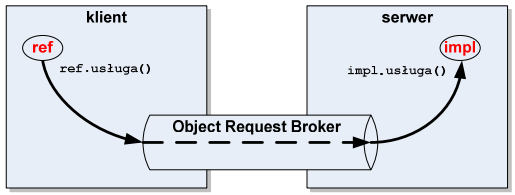
OMG IDL	Java	C++
short, long, double	short, int, long	short, long, double
any	org.omg.CORBA.Any	any class
string, wstring	java.lang.String	char*, wchar_t*
sequence	array	class
typedef	helper class	typedef
interface	class	class
exception	class	class
module	package	namespace

Object Request Broker – wprowadzenie

- ORB – centralna szyna komunikacji
- Najważniejsza cecha ORB – przezroczystość
  - Lokalizacja obiektu
  - Implementacja obiektu
  - Stan działania obiektu
  - Komunikacja między obiektami
- Zarządzanie referencjami do obiektów
  - Nieczytelne, niemodyfikowalne, nieusuwalne
  - Pozyskiwanie referencji
    - tworzenie obiektu
    - usługi katalogowe
    - serializacja referencji do pliku lub bazy danych

ORB - usługodawcy

- Usługodawca: konkretna implementacja operacji obiektu CORBA wyspecyfikowanych w deskryptorze IDL
  - Wiele referencji odwzorowywanych na usługodawcę
  - Jedna referencja odwzorowywana na wielu usługodawców
  - Obiekt może nie mieć aktywnych usługodawców
  - Obiekt może wcale nie mieć usługodawców



ORB – lista dostępnych implementacji

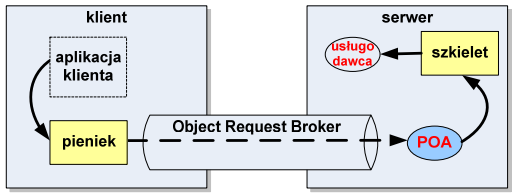
- IIOP.NET, http://iiop-net.sourceforge.net/
- JacORB, http://www.jacorb.org/
- Java IDL, http://java.sun.com/products/jdk/idl/
- MICO is CORBA, http://www.mico.org/
- omniORB, http://omniorb.sourceforge.net/
- OpenORB, http://openorb.sourceforge.net/
- TheAceORB, http://www.cs.wustl.edu/~schmidt/TAO.html
- VisiBroker, http://www.borland.com/us/products/visibroker

Usługi CORBA

- Life Cycle Service
- Persistence Service
- Naming Service
- Event Service
- Concurrency Control Service
- Transaction Service
- Relationship Service
- Query Service
- Properties Service
- Time Service

POA – Portable Object Adapter

- Odpowiedzialność adaptera obiektów
  - Generowanie i interpretacja referencji do obiektów
  - Wywoływanie metod
  - Bezpieczeństwo interakcji między obiektami
  - Aktywacja i dezaktywacja obiektów
  - Rejestrowanie implementacji interfejsów



Definicja i prekompilacja interfejsu

messageBox.idl

```
module MessageModule {
    typedef sequence<string> MessageSeq;
    interface MessageBox {
        exception boxFull {};
        attribute string reply;
            string leaveMessage(in string msg) raises (boxFull);
        MessageSeq getMessages();
    };
};
```

```
C:\> idlj.exe -fserver messageBox.idl
```
1

```
C:\> orbd.exe -ORBInitialPort 1050
```
2

Automatycznie generowane pliki

[dir] MessageModule
- MessageBox.java
- MessageBoxOperations.java
- MessageBoxPOA.java
- MessageSeqHelper.java
- MessageSeqHolder.java
- [dir] MessageBoxPackage
  - boxFull.java
  - boxFullHelper.java
  - boxFullHolder.java

Przygotowanie usługi 1/2

MessageBoxImpl.java

```
public class MessageBoxImpl extends MessageBoxPOA {
  public MessageBoxImpl(String name, int max) { … }

  protected String reply;
  protected int max;
  protected Vector messages;
  private ORB orb;

  public String leaveMessage(String msg)
    throws MessageModule.MessageBoxPackage.boxFull {
    if (messages.size() > max)
      throw new MessageModule.MessageBoxPackage.boxFull();
    messages.addElement(msg);
    return reply(); }
  …
```

składowe klasy usługi

lokalny broker

implementacja usługi

Przygotowanie usługi 2/2

MessageBoxImpl.java

```
  …
  public void setORB(ORB orb) { this.orb = orb; }

  public void reply(String reply) { this.reply = reply; }

  public String reply() { return this.reply; }

  public String[] getMessages() {
    String[] _messages = new String[messages.size()];
    messages.copyInto(_messages);
    messages = new Vector(max);
    return _messages;
  }
}
```

Przygotowanie serwera 1/2

MessageServer.java

```
…
public static void main(String[] args) {
  try {
    Properties props = new java.util.Properties();
    props.put("org.omg.CORBA.ORBInitialPort", "1050");
    props.put("org.omg.CORBA.ORBInitialHost", "localhost");
```

połączenie z brokerem

```
    ORB orb = ORB.init(args,props);
    POA rootpoa =
      POAHelper.narrow(orb.resolve_initial_references("RootPOA"));
    rootpoa.the_POAManager().activate();
    …
```

Przygotowanie serwera 2/2

MessageServer.java

```
…
  MessageBoxImpl messageBoxImpl = new MessageBoxImpl("myMessages",5);
  messageBoxImpl.setORB(orb);

  org.omg.CORBA.Object ref =
    rootpoa.servant_to_reference(messageBoxImpl);
  MessageBox box = MessageBoxHelper.narrow(ref);

  org.omg.CORBA.Object objRef =
    orb.resolve_initial_references("NameService");
  NamingContextExt ncRef = NamingContextExtHelper.narrow(objRef);

  NameComponent path[] = ncRef.to_name("MessageServer");
  ncRef.rebind(path, box);
  orb.run();
```

Przygotowanie klienta 1/3

```
C:\> idlj.exe -fclient messageBox.idl
```

[dir] MessageModule
- MessageBoxHelper.java
- MessageBoxHolder.java
- _MessageBoxStub.java

Przygotowanie klienta 2/3

MessageClient.java

```
…
public static void main(String[] args) {
  try {
    Properties props = new java.util.Properties();
    props.put("org.omg.CORBA.ORBInitialPort", "1050");
    props.put("org.omg.CORBA.ORBInitialHost", "localhost");

    org.omg.CORBA.ORB orb = org.omg.CORBA.ORB.init(args, props);
    org.omg.CORBA.Object objRef =
      orb.resolve_initial_references("NameService");
    NamingContext ncRef = NamingContextHelper.narrow(objRef);
```

usługa nazewnicza

```
    NameComponent nc = new NameComponent("MessageServer", "");
    NameComponent path[] = { nc };
    MessageBox box = MessageBoxHelper.narrow(ncRef.resolve(path));
```

znalezienie usługi MessageServer

Przygotowanie klienta 3/3

MessageClient.java

```
  …
  if (action.equals("reply")) { box.reply(param); }
  else if (action.equals("leave")) {
    try {
      System.out.println(box.leaveMessage(param));
    } catch (MessageModule.MessageBoxPackage.boxFull e) {
        e.printStackTrace(); }
  }
  else if (action.equals("get")) {
    String[] messages = box.getMessages();
    for (int i = 0; i < messages.length; i++)
      System.out.println("  " + messages[i]);
  }
  …
```

Materiały dodatkowe

- CORBA Faq, www.omg.org/gettingstarted/corbafaq.htm
- Object Managemet Group, http://www.omg.org
- "Java Programming with CORBA", G.Brose, A.Vogel, K.Duddy, Wiley Computer Publishing, 2001
- Introduction to CORBA, http://java.sun.com/developer/onlineTraining/corba/