Софийски университет “Св. Климент Охридски”

Факултет по Математика и Информатика

Катедра "Информационни технологии"

# ДИПЛОМНА РАБОТА

Тема:

## Приложение на Common Object Request Broker Architecture (CORBA) в биомедицината

**Дипломант:** Аглика Петрова Иванова

**Специалност:** Информатика

**Факултетен номер:** 42356

**Научен ръководител:** доц. Антоний Попов

**София,**
**07.2005 г.**

# Съдържание

# 1. Увод

Биологичните разработки днес зависят от широк диапазон от софтуер, взаимодействащ си с голям брой коренно различни източници на данни. Съществуват няколкостотин важни източника, които се различават както по големина, така и по сфера на действие. Многобройни групи от учени разработват и публикуват тези биологични данни в най-различни формати. Жизнено важно е да се измисли ефективен механизъм, който да позволява да се използват данни от тези различни източници.

През последните години стана ясно, че е от съществено значение да бъде открит начин, по който различните софтуери, хардуери и източници на данни да работят гладко помежду си, или казано по друг начин как да бъде постигната интероперативност. Направени бяха много опити да се осигури стандартен формат данни, но никой от тях не бе широко възприет. Една от ключовите концепции в резултат на тези усилия бе установяването на важната роля на метаданните – данни за данните. Търсените артибути на данните могат да бъдат строго определени чрез използването на подходящ метаезик. Метаезикът е използваем от машина, а програма, която няма никаква предишна информация за определена съвкупност от данни е в състояние да чете информацията и да я използва за разбор на данните. Предимството на този подход се състои в това, че един метаезик може да бъде използван да дефинира атрибутите на широк диапазон от различни съвкупности от данни. Изискванията за дадена спецификация са били обект на много спорове, като целите се концентрират по-скоро върху детайлите по изпълнението, отколкото върху това как точно тези деатйли са употребени. Това доведе до разработката и приложението на множество различни езици за дефиниране в сферата на биологичните изследвания, като те са или специфично проектирани да описват биологични данни (SDDL) или са по-общи езици за дефиниране (ASN.1, CIF, ACE, OPM). При всички случаи поддръжката на тези езици и свързаните с тях софтуерни инструменти се изпълнява от отделни групи специалисти в допълнение към останалите им дейности.

Темата на настоящата дипломна работа е разработване на сайт, запознаващ читателите и потребителите с приложението на Общата архитектура за посредници при запитвания за обекти Common Object Request Broker Architecture (CORBA) в биоинформатиката. Работата има за цел да се направи обзор и практическо описание на използването на CORBA във важни задачи от биоинформатиката, както и да осигури добро разбиране на основните механизми на архитектурата, да даде общ поглед на нейните компоненти и да запознае читателите и потребителите с терминологията, използвана в OMG документацията. Ще бъдат разгледани някои от приложенията на CORBA в

медицинските пациентски регистри и в телероботиката като пример за употребата на CORBA - базирани приложения в биомедицинската информатика.

Настоящата разработка е насочена към българските потребители и студенти и тъй като преведената литература по темата е твърде оскъдна, стремежът е начина на поднасяне на информацията да е максимално интересен и необременяващ. За това способства структурата на сайта, даваща възможност на всеки посетител сам да избере информацията, от която се интересува по бърз и удобен начин. Дипломната работа е разделена на няколко основни части, обособени съответно в глави и подглави.

# 2. Въведение в CORBA

## 2.1. Общи понятия за Common Object Request Broker Architecture (CORBA)

Общата архитектура за посредници при запитвания за обекти CORBA (Common Object Request Broker Architecture) е общ стандарт за работа с разпределени обекти. Счита се, че това е важна нова разработка в полето на биоинформатиката. CORBA е разработена от Група за управление на обекти (Object Management Group, OMG), голям консорциум, който включва главните софтуерни компании (SUN, DEC, IBM, Apple, Hewlett-Packard и т.н.), както и крайни потребители.[1]

Общата архитуктура за посредници при запитвания за обекти (CORBA) дефинира набор от стандарти, който образува свързана рамка, в която независимите източници на данни и техните услуги могат да бъдат достигнати. В тези стандарти влизат формалният език, езика за интерфейсна дефиниция (IDL), в който са специфицирани данните и услугите и ORB – посредник при запитвания за обекти, който е необходим, за да се реализират тези услуги. На практика те осигуряват главния компонент, който разделя целите за достъп до данни от въпросите за организация на данни. Тази посредническа стратегия позволява множество методи за достъп до данни да бъдат поддържани и да бъдат достъпни като стандартни услуги. Това позволява подобрения и усъвършенстване на замислената база данни, употребена изцяло без намеса в данните и без загуба на информация. По този начин могат да бъдат внедрени нови технологии и услуги и да бъдат направени общодостъпни, без да бъде застрашена стабилността на другите интерфейси на базата данни. Унифицирането на стандартите като CORBA ще бъде от голямо значение при разработката на следващото поколение интегрирани приложения за нуждите на биологията.

CORBA позволява взаимовръзката между обекти и приложения без значение от :

- компютърния език на приложенията, които осигуряват или използват обектите
- машинната архитектура на компютрите, които са свързани
- географското положение на компютрите (връзка чрез интернет)

По друг начин може да се каже :

За обектно-ориентираното програмиране CORBA е това, което е World Wide Web (WWW) за документите. [1]

## 2.2. Описание на CORBA

### 2.2.1. Група за управление на обекти (Object Management Group - OMG)

Групата за управление на обекти (http://www.omg.org/) е консорциум в компютърната индустрия, създаден през 1989г. с цел популяризирането на теорията и практиката на обектната технология в разпределените компютърни системи. На практика, стремежът на този консорциум е да намали сложността, да намали цените и да ускори въвеждането на нови софтуерни приложения. В началото OMG бе формирана от 13 компании, като в последствие членската маса нараства до над 600 софтуерни компании, разработчици и потребители. [2]

OMG реализира своите цели чрез създаване на стандарти, които позволяват взаимодействието и преносимостта на разпределените обектно-ориентирани приложения. Те не създават софтуер или указания за реализация, а само спецификации, като за тях се използват идеите на членовете на OMG, които отговарят на RFI (Request for Infirmation)- Заявка за информация и RFP (Request for Proposals)- Заявка за предложение. Силата на този подход идва от факта, че повечето софтуерни компании, заинтересовани от развитието на разпределените обектно-ориентирани технологии са сред членовете на OMG.

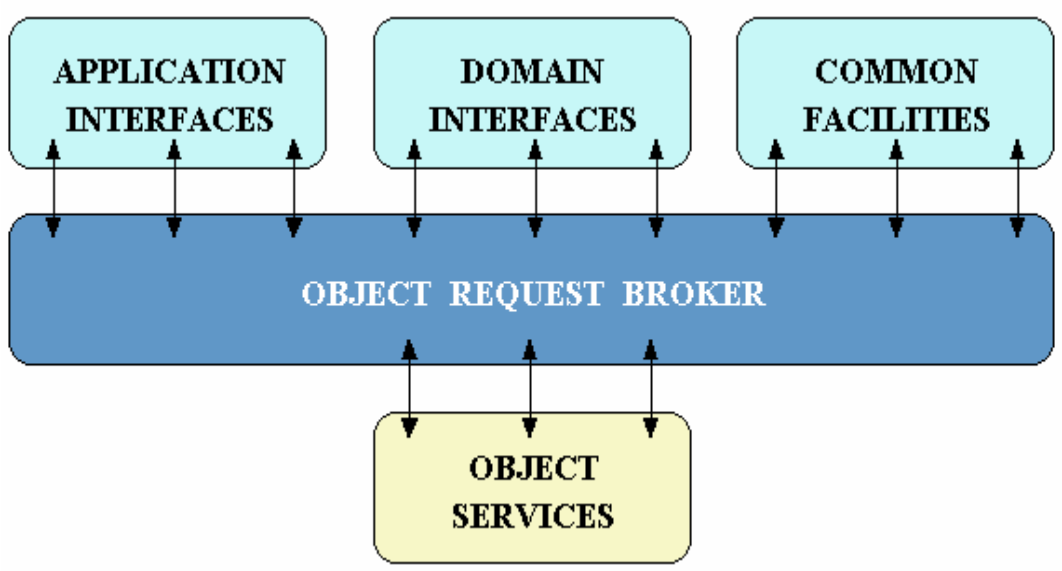
### 2.2.2. Архитектура за управление на обекти - Object Management Architecture (OMA)

OMA представлява поглед от високо ниво на напълно разпределена среда. Тя се състои от четири компонента, които грубо могат да бъдат разделени на две части: системно-ориентирани компоненти (Object Request Broker - Посредник при запитвания за обекти и Object services - Услуги на обектите) и компоненти, ориентирани към приложенията (Application Objects - Приложни обекти и Common facilities - Общи средства)

ORB (Object Request Broker) е онзи измежду всички компоненти, който образува основата на OMA и организира всички комуникации между нейните компоненти. Той позволява на обектите да си взаимодействат в хетерогенни разпределени среди,

независимо от платформите, на които тези обекти се основават и техниките, използвани за тяхната реализация. За изпълнението на тази задача ORB си служи с Object Services, които отговарят за общото управление на обектите, като например създаването на обекти, контрол на достъпа, пазенето на пътя до преместените обекти и т.н. Common Facilities (Общи средства) і Application Objects (Приложни обекти) са компоненти, които са по-близки до потребителя и при техните функции те извикват услугите на системните компоненти.

Следващата схема показва основните компоненти в модела на архитектурата на OMG. По-долу са представени описанията на основните компоненти [3].




Фигура 1. Описание на модел на архитектурата на OMG

**Услуги на обектите - Object Services** - Това са независими от домейните интерфейси, които се използват от много програми за разпределени обекти. Така например почти винаги има нужда от услуга, осигуряваща откриването на други налични услуги без значение от домейна на приложението. Това са две от примерните обектни услуги, които изпълняват тази роля:

- Именуваща услуга -- позволява на клиентите да намират обектите на базата на техните имена;
- Trading услуга -- позволява на клиентите да намират обектите на базата на техните особености.

Това са също така и спецификации на обектни услуги за управление на жизнения цикъл, сигурността, транзакциите и нотификациите на събития, така както и много други.

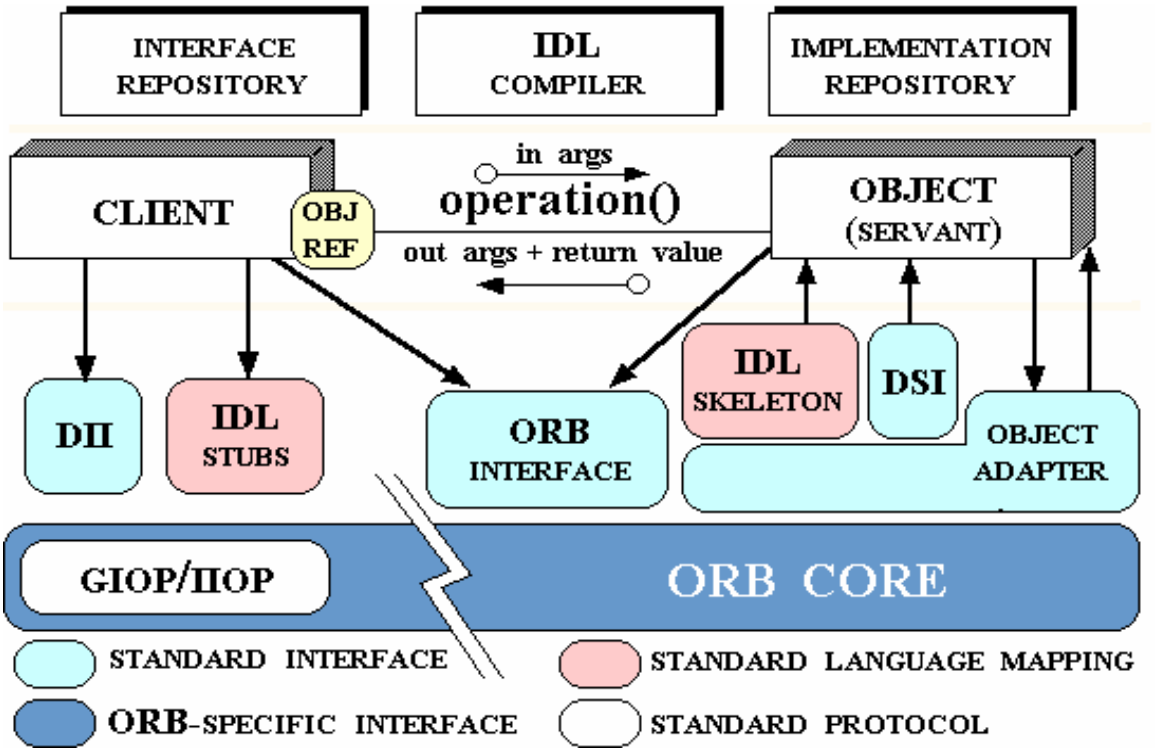
**Общи средства - Common facilities** - Подобно на интерфейсите на обектните услуги, тези интерфейси са също така хоризонтално-ориентирани, но за разлика от обектните услуги, те са ориентирани към приложенията на крайния потребител. Пример за такава функционалност е Функционалността на Разпределеният Документен Компонент-Distributed Document Component Facility (DDCF), което представлява Обща функционалност на съставните документи, базирана на OpenDoc. DDCF позволява представянето и обмена на обекти, базирано на документния модел, като например улесняването на връзките на spreadsheet обект е рапортен документ.

**Интерфейси на домейни - Domain Interfaces** - Тези интерфейси изпълняват ролята, подобна на обектните услуги, но те са ориентирани към специфичните домейни на приложенията. Така например един от първите OMG RFP (Request for Proposal), издадени за домейн интерфейс е на PDM (Product Data Management). Ще бъдат издадени и други OMG RFP в телекомуникационните, медицинските и финансовите сфери.

**Интерфейси на приложения - Application Interfaces** - Тези интерфейси са разработени специално за определено приложение. Тъй като те са специфични за даденото приложение и тъй като OMG не разработва приложения (само спецификации), тези интерфейси не са стандартизирани. Ако с времето се окаже, че някои определени функции изпъкнат с широка уптреба в определен домейн на приложение, то те могат да бъдат кандидати за по-нататъшна OMG стандартизация.

## 2.3. Архитектура на CORBA ORB

Следващата графика илюстрира основните компоненти на архитектурата на CORBA ORB. Описанието на тези компоненти е на разположение под фигурата: [3]




Фигура 2. Архитектура на CORBA ORB

- *Обект* — в CORBA програмирането това е съвкупност от идентичност, интерфейс и реализация, която е позната също така и под името Servant

- *Servant* – реализация на програмния език, която дефинира операциите за поддръжка на IDL интерфейса. Servant – и могат да бъдат написани на различни езици, включително и C, C++, Java, Smalltalk и Ada.

- *Клиент* — това е програмната единица, която извиква операциите на обектните реализации. Достъпът до услугите на отдалечени обекти трябва да е ясен за извикващия ги. В оптималния случай той трябва да бъде лесен като извикването на метод за обект, напр. obj->op(args). Останалите компоненти на горепосочената схема спомагат за поддържането на това ниво на яснота.

- *Посредник при запитвания за обекти* (Object Request Broker, ORB)—ORB осигурява механизъм за ясна комуникация между клиентските заявки и целеви обектни реализации. ORB опростява разпределеното програмиране чрез разединяването на клиента от детайлите на извикванията на методи. Това прави клиентската заявка да изглежда като извикване на локална процедура. Когато клиентът извика определена

операция, ORB се грижи за това обектната реализация да бъде открита, изрично да бъде активирана ако е неодходимо, да достави заявката до обекта и да върне отговор обратно на извикващия операцията.

- *Интерфейс на ORB* — ORB е логическа свързаност, която може да бъде изпълнена по различни начини (като един или повече процеси или съвкупност от библиотеки). За да бъдат разделени приложенията от детайлите на реализацията, CORBA спецификацията дефинира абстрактен интерфейс за ORB. Този интерфейс осигурява различни спомагателни функции като конвертирането на обектни референции в стрингове и обратно, създаването на списъци с аргументи за заявки, направени чрез Интерфейса за динамично извикване, описан по-долу.

- CORBA IDL *stub* – ове и *скелети*. CORBA IDL stub – овете и скелетите служат като "спойка" между клиента и сървърските приложения и съответно ORB –а. Трансформацията между CORBA IDL дефинициите и целевият програмен език е автоматизирана от CORBA IDL компилатор. Употребата на компилатор намалява възможните несъответствия между клиентските stub- ове и сървърските скелети и увеличава възможностите за автоматично компилирани оптимизации.

- *Динамично извикване на интерфейс* (Dynamic Invocation Interface, DII) – Този интерфейс позволява на клиента директно да достига до съществените механизми за заявка, осигурени от ORB. Приложенията използват DII, за да правят заявки към обектите динамично, без да изискват свързването със специфичен stub на IDL интерфейса. За разлика от IDL stub – овете (които позволяват RPC заявки), DII също така позволява на клиентите да правят не-блокиращи отсрочени синхронизирани (отделни операции по изпращане и получаване) и еднопосочни (само за изпращане) извиквания.

- *Динамичен скелетен интерфейс* (Dynamic Skeleton Interface, DSI) – От страна на сървъра това е аналогия на DII от клиентската страна. DSI позволява на ORB да доставя заявки към обектни реализации, за който по време на компилацията няма информация за типа на обекта, за който се изпълнява. Когато прави заявката клиентът не знае дали реализацията й типово-специфични IDL скелети или използва динамичните скелети

- *Обектен адаптор* – Той помага на ORB за доставянето на заявките към обекта като активира обекта. По-важното е че обектния адаптор свързва обектната реализация с