

ЛЕКЦІЯ 5. ТЕХНОЛОГІЧНИЙ СТАНДАРТ CORBA

План

1. Поняття CORBA
2. Об'єктна модель CORBA
3. Брокер об'єктних запитів (Object Request Broker ORB)
4. Базовий об'єктний адаптер
5. Мова опису інтерфейсів
6. Архітектура інформаційної системи на основі CORBA
7. Порівняння CORBA з іншими технологіями

1. Поняття CORBA

CORBA (Common Object Request Broker Architecture) – це набір відкритих специфікацій інтерфейсів, що визначає архітектуру технології міжпроцесної взаємодії і незалежного маніпулювання об'єктами. Розробниками технології інтерфейсів є організації OMG і X/Open.

Завдання CORBA – інтегрувати розподілені системи, дати можливість програмам, що написані різними мовами та працюють у різних вузлах мережі, взаємодіяти одна з одною так само просто, наче вони знаходяться в адресному просторі одного процесу.

Реалізувати технологію відповідно до специфікацій CORBA може хто завгодно. Створені програмні продукти, природно, вже не є відкритими, а стають комерційними продуктами

CORBA дозволяє розглядати всі застосування в розподіленій системі як об'єкти. Причому об'єкти можуть одночасно грати роль і клієнта, і сервера: роль клієнта, якщо об'єкт є ініціатором виклику методу іншого об'єкту; роль сервера, якщо інший об'єкт викликає на ньому який-небудь метод. Об'єкти-сервери зазвичай називають реалізацією об'єктів. Практика показує, що більшість об'єктів одночасно виконують роль і клієнтів, і серверів, почерзі викликаючи методи інших об'єктіві відповідаючи на виклик ззовні. Використовуючи CORBA, тим самим, є можливість будувати набагато гнучкіші системи, ніж системи клієнт-сервер, засновані на дворівневій чи трьохрівневій архітектурі.

Головними компонентами моделі CORBA є:

- брокер об'єктних запитів (Object Request Broker);
- мова визначення інтерфейсів (Interface Definition Language).

У специфікацію CORBA входить також декілька додаткових, але важливих служб (сервісів):

- служба динамічного формування запитів (DII);
- служба репозиторія інтерфейсів (IR);
- служба динамічної обробки запитів (DSI);

- служби особистих брокерів запитів (GIOP).

Характерні особливості розробок кросплатформних розподілених ПС за технологією CORBA полягають в наступному:

- *Мова опису інтерфейсів OMG IDL* дозволяє визначити інтерфейс, незалежний від мови реалізації.
- *Високий рівень абстракції CORBA* в семирівневій моделі відкритих систем позбавляє програміста від роботи з низькорівневими мережевими протоколами.
- Програмісту не потрібна інформація про місце сервера в мережевій інформаційній системі і способі його активації.
- Розробка клієнтської програми не залежить від серверної операційної системи і апаратної платформи.

2. Об'єктна модель CORBA

Об'єктна модель CORBA визначає взаємодію між клієнтами і серверами. Клієнти – це застосування, які використовують сервіси, що надаються серверами. Об'єкти-сервери містять набір сервісів, що розділяються між багатьма клієнтами. Операція вказує запрошуваний сервіс. Інтерфейси об'єктів описують операції, які можуть бути викликані клієнтами певного об'єкту. Реалізації об'єктів – це застосування, які виконують сервіси, що викликаються клієнтами.

3. Брокер об'єктних запитів (Object Request Broker ORB)

Головний компонент CORBA – брокер об'єктних запитів (ORB – Object Request Broker) – програмне забезпечення, що забезпечує зв'язок між об'єктами. Його завданням є надання механізму виконання запиту об'єкту-клієнта: пошук об'єкту, до якого відноситься даний запит, передача необхідних даних, підготовка об'єкту до обробки. Брокер забезпечує прозору взаємодію клієнтського і серверного застосувань. Для розробника виклик методів віддалених об'єктів не відрізняється від звичайних локальних викликів.

Стандарт CORBA визначає, яким чином програмні компоненти, розподілені по мережі, можуть взаємодіяти один з одним незалежно від операційних систем і мов реалізації.

ORB дозволяє:

- знайти віддалений об'єкт по Об'єктному Посиланню (IOR - Interoperable Object Reference);
- викликати метод віддаленого об'єкту, передавши йому вхідні параметри (маршалінг параметрів - marshaling parameters);
- отримати значення і параметри, які повертаються від сервера (демаршалінг параметрів - unmarshaling parameters).

Обробка викликів різних видів відбувається різними способами. Виклик віддаленого об'єкту обробляється особливими методами, визначеними в CORBA-специфікації. Вони формують по зробленому запиту низькорівневе представлення, залежне від використовуваних апаратно-програмних засобів.

Клієнт може викликати методи об'єктів за допомогою ORB декількома способами. Можливі варіанти взаємодії об'єктів показані на рис. 5.1.

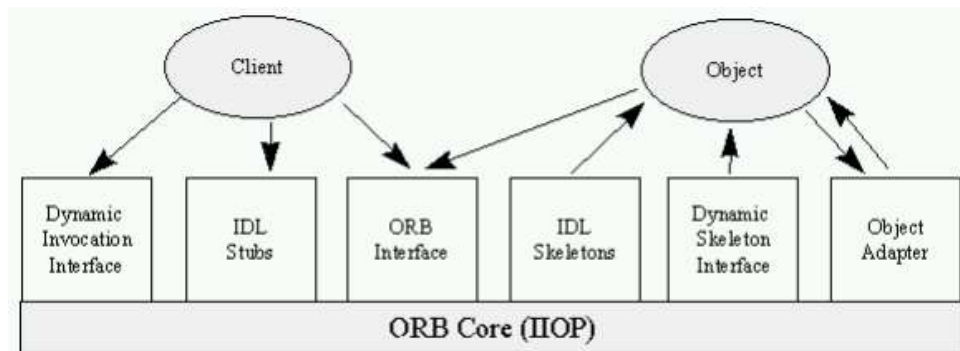


Рис. 5.1 Взаємодія об'єктів в CORBA

Dynamic Invocation Interface (DII): дозволяє клієнту знаходити сервери і викликати їх методи під час роботи системи.

IDL Stubs: визначає, яким чином клієнт виконує виклик сервера.

ORB Interface: загальні як для клієнта, так і для сервера служби (сервіси).

IDL Skeleton: забезпечує статичні інтерфейси для об'єктів певного типу.

Dynamic Skeleton Interface: загальні інтерфейси для об'єктів, незалежно від їх типу, які не були визначені в IDL Skeleton.

Object Adapter: здійснює комунікаційну взаємодію між об'єктом і ORB.

Брокер об'єктних запитів є проміжним програмним забезпеченням, призначеним для об'єднання інформаційних ресурсів розподіленої неоднорідної системи, дозволяючи одній частині системи не піклуватися про фізичне розташування інших частин (об'єктів) системи.

На ринку представлені ORB різних виробників (наприклад, VisiBroker, WebLogic), але всі вони відповідають єдиній специфікації CORBA. Тому в принципі CORBA дозволяє будувати розподілені системи, одночасно використовуючи ORB різних виробників, і будуючи систему одночасно на різних платформах і різних мережевих протоколах.

Переваги ORB

Теоретично CORBA визначається як найкраща клієнт/сервер middleware-система, але на практиці вона задовільна лише настільки, наскільки задовільні продукти, що її реалізують. До основних комерційних ORB відносяться: Orbix від фірми IONA Technologies; DSOM від IBM; Object Broker від Digital; JOE від Sun; Visibroker від Visigenic та Netscape; ORB+ від HP.

Переваги кожної CORBA ORB такі:

- *статистичні та динамічні виклики методів.* CORBA ORB надає можливість або статично визначити виклики методів прямо під час компіляції, або знаходити їх динамічно, але вже під час роботи програми.

- *відображення та мови високого рівня.* CORBA ORB дає змогу викликати методи у серверних компонент, використовуючи будь-який з певного списку мов високого рівня – C, C++, SmallTalk, Java, Ada. Абсолютно неважливо, на якій мові написані об'єкти. CORBA відділяє інтерфейси від реалізації і надає мово незалежні типи даних, що дає змогу здійснювати виклик методів, минаючи границі якоїсь мови програмування та конкретної операційної системи.

- *система, що здатна до самоопису.* З використанням своїх метаданих, CORBA дає змогу описувати інтерфейс будь-якого сервера, відомого системі. Кожна CORBA ORB повинна підтримувати Репозитарій Інтерфейсів, який зберігає необхідну інформацію, яка описує синтаксис інтерфейсів, підтримуваних серверами. В своїй роботі клієнти використовують ці метадані для здійснення викликів до серверів.

- *прозорість.* ORB може виконуватися як сам собою (наприклад, на портативному комп'ютері), оскільки в оточенні всіх інших ORB, з якими вона взаємодіє шляхом CORBA 2.0 IIOP (Internet Inter- ORB Protocol) протоколу. ORB може здійснювати між об'єкту взаємодію також для одного процесу, як і для кількох процесів, які виконуються на одній машині, та для процесів, виконання яких відбувається в мережі, під різними операційними системами. Реалізація цих взаємодій ніяк не зачіпає самі об'єкти. При використанні технології CORBA, розробник не має турбуватися про розташування серверів, запуск (активування) об'єктів, вирівнювання розміру змінних в залежності від платформи та операційної системи, та про те, як здійснюється передача повідомлень. Рішення всіх цих задач бере на себе продукт, який підтримує стандарт CORBA.

- *вбудована безпека.* Всі свої запити ORB доповнює певною контекстною інформацією, яка забезпечує збереження даних.

- *поліморфізм при виклику методів.* ORB не просто викликає віддалений метод – ORB викликає метод на віддаленому об'єкті. Тобто виконання одних і тих же функцій на різних об'єктах призведуть до різних дій, в залежності від типу об'єкта.

В архітектурі CORBA кожен об'єкт, методи якого доступні іншим об'єктам (зазвичай його називають CORBA-об'єктом), має унікальне по всій доступній мережі Об'єктне Посилання (IOR -Interoperable Object Reference), по якому до нього можна звернутися. Шукати CORBA-об'єкти можна як по IOR, так і по символічних іменах, якщо вони зареєстровані (зазвичай при створенні) в спеціальному сервісі імен (NameService). Для звернення до методів CORBA-об'єкта останній має відкритий для всіх останніх CORBA-об'єктів інтерфейс.

Інтерфейси CORBA-об'єктів прийнято описувати на спеціальній, визначеній специфікацією CORBA мові IDL (Interface Definition Language). Виробники ORB поставляють разом з ORB також і утиліти, що перетворюють описи інтерфейсів CORBA- об'єктів в конструкції відповідних мов програмування.

4. Базовий об'єктний адаптер

Специфікація OMG CORBA визначає базовий об'єктний адаптер (Basic Object Adapter - BOA), який має бути реалізований в усіх брокерах запитів.

Базовий об'єктний адаптер – це набір інтерфейсів для створення посилань на віддалені об'єкти, реєстрації об'єктів, авторизації запитів і активізації застосувань.

Основна функція об'єктного адаптера, який використовується для реалізації CORBA-об'єкта, – забезпечення доступу до служб брокера об'єктних запитів. Об'єктний адаптер надає всі низькорівневі засоби для зв'язку об'єкту з його клієнтами. До числа цих засобів входять:

- генерація посилань на віддалені об'єкти;
- виклик методів, визначених в IDL;
- забезпечення безпеки взаємодії;
- активація і деактивація об'єктів;
- встановлення відповідності між посиланнями на віддалені об'єкти і реальними екземплярами об'єктів;
- реєстрація об'єктів.

Загалом, для кожного типа застосувань передбачається створення власного об'єктного адаптера. Базовий об'єктний адаптер є рішенням першочергової задачі забезпечення зв'язку між реалізацією об'єкту і брокером запитів. Для організації взаємодії між ORB і, наприклад, системою управління базами даних має бути розроблений свій об'єктний адаптер.

5. Мова опису інтерфейсів

Мова опису інтерфейсів (IDL) – ключовий елемент специфікації CORBA. Вона є засобом, за допомогою якого можуть бути визначені операції, що викликаються клієнтами у віддалених серверних об'єктах. IDL – повністю об'єктно-орієнтована мова, що підтримує інкапсуляцію, спадкоємство (у тому числі множинне) і поліморфізм.

Синтаксис IDL подібний до синтаксису мови C++, що спрощує його вивчення великою кількістю розробників, що володіють C++. В той же час, IDL є потужною мовою, що дозволяє описувати інтерфейси CORBA об'єктів будь-якої складності. Умову введені всі конструкції, необхідні для відображення об'єктних властивостей застосувань.

Опис інтерфейсу на IDL складається із заголовка і тіла. Заголовок містить назву інтерфейсу і вказує успадковані інтерфейси. Тіло складається з оголошень констант, типів, виняткових ситуацій, атрибутів і операцій.

IDL – не повна мова програмування. Це чисто декларативна мова, що описує інтерфейс і не містить жодної реалізації.

IDL-специфікації можуть бути відкомпільовані в заголовочні файли і спеціальні прототипи серверів, які можуть використовуватися безпосередньо програмістом. Тобто IDL-визначені методи можуть бути написані, а потім виконані, на будь-якій мові, для якої існує відображення з IDL.

Відображення також є предметом стандартизації OMG. Відомі стандартизовані відображення для Ada, C, C++, Lisp, Smalltalk, Java, COBOL, PL/I і Python (рис. 5.2).

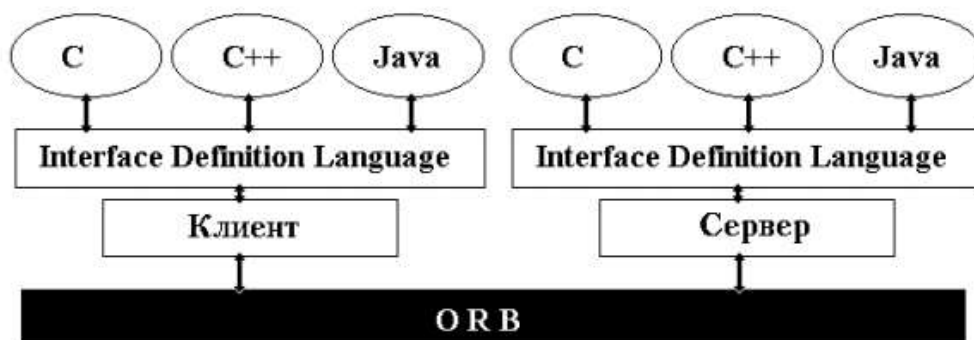


Рис. 5.2 CORBA IDL відображення в моделі клієнт/сервер

За допомогою IDL можна описати і атрибути компонента, і батьківські класи, які вона успадковує, і виключення, що викликаються, і, нарешті, методи, що визначають інтерфейс, причому з описом вхідних і вихідних параметрів.

6. Архітектура інформаційної системи на основі CORBA

Трирівнева архітектура інформаційних систем, згідно специфікацій OMG, включає системи управління даними, мережі взаємодіючих CORBA-об'єктів і інтерфейси користувача для представлення даних.

Вочевидь, що для більшості ІС необхідна деяка множина загально-системних об'єктних сервісів (служб), які не залежать від предметної області і забезпечують базову функціональність для управління розподіленими об'єктами. З метою полегшення створення розподілених застосунків консорціум OMG стандартизував основні системні служби (сервіси):

Служба іменування (Naming service) служить для управління посиланнями на CORBA-об'єкти і їх зберігання. Її основне завдання полягає в тому, щоб універсальним чином організувати з'єднання об'єктів один з одним. Служба іменування оперує зі сховищем об'єктних посилань. Звернення до неї виконується для отримання потрібного об'єктного посилання, що ідентифікується по зрозумілому розробникові імені об'єкту.

Служба подій (Event service) забезпечує підтримку асинхронної взаємодії застосунків.

Служба зберігання об'єктів (Persistence service) надає набір універсальних інтерфейсів для збереження екземплярів об'єктів в довготривалій пам'яті. Служба розроблена таким чином, що можлива її реалізація на основі об'єктної бази даних.

Служба управління транзакціями (Transaction service) підтримує багатомодель транзакцій, включаючи вкладені транзакції. Необхідна для коректної роботи застосунків в розрахованому на багато користувачів режимі.

Служба зв'язку (Relationship service) реалізує логічні зв'язки між CORBA-об'єктами. Служба визначає два додаткові типи об'єктів: зв'язок і роль. Роль є CORBA-об'єкт, що відображає характер зв'язку, а зв'язок характеризує залежності об'єктів прикладної області.

Служба управління розподіленими ресурсами (Concurrency control service) дозволяє клієнтам координувати свої дії при використанні ресурсів, що розділяються. Управління розподіленими ресурсами здійснюється за допомогою блокувань. Кожне

блокування асоціюється з єдиним ресурсом і єдиним клієнтом. Служба визначає декілька режимів блокувань, які відповідають різним способам доступу.

Служба зовнішнього представлення (Externalization service) формує копію CORBA-об'єкта у вигляді деякого зовнішнього представлення – файлу, елементу бази даних тощо.

7. Порівняння CORBA з іншими технологіями

Порівняння CORBA і DCOM

CORBA претендує на спробу створення стандартної платформи проміжного рівня для спільної роботи застосувань різних виробників.

Основною метою розробки DCOM було розширення функціональних можливостей при збереженні сумісності з існуючими версіями, які були в ранніх версіях Windows.

По різному реалізовані і інтерфейси. CORBA надає стандартну мову IDL, на якій виконуються визначення інтерфейсів з подальшим перетворенням на тексти програм на вибраній мові програмування. У DCOM використовуються бінарні інтерфейси (табличні). При такому підході інтерфейси об'єктів визначаються незалежно від мови програмування.

Порівняння RPC і CORBA

Чим механізм викликів CORBA відрізняється від механізму RPC? Ці механізми схожі, але проте між ними є серйозні відмінності реалізації. За допомогою RPC можна викликати певну функцію. А за допомогою ORB можна викликати метод в певного об'єкту. Різні об'єкти класів можуть по-різному відповідати на виклик одного і того ж методу. Оскільки кожен об'єкт управляє своєю власною інформацією, то метод буде викликаний на суто конкретних даних.

В разі RPC, буде виконаний лише якийсь конкретний фрагмент коду сервера, який і взаємодіє з даними сервера. Всі функції з однаковими іменами будуть виконані абсолютно однаково. У RPC відсутня конкретизація викликів, в тому сенсі, в якому це відбувається в ORB. У ORB всі виклики функцій відбуваються до конкретних об'єктів, тим самим, результати цих функцій можуть бути абсолютно різні. Виклики функцій обробляються в специфічній для кожного окремого об'єкту формі.

Контрольні запитання і завдання для самостійної роботи

1. Дайте визначення поняттю CORBA.
2. Назвіть головні елементи CORBA.
3. Що таке брокер об'єктних запитів?
4. Що таке розподілені дані застосунків?
5. Перелічіть найбільш відомі реалізації CORBA.
6. У чому полягає кросплатформність моделі CORBA
7. Архітектура управління об'єктами в CORBA