

Распределенные вычислительные системы

Лекция №5: Стандарты OMG, Архитектура CORBA, ORB

Алексей В. Бурдаков, к.т.н.
burdakov@usa.net

План лекций

Дата	Тема
06.09	1. Вводная лекция: распределенные системы
13.09	2. Эволюция распределенных технологий, метамодель
20.09	3. Принципы ПО среднего слоя
27.09	4. Стандарты OMG, Архитектура CORBA, ORB
04.10	5. <u>COM, Java/RMI</u>
11.10	ПК1
18.10	6. Определение местонахождения распределенных объектов
25.10	7. Долговременное хранение распределенных объектов
01.11	8. Распределенные объектные транзакции
08.11	9. Безопасность
15.11	10. Расширенное взаимодействие
22.11	ПК2
29.11	Вакантно
06.12	Вакантно
13.12	Зачет/Экзамен

5-2

План лекции

- Анонс рейтинга
- Java/RMI
- COM

5-3

Анонс рейтинга

- 01.11 планируется проведения рейтинга
- Рейтинг
 - 20-30 минут
 - Каждому по несколько вопросов
 - Случайная выборка из общего пула в 60-70 вопросов (пул формируется)
- 20% лучших получают преимущество на зачете

5-4

План лекции

- Анонс рейтинга
- Java/RMI
- COM

5-5

Язык Java

- Java – объектно-ориентированный язык программирования, созданный Sun Microsystems
- Гибридный язык программирования, который и компилируется и интерпретируется
- Компилятор создает байт-код, который после интерпретируется виртуальной машиной

5-6

Цели RMI

- В Java 1.0 взаимодействие между объектами ограничено одной виртуальной машиной
- RMI (Remote Method Invocation – Удаленный вызов методов) поддерживает взаимодействие между различными виртуальными машинами (потенциально между различными узлами)
- Обеспечивает тесную интеграцию с Java
- Минимизация изменений в язык Java
- Работает в гетерогенных средах

5-7

Объектная модель Java

- Интерфейсы и удаленные объекты
- Классы
- Атрибуты
- Операции
- Исключения
- Наследование

5-8

Интерфейсы Java и удаленные объекты

- Java уже включает концепцию интерфейсов в язык (различаются классы и интерфейсы)
- RMI не имеет специального языка определения интерфейсов
- Предопределенный интерфейс Remote
- Удаленные интерфейсы расширяют интерфейс Remote
- Удаленные классы реализуют удаленные интерфейсы
- Удаленные объекты являются экземплярами удаленных классов

5-9

Пример удаленного интерфейса Java

Имя пакета *Имя интерфейса* *Декларация как удаленного*

```
package soccer;
interface Team extends Remote {
public:
    String name() throws RemoteException;
    Trainer[] coached_by() throws RemoteException;
    Club belongs_to() throws RemoteException;
    Players[] players() throws RemoteException;
    void bookGoalies(Date d) throws RemoteException;
    void print() throws RemoteException;
};
```

Удаленная операция

5-10

Атрибуты

- RMI не поддерживает атрибуты
- Атрибуты должны быть представлены как пары Set и Get операций разработчиком
- Пример:

```
interface Club extends Organization, Remote {
public:
    int noOfMembers() throws RemoteException;
    Address location() throws RemoteException;
    Team[] teams() throws RemoteException;
    Trainer[] trainers() throws RemoteException;
    ...
};
```

Операция get для получения значения атрибута

5-11

Комбинация классов и удаленных интерфейсов

```
interface Organization {
private:
    String name() RemoteException;
};
class Address {           Club возвращает объект Адрес
public:
    String street;
    String postcode;
    String city;
};
interface Club extends Organization, Remote {
public:
    int noOfMembers() throws RemoteException;
    Address location() throws RemoteException;
    Team[] teams() throws RemoteException;
    Trainer[] trainers() throws RemoteException;
    void transfer(Player p) throws RemoteException;
};
```

Club делает операцию name() Доступной удаленно

5-12

Операции и передача параметров

- Операции называются методами
- Удаленные вызовы:
 - Атомарные типы передаются по значению
 - Удаленные интерфейсы передаются по ссылке
 - Не- удаленные объекты передаются по значению

```
class Address {
    public:
        String street;
        String postcode;
        String city;
};
interface Club extends Organization, Remote {
    public:
        Address location() throws RemoteException;
        ...
};
```

Возвращает копию address!

5-13

Операции и передача параметров

Параметр	Атомарный тип	Неудаленный объект	Удаленный объектный тип
Локальный	CBV	CBR	CBR
Удаленный	CBV	CBV	CBR

5-14

Заявки

- Семантика однократного выполнения, т.е. если нет ошибки, то метод выполнен успешно
- В отличии от CORBA и COM, поддерживается только статические вызовы (т.е. до компиляции приложения необходимо знать с кем будет взаимодействовать клиентский объект)

5-15

Исключения

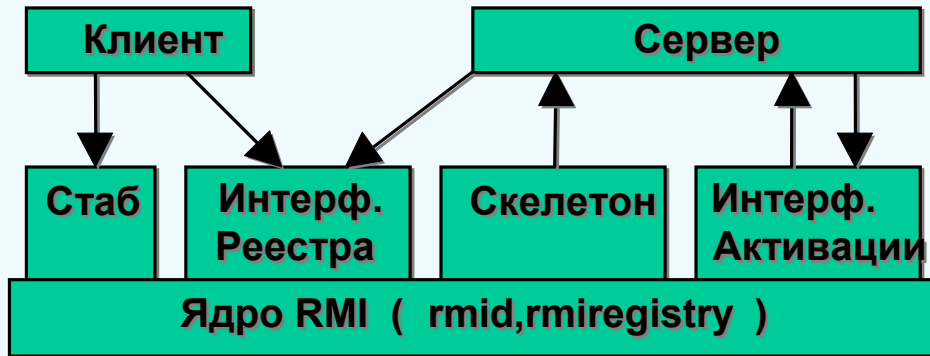
- Предопределенные исключения RemoteException
- Типо-зависимые исключения
- Пример:

Типо-зависимые исключения

```
class PlayerBooked extends Exception {};  
interface Team extends Remote {  
    public:                Операция декларирует, что она может  
                           вызвать это исключение  
        ...  
        void bookGoalies(Date d) throws ←  
            RemoteException, PlayerBooked;  
        ...  
};
```

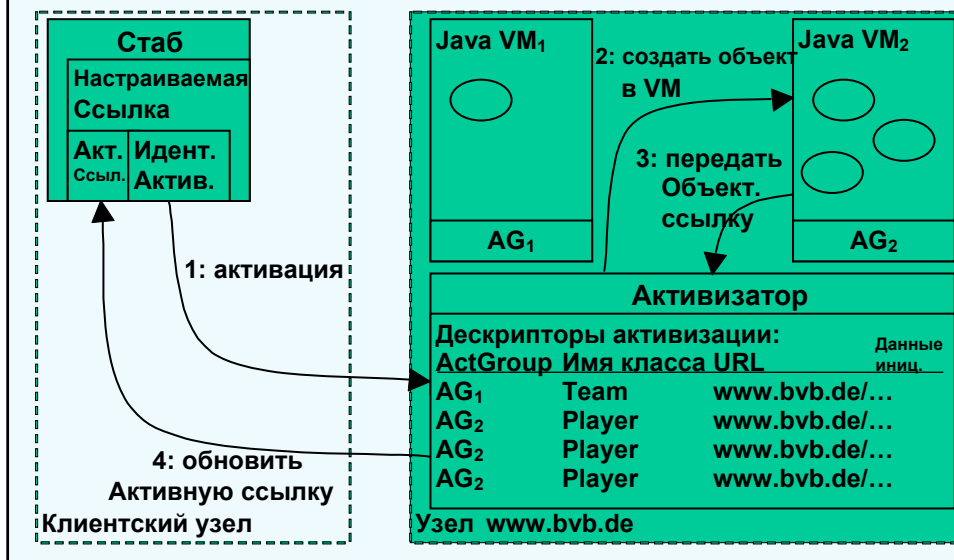
5-16

Архитектура



5-17

Активация в Java



Обсуждение

- Использование различной семантики вызова для удаленных и локальных объектов означает неполную прозрачность доступа
- Стандартный механизм ссылок в Java не обеспечивает прозрачности местонахождения
- В этом отношении Java/RMI стоит ниже нежели COM & CORBA

5-19

План лекции

- Анонс рейтинга
- Java/RMI
- COM

5-20

COM: История

- Первая версия COM представлена в 1993 г.
- Поддержка распределенности представлена в Windows NT 4.0
- Различные названия – Network/OLE, DCOM, ActiveX
- Построена на основе OSF/DCE

5-21

COM: Цели

- Компонентная модель, обеспечивающая двоичную инкапсуляцию и двоичную совместимость
 - Двоичная инкапсуляция: Клиенты не должны быть перекомпилированы если серверные объекты изменяются (в противовес синтаксической инкапсуляции)
 - Двоичная совместимость: Клиентские и серверные объекты могут быть разработаны с помощью различных средств и языков
- COM – собственный де-факто стандарт

5-22

COM: Объектная модель и определение интерфейса

- Интерфейсы
- Реализации и объекты
- Классы
- Атрибуты
- Операции
- Запросы
- HRESULTS
- Наследование

5-23

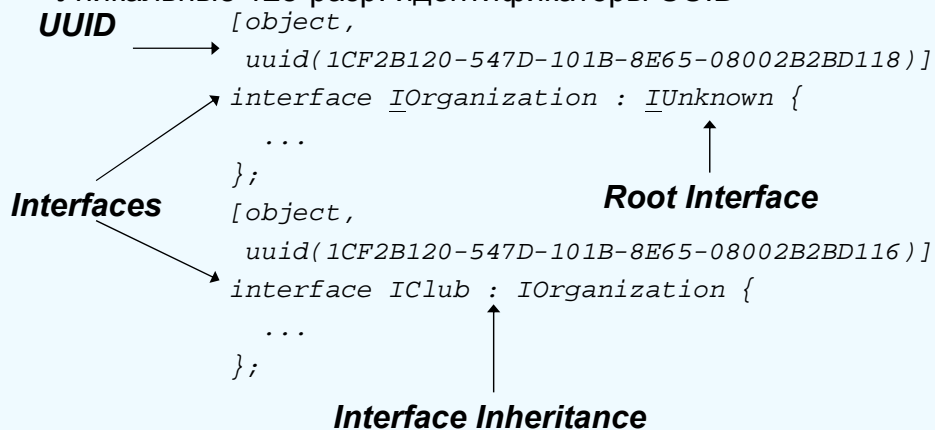
COM: Microsoft IDL (MIDL)

- Язык, выражающий концепции COM
- MIDL:
 - Язык независимый от языка программирования
 - В основе лежит OSF/RPC IDL
 - Вычислительно неполон
 - Не позволяет ограничивать область видимости с помощью модулей
- Доступно связывание с различными языками: Microsoft C++, Visual Basic, J++, а также средства других производителей – Delphi, PowerBuilder, и т.д.

5-24

COM: Интерфейсы

- Интерфейс определяет протокол, посредством которого объекты-клиенты связываются с объектами-серверами
- Уникальные 128-разр. идентификаторы UUID



5-25

COM: Реализации

- Реализует интерфейсы на языках программирования (C++)

```
#include "Soccer.h"
class Player : public IPlayer {
private:
    char* name;
    short Number;
protected:
    virtual ~TrainerPlayer(void);
public:
    TrainerPlayer(void);
    IMPLEMENT_UNKNOWN(TrainerPlayer)
    BEGIN_INTERFACE_TABLE(TrainerPlayer)
    IMPLEMENTS_INTERFACE(ITrainer)
    IMPLEMENTS_INTERFACE(IPlayer)
    END_INTERFACE_TABLE(TrainerPlayer)
    void book(); // IPlayer methods
};
```

5-26

COM: Объекты

- Объект – экземпляр реализации COM и представляет собой экземпляр класса в соответствующем языке программирования
- Ссылки на объекты COM называются указателями на интерфейсы
- Указатели на интерфейсы ссылаются на области в основной памяти
- Если объект удален, то ссылка указывает на гроху
- Ссылки обеспечивают прозрачность расположения
- Объектные ссылки постоянны (persistent)

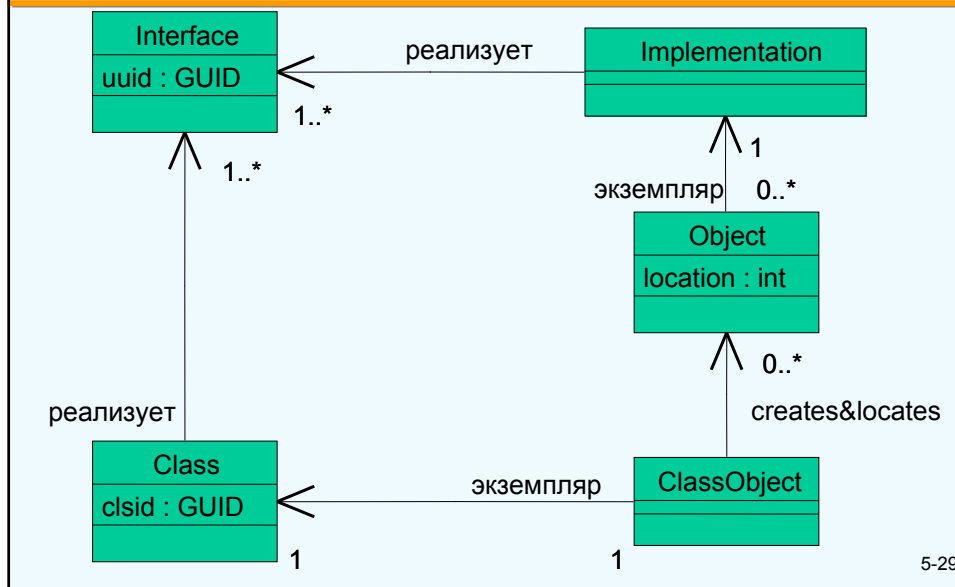
5-27

COM: Классы

- Именованные реализации, представляющие типы, для которых возможно создание экземпляров
- Один или несколько интерфейсов
- Основной механизм для создания объектов COM
- Может вернуть указатель на интерфейс к объекту COM

5-28

COM: Объекты, интерфейсы и классы



COM: Типы

- Атомарные типы (платформо-независимые)
 - Byte, small, short, long, float, double, char, enum, boolean
- Конструкторы типов (меньше чем в CORBA)
 - Struct и array (фиксированной, переменной длины и комбинированные)

5-30

COM: Атрибуты

- COM не поддерживает декларацию атрибутов
- Атрибуты должны быть представлены как операции установки и чтения (set / get)
- COM использует специальное ключевое слово для обозначения такой операции
- Пример:

```
interface IOrganization : IUnknown {  
    [propget] HRESULT Name([out] BSTR val);  
};
```

5-31

COM: Операции

```
interface IClub : IOrganization {  
    [propget] HRESULT NoOfMembers([out] short *val);  
    [propget] HRESULT Address([out] ADDRESS *val);  
    [propget] HRESULT Teams([in] long cMax, [out] long *pcAct,  
        [out, size_is(cMax), length_is(*pcAct)] ITeam *val);  
    [propput] HRESULT Teams([in] long cElems,  
        [in, size_is(cElems)] ITeam *val);  
    [propget] HRESULT Trainers([out] ITrainer *val[3]);  
    [propput] HRESULT Trainers([in] ITrainer *val[3]);  
    HRESULT transfer([in] IPlayer *p);  
};
```

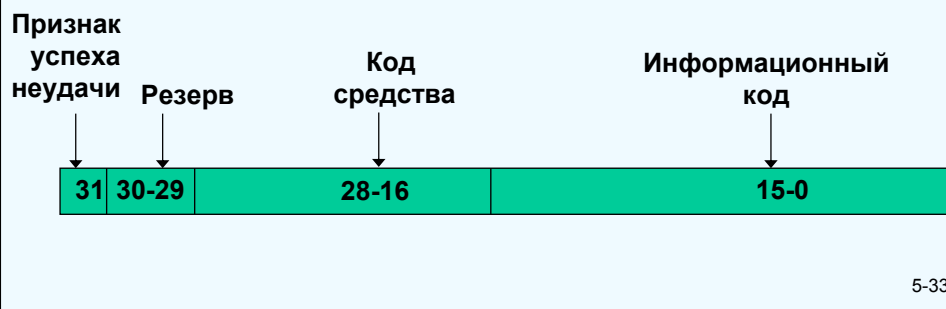
Вид параметра *Список параметров*

Возвращает значение, указывающее успех/неудачу *Имя операции* *Параметры, например указатель на интерфейс*

5-32

COM: HRESULTS

- HRESULTS – 32-битные целые
- Результаты выполнения операций возвращаются через входные параметры
- Структурированы в виде 4-х полей

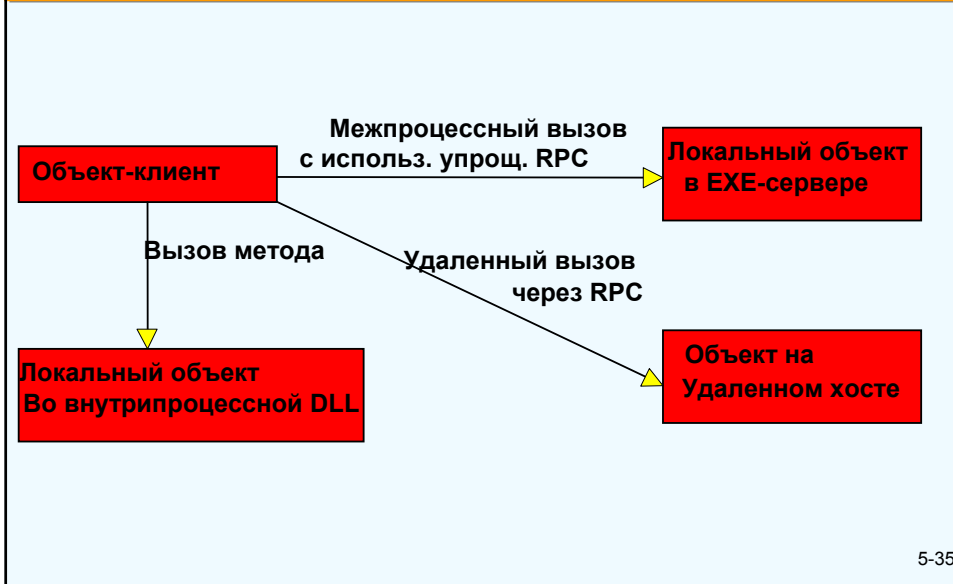


Выполнение операции COM

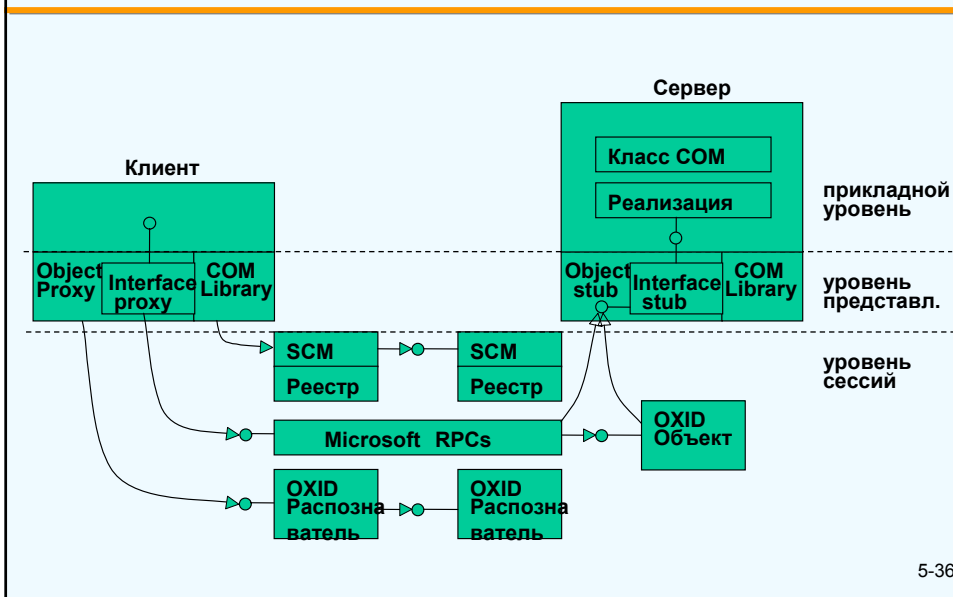
- Выполнение определяется клиентскими объектами
 - Выполнение определяет:
 - Указатель на интерфейс серверного объекта
 - Имя вызываемой операции
- Выполнение проводится синхронно
- Выполнение может быть:
 - Статическим
 - Динамическим (через интерфейс IDispatch)
- Клиенты должны проверять HRESULTS

5-34

Три варианта реализации запроса



Архитектура распределенного COM



Литература / Internet источники

- В. Эммерих *Конструирование распределенных объектов*. - М.:Мир. - 2002.
- М.Р. Когаловский *Энциклопедия технологий баз данных*. - М.: ФС. - 2002.
- Ю.А. Григорьев, А.Д. Плутенко. - *Жизненный цикл проектирования распределенных баз данных*. - Благовещенск. - 1999.
- Э. Таненбаум, М. Ван Стеен *Распределенные системы. Принципы и парадигмы*. – СПб.: Питер, 2003. – 877 с.: ил. – (Серия «Классика computer science»)
- www.omg.org

5-37