

Распределенные информационные системы

Технология DDS

IDL

Вопросы

- Введение:
 - Идея IDL
 - Основные реализации
- OMG IDL
 - Базовые типы данных

Общее понятие

Реализации

ВВЕДЕНИЕ

Язык описания интерфейсов – зачем? – 1/2

- Одним из способов абстрактного представления РИС является восприятие ее как некой «распределенной программы» (РП)
 - Вариант, облегчающий восприятие РП – парадигма удаленного вызова процедур (RPC)
 - ~~Люди – меркантильные ленивые св...~~
Разработчикам требуется снизить затраты времени на построение связей между локальным и удаленным представлением процедур
 - Создаются автоматические генераторы заглушек / прокси

IDL – зачем? – 2/2

- Необходим [независимый от ЯП и платформы] способ описания функций (интерфейсов доступных удаленно классов)
- Создают язык описания интерфейсов (Interface Description / Definition Language – IDL)

Примечание: Л – Логика!

IDL: общие замечания

- IDL имеет множество реализаций, чьи выразительные свойства соответствуют конкретным технологиям RPC
- Синтаксис конкретной реализации IDL близок «обычной» для технологии, в которой используется. Основными вариантами являются
 - Синтаксис схожий с C++
 - XML документ

IDL: Реализации идеи – 1/2

- Наиболее интересные реализации IDL:
 - IDL specification language:
 - Один из первых / первый IDL
 - Описан в 1983 году
 - Использовался в системах компиляции языка Ада
 - Microsoft IDL (MIDL)
 - Описывает процедуры COM/DCOM
 - COM IDL
 - Описывает интерфейсы между модулями COM
 - Используется в технологии DCE/RPC (Distributed Computing Environment / Remote Procedure Calls) подсистеме RPC разработанной для DCE (Distributed Computing Environment – среда распределённых вычислений)
 - Поддерживается The Open Group

IDL: Реализации идеи – 2/2

- Наиболее интересные реализации IDL:
 - AIDL
 - Платформа Android
 - ЯП Java
 - Поддерживает локальные и удаленные вызовы процедур
 - Доступна через JNI (Java Native Interface)
 - Web Services Description Language (WSDL)
 - Описывает RPC доступный через веб (т.е. по протоколу HTTP)
 - Open API, RSDL (RESTful Service Description Language)
 - Описывают REST сервисы
 - OMG IDL
 - Стандарт OMG
 - Используется технологиями CORBA (для описания интерфейсов) и DDS (для описания структур данных)

Профили DDS в OMG IDL

Базовые типы данных

OMG IDL

Профили DDS в OMG IDL

- Стандарт OMG IDL включает следующие подмножества описаний, касающиеся DDS:
 - Plain DDS Profile
 - Описывает минимальный перечень того, что поддерживается DDS
 - Включает описание **типов данных**
 - Extensible DDS Profile
 - Дополняет Plain DDS Profile форматами описаний, необходимыми для использования расширяемых и динамических топиков
 - RPC over DDS Profile
 - Задаёт форматы описаний, позволяющие описывать сервисы доступные как RPC over DDS

Типы данных

- Типы данных IDL могут быть простыми и сконструированными (constructed)
- Определения могут ссылаться друг на друга:
 - Прямо адресовать простые типы данных
 - Использовать имена, предварительно описанные в документе
 - Сконструированные
 - Поименованные (`typedef`)
 - Заданные анонимно (без присвоения отдельного имени) указанием размера (ограничения)

Базовые типы данных (Basic Types)

- Базовые типы данных включают:
 - Типы целых чисел
 - Типы чисел с плавающей точкой
 - Типы символов
 - Типы широких символов
 - Логический тип
 - Тип октет

Базовые целые – 1/2

OMG IDL	C++	Размер	Значения
short	short	16 бит	$-2^{15} \dots 2^{15}-1$
unsigned short	unsigned short	16 бит	$0 \dots 2^{16}-1$
long	long	32 бит	$-2^{31} \dots 2^{31}-1$
unsigned long	unsigned long	32 бит	$0 \dots 2^{32}-1$
long long	long long	≥ 64 бит	$-2^{63} \dots 2^{63}-1$
unsigned long long	unsigned long long	≥ 64 бит	$0 \dots 2^{64}-1$

Базовые целые – 2/2

- Однобитного целого нет (см. char и octet)
- Применяется тот тип данных платформы, который точно вмещает указанный диапазон значений
 - Например: Если [на платформе] для short нет отдельного 16-битного целого типа, то используется следующий по размеру
- long – 32-битный тип данных
- long long – 64-битное целое
 - Если платформа не поддерживает тип, выдается ошибка компиляции

Базовые с плавающей точкой – 1/2

OMG IDL	C++	Размер	Значения
float	float	32 bits	IEEE single-precision floating point numbers
double	double	64 bits	IEEE double-precision floating point numbers
long double	long double (?)	≥ 79 бит	IEEE double-extended floating point number: - целое со знаком ≥64 бит - степень ≥15 бит

Базовые с плавающей точкой – 2/2

- long double: Может отражаться на:
 - 80-битный формат повышенной точности (80-bit extended format)
 - 128-битный формат Quadruple precision
 - Если платформа не поддерживает, выдается ошибка компиляции
 - NB! При описании типов с плавающей точкой стандарт OMG IDL ссылается на IEEE 754-1985
 - Был заменен → IEEE 754-2008 → IEEE 754-2019
 - 80-битный формат соответствует понятию «Extended and extendable precision» из актуальной версии стандарта

Базовые символьные – 1/2

OMG IDL	C++	Размер	Значения
char	unsigned char	8 бит	Любой символ из однобайтной кодировки
wchar	wchar	Зависит от платформы	Произвольные кодовые наборы

Базовые символьные – 2/2

- `char`: 8-бит, представляющие:
 - Один символ из любой однобитной кодировки (NB! Сама кодировка не указывается → проблемы с кросс-платформенностью)
 - Если используется как элемент массива, то задает символ из [возможно] многобайтной кодировки
- `wchar`: задает широкий символ из произвольной кодировки. Размер зависит от платформы

Базовые: октет и логический – 1/2

OMG IDL	C++	Размер	Значения
octet	unsigned char	≥ 8 bits	0x0 ... 0xff
boolean	bool	unspecified	{TRUE FALSE}

Базовые: октет и логический – 2/2

- `octet`:
 - Предназначен для передачи бинарных данных между системами (использование `char` недопустимо!)
 - В C++ нет отдельного типа данных для 8-битного целого
- `boolean`:
 - OMG IDL: представляет собой перечислимый тип с двумя значениями `true` и `false`
 - C++: `bool` строится поверх целого числа и ненулевое часто используется как код ошибки при возврате функции

IDL types – параметризованные

- Параметризованные типы состоят из определения шаблона, параметризованого
 - Определяется во время выполнения.
Технически, содержит описание [назначенного] типа (TypeCode) и значение
 - `std::any` появился только в спецификации C++17

Built-in IDL types – другие – 2/4

OMG IDL	C++	Размер	Значения
string	std::string (?)	Переменный	ISO Latin-1, except NUL
string<bound>	std::string (??)	Переменный, не более <bound>	ISO Latin-1, except NUL
any	std::any	Переменный	Универсальный контейнер

Built-in IDL types – замечания – 4/4

- any
 - Определяется во время выполнения.
Технически, содержит описание [назначенного] типа (TypeCode) и значение
 - `std::any` появился только в спецификации C++17

Extended built-in IDL types – 1/

OMG IDL	C++	Размер	Значения
wstring	std::wstring	Переменный	Произвольные кодовые наборы
fixed < digit-size, scale >	NA	≥ 31 значащих цифр	Число с фиксированной точкой