НАЦИОНАЛЬНЫЙ СТАНДАРТ РОССИЙСКОЙ ФЕДЕРАЦИИ

## Информационная технология

# ОТКРЫТАЯ РАСПРЕДЕЛЕННАЯ ОБРАБОТКА БАЗОВАЯ МОДЕЛЬ

## Часть 1

## Основные положения

Издание официальное

БЗ 4—2002/60

ГОССТАНДАРТ РОССИИ
Москва

gost.e-ecolog.ru

Предисловие

1 РАЗРАБОТАН Государственным научно-исследовательским и конструкторско-технологическим институтом «ТЕСТ» Министерства Российской Федерации по связи и информатизации

ВНЕСЕН Министерством Российской Федерации по связи и информатизации

2 УТВЕРЖДЕН И ВВЕДЕН В ДЕЙСТВИЕ Постановлением Госстандарта России от 4 февраля 2004 г. № 51-ст

3 Настоящий стандарт идентичен международному стандарту ИСО/МЭК 10746-1—98 «Информационная технология. Открытая распределенная обработка. Базовая модель. Часть 1. Основные положения»

4 ВВЕДЕН ВПЕРВЫЕ

© ИПК Издательство стандартов, 2004

Настоящий стандарт не может быть полностью или частично воспроизведен, тиражирован и распространен в качестве официального издания без разрешения Госстандарта России

Содержание

1 Область применения . . . 1
2 Нормативные ссылки . . . 1
3 Определения . . . 2
3.1 Определения настоящего стандарта . . . 2
3.2 Определения из других стандартов . . . 2
4 Сокращения . . . 5
5 Соглашения . . . 5
6 Стандартизация ОРО . . . 6
6.1 Цели и причины . . . 6
6.2 Реализация . . . 7
6.2.1 Моделирование объектов . . . 7
6.2.2 Спецификации точек зрения . . . 7
6.2.3 Прозрачность распределения . . . 8
6.2.4 Соответствие . . . 8
6.3 Стандарты . . . 8
6.3.1 Базовая модель . . . 8
6.3.2 Специфические стандарты . . . 9
7 Основные положения . . . 9
7.1 Базовые понятия моделирования . . . 10
7.1.1 Объекты . . . 10
7.1.2 Интерфейсы и точки взаимодействия . . . 10
7.1.3 Поведение и состояние . . . 11
7.2 Понятия спецификаций . . . 11
7.2.1 Композиция и декомпозиция . . . 11
7.2.2 Поведенческая совместимость . . . 12
7.2.3 Тип и класс . . . 12
7.2.4 Шаблоны . . . 12
7.2.5 Роли . . . 12
7.2.6 Базовые и производные классы . . . 13
7.3 Организационные понятия . . . 13
7.3.1 Группы и области . . . 13
7.3.2 Наименование . . . 13
7.3.3 Контракт . . . 14
7.3.4 Взаимодействие и связывание . . . 14
8 Архитектура . . . 14
8.1 Основы архитектуры . . . 14
8.1.1 Точки зрения . . . 15
8.1.2 Прозрачности распределения . . . 15
8.2 Предпринимательский язык . . . 16
8.3 Информационный язык . . . 18
8.4 Вычислительный язык . . . 19
8.4.1 Вычислительные интерфейсы . . . 21
8.4.2 Модель связывания . . . 21
8.4.3 Типы и подтипы вычислительных интерфейсов . . . 23
8.4.4 Переносимость . . . 24
8.5 Инженерный язык . . . 24
8.5.1 Кластеры, капсулы и узлы . . . 24
8.5.2 Каналы . . . 26
8.5.3 Указатель интерфейса . . . 28
8.5.4 Связывание . . . 28
8.5.5 Установление канала . . . 29
8.5.6 Интерфейсы административного управления . . . 30
8.5.7 Пересечения . . . 30
8.5.8 Точки соответствия . . . 31

8.6 Технологический язык . . . 31
8.7 Согласованность между точками зрения . . . 32
8.7.1 Согласованность предпринимательской точки зрения с другими . . . 33
8.7.2 Соответствия между вычислительной и инженерной спецификациями . . . 34
8.8 Функции ОРО . . . 37
8.8.1 Функции управления . . . 37
8.8.2 Функции координации . . . 37
8.8.3 Функции хранилища . . . 38
8.8.4 Функции безопасности . . . 38
8.9 Прозрачности распределения ОРО . . . 39
8.9.1 Прозрачность доступа . . . 39
8.9.2 Прозрачность отказа . . . 39
8.9.3 Прозрачность положения . . . 39
8.9.4 Прозрачность миграции . . . 39
8.9.5 Прозрачность постоянства . . . 39
8.9.6 Прозрачность перемещения . . . 40
8.9.7 Прозрачность дублирования . . . 40
8.9.8 Прозрачность транзакции . . . 40
9 Оценка соответствия . . . 40
9.1 Оценка соответствия и процесс разработки . . . 40
9.2 Оценка соответствия — рассматриваемые взаимосвязи . . . 41
9.3 Точки соответствия и относящиеся к ним понятия . . . 41
9.4 Спецификации соответствия ОРО . . . 42
9.4.1 Уровень абстракции . . . 42
9.4.2 Использование кратных опорных точек . . . 42
9.5 Влияние языков точек зрения на соответствие . . . 43
9.6 Деятельность по оценке соответствия . . . 43
10 Административное управление системами ОРО . . . 43
10.1 Области административного управления . . . 44
10.2 Политика административного управления . . . 44
10.3 Моделирующие структуры административного управления . . . 45
11 Использование стандартов в системах ОРО . . . 45
11.1 Предпринимательская точка зрения . . . 45
11.1.1 Предпринимательская спецификация . . . 45
11.1.2 Применение стандартов . . . 46
11.2 Информационная точка зрения . . . 47
11.2.1 Информационная спецификация . . . 47
11.2.2 Применение стандартов . . . 47
11.3 Вычислительная точка зрения . . . 47
11.3.1 Вычислительная спецификация . . . 47
11.3.2 Применение стандартов . . . 48
11.4 Инженерная точка зрения . . . 48
11.4.1 Инженерная спецификация . . . 48
11.4.2 Применение стандартов . . . 49
11.5 Технологическая точка зрения . . . 50
11.5.1 Технологическая спецификация . . . 50
11.5.2 Применение стандартов . . . 50
12 Примеры спецификаций ОРО . . . 51
12.1 Мультимедийная система конференций . . . 51
12.1.1 Введение . . . 51
12.1.2 Предпринимательская спецификация . . . 52
12.1.3 Информационная спецификация . . . 53
12.1.4 Вычислительная спецификация . . . 55
12.1.5 Инженерная спецификация . . . 57
12.1.6 Технологическая спецификация . . . 58

12.2 Многостороннее связывание аудио/видеопотока . . . . . 58
12.2.1 Общее описание . . . . . 59
12.2.2 Предпринимательская спецификация . . . . . 59
12.2.3 Информационная спецификация . . . . . 62
12.2.4 Вычислительная спецификация . . . . . 64
12.2.5 Инженерная спецификация . . . . . 65
12.2.6 Технологическая спецификация . . . . . 68
12.3 Пример административного управления — метрический объект . . . . . 68
12.3.1 Предпринимательская спецификация . . . . . 68
12.3.2 Информационная спецификация . . . . . 70
12.3.3 Вычислительная спецификация . . . . . 71
12.4 Пример базы данных . . . . . 72
12.4.1 Предпринимательская спецификация . . . . . 72
12.4.2 Информационная спецификация . . . . . 72
12.4.3 Вычислительная спецификация . . . . . 72
Приложение А Библиография . . . . . 75

НАЦИОНАЛЬНЫЙ СТАНДАРТ РОССИЙСКОЙ ФЕДЕРАЦИИ

**Информационная технология**

# ОТКРЫТАЯ РАСПРЕДЕЛЕННАЯ ОБРАБОТКА БАЗОВАЯ МОДЕЛЬ

**Часть 1**

**Основные положения**

Information technology. Open Distributed Processing. Reference Model. Part 1. Overview

**Дата введения 2005—01—01**

## 1 Область применения

Настоящий стандарт содержит:

- введение и обоснование открытой распределенной обработки (ОРО),
- обзор базовой модели открытой распределенной обработки (БМ-ОРО) и объяснение ее ключевых понятий,
- руководство по применению БМ-ОРО.

Настоящий стандарт содержит общий обзор и подробное объяснение ОРО и может использоваться различными способами в сочетании с другими стандартами данной серии:

а) при ограничении только настоящим стандартом для понимания значения ОРО для вашей организации следует прежде всего обратить внимание на раздел 6;

б) если вы намерены полностью изучить БМ-ОРО, то вам так же следует ознакомиться с разделом 6 до перехода к ГОСТ Р ИСО/МЭК 10746-2 и ГОСТ Р ИСО/МЭК 10746-3;

в) при использовании ГОСТ Р ИСО/МЭК 10746-2 и ГОСТ Р ИСО/МЭК 10746-3 следует пользоваться разделами 7—10 настоящего стандарта, где даны разъяснения различных понятий;

г) после изучения ГОСТ Р ИСО/МЭК 10746-2 и ГОСТ Р ИСО/МЭК 10746-3 следует ознакомиться с разделами 11 и 12 настоящего стандарта, где обсуждается их использование в спецификациях систем ОРО и приведены некоторые примеры применения понятий ОРО для спецификации систем.

## 2 Нормативные ссылки

В настоящем стандарте использованы ссылки на следующие стандарты:

ГОСТ Р ИСО/МЭК 7498-1—97 Информационная технология. Взаимосвязь открытых систем. Базовая эталонная модель. Часть 1. Базовая модель

ГОСТ Р ИСО/МЭК 9646-1—93 Информационная технология. Взаимосвязь открытых систем. Методология и основы аттестационного тестирования для ВОС. Часть 1. Общие принципы

ГОСТ Р ИСО/МЭК 10746-2—2000 Информационная технология. Открытая распределенная обработка. Базовая модель. Часть 2. Модель

ГОСТ Р ИСО/МЭК 10746-3—2001 Информационная технология. Открытая распределенная обработка. Базовая модель. Часть 3. Архитектура

ГОСТ Р ИСО/МЭК 10746-4—2004 Информационная технология. Открытая распределенная обработка. Базовая модель. Часть 4. Архитектурная семантика

ИСО/МЭК 10164-11—94* Информационная технология. Взаимосвязь открытых систем. Административное управление системами. Часть 11. Метрические объекты и атрибуты

* Международный стандарт — во ВНИИКИ Госстандарта России.

**Издание официальное**

## 3 Определения

**3.1 Определения настоящего стандарта**

В настоящий стандарт не введены новые определения.

**3.2 Определения из других стандартов**

В настоящем стандарте используют следующие термины, определенные в ГОСТ Р ИСО/МЭК 10746-2:

- **абстракция;**
- **действие;**
- **шаблон действия;**
- **деятельность;**
- **архитектура;**
- **элементарный(ая);**
- **базовый класс;**
- **поведение (объекта);**
- **поведенческая совместимость;**
- **связывание;**
- **связывающее поведение;**
- **цепочка (действий);**
- **класс;**
- **объект-клиент;**
- **коммуникация;**
- **согласованность;**
- **составной объект;**
- **композиция;**
- **конфигурация (объектов);**
- **точки соответствия;**
- **объект-потребитель;**
- **контракт;**
- **контрактный контекст;**
- **создание;**
- **данные;**
- **декомпозиция;**
- **удаление;**
- **производный класс;**
- **прозрачность распределения;**
- **категория;**
- **среда (объекта);**
- **контракт среды;**
- **ошибка;**
- **устанавливающее поведение;**
- **отказ;**
- **неисправность;**
- **идентификатор;**
- **информация;**
- **инициирующий объект;**
- **экземпляр;**
- **реализация;**
- **взаимодействие;**
- **интерфейс;**
- **сигнатура интерфейса;**
- **внутреннее действие;**
- **опорная точка взаимодействия;**
- **введение (<X>);**
- **инвариант;**

- **положение в пространстве;**
- **информация (административного) управления;**
- **имя;**
- **разрешение имени;**
- **область наименования;**
- **уведомление;**
- **объект;**
- **обязательство;**
- **стандарт ОРО;**
- **система ОРО;**
- **воспринимаемая опорная точка;**
- **разрешение;**
- **постоянство;**
- **политика;**
- **переносимость;**
- **объект-создатель;**
- **программируемая опорная точка;**
- **запрещение;**
- **качество услуги;**
- **опорная точка;**
- **уточнение;**
- **отвечающий объект;**
- **роль;**
- **объект-сервер;**
- **состояние;**
- **подкласс;**
- **подтип;**
- **система;**
- **шаблон класса;**
- **завершающее поведение;**
- **связка;**
- **торг;**
- **тип;**
- **развязывание;**
- **точка зрения.**

В настоящем стандарте используют следующие термины, определенные в ГОСТ Р ИСО/МЭК 10746-3:

- **язык <точки зрения>;**
- **информация управления доступом;**
- **прозрачность доступа;**
- **объявление;**
- **базовый инженерный объект;**
- **связник;**
- **связывающий объект;**
- **капсула;**
- **менеджер капсулы;**
- **канал;**
- **контрольная точка;**
- **взятие контрольной точки;**
- **менеджер кластера;**
- **шаблон кластера;**
- **коммуникационный интерфейс;**
- **общность;**
- **составное связывающее действие;**
- **вычислительная точка зрения;**
- **деактивация;**

- **динамическая схема;**
- **инженерная точка зрения;**
- **предпринимательская точка зрения;**
- **явное связывание;**
- **прозрачность отказа;**
- **федерация;**
- **поток;**
- **сокрытие;**
- **неявное связывание;**
- **информационная точка зрения;**
- **пересечение;**
- **опрос;**
- **инвариантная схема;**
- **вызов;**
- **прозрачность размещения;**
- **миграция;**
- **прозрачность миграции;**
- **узел;**
- **ядро;**
- **функция ОРО;**
- **операционный интерфейс;**
- **сигнатура операционного интерфейса;**
- **прозрачность постоянства;**
- **элементарное связывающее действие;**
- **протокольный объект;**
- **реактивация;**
- **восстановление;**
- **прозрачность перемещения;**
- **релокатор;**
- **схема дублирования;**
- **прозрачность дублирования;**
- **уполномоченный по безопасности;**
- **область безопасности;**
- **сигнал;**
- **интерфейс сигнала;**
- **сигнатура интерфейса сигнала;**
- **статическая схема;**
- **интерфейс потока;**
- **сигнатура интерфейса потока;**
- **заглушка;**
- **цель;**
- **технологическая точка зрения;**
- **завершение;**
- **прозрачность транзакции;**
- **подтверждение.**

В настоящем стандарте используют следующие термины, определенные в ГОСТ Р ИСО/МЭК 9646-1:

- **заявка о соответствии реализации;**
- **дополнительная информация о реализации для тестирования;**
- **пункт контроля и наблюдения.**

В настоящем стандарте используют следующие термины, определенные в ГОСТ Р ИСО/МЭК 7498-1:

- **открытая система;**
- **абстрактный синтаксис;**
- **синтаксис передачи.**

## 4 Сокращения

В настоящем стандарте использованы следующие сокращения:
АВУ — архитектура верхних уровней;
АТИС — архитектура телекоммуникационной информационной сети;
БИО — базовый инженерный объект;
БМ-ОРО — базовая модель открытой распределенной обработки;
ВОС — взаимосвязь открытых систем;
ВПК — вызов прикладной категории;
ВПП — вызов прикладного процесса;
ГИП — графический интерфейс пользователя;
ГУО — группа (административного) управления объектами;
ДИТР — дополнительная информация для тестирования реализации;
ЗСР — заявка о соответствии реализации;
ИТ — информационная технология;
ИЧК — интерфейс человек—компьютер;
КУ — качество услуги;
МИУ — модель информации (административного) управления;
ММО — метод моделирования объектов;
ММСК — мультимедийная система конференций;
МФО — методы формального описания;
ОПУ — объект прикладной услуги;
ОРО — открытая распределенная обработка;
П-профиль — прикладной профиль;
ПД — период дробления;
ПК — прикладная категория;
ПКН — пункт контроля и наблюдения;
ПОИУ — протокол общей информации (административного) управления;
ПП — прикладной процесс;
ППИ — прикладной программный интерфейс;
СОС — среда открытой системы;
СПУ — структура прикладного уровня;
Т-профиль — транспортный профиль;
УВП — удаленный вызов процедуры;
УДБД — удаленный доступ к базе данных;
УОИУ — услуга общей информации (административного) управления;
ФОП — Фонд открытого программного обеспечения;
Ф-профиль — профиль формата и представления;
ЯО — язык определений;
ЯОИ — язык определения интерфейсов;
ACID — Atomicity Consistency Isolation Durability (элементарная согласованная продолжительность изоляции);
CAD — Computer Aided Design (проектирование, основанное на компьютере);
CD — Compack Disk (компактный диск).

## 5 Соглашения

В настоящем стандарте используют следующие соглашения.
1) Первое использование в разделах 7 и 8 формальных терминов, определенных в ГОСТ Р ИСО/МЭК 10746-2 и ГОСТ Р ИСО/МЭК 10746-3, выделено курсивом.
2) В примере раздела 12 использованы графические соглашения ММО по Rumbaugh 91[1].
3) На диаграммах:
- объекты изображены как овалы и кружки;
- символом « ⊥ » около объекта изображен интерфейс.

## 6 Стандартизация ОРО

### 6.1 Цели и причины

Целью стандартизации ОРО является разработка стандартов, позволяющих реализовать преимущества услуг распределенной обработки информации в среде неоднородных ресурсов ИТ и нескольких организационных областях. Эти стандарты направлены на ограничение спецификаций систем и обеспечение для них инфраструктуры, которая снимает трудности, унаследованные от проектирования и программирования распределенных систем.

Распределенные системы важны вследствие растущей потребности взаимодействия систем обработки информации. Эта необходимость возникает в связи с такими организационными тенденциями, как, например, уменьшение размеров организаций, что требует информационного обмена как между группами внутри организации, так и между взаимодействующими организациями. Достижения технологии делают такой обмен возможным, отвечая на новые требования развитием сетей информационных услуг и персональных рабочих станций и допуская конструкции приложений, распределенных по большой конфигурации взаимосвязанных систем.

Для того чтобы управлять распределенной системой и эксплуатировать ее (например, использовать ее потенциал для оптимизации доступности, производительности, надежности и стоимости), организации должны иметь дело с рядом ключевых характеристик распределения системы, таких как:

- **удаленность.** Компоненты распределенной системы могут быть широко распределены в пространстве; взаимодействие может быть локальным или удаленным;
- **конкуренция.** Любой компонент распределенной системы может работать параллельно с любыми другими компонентами;
- **отсутствие глобального состояния.** Глобальное состояние распределенной системы не может быть точно определено;
- **частичные отказы.** Любой компонент распределенной системы может отказать независимо от любых других компонентов;
- **асинхронность.** Коммуникационная деятельность и обработка не управляются едиными глобальными часами. Нельзя считать, что взаимосвязанные изменения в распределенной системе происходят как единый акт;
- **неоднородность.** Нет гарантий того, что компоненты распределенной системы построены по единой технологии, а набор различных технологий определенно будет меняться со временем. Неоднородность проявляется в разных местах: в технических средствах, операционных системах, коммуникационных сетях и протоколах, языках программирования, приложениях и т. п.;
- **автономность.** Распределенная система может быть поделена между несколькими автономными уполномоченными по административному управлению и контролю, без единого центра контроля. Степень автономности определяет, насколько ресурсы обработки и соответствующие устройства (принтеры, запоминающие устройства, графические дисплеи, аудио устройства и т. п.) находятся под контролем отдельных организационных единиц;
- **эволюция.** В течение своего жизненного цикла распределенная система столкнется со многими изменениями, которые мотивируются техническим прогрессом, обеспечивающим лучшую производительность за лучшую цену, стратегическими решениями о новых целях, новыми типами приложений;
- **мобильность.** Источники информации, узлы обработки и пользователи могут перемещаться физически. Программы и данные также могут перемещаться по узлам, например для борьбы с физическими перемещениями или для оптимизации производительности.

Построение таких систем является непростой задачей. Ее решение требует определенной архитектуры и, так как единственное инженерное решение не может удовлетворить все требования, это должна быть гибкая архитектура. Более того, так как единственный поставщик не имеет ответов на все вопросы, то существенно, что архитектура и любые функции, необходимые для реализации архитектуры, определяются набором стандартов, и несколько поставщиков могут сотрудничать при обеспечении распределенных систем. Такие стандарты позволяют построить систему, которая:

- является **открытой,** обеспечивающей как переносимость (выполнение компонентов системы в разных узлах обработки без модификации), так и взаимодействие (осмысленное взаимодействие между компонентами, возможно, размещенными в разных системах);

- является **интегрированной,** включающей в себя различные системы и ресурсы в целом, без дорогостоящих доработок. В их число могут входить системы с различными архитектурами и различные ресурсы с разными возможностями. Интеграция помогает иметь дело с неоднородностью;

- является **гибкой,** способной как эволюционировать, так и приспосабливать существующие и продолжающиеся операции унаследованных систем. Открытая распределенная система должна быть в состоянии столкнуться с изменениями во времени, например она должна допускать динамическую переконфигурацию для приспособления к изменившимся обстоятельствам. Гибкость способствует мобильности;

- является **модульной,** допускающей автономность частей системы при сохранении их взаимосвязанности. Модульность является основой для гибкости;

- может быть **включена в федерацию,** что позволяет для достижения единой цели скомбинировать систему с другими системами из разных технических или административных областей;

- является **управляемой,** допускающей мониторинг, контроль и административное управление ресурсами системы для обеспечения конфигурации, КУ и политики учета;

- удовлетворяет требуемому **КУ,** обеспечивая, например, доступность и надежность в контексте удаленных ресурсов и взаимодействий и, одновременно, устойчивость к отказам, позволяя остальной части распределенной системы сохранять работоспособность при отказе некоторой ее части. Обеспечение устойчивости к отказам (и независимости в целом) необходимо в больших распределенных системах, где маловероятно, что все части системы всегда будут работать одновременно;

- является **безопасной,** гарантируя, что средства системы и данные защищены от неавторизованного доступа. Требованиям безопасности сложнее удовлетворить из-за удаленности взаимодействий и мобильности частей системы и ее пользователей;

- обеспечивает **прозрачность,** маскируя от приложений детали и различия в методах, которые используются для разрешения проблем, связанных с распределенностью системы. Это — центральное требование, возникающее из необходимости обеспечить конструкцию распределенных приложений. Аспекты распределения, которые должны быть замаскированы (в целом или частично), включают в себя: неоднородность поддерживающего программного и технического обеспечения, размещение и мобильность компонентов, методы достижения требуемого уровня КУ в случае отказов (например, дублирование, миграция, взятие контрольной точки и т. п.).

**6.2 Реализация**

Стандартизация ОРО имеет четыре основных составляющих:

- объекты, моделирующие подход к спецификации системы;

- спецификация системы в терминах отдельных, но взаимосвязанных спецификаций точек зрения;

- определение инфраструктуры системы, обеспечивающей прозрачность распределения для приложений системы;

- основные положения проверки соответствия системы.

6.2.1 Моделирование объектов

Моделирование объектов обеспечивает формализацию проверенной практики проектирования абстракций и инкапсуляций. Абстракция позволяет отделить описание функционирования системы от деталей ее реализации. Инкапсуляция позволяет скрыть от пользователей неоднородность, локализацию отказов, реализацию безопасности и методы предоставления услуг.

Понятия моделирования объектов охватывают:

- основные моделирующие понятия, обеспечивающие строгие определения минимального набора понятий (действие, объект, взаимодействие, интерфейс), которые образуют основу для описаний систем ОРО и применяются для всех точек зрения;

- специфицирующие понятия, относящиеся к таким терминам, как тип и класс, которые необходимы для утверждений о спецификациях и взаимосвязях между ними. Они обеспечивают общие средства проектирования и установления требований к языкам спецификаций;

- структурирующие понятия, построенные на основе моделирующих и специфицирующих понятий; они нацелены на рекуррентные структуры в распределенных системах и охватывают такие понятия, как политика, наименование, поведение, зависимость и коммуникация.

6.2.2 Спецификации точек зрения

Точка зрения (на систему) является абстракцией, которая позволяет связать спецификацию системы в целом с конкретным множеством понятий. Были выбраны пять точек зрения, которые, будучи простыми и полными, охватывают все области проектирования архитектуры. Этими точками зрения являются:

- предпринимательская точка зрения, которая сконцентрирована на целях, сфере применения и политике, управляющей деятельностью специфицируемой системы в организации, частью которой она является;

- информационная точка зрения, которая сконцентрирована на видах информации, обрабатываемой системой, и ограничениях на использование и интерпретацию этой информации;

- вычислительная точка зрения, которая сконцентрирована на функциональной декомпозиции системы на множество объектов, взаимодействующих через интерфейсы; такая декомпозиция нужна для распределения системы;

- инженерная точка зрения, которая сконцентрирована на инфраструктуре, требуемой для поддержки распределения системы;

- технологическая точка зрения, которая сконцентрирована на выборе технологии для поддержки распределения системы.

Для каждой точки зрения имеется соответствующий язык точки зрения, который может использоваться для выражения спецификации системы с этой точки зрения. Понятия моделирования объектов дают общий базис для языков точек зрения и делают возможной идентификацию взаимосвязей между спецификациями различных точек зрения и проверку соответствия между представлениями системы с разных точек зрения.

6.2.3 Прозрачность распределения

Прозрачности распределения позволяют скрыть от приложений сложность, связанную с распределением системы, если она не имеет отношения к задачам приложений. Например:

- прозрачность доступа маскирует различия между системами в представлении данных и методах вызова;

- прозрачность размещения маскирует необходимость приложению иметь информацию о размещении для вызова услуги;

- прозрачность перемещения маскирует от приложения перемещение услуги;

- прозрачность дублирования маскирует тот факт, что для обеспечения надежности и доступности может существовать несколько копий услуги.

Стандарты ОРО определяют функции и структуры для реализации прозрачностей распределения. Однако во многих случаях, по соображениям реализуемости и стоимости, будут использованы только некоторые выбранные виды прозрачности. Таким образом, соответствующая система ОРО должна реализовывать те прозрачности, которые она поддерживает в соответствии с нужными стандартами, но не обязана обеспечивать все прозрачности.

6.2.4 Соответствие

Основные характеристики неоднородности и эволюция системы подразумевают, что отдельные части распределенной системы могут быть приобретены по отдельности от разных поставщиков. Следовательно, для удовлетворения спецификациям системы важно, чтобы поведение разных частей системы было четко определено и чтобы было возможно установить ответственность за каждый отказ.

На эти вопросы ориентированы основные положения, определенные для оценки соответствия. Эти положения охватывают:

- идентификацию в наборе спецификаций точек соответствия, в которых может быть проведено наблюдение соответствия;

- определение классов точек соответствия;

- спецификация характера заявлений о соответствии, которые могут быть сделаны с каждой из точек зрения, и взаимосвязь между ними.

**6.3 Стандарты**

6.3.1 Базовая модель

БМ-ОРО обеспечивает основные положения для стандартизации ОРО. Она состоит из двух основных частей:

- ГОСТ Р ИСО/МЭК 10746-2, который определяет понятия и аналитические основы для описания распределенных систем обработки, включая основные положения для оценки соответствия;

- ГОСТ Р ИСО/МЭК 10746-3, который определяет, как специфицируются системы ОРО, и инфраструктуру, обеспечивающую прозрачности распределения.

ГОСТ Р ИСО/МЭК 10746-4 дополняет упомянутые выше части формальной интерпретацией моделирующих понятий и языков точек зрения в терминах существующих методов формального описания.

БМ-ОРО является родовой в том смысле, что не зависит от (а значит, и применима к) сферы применения, использующей или требующей технологии распределительных систем. В некоторых конкретных областях применения будет необходимо уточнить и специализировать БМ-ОРО для соответствия конкретным потребностям, разработав:

- специфичные базовые модели, которые охватывают отдельные виды деятельности, используя понятия и общие функции, определенные в БМ-ОРО, и определив дополнительные концептуальные детали и специфические функции; примером является архитектура телекоммуникационных информационных сетей (АТИС);

- стандарты для реализации специфических функций, необходимых для конкретных приложений и, возможно, идентифицированных в специфичной базовой модели; примером является интерфейс телефонного вызова.

Будучи родовой, БМ-ОРО позволяет интегрировать технологии распределенных систем в экономически эффективные технические решения для промышленных потребностей. В частности, в случае архитектур, опубликованных Фондом открытого программного обеспечения (ФОП) и Группой административного управления объектами (ГУО), для разъяснения, как специфицированные ими для поддержки распределенных систем функции работают совместно, подход ОРО использует такие понятия, как федерация, прозрачность и административное управление системой и, определяя тонкую структуру опорных точек, обеспечивает интеграцию функций разных источников.

6.3.2 Специфические стандарты

В рамках общего подхода БМ-ОРО идентифицированы следующие четыре категории стандартов:

- дополнительные архитектурные понятия, которые уточняют БМ-ОРО в таких конкретных областях, как наименование, безопасность и оценка соответствия;

- стандарты нотаций, которые определяют обозначения для выражения спецификаций различных аспектов интеграции и распределения системы, а также правила для взаимосвязи различных спецификаций;

- стандарты для компонентов, которые определяют отдельные функции ОРО и наборы тесно связанных функций, которые могут быть реализованы как единое программное или техническое средство;

- стандарты для композиции компонентов, которые определяют скоординированное использование нескольких компонентов для достижения некоторой цели системы в целом, такой как обеспечение конкретной прозрачности.

Примечание — Некоторые стандарты могут специфицировать как компоненты, так и их композицию (в таком случае нужная возможность может быть реализована непосредственно). Другие стандарты могут создавать основу для ряда стандартов на композиции компонентов, например на стандарт для релокатора ОРО могли бы ссылаться стандарты для композиций компонентов, обеспечивающих прозрачность положения или миграции.

БМ-ОРО предоставляет основу для стандартов на компоненты и их композиции, обеспечивающие функции ОРО, которые допускают различные подходы к их реализации. Такая гибкость необходима, если основные понятия должны иметь приемлемое время жизни, включая в себя новые разработки по мере их появления. Таким образом, может существовать специфический стандарт или набор стандартов, устанавливающий одно конкретное решение для обеспечения некоторого требования ОРО путем конкретного выбора, необходимого для реализации открытого продукта, и могут существовать несколько таких стандартов, соответствующих разным проектным решениям. Со временем, по мере появления новых технологий, могут появиться новые поколения стандартов в рамках той же самой БМ-ОРО.

## 7 Основные положения

В ГОСТ Р ИСО/МЭК 10746-2 определен набор понятий моделирования, которые обеспечивают основу для описания *архитектуры систем ОРО,* установленной в ГОСТ Р ИСО/МЭК 10746-3. Эти понятия распадаются на три категории:

- базовые понятия моделирования, которые вводят общую объектно-ориентированную модель. В общем случае система ОРО может быть описана как совокупность связанных, взаимодействующих *объектов;*

- понятия спецификаций, которые не являются внутренними для распределенных систем, но позволяют пользователю описывать и обсуждать спецификации систем ОРО. Они устанавливают требования к языку спецификаций, который используется для спецификации систем ОРО, а так как эти требования не зависят от языка, то они могут быть применены к любому языку спецификаций или программирования;

- организационные понятия, охватывающие организацию и свойства систем и объектов, *политику, наименование, поведение* и управление, которые соответствуют представлениям и структурам, применяемым в общем случае для проектирования и описания распределенных систем.

ГОСТ Р ИСО/МЭК 10746-2 обеспечивает также общую структуру для понимания *соответствия* в системах ОРО, выраженного в терминах общей объектной модели. Эта структура обсуждается в разделе 9.

**7.1 Базовые понятия моделирования**

7.1.1 Объекты

Спецификации систем ОРО формулируются в терминах объектов. Объект является представлением любой *категории* реального мира. Он содержит *информацию* и предоставляет услуги. Система строится из взаимодействующих объектов. Объект характеризуется тем, что отличает его от других объектов, а так же *инкапсуляцией, абстракцией* и поведением.

Инкапсуляция является свойством, которое состоит в том, что информация, содержащаяся в объекте, доступна только через *взаимодействия* в *интерфейсах,* поддерживаемых объектом. Так как объекты инкапсулированы, то взаимодействия не имеют скрытых эффектов. Это означает, что взаимодействие с одним объектом не может повлиять на *состояние* другого без какого-либо вторичного взаимодействия с этим объектом. Таким образом, любое изменение состояния объекта может произойти в результате *внутреннего действия* объекта или взаимодействия объекта со *средой.*

Абстракция подразумевает, что подробности внутреннего строения объекта скрыты от других объектов; она имеет существенное значение для работы в неоднородной среде, допуская, что различные услуги обеспечиваются разными путями, используя разные методы и технологии, допуская *переносимость* и возможность взаимодействия.

Абстракция также образует жесткое разделение объектов, позволяющее заменять и модифицировать их без изменения их среды при условии, что они продолжают обеспечивать услуги, ожидаемые их средой (т.е. они являются обратно совместимыми). Такой подход к расширению важен в больших, неоднородных, распределенных средах, которые по своей природе являются непрерывно эволюционирующими. Объектная модель обеспечивает модульность и возможность построения новых модулей из существующих: эти возможности существенны для построения гибких систем и поощрения повторного использования элементов для повышения производительности.

Объектная модель ОРО является общей и содержит минимальное количество допущений. Например:

- объекты могут быть произвольной сложности (например, большими, как телефонная сеть, или малыми, как целое число);

- объекты могут демонстрировать произвольное (инкапсулированное) поведение и иметь любой уровень внутреннего параллелизма;

- взаимодействия между объектами не ограничиваются и могут включать в себя, например, асинхронные и многоканальные синхронные взаимодействия.

7.1.2 Интерфейсы и точки взаимодействия

Объекты могут взаимодействовать только через интерфейсы, а интерфейс представляет часть поведения объекта, относящуюся к конкретному подмножеству его возможных взаимодействий. Каждый интерфейс идентифицируется набором взаимодействий, в которых может участвовать объект. Эти взаимодействия не обязательно происходят с другими объектами: объект может взаимодействовать сам с собой. Важной характеристикой понятия «объект» в ОРО является то, что объект может иметь несколько интерфейсов. Причиной рассмотрения кратных интерфейсов является функциональное разделение и распределение. Функциональное разделение может быть понято, например, в контексте административного управления системы, где управляющие и неуправляющие взаимодействия обычно разделены по разным интерфейсам. При работе с распределенными объектами разделение необходимо, когда интерфейсы образуют точки доступа к объекту, которые расположены в различных точках пространства.

Как следствие определения нескольких интерфейсов для объекта, на взаимодействие через любой из интерфейсов могут влиять взаимодействия через другие интерфейсы, которые не обязательно определены по отдельности.

Интерфейс существует в *точке взаимодействия,* которая в любой момент времени связана с некоторой точкой в пространстве. Несколько интерфейсов могут существовать в данной точке взаимодействия, а точка взаимодействия может быть мобильной. Смысл точек в пространстве и времени (и способ, которым они представляются) зависит от языка, на котором написана спецификация.

7.1.3 Поведение и состояние

Поведение объекта является совокупностью *действий,* в которых может участвовать объект, вместе с набором ограничений на то, когда эти действия могут происходить. Объектная модель не ограничивает вид и характер поведения объекта. Действия могут быть взаимодействиями объекта с его средой или внутренними действиями объекта.

Состояние и поведение являются взаимосвязанными понятиями. Состояние объекта является положением объекта в данный момент, которое определяет будущие потенциальные последовательности действий, в которых этот объект может участвовать. С другой стороны, действия приводят к изменениям состояния и, следовательно, текущее состояние объекта определяется его предшествующим поведением. Конечно, действия объекта, которые фактически будут предприняты, не полностью определяются его текущим состоянием; они зависят от того, какие действия подготовила среда для этого объекта.

**7.2 Понятия спецификаций**

7.2.1 Композиция и декомпозиция

*Композиция* и *декомпозиция* могут быть использованы для построения спецификации распределенной системы в виде набора спецификаций, каждая из которых относится к своему уровню абстракции. Это позволяет разделить спецификацию сложной распределенной системы на спецификации нескольких простых объектов, которые, в свою очередь, могут быть разделены на более низком уровне абстракции.

Процессы композиции и декомпозиции обеспечивают иерархическую спецификацию распределенного приложения. В этой иерархии композиций *классы* объектов на верхних уровнях собираются из *конфигураций* классов объектов-компонентов более низких уровней. Таким образом, композиция является мощным моделирующим понятием, которое позволяет рассматривать подсистему как единый объект высокого уровня.

Например, диаграмма на рисунке 1 показывает композицию двух объектов: *C* и *D*. Для того чтобы образовать композицию *C* с *D*, поведение *C* должно быть определено таким образом, что он надлежащим образом взаимодействует с *D*. Интерфейс между *C* и *D* может быть простым, как единственное взаимодействие (например, передача информации от *C* к *D*), или может быть более сложным поведением (таким, как последовательность взаимодействий, при которых *C* вызывает операции с возвратом результатов). Если *C* и *D* объединяются в один объект, то взаимодействия между *C* и *D* оказываются скрытыми и становятся внутренними действиями составного объекта. Левый интерфейс на рисунке представляет интерфейс составного объекта.

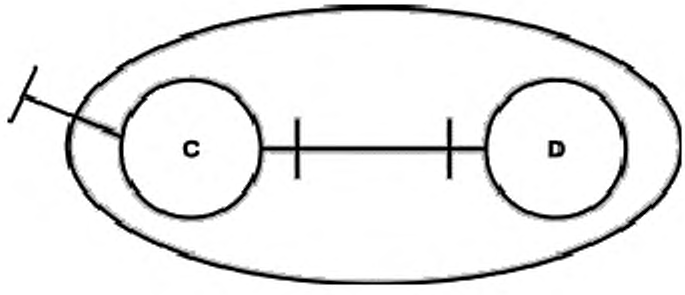



Рисунок 1 — Композиция объектов

Композиция объектов приводит к композиции состояний и поведений и, следовательно, можно говорить о составном поведении и составном состоянии.

Иерархия композиций ортогональна иерархии *подклассов,* обсуждаемой в 7.2.3, и их не следует путать. В общем случае нет связи иерархии подклассов между классами объектов-компонентов и классами составных объектов. Примером является случай, когда составной объект относится к

подклассу класса, к которому относятся объекты-компоненты (в частности, когда объекты представляют коммуникационные услуги, а составной объект добавляет возможности одному из своих компонентов).

7.2.2 Поведенческая совместимость

Говорят, что один объект является поведенчески совместимым с другим в некоторой среде, если первый объект может заменить второй так, что среда не сможет выявить различие. Любая конкретная интерпретация *поведенческой совместимости* будет налагать ограничения на допустимое поведение среды. Общий подход заключается в принятии того, что среда ведет себя как тестер для исходного объекта. Это означает, что среда должна быть в состоянии полностью осуществить исходное поведение, но не более того. Такое предположение является существенным, когда нужно рассмотреть поведенческую совместимость в контексте некоторой неизвестной среды.

7.2.3 Тип и класс

Тип является предикатом (т.е. свойством или множеством свойств) совокупности (объектов, интерфейсов и пр.). Например, «является красным» — это тип. Говорят, что нечто удовлетворяет типу или относится к типу, если для него справедлив предикат. Элементы могут быть весьма различны, но будут удовлетворять одному и тому же типу; для этого они должны демонстрировать свойства, предписанные типом. Например, конкретный флаг, конкретный кирпичный дом и конкретный спортивный автомобиль могут быть красными.

Типы неявно подразделяют элементы на множества, называемые классами; класс является совокупностью элементов со свойствами, предписанными типом.

Понятие типа очень общее и может быть специализировано различными способами. Оно полезно в любом контексте, где необходимо рассматривать или проверять свойства элементов (например, при *торге* или *связывании*).

Понятия типа и класса приводят к естественным иерархиям класс/подкласс и тип/подтип. Различие класс/подкласс соответствует интуитивному различию в теории множеств между множеством и подмножеством. Один класс является подклассом другого только в том случае, если первый является подмножеством последнего. Один тип является подтипом другого, если предикаты первого типа влекут за собой предикаты второго.

Подклассы и подтипы тесно связаны. Для каждого типа имеется соответствующий класс (который может быть пустым). Таким образом, если есть два типа Т1 и Т2, то должны быть и два соответствующих класса С1 и С2. Если Т1 является подтипом Т2, то С1 является подклассом С2.

7.2.4 Шаблоны

*Шаблон* описывает совокупность элементов (объектов, интерфейсов и пр.) достаточно подробно для реализации из него нового элемента.

Когда шаблон описывает множество объектов, то он описывает такие характеристики, как параметры состояния, операции и поведение. Обычно *реализация* объекта вызывает установку начального состояния, например объект «буфер» должен быть создан с пустым содержимым.

Понятие поведенческой совместимости применимо и к шаблонам объектов в том смысле, что имеется поведенческая совместимость между двумя шаблонами объектов, если объекты, реализованные по этим шаблонам, являются поведенчески совместимыми.

*Тип шаблона* является предикатом, определенным в шаблоне. Тип шаблона удовлетворяется всеми реализациями шаблона и в общем случае может удовлетворяться некоторыми другими элементами, если они удовлетворяют тем же требованиям, что и реализации. Например, тип шаблона может быть определен так, что объекты, реализованные из разных шаблонов, но удовлетворяющие этому типу шаблона, демонстрируют поведенческую совместимость.

Каждый тип шаблона приводит к возникновению *класса шаблонов* — множества *экземпляров* типа шаблона. Классы шаблонов могут образовывать иерархии подклассов в соответствии с отношениями тип/подтип между типами шаблонов.

7.2.5 Роли

Роль идентифицирует в шаблоне для составного объекта поведение, которое должно быть связано с одним из объектов-компонентов.

Роль может соответствовать подмножеству полного поведения объекта-компонента. Когда объект рассматривается в терминах роли, то рассматриваются только действия указанного подмножества, абстрагируясь от других действий, возможно относящихся к другим ролям. Объект-компонент может иметь в данное время несколько ролей, в зависимости от его взаимодействий, и может иметь разные роли в разное время. Эти роли могут быть связаны с интерфейсами.

Например, объект может иметь обычное функционирование или порученную роль (т.е. его назначение), а для процесса административного управления может иметь роль административного управления (т.е. поведение, необходимое для мониторинга и управления поведением порученной роли). Каждая роль имеет свой собственный интерфейс; порученная роль связана с порученным интерфейсом, а роль административного управления связана с интерфейсом административного управления.

7.2.6 Базовые и производные классы

Понятия *базового* и *производного класса* основаны на общем понятии изменения шаблонов, названном *нарастающей модификацией*. Нарастающая модификация является получением нового шаблона путем изменения существующего. Новый шаблон называется производным, а исходный — базовым шаблоном. Экземпляры исходного и производного шаблонов называются, соответственно, базовым и производным классами.

В общем случае нарастающая модификация, в которой допустимы замены, приводит к иерархии, отличной от иерархии класс/подкласс.

Например, рассмотрим класс шаблонов С1 красных автомобилей, определенный шаблоном Temp1, который содержит строку:

ЦВЕТ = КРАСНЫЙ.

В классе шаблонов С2 эта строка заменена на

ЦВЕТ = СИНИЙ.

Класс шаблонов С2 является производным класса С1, но не является его подклассом.

В некоторых случаях реализация производного класса может быть основана на реализации базового класса. Тогда говорят о иерархии реализаций, которая позволяет совместно использовать исполняемый код программ. Однако в распределенной среде это может вызвать проблемы, так как изменения кода базового класса должно быть распространено для обновления всех реализаций производного класса; таким образом, в системах ОРО не требуется поддерживать иерархию реализаций.

**7.3 Организационные понятия**

7.3.1 Группы и области

*Группа* является набором объектов, собранных вместе по организационным причинам или потому, что поведение объектов имеет общие характеристики (например, в репликационной группе они могут заменять друг друга, в коммуникационной группе они участвуют в одном и том же взаимодействии). Понятие группы является родовым и допускает спецификации различных видов групп, которые могут использоваться в распределенных системах для таких разных целей, как *отказоустойчивость*, доступность и поддержка приложения (например, в приложениях конференций).

*Область* является конкретной формой группы, в которой конкретные аспекты поведения объектов управляются одним уполномоченным. Например, в *области безопасности* применяемая к поведению объектов политика безопасности устанавливается одним *уполномоченным по безопасности*. Концепция области допускает введение в распределенных системах понятий автономности, контроля и уполномоченного. Концепция области перекрывает самые различные потребности, так как распределенные системы используют разные виды областей (например, области безопасности, административного управления, *наименования*).

7.3.2 Наименование

Наименование необходимо для различения компонентов распределенной системы и доступа к ним; следовательно, оно является фундаментальным элементом конструкции распределенной системы.

Для работы в условиях неоднородности, автономности и *федеративности* необходимы зависящее от контекста наименование и администратитвное управление *именами*. Они обеспечивают гибкость и эволюционирование, допуская независимое развитие систем имен и их произвольных комбинаций, включая существующие независимые системы. Кроме того, они допускают неоднородность систем имен на нескольких уровнях, например формы имен, присваивания имен, политика наименования и стратегия *разрешения имени* могут быть разными в разных системах.

Имена используются для указания категорий в данном контексте. Могут быть обстоятельства, при которых одно имя указывает на несколько категорий. Имя, которое недвусмысленно указывает на одну категорию, называется *идентификатором*.

Концепция наименования в ГОСТ Р ИСО/МЭК 10746-2 не устанавливает полной схемы наименования в ОРО. Такая схема является предметом самостоятельной стандартизации.

7.3.3 Контракт

*Контракт* является соглашением, которое управляет кооперацией между несколькими объектами и содержит *обязательство, разрешения, запрещения* и *ожидания,* связанные с этими объектами. Таким образом, контракт является общим понятием для характеристики и регулирования кооперации между объектами.

Всякий раз, когда между объектами существует кооперация (взаимодействие), между ними существует некоторый контракт. В случаях, когда контракт в некоторый момент может быть согласован и позднее — завершен, имеется динамическая спецификация конфигурации объектов. Хотя кооперация между объектами возможна постоянно, правила, устанавливаемые контрактом, ограничивают возможную кооперацию некоторым текущим, временным поведением.

Часто контракты являются производными от правил *языка точки зрения* и не обязательно должны быть выражены явно. Например, в случае вычислительного языка клиент обязан не вызывать операцию, которая не определена в интерфейсе сервера. Явно должны быть выражены только контракты, устанавливающие дополнительные ограничения.

Например, контракт может устанавливать:

- роли объектов и обязательства для этих ролей, т. е. ожидаемое кооперативное поведение;
- вопросы *качества услуг (КУ)* кооперации объектов (вопросы зависимости, корректности и пр.);
- виды поведения, *нарушающие* контракт.

*Контракт среды* является частным видом контракта между объектом и его средой. Этот контракт описывает требования, предъявляемые объектом к среде и средой к объекту. В частности, в нем рассматриваются ограничения КУ.

7.3.4 Взаимодействие и связывание

*Связывающее поведение* устанавливает *контрактный контекст* (связывание) между интерфейсами и позволяет объектам кооперироваться. Связывание может существовать на нескольких уровнях абстракции.

*Взаимодействие* является взаимосвязью, которая существует между объектами, кооперированными на основе связывания. Когда имеет место взаимодействие, объекту известно, что другой объект взаимодействия подчиняется контракту. Объект может вовлечен одновременно в несколько взаимодействий; в этом случае для каждого из них имеется соответствующий контракт.

Примечание — В примере следующего абзаца операции Bind, BindResponse, Unbind и прикладной контекст используются в смысле ВОС.

Примером установления взаимодействия является связывающее поведение ассоциации ВОС. *Устанавливающее поведение* обеспечивается операцией Bind, которая включает в себя отправку инициирующим прикладным объектом параметров контракта отвечающему прикладному объекту. *Отвечающий объект* использует операцию BindResponse (возможно, с другими параметрами контракта), устанавливая тем самым взаимодействие. Контрактный контекст для взаимодействия задается прикладным контекстом, согласованным при обмене Bind и BindResponse. Когда стороны решат закончить взаимодействие, они инициируют *завершающее поведение,* вызвав операцию Unbind.

## 8 Архитектура

В ГОСТ Р ИСО/МЭК 10746-3 установлены требования, которые должны быть справедливыми для системы ОРО. В этом стандарте на основе понятий и терминологии ГОСТ Р ИСО/МЭК 10746-2 определены:

- основы архитектуры для построения спецификаций систем ОРО в терминах точек зрения, *спецификаций точек зрения* и *прозрачностей распределения;*
- набор языков, в терминах которых могут быть выражены спецификации различных точек зрения;
- инфраструктура системы, обеспечивающая *прозрачности распределения* для приложений системы.

### 8.1 Основы архитектуры

Распределенные системы могут быть очень большими и сложными, и самые различные влияющие на их проекты подходы могут быть отражены в спецификации, которая должна иметь структуру, допускающую успешное управление. Удачная структура должна позволять проектировать от-

дельные части системы независимо друг от друга, если они являются независимыми, но она же должна позволять четко идентифицировать те места, где разные аспекты проекта ограничивают друг друга. Для достижения этого в архитектуре ОРО используют два главных структурирующих подхода: определение точек зрения и определение прозрачностей.

8.1.1 Точки зрения

Точка зрения является частью спецификации всей системы, введенной для объединения конкретных порций информации, относящихся к некоторой конкретной области рассмотрения при проектировании системы. Система ОРО есть нечто, что может быть рассмотрено; таким образом, например, она может быть и системой обработки информации в некоторой организации, и конкретным компонентом (техническим или программным) такой системы. Точки зрения не являются абсолютно независящими друг от друга: ключевые элементы каждой из них идентифицируются как связанные с элементами других точек зрения. Однако точка зрения достаточно независима для того, чтобы упростить рассмотрение полной спецификации.

Каждая точка зрения может быть связана со всеми другими. Они не образуют фиксированной последовательности, как набор уровневых протоколов, и не задают фиксированного порядка, соответствующего некоторой методологии проектирования. Архитектура выражается в терминах полного набора взаимосвязанных точек зрения, без указания того, как должна строиться полная спецификация для данной системы.

В БМ-ОРО определены пять точек зрения на систему и ее среду:

а) *Предпринимательская точка зрения,* которая сфокусирована на целях, области применения и политике системы.

б) *Информационная точка зрения,* которая сфокусирована на семантике информации и осуществляемой обработке информации.

в) *Вычислительная точка зрения,* которая допускает распределение путем функциональной декомпозиции системы на объекты, взаимодействующие через интерфейсы.

г) *Инженерная точка зрения,* которая сфокусирована на методах и функциях, требуемых для обеспечения распределенного взаимодействия между объектами системы.

д) *Технологическая точка зрения,* которая сфокусирована на выборе технологии в этой системе.

Для представления системы ОРО с конкретной точки зрения необходимо определить структурированный набор понятий, в терминах которого может быть выражено это представление (или спецификация). Такой набор понятий предоставляет язык для написания спецификаций систем с этих точек зрения, и спецификации образуют модель системы в терминах понятий. Термины каждого *языка точки зрения* и правила применения этих терминов определены с помощью установленных в ГОСТ Р ИСО/МЭК 10746-2 моделирующих понятий. Каждый язык обладает достаточными выразительными средствами для спецификации с соответствующей точки зрения *функций ОРО*, приложений и политики. Когда спецификации системы соответствуют этим языкам, проектируемая система является системой ОРО, по крайней мере с архитектурной точки зрения.

В БМ-ОРО разные языки точек зрения отличаются по строгости налагаемых ими ограничений. Языки тех точек зрения, сконцентрированных на организации распределения и обеспечении общих решений (вычислительная и инженерная точки зрения), устанавливают значительное количество ограничений, которые должны соблюдаться, и тем самым гарантируют взаимодействие (и переносимость) между компонентами. С другой стороны, так как БМ-ОРО является родовой, то в предпринимательском и информационном языках установлено всего несколько правил, и они ограничиваются набором основных понятий и руководств относительно области применения и информационного моделирования распределенных систем.

Более строгие ограничения для предпринимательского и информационного языков будут определены в последующих стандартах применительно к спецификациям систем с конкретными областями использования. Например, для любых систем торгов предпринимательская и информационная спецификации ограничиваются положениями стандартов по торгам.

8.1.2 Прозрачности распределения

При проектировании распределенной системы становится очевидной необходимость рассмотрения ряда вопросов, являющихся непосредственным результатом распределения: компоненты системы являются неоднородными, они могут отказывать независимо друг от друга, они находятся в разных и, возможно, изменяющихся местах и т. п. Эти вопросы могут быть решены либо непосредственно, как часть прикладного проекта, либо могут быть выбраны стандартные решения, основанные на хорошей практике.

Если выбран стандартный метод, то проектировщик приложения работает в среде, которая прозрачна относительно данного конкретного вопроса; говорят, что стандартный метод обеспечивает *прозрачность распределения*. Проектировщик приложения просто выбирает, какую он хочет принять прозрачность распределения и где в проекте она должна применяться.

Применение прозрачности распределения может непосредственно привести к повторному использованию программного обеспечения. Выбор прозрачностей распределения в спецификации системы может привести к автоматическому включению в систему хорошо проверенных реализаций стандартных решений с помощью используемых инструментов построения системы, таких как компиляторы, редакторы связей, менеджеры конфигурации. Проектировщик выражает системные требования в виде упрощенных утверждений системы и свойств прозрачности распределения, которые должны использоваться.

В БМ-ОРО определены следующие прозрачности распределения:

а) *прозрачность доступа,* которая маскирует различия в представлении данных и методах вызова, обеспечивая взаимодействие между объектами. Она решает многие из проблем взаимодействия между неоднородными системами и, в общем случае, обеспечивается по умолчанию;

б) *прозрачность отказа,* которая маскирует от объекта отказ и возможное *восстановление* других объектов (или его самого), обеспечивая отказоустойчивость. Когда она обеспечивается, проектировщик может работать в идеализированной среде, где соответствующие классы отказов не встречаются;

в) *прозрачность положения,* которая маскирует использование информации о *положении в пространстве* при идентификации и связывании с интерфейсами. Она обеспечивает логический вид наименования, не зависящий от фактического физического размещения;

г) *прозрачность миграции,* которая маскирует от объекта возможности системы по изменению положения этого объекта. Миграция часто используется для достижения сбалансированности загрузки и уменьшения времени ожидания;

д) *прозрачность перемещения,* которая маскирует перемещение интерфейса от других, связанных с ним интерфейсов. Перемещение позволяет продолжить операцию системы, даже когда миграция или перемещение некоторых объектов создает временную несогласованность в виде, в котором объекты представляются их пользователям;

е) *прозрачность дублирования,* которая маскирует использование для обеспечения интерфейса группы поведенчески взаимно совместимых объектов. Дублирование часто используется для повышения производительности и доступности;

ж) *прозрачность постоянства,* которая маскирует от объекта *деактивацию* и *реактивацию* других объектов (или его самого). Деактивация и реактивация часто используются для обеспечения *постоянства* объекта, когда система не в состоянии непрерывно его обеспечивать обработкой, памятью или функциями коммуникации;

и) *прозрачность транзакций,* которая для достижения согласованности маскирует координацию деятельностей в конфигурации объектов.

В любой спецификации системы определение прозрачности включает в себя как набор требований, так и прозрачность распределения, которая им удовлетворяет. Набор требований устанавливает, где необходима прозрачность распределения (т. е. на какие взаимодействия она влияет). Это может быть просто утверждение о том, что прозрачность применяется по всей системе, или это может быть более селективное утверждение, включающее в себя конкретные интерфейсы и определяющее, например, взаимодействия, которые образуют транзакцию на выбранных объектах и интерфейсах и должны обеспечиваться дублированием. Решение имеет вид набора правил для преобразования спецификации запрошенной прозрачности распределения в спецификацию, в которой выбранные объекты и интерфейсы расширяются для включения методов обеспечения прозрачности.

**8.2 Предпринимательский язык**

Предпринимательский язык вводит основные понятия, необходимые для представления системы ОРО в контексте предприятия, в которой она работает. Предпринимательская спецификация предназначена для выражения целей и ограничений политики рассматриваемой системы. Для этого система представляется одним или несколькими предпринимательскими объектами в изображающем предприятие *сообществе* предпринимательских объектов и ролями, в которых эти объекты участвуют. Эти роли представляют, например, пользователей, владельцев и поставщиков информации, обрабатываемой системой. Создание отдельной точки зрения для выражения этой информации отделяет спецификацию задач системы от способа, которыми система эти задачи решает.

Одной из ключевых идей предпринимательского языка является идея контракта, связывающего исполнителей разных ролей в сообществе и выражающего их взаимные обязательства. Контракт может выражать общие цели и ответственность различных ролей в сообществе, таких как производитель и потребители, правительственная организация и ее клиенты.

При необходимости предпринимательская спецификация может формулировать вопросы владения ресурсами и ответственности (при оплате товаров и услуг) для идентификации, например ограничений на методы учета и безопасности в инфраструктуре, которая поддерживает систему.

Конкретным видом сообщества является федерация, которая представляет собой собранные вместе несколько групп, отвечающих разным уполномоченным (и, следовательно, представляемых как разные области), для того, чтобы они могли скооперироваться для достижения некоторой цели. Так как эволюция распределенных систем будет быстро приводить к слиянию существующих раздельно управляемых подсистем для совместного использования информации или обеспечения коммерческих интересов, то спецификация *создания* федераций и формулирование правил, по которым они управляются, образует важную часть спецификации системы с предпринимательской точки зрения.

Собранные в федерацию области могут быть административными (например, каждая из них управляется конкретным уполномоченным по безопасности или административному управлению) или технологическими (например, каждая из них является каким-то выбранным программным или техническим средством). Спецификация федерации включает в себя спецификации целей взаимодействия между разными областями и политики, управляющей этим взаимодействием.

Федерация административных областей относится к взаимодействию между областями в одном и том же или разных предприятиях с целью обеспечивать совместное использование, интеграцию или разделение ресурсов и приложений среди разных систем в ответ на потребности пользователя. Федерация технологических областей относится к интеграции различных системных архитектур и систем с разными ресурсами и возможностями; она обеспечивает модульность, которая позволяет наращивать систему, не отбрасывая существующие приложения. Эти два типа федерации часто совпадают, так как различия в администрировании могут привести к различию в выборе технологии.

Для административных областей один или оба администратора могут захотеть, в дополнение к контролю самими объектами, установить свой собственный контроль доступа для таких целей, как безопасность, учет или мониторинг. Административные границы также являются теми точками, где происходят изменения ответственности за административное управление такими вопросами, как распределение ресурсов и гарантии зависимости.

Политика, управляющая операциями федерации, включает в себя политику, управляющую взаимодействием. Таким образом, может оказаться необходимым, чтобы спецификация федерации была увязана со спецификацией возможностей *пересечения* в инженерном описании, а цели и политика федерации устанавливали ограничения на возможности пересечения.

Предпринимательская спецификация устанавливает политику, управляющую поведением определяемых ею сообществ. Эта политика определяет действия предпринимательских объектов, составляющих сообщества, и сконцентрирована на установлении и выполнении обязательств (например, запроса доставки, осуществление доставки) и разрешении или запрещении действий (например, авторизируя или отвергая доступ к возможностям системы). Политика может относиться к следующему:

а) организации сообщества в терминах ролей и назначения ролей предпринимательским объектам. Например, **правила сообщества** могут устанавливать:

- назначение ролей и ответственности предпринимательским объектам внутри сообщества,
- то, как предпринимательские объекты связаны в структуре сообщества (например, иерархия или равноправие).

Предпринимательская спецификация может включать в себя **коммерческие правила,** которые выражают:

- предпринимательство как коммерческую категорию,
- требования учета,
- развитие бизнеса для достижения своих целей;

б) допустимым взаимодействиям между предпринимательскими объектами, играющими разные роли (т. е. к контролю доступа). Например, **правила безопасности** могут определять:

- взаимосвязи роль — *деятельность* — объект, требования целостности и конфиденциальности для деятельностей и объектов,

- правила для выявления угроз безопасности,
- правила защиты от угроз безопасности,
- правила для ограничения вреда, вызванного нарушением безопасности;

в) ответственности, делегируемой предпринимательским объектам. Например, **правила описания уполномоченного** используются для присвоения:

- привилегий предпринимательским объектам (доверение),
- разрешение или запрещение действий предпринимательских объектов;

г) учету использования ресурсов. Например, **правила использования ресурсов** определяют ограничения, которые могут быть внешним образом установлены для:

- регулирующих органов,
- рыночных запросов,
- среды,

в зависимости от того, используется ресурс:

- общедоступно,
- частно,
- третьей стороны;

д) владению ресурсами. Например, **правила передачи** могут устанавливать обмен владением и(или) ответственностью за ресурсы между предпринимательскими объектами;

е) членству в федерации. Например, предпринимательская спецификация может включать в себя **правила области,** которые устанавливают:

- правила членства в области,
- правила взаимодействия между областями одного типа,
- правила наименования областей.

Предполагается, что для выражения предпринимательских спецификаций для разных конкретных организационных структур и коммерческих задач будут существовать различные нотации. БМ-ОРО требует, чтобы соответствующая спецификация была создана, но устанавливает мало ограничений на то, как это должно быть сделано.

Оценка соответствия системы предпринимательской спецификации включает соответствующие требования (например, время ответа для выполнения обязательства), выраженные в спецификации для установления наблюдения поведения системы в *точках соответствия,* идентифицированных в инженерной и технологической спецификациях, и оценку степени согласованности между требованиями и наблюдениями.

**8.3 Информационный язык**

Отдельные компоненты распределенных систем должны иметь общее понимание информации, которой они обмениваются при взаимодействии, иначе система не будет себя вести так, как ожидается. Некоторые из этих элементов информации могут тем или иным способом обрабатываться многими объектами системы. Для того чтобы гарантировать, что интерпретации этих элементов согласованы, информационный язык определяет понятия для спецификации смысла хранящейся в элементах информации и их обработки (системой ОРО) независимо от самих функций обработки информации, которые должны применяться.

Хранящаяся в системе ОРО информация о категориях реального мира, включая саму систему ОРО, представлена в информационной спецификации в терминах информационных объектов, из взаимосвязей и поведения. Основные информационные элементы представлены элементарными информационными объектами. Более сложная информация представлена составными информационными объектами, выражающими взаимосвязи в множестве составляющих информационных объектов.

Как и при обычном моделировании данных, информационная спецификация включает в себя набор взаимосвязанных схем, а именно *инвариантную, статическую* и *динамическую.*

*Инвариантная схема* выражает взаимосвязи между информационными объектами, которые почти всегда должны быть справедливыми при любом допустимом поведении системы. Например, инвариантная схема для банковского счета должна устанавливать, что баланс всегда должен быть неотрицательным, так как банк не позволяет превышать наличный счет.

*Статическая схема* выражает утверждения, которые должны быть справедливы в один момент времени. Обычное использование статической схемы — спецификация начального состояния информационного объекта. Например, начальное состояние объекта «банковский счет» состоит из текущего баланса 0$ и суммы расходов за текущий день, которая также равна 0$. Другая статическая

схема может быть использована для описания того, что сумма расходов за день равна 0$ каждую полночь; эта статическая схема не устанавливает ограничений на текущий счет в полночь.

*Динамическая схема* устанавливает, как может эволюционировать информация по мере работы системы. Например, для банковского счета требуются динамические схемы для депозита, расходов, оплаты и изменения вкладов. Динамическая схема может быть применена только при определенных обстоятельствах (которые могут быть специфицированы с помощью статической схемы). Например, динамическая схема для расхода N$ может устанавливать, что текущий счет уменьшается на N$ при условии, что общее количество расходов не превышает 500$. Динамическая схема не может специфицировать получающееся состояние, которое нарушает инвариантные ограничения, т. е. могут расходоваться только деньги со счета.

В дополнение к описанным изменениям состояния динамическая схема может создавать и удалять объекты-компоненты. Это позволяет всю информационную спецификацию системы ОРО моделировать как один (составной) информационный объект.

Эти схемы могут применяться ко всей системе или к отдельным областям внутри нее. В частности, в больших и быстро эволюционирующих системах согласование и образование федераций отдельных информационных областей будет одной из больших задач, которые необходимо решать для административного управления информацией.

Схемы для составных информационных объектов не обязательно должны относиться ко всем компонентам информационного объекта. Эти схемы могут быть составлены из схем отдельных компонентов при условии, что такая композиция имеет смысл. Инкапсуляция информационных объектов связана с уровнем абстракции рассматриваемого описания. Например, на подходящем уровне абстракции схемы для составных информационных объектов могут относиться к внутреннему строению их компонентов, хотя на более высоком уровне абстракции это может быть невозможным. Тем самым допускается спецификация таких сложных фраз, как «телефонный номер клиента, расходы которого сегодня превысили 400$».

Некоторые элементы, видимые с предпринимательской точки зрения, могут быть видны и с информационной точки зрения, и наоборот. Например, деятельность, видимая с предпринимательской точки зрения, может проявляться в информационной точке зрения как спецификация некоторой обработки, которая вызывает переход состояния информационной категории.

Разные нотации для информационных спецификаций моделируют свойства информации разными способами. Основное внимание может уделяться классификации и переклассификации типов информации или состояниям и поведению объектов. В некоторых языках спецификаций элементарные информационные объекты представляются как значения. Применяемые подходы зависят от метода моделирования и используемой нотации.

Оценка соответствия системы информационной спецификации включает в себя соотнесение выраженных в спецификации требований (например, в инвариантной схеме) с набором наблюдений поведения системы в точках соответствия, идентифицированных в инженерной и технологической спецификациях, и оценку степени согласованности между требованиями и наблюдениями.

**8.4 Вычислительный язык**

Вычислительная точка зрения сконцентрирована непосредственно на распределении обработки, но не на методах взаимодействия, которые позволяют осуществить распределение. Вычислительная спецификация разлагает систему на объекты, выполняющие отдельные функции и взаимодействующие через четко определенные интерфейсы. Это обеспечивает основу для рассмотрения того, как распределить работу, которая должна быть выполнена, так как интерфейсы могут быть распределены независимо при условии, что в инженерной спецификации могут быть определены методы коммуникации для обеспечения поведения в этих интерфейсах.

Сердцевиной вычислительного языка является модель объекта, определяющая:

- вид интерфейса, который может иметь объект;
- способы, которыми интерфейсы могут быть связаны, и виды взаимодействий, которые могут через них происходить;
- действия, которые может осуществлять объект; в частности, создание новых объектов и интерфейсов и установление связывания.

Вычислительная модель объекта обеспечивает основу для гарантии того, что языки спецификаций, языки программирования и методы коммуникации являются согласованными, обеспечивая тем самым открытое взаимодействие и переносимость компонентов.

Вычислительный язык позволяет разработчику спецификации выразить ограничения на распределение приложения (в терминах контрактов сред, связанных с отдельными интерфейсами и связываниями интерфейсов) без спецификации фактической степени распределения. Это гарантирует, что приложения не содержат необъявленных предположений, влияющих на распределения их компонентов. Поэтому конфигурация и степень распределения технического обеспечения, на котором работают приложения ОРО, легко могут быть изменены, в зависимости от ограничений среды, без существенного влияния на прикладное программное обеспечение.

Вычислительный язык не исключает использования в распределенной среде программного обеспечения, спроектированного для централизованных систем. Этим допускается инкапсуляция существующих приложений в качестве (нераспределенных) компонентов больших распределенных приложений. Благодаря этому возможен эволюционный подход к обеспечению распределения и сохранения капитальных вложений в существующее программное обеспечение.

Взаимодействия между вычислительными объектами являются существенно асинхронными и могут быть в трех разных формах:

- *операции,* которые похожи на процедуры и вызываются на предназначенных для них интерфейсах;
- *потоки,* которые являются абстракциями непрерывных последовательностей данных между интерфейсами;
- *сигналы,* которые являются элементарными взаимодействиями.

Операции отражают парадигму клиент/сервер. Операция является взаимодействием между *объектом-клиентом* и *объектом-сервером,* при котором запрашивается (вызывается) выполнение некоторой функции сервером. Имеется два вида операций:

- *запросы,* при которых сервер возвращает ответ (*завершение*) на запрос клиента;
- *сообщение,* при котором ответ клиенту не возвращается.

Понятие завершения обобщает понятия результата и исключительного состояния, которые существуют во многих объектно- и необъектно-ориентированных языках программирования.

Выполнение операций происходит в пространстве и во времени. Следовательно, когда операция отказывает, то отказ не обязательно происходит для всех участников и может наблюдаться ими в разное время. Возможности клиента наблюдать отказ и предпринимать какие-то действия различны для запросов и сообщений.

В случае запроса двусторонняя связь гарантирует как то, что клиент имеет подтверждение выполнения функции, так и то, что, если *связка* деятельностей клиента привела к *цепочке* вызовов, сервер будет отвечать на запросы клиента в порядке их поступления.

В случае сообщения гарантии и порядок осуществления определяются теми контрактами сред, которые применяются к операциям.

Потоки могут использоваться для моделирования, например, потока аудио- или видеоинформации в мультимедийном приложении или в ориентированных на голос телекоммуникационных услугах, а также для моделирования непрерывного потока периодического чтения датчика в приложении управления процессом. Поток характеризуется именем и типом, который задает природу и формат данных. В вычислительной модели точная семантика потока остается неопределенной. Фактически может быть много различных семантик для потока в зависимости от прикладной области.

Сигналы являются низшим уровнем описания взаимодействий между вычислительными объектами. Сигнал — это парное, элементарное, совместное действие, приводящее к односторонней передаче от *инициирующего* вычислительного объекта к *отвечающему* (в данном контексте «отвечающий» означает «принимающий передачу»). Это значит, что:

- сигнал происходит в определенный момент времени и, следовательно, является опорной точкой для целей измерения (например, при наблюдениях КУ);
- отказ идентичен и видим для всех участников.

Во многих случаях сигнал является ответом (в терминах реализации) на наблюдаемое событие в некотором физическом месте, однако определение этого понятия не исключает реализацию сигналов методами транзакций, которые дают необходимые гарантии поведения.

Операции или потоки могут быть объяснены в терминах нескольких сигналов. Например, запрос можно понимать как последовательность сигналов: посылка вызова (объектом-клиентом), получение вызова (объектом-сервером), посылка завершения (сервером), получение завершения (клиентом). Напротив, так как точная семантика потоков не устанавливается в вычислительной модели, отображение их в сигналы не определяется. Моделирование операций или потоков в терминах сигналов

становится необходимым для определения сквозных характеристик КУ, операции многостороннего связывания и связываний разных видов интерфейсов (например, потока с *интерфейсом операций*).

8.4.1 Вычислительные интерфейсы

Вычислительный интерфейс характеризуется сигнатурой, поведением и контрактом среды.

Сигнатура зависит от типа интерфейса, который может быть интерфейсом операций, потоков и сигналов.

*Интерфейс операций* имеет сигнатуру, которая определяет набор операций, поддерживаемых на интерфейсе, и какую роль (клиента или сервера) имеет интерфейс для этого набора операций.

*Интерфейс потоков* имеет сигнатуру, которая определяет набор потоков, поддерживаемых на интерфейсе, и, для каждого потока, какую роль (создателя или потребителя) имеет интерфейс.

*Интерфейс сигналов* имеет сигнатуру, которая определяет набор сигналов, поддерживаемых на интерфейсе, и, для каждого сигнала, какую роль (инициирующую или принимающую) имеет интерфейс.

Поведение описывается связанными с интерфейсом допустимыми последовательностями действий вычислительного объекта. Поведение может включать в себя внутренние действия объекта и может ограничиваться средой объекта, в частности, взаимодействиями через другие интерфейсы.

Существенно, что каждая спецификация интерфейса содержит контракт среды, который устанавливает набор ограничений КУ, налагаемых на вычислительный объект его средой. Если среда (другие вычислительные объекты и обсепечивающая инфраструктура) предоставляет требуемый уровень КУ, то гарантируется (проектом), что сам объект предоставляет определенный уровень КУ. КУ, специфицированное для интерфейса, выражает как его требования к среде, так и КУ, демонстрируемое объектом в среде, удовлетворяющей этим требованиям. Вычислительным языком не предписывается конкретная нотация для спецификации КУ.

8.4.2 Модель связывания

Взаимодействие между данными вычислительными интерфейсами возможно, только если между ними произошло связывание (т. е. некоторая часть связи). Вычислительный язык специфицирует действия *явного связывания* для интерфейсов как операций, так и потоков. В случае интерфейсов операций он также специфицирует, что связывание может быть *неявным* для того, чтобы можно было использовать нотации, которые не предоставляют выражений для связывающих действий. *Неявное связывание* может иметь место только для интерфейсов операций, так как в других случаях не самоочевидно, где находится инициатива связывания относительно последующих взаимодействий.

Когда связывание неявное, вызов объектом-клиентом приводит к связыванию соответствующего интерфейса клиента с интерфейсом сервера и осуществлению операции (запроса или сообщения). Вычислительный язык оставляет неопределенным, удаляется или нет интерфейс клиента в конце процесса. Неявное связывание не имеет средств указания контракта среды для связывания.

Когда связывание явное, оно определяется в терминах двух видов связывающих *действий*: *элементарного* и *составного связывания*. Связывающие действия применимы только в контексте явного связывания.

Действие элементарного связывания позволяет связать два интерфейса одного и того же или разных вычислительных объектов. Интерфейсы должны быть одного и того же типа, но оба они могут быть интерфейсами операций, потоков или сигналов. Действие элементарного связывания осуществляется одним из рассматриваемых объектов и приводит к установлению на каждом интерфейсе информации, необходимой для взаимодействия, т. е. идентифицирующей другой, участвующий во взаимодействии, интерфейс. Определение явного *развязывающего* действия не требуется, но очевидно, что удаление любого интерфейса приводит и к удалению связывания. Действие элементарного связывания требует, чтобы рассматриваемые интерфейсы были одного и того же типа и имели дополнительные типы сигнатур и роли (например, один — клиент, другой — сервер).

Действие составного связывания позволяет связать два или несколько интерфейсов одного и того же или разных типов с помощью *связующего объекта* (см. рисунок 2).

Это действие может осуществляться одним из участвующих в связывании вычислительных объектов или вычислительным объектом, отделенным от этого связывания. Оно приводит к реализации вычислительного объекта для обеспечения связывания (связующего объекта). Связующий объект реализует соответствующий набор интерфейсов и использует действия элементарного связывания этих интерфейсов с нужными. Он также реализует управляющие интерфейсы, через которые можно управлять его операциями, и возвращает идентификаторы интерфейсов инициирующему вычислительному объекту.

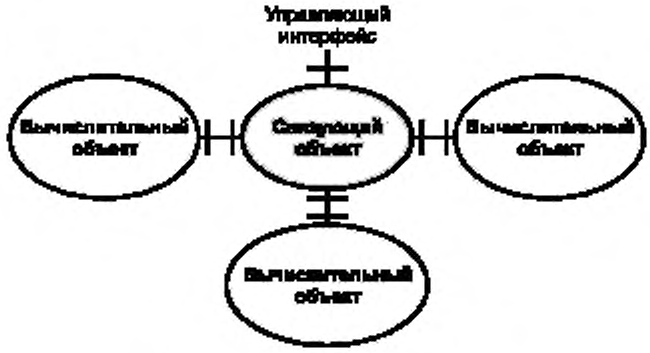



Рисунок 2 — Составное связывание

Поведение связующих объектов отображает обеспечиваемую им коммуникационную семантику, а вычислительная модель не ограничивает типы связующих объектов, отражая тот факт, что могут существовать различные коммуникационные структуры между объектами. Тем не менее полезные классы связующих объектов могут быть стандартизованы в зависимости от классов приложений. В частности, связующие объекты могут устанавливать операции многостороннего или сложного связывания (например, между интерфейсами операций и потоков разных типов).

Как и любые другие объекты, связующие объекты могут уточняться утверждениями о КУ, которые ограничивают их корректное поведение (например, ограничивают общее время коммуникационной задержки или общую задержку синхронизации на интерфейсе получателя). Когда сделаны такие утверждения о КУ, интерфейсы, участвующие в элементарных связываниях связующего объекта, должны быть интерфейсами сигналов, так как элементарная природа сигналов позволяет установить в пространстве и времени точки, в которых можно наблюдать КУ.

Управляющие интерфейсы связующего объекта делают возможным *удаление* связывания, управляющие операциями объекта и качеством предоставляемых им услуг. Примерами предоставляемых ими возможностей являются:

а) управление *сообщениями об ошибках,* которые разрушают связующий объект; этим допускается спецификация интерфейса, через который объект вызывает операцию сообщения, когда отказы разрывают связывания;

б) управление динамическим многосторонним связыванием, позволяющее добавлять новых пользователей и удалять существующих;

в) групповой вызов, делающий возможными в вычислительном языке множественные элементарные вызовы и позволяющий добавлять и удалять членов группы;

г) управление КУ, относящимся к связыванию, позволяющее работать с конкретными характеристиками КУ; в частности, этот вид управления был бы очень полезен для связывания потоков мультимедийных приложений;

д) сообщение о событии, интересующем приложение; например, сообщение может сигнализировать о начале или конце периода звучания в звуковом потоке.

В случае связывания интерфейсов потоков можно абстрагироваться от специфических для приложения правил составления потока. В простейшем случае связывание будет представлять один поток от создателя интерфейса к его потребителю (например, от звукозаписывающей системы к динамику). Однако правила составления могут быть более сложными.

а) Полный дуплексный путь может создаваться и управляться как одно связывание; результирующие потоки соединяют создателя интерфейса на вычислительном объекте с потребителем интерфейса на другом объекте.

б) Несколько полных дуплексных интерфейсов могут быть соединены связующим объектом, который инкапсулирует правила системы конференций, позволяя распространить поток от выбранного *создателя* (текущего докладчика) ко всем потребителям. Различная степень прикладного управления может осуществляться через управляющий интерфейс связывания для обеспечения явного управления потоком.

в) Потоки от нескольких создателей могут быть объединены в составной поток для единственного потребителя. Например, видеопоток от одного источника и звуковой поток от другого могут

быть скомбинированы в единый телевизионный поток изображения и соответствующего комментария. Тогда управляющий интерфейс может позволить манипулировать инженерным механизмом синхронизации потока.

Интерфейс может быть связан многократно. В случае неявного связывания кратное связывание интерфейса требует, чтобы каждое связывание было идентифицировано сервером интерфейса. В случае явного связывания кратное связывание требует, чтобы кратное связывание было идентифицировано связующим объектом.

8.4.3 Типы и подтипы вычислительных интерфейсов

Интерфейсы в вычислительном языке являются строго типизированными для максимально ранней проверки согласованности распределенных программ и вычислительного языка ОРО. Типы интерфейсов связаны отношением образования подтипа, которое определяет минимальные условия, необходимые для обеспечения осмысленного взаимодействия объектов.

Тип сигнатуры интерфейса определяет форму и вид взаимодействий, доступных через интерфейс, а сигнатура подтипа устанавливает минимальные требования, при выполнении которых один интерфейс может заменить другой. Правила основаны на семантике взаимодействий вычислительных интерфейсов и достаточны для того, чтобы гарантировать, что замененный интерфейс может согласованно интерпретировать структуру любого происходящего взаимодействия. Конечно, также необходимо, чтобы интерфейсы согласовывались в терминах семантики передаваемых данных, но невозможно определить общие правила, передающие такое согласование.

*Сигнатура интерфейса сигналов* определяет для каждого сигнала в интерфейсе его имя, параметры, а также является ли рассматриваемый интерфейс *инициатором* или ответчиком. *Сигнатура интерфейса операций* определяет для каждого вида операций в интерфейсе имя операции, количество и типы ее аргументов, а также (для запроса) набор возможных результатов операции (завершений). Для каждого завершения определяются его имя, количество и типы его аргументов.

Правила образования подтипов для сигнатур интерфейсов операций и сигналов определены в ГОСТ Р ИСО/МЭК 10746-3, приложение А. Семантика взаимодействия для интерфейсов операций и потоков может быть выражена не только в терминах операций и потоков, но и в терминах сигналов, так как сигналы обеспечивают основные блоки, из которых строится модель любого типа взаимодействий более высокого уровня.

Тип интерфейса потоков определяется в терминах набора составляющих потоков, имеющих базовые типы, которые поддерживаются доступными нижележащими средствами. Примерами базовых типов являются отдельные звуковые или видеопотоки. Каждый поток-компонент имеет единственное направление, входящее в связывание или исходящее из него. Потоки-компоненты описанием типа организуются в сигнатуру потока (так же, как аргументы организуются в сигнатуру в интерфейсе операций). Интерфейс потоков может состоять из нескольких взаимосвязанных потоков, направленных в одну или в разные стороны. В случае потоков тип интерфейса сам может быть использован для представления более сложного потока так, что типы, построенные из последовательностей потоков, могут образовывать многоуровневую иерархию.

Примерами типов интерфейсов потоков являются:

а) звуковой поток от аудиоисточника;

б) звуковой поток к устройству воспроизведения;

в) полностью дуплексный речевой тип, в котором есть один входящий и один исходящий потоки для представления пользовательской точки зрения на звуковые аспекты услуг телефонии;

г) составной телевизионный сигнал, состоящий из аудио- и видеокомпонентов;

д) более сложные ориентированные на приложение типы, в которых несколько аудио- и видеопотоков комбинируются для представления потоков в системе виртуальной реальности.

Тот факт, что каждый поток в интерфейсе потоков имеет направление, подразумевает, что типы интерфейсов существуют парами, содержащими разнонаправленные потоки. Однако, если тип интерфейса имеет эквивалентные наборы входящих и исходящих потоков, то два взаимосвязанных типа становятся эквивалентными. Нотация для определения интерфейсов потоков может допускать в таких случаях сокращения или одновременное определение двух взаимосвязанных форм.

Любая система типов интерфейсов потоков будет иметь соответствующие правила образования подтипов. Они отличаются от правил образования подтипов для интерфейсов операций тем, что построены таким образом, чтобы обеспечить коммуникацию между вычислительными объектами, интерфейсы потоков которых предоставляют разные возможности. Например, звуковой интерфейс может рассматриваться как подтип составного звукового и видеоинтерфейсов, позволяя удаленно-

му пользователю телефона связаться с пользователем видеотелефонной системы. Оптимальная форма образования подтипа будет зависеть от приложения, так что выбор подходящего варианта является частью процесса проектирования.

В общем случае правила образования подтипов потока могут быть получены в два этапа:

а) идентификация соответствий между элементарными потоками в двух типах, решение, являются ли найденные соответствия существенными для взаимоотношения подтипа;

б) сравнение типов элементарных потоков, включая КУ, для определения, существует ли взаимоотношение подтипа.

Невозможно определить общие правила образования подтипов для интерфейсов потоков, так как они зависят от деталей взаимодействий, абстракциями которых являются рассматриваемые потоки.

8.4.4 Переносимость

Вычислительный язык определяет действия, которые может осуществлять объект, и перечисляет возможные режимы отказов этих действий. Таким образом, вычислительный язык определяет ориентированную на объекты модель программирования для родовой виртуальной машины, которая реализуется инженерными и технологическими правилами.

Могут быть определены различные наборы правил переносимости, каждое из которых специфицирует конкретное подмножество действий, определяемых вычислительной моделью программирования. Набор правил переносимости идентифицирует требования к вычислительной нотации для обеспечения переносимости объектов между разными средами. Сама БМ-ОРО определяет базовую и полную среды переносимости, зависящие от набора поддерживаемых действий.

**8.5 Инженерный язык**

Инженерный язык сфокусирован на способах достижения взаимодействия объектов и на необходимых для этого ресурсах. Он определяет понятия для описания инфраструктуры, требуемой для обеспечения селективной прозрачности распределения взаимодействий между объектами, и правила для структурирования коммуникационных *каналов* между объектами и структурирования систем для административного управления ресурсами.

Тогда как вычислительная точка зрения сконцентрирована на том, когда и почему взаимодействуют объекты, инженерная точка зрения сконцентрирована на том, как они взаимодействуют. В инженерном языке основным вопросом является обеспечение взаимодействий между вычислительными объектами. Как следствие, существует непосредственная связь между описаниями с этих двух точек зрения: вычислительные объекты с инженерной точки зрения представляются как *основные инженерные объекты,* в вычислительные связывания, явные и неявные — как каналы или локальные связывания.

Понятия и правила являются достаточными для спецификации внутренних интерфейсов в инфраструктуре, позволяют определить различные точки соответствия для различных прозрачностей и допускают стандартизацию родовой инфраструктуры, в которой могут размещаться стандартные модули прозрачности.

8.5.1 Кластеры, капсулы и узлы

Инженерный язык имеет дело с базовыми инженерными объектами и с различными другими объектами, поддерживающими базовые. Он связывает эти объекты для обеспечения доступности системных ресурсов, идентифицируя вложенные последовательности групп.

На внешнем уровне инженерные объекты физически локализованы и связаны с ресурсами обработки путем группировки в *узлы,* которые можно представлять себе как независимо управляемые компьютерные системы. Узел может быть чем-либо, что имеет целостный вид ресурсов (до тех пор, пока проектировщик системы может рассматривать их как единое целое). Таким образом, тесно связанные системы параллельной обработки могут рассматриваться как узел до тех пор, пока они имеют одну политику составления расписания и выделения ресурсов, т. е. как одна операционная система.

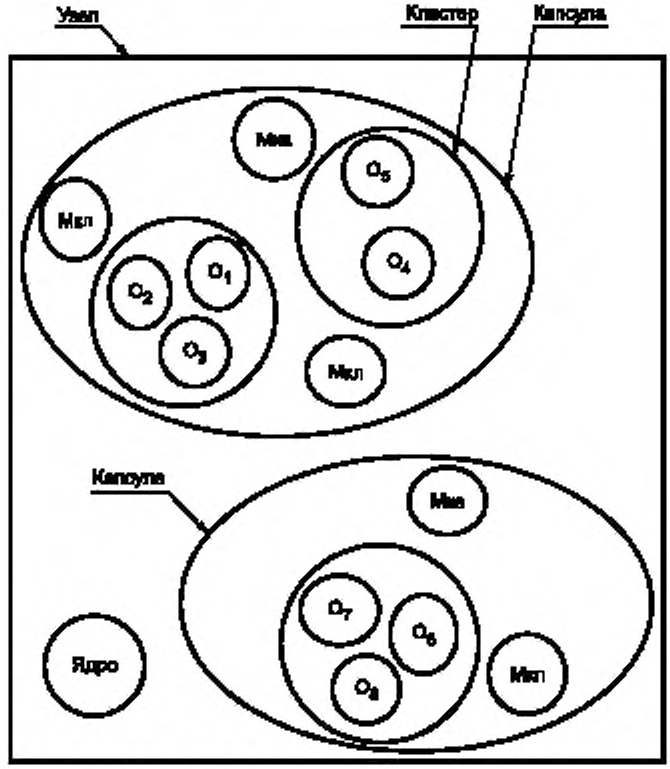
Узел находится под управлением *ядра*, которое отвечает за инициализацию, создание групп инженерных объектов, построение доступных коммуникационных возможностей и предоставление таких базовых услуг, как таймер и источник уникальных идентификаторов.

В ядре могут находиться несколько *капсул.* Капсула занимает память и совместно использует ресурсы обработки ядра. Ее можно представлять в терминах традиционных защищенных процессов со своим собственным адресным пространством. Таким образом, капсула является защищенным

блоком и, в общем случае, наименьшим блоком независимой от обработки отказов операционной системой. Существует ассоциированный с каждой капсулой специальный инженерный объект, называемый *менеджером капсулы*, и (при описании) капсула управляется взаимодействиями с ее менеджером.

Капсула обычно содержит много инженерных объектов; объекты объединяются в капсулы для уменьшения стоимости взаимодействия объектов. Причиной является то, что коммуникация между традиционными процессами медленна и дорога из-за проверок, которые необходимо проводить; однако *подтверждения* могут быть доверены вычислительным средствам, образующим капсулу, а структура взаимодействий между тесно связанными инженерными объектами может быть достаточно расширена для того, чтобы позволить им совместно использовать ресурсы. Ресурсы в капсуле управляются с помощью некоторого языка, специфичного для системы.

Наименьшей группой инженерных объектов является кластер в капсуле. Объекты объединяются в кластер для того, чтобы уменьшить стоимость работы с ними. Кластеры, капсулы и узлы показаны на рисунке 3. Для всех инженерных объектов в капсуле может быть одновременно получена *контрольная точка;* они все вместе могут быть переданы на постоянное хранение, реактивированы или перемещены в другой узел. Такие действия над всем кластером в виде одной операции открывают способ управления очень тонкой структурой, ориентированной на объекты системы за приемлемую цену. Например, система географической информации может рассматривать данные об отдельных точках на карте как инженерные объекты, но не в состоянии обеспечить приемлемую стоимость обработки, если каждый из этих объектов существует совершенно самостоятельно. Связь между инженерными объектами в кластере может быть сильно оптимизирована, так как объекты создаются вместе, на одном языке и вместе хранятся.




Мкк - менеджер капсулы;
Мкл - менеджер кластера

Рисунок 3 — Капсулы, кластеры и узлы

Следовательно, взаимодействие в кластере может поддерживаться вызовом локального метода или аналогичным способом.

Кластер управляется, а действия над ним инициализируются взаимодействиями с соответствующим объектом — *менеджером кластера.*

8.5.2 К а н а л ы

Когда взаимодействуют инженерные объекты из разных кластеров, то необходимо хорошее взаимодействие поддерживающих средств. Даже если объекты в текущий момент находятся в одной капсуле или узле, необходимы средства для обработки завершения, отказа или перемещения одного из них. Набор необходимых для этого средств образует канал, который строится из нескольких взаимодействующих инженерных объектов (см. рисунок 4).

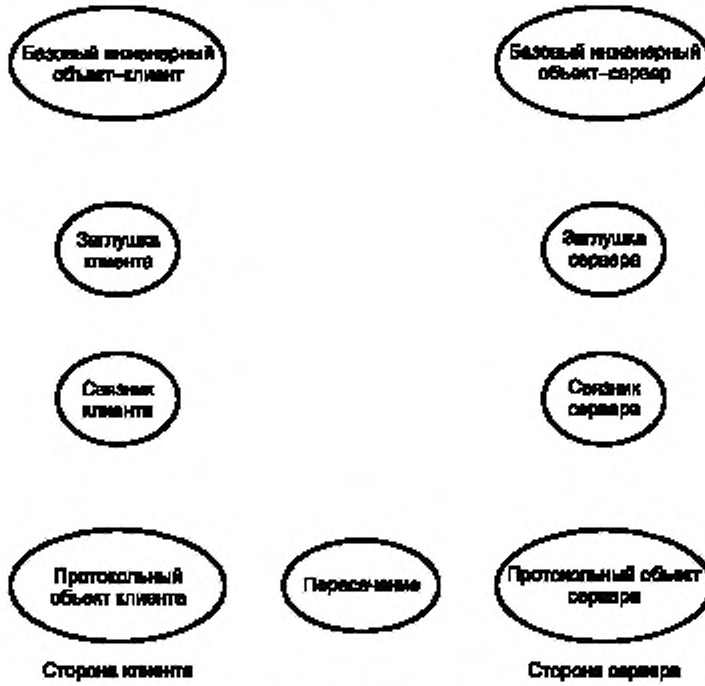



Рисунок 4 — Пример канала клиент—сервер

Инженерные объекты в канале на основании выполняемой ими работы подразделяют на три типа. *Заглушки* ориентированы на передачу информации при взаимодействии, *связники* — на обеспечение ассоциации между базовыми инженерными объектами, соединенными каналом, *протокольные объекты* управляют фактической связью.

Заглушки взаимодействуют непосредственно с поддерживаемыми базовыми инженерными объектами и осуществляют такие функции, как подстановка и анализ параметров или регистрация информации о взаимодействии. Таким образом, заглушки должны иметь доступ к информации о типе взаимодействия или, в более общем случае, о типе поддерживаемого интерфейса. Этим они отличаются от связников и протокольных объектов, которые передают сообщение целиком, без анализа их внутренней структуры. В зависимости от проекта системы заглушка может быть непосредственно связана с конкретным инженерным объектом или может совместно использоваться несколькими такими объектами. В общем случае совместное использование подразумевает необходимость передачи дополнительной информации для идентификации поддерживаемых объектом с целью их различения.

Связники отвечают за решение многих проблем распределения. Связник устанавливает связывание при создании канала, а в дальнейшем поддерживает его сквозную целостность. Это означает, что связник обрабатывает изменения конфигурации, отказы связи или объекта и сохраняет след

другой оконечной точки, если объект переместился или отказал и был заменен (процесс перемещения объекта). Таким образом, связники участвуют в обеспечении многих видов прозрачности распределения.

Протокольные объекты обеспечивают связь достаточного качества и надежности между обслуживаемыми связниками. Кроме обработки используемых парных протоколов, они, при необходимости, обеспечивают доступ к поддерживающим услугам, таким как услуги справочника для преобразования адреса.

Любой из этих трех видов инженерных объектов может сам нуждаться в коммуникации с другими частями системы для получения необходимой в их работе информации и предоставлении *информации административного управления* другим инженерным объектам. Такая коммуникация сама может нуждаться в различных прозрачностях распределения и, следовательно, коммуникация от этих объектов к другим осуществляется с помощью каналов; с такой точки зрения инженерные объекты в одном канале играют роль базовых инженерных объектов в другом. Аналогично, любой из этих объектов может поддерживать управляющие интерфейсы, через которые ими можно управлять. Например, протокольный объект может предоставлять управляющий интерфейс, через который устанавливается нужное качество услуги.

В случаях, когда канал пересекает какую-либо техническую или организационную границу, могут потребоваться дополнительные проверки или преобразования для согласования требований двух сторон. Эти функции осуществляются пересечениями (описанными в 8.5.7), которые составляют часть канала. Им может потребоваться осуществить преобразование формата или протокола, обеспечить учет или проверку управления доступом. Пересечение может быть надстройкой над протокольными объектами, связниками и заглушками в зависимости от характера выполняемой им работы.

На рисунке 4 для простоты была показана конфигурация заглушек, связников, протокольных объектов и пересечения, обеспечивающая единственный канал между двумя базовыми инженерными объектами. В общем случае такая конфигурация может поддерживать каналы между несколькими парами базовых инженерных объектов (рисунок 5) или каналы с несколькими оконечными точками, обеспечивающие различные виды групповой (рисунок 6) или многоцелевой коммуникации. В последнем случае за координацию коммуникации отвечают связники, но многоцелевые средства могут предоставляться либо связниками, либо протокольными объектами, в зависимости от имеющейся технологии. Многоконечные каналы используют для обеспечения прозрачности дублирования.

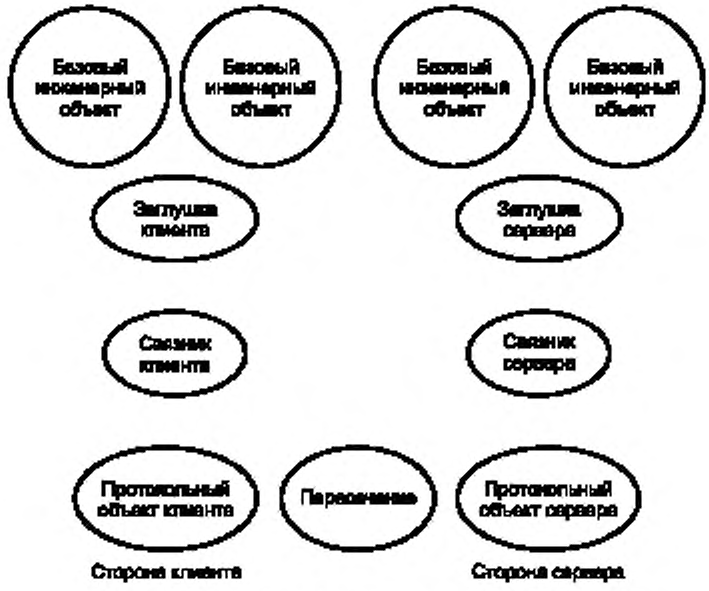



Рисунок 5 — Пример конфигурации с кратными каналами

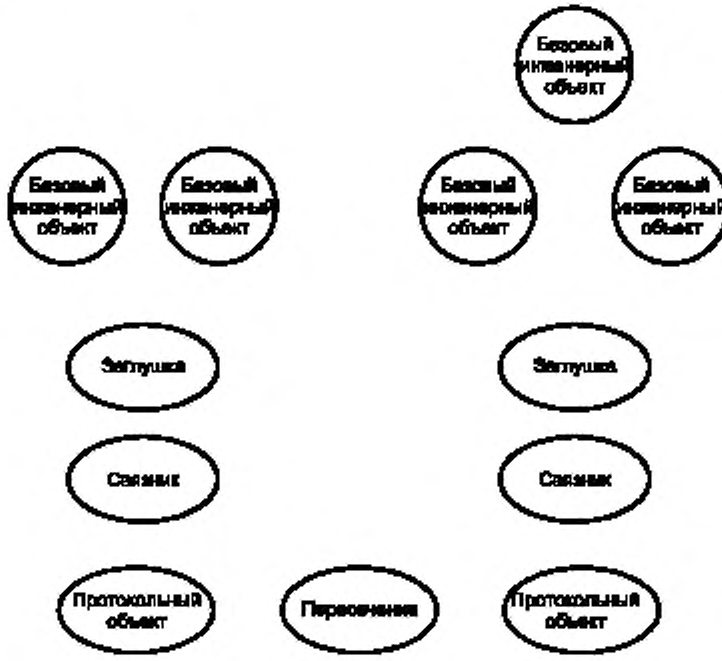



Рисунок 6 — Пример конфигурации группового канала

8.5.3 Указатель интерфейса

При создании интерфейса для него создается указатель. В этом процессе участвует ядро как для того, чтобы сделать интерфейс недвусмысленным, так и для того, чтобы инженерным объектам узла, участвующим в связывании, были выделены и проинициализированы достаточные ресурсы.

Указатель интерфейса является ключом для доступа к большому количеству информации. Задавая такой указатель, можно получить тип интерфейса, коммуникационный адрес, по которому может быть инициализировано связывание через этот интерфейс, и другую информацию об ожидаемом поведении заглушек, связников и протокольных объектов в канале, что необходимо для успешности последующего связывания. Указатель является начальной точкой для вызова функций, необходимых при обработке ошибок; знание указателя интерфейса делает возможным контакты с соответствующим *релокатором*.

Однако сказанное не подразумевает, что вся перечисленная информация кодируется как часть указателя интерфейса: это могло бы сделать указатель слишком большим для обработки. Требование архитектуры состоит в том, что должно быть какое-либо предписание для получения необходимой информации, начиная с указателя интерфейса, но точный вид предписания в терминах декодирования и запросов к другим инженерным объектам может выбираться по-разному в различных проектах систем.

Кроме этого различия проектов, могут иметь место различия из-за существования кратных областей наименования и выделения указателей относительно этих областей. Поэтому пересечениям (или другим инженерным объектам в канале) может потребоваться преобразования указателей интерфейсов при переходах через границы областей.

8.5.4 Связывание

Имеется два вида инженерных связываний. В пределах кластера или между инженерными объектами, которые кооперируются в узле для обеспечения канала, существуют локальные связывания, которые обеспечиваются специфичными для системы методами. В архитектуре такие связывания рассматриваются как элементарные. С другой стороны, связывания, поддерживаемые каналами, обеспечивают соответствующие прозрачности распределения; они называются распределенными связываниями, и в общем случае в их создании участвуют некоторые взаимодействия между несколькими узлами для установления канала.

8.5.5 Установление канала

Для того чтобы установить связывание между двумя инженерными объектами, необходимо иметь возможность идентифицировать и описать интерфейсы этих объектов. Для идентификации и достаточно подробного для связывания описания инженерных интерфейсов используют указатели интерфейсов. Указатель интерфейса может быть передан через одно или несколько взаимодействий между инженерными объектами. Указатель интерфейса позволяет любому объекту, который его получил, инициировать связывание с этим интерфейсом без какой-либо дополнительной информации.

Инженерный объект, который хочет передать указатель одному из своих интерфейсов, запрашивает у ядра создание указателя интерфейса (через операцию на интерфейсе административного управления узла). Указатель интерфейса идентифицирует интерфейс и предоставляет информацию, необходимую для создания связывания с ним.

Тип инженерного интерфейса передает тип вычислительного интерфейса, который он реализует, и необходимую для его обеспечения конфигурацию заглушек, связников, протокольных объектов и пересечений (описанных ниже в 8.5.7). Когда связываются инженерные интерфейсы с совместимыми типами вычислительных интерфейсов, установление связывания согласует точную конфигурацию канала, необходимую для обеспечения всех участвующих интерфейсов. Может оказаться возможной оптимизация этой конфигурации. Например, если интерфейсы поддерживают одно и то же представление данных, то нет необходимости создавать конфигурацию канала с инженерными объектами для преобразования представления данных, хотя остается необходимость помещать в сообщения копии аргументов или результатов.

Связывания между интерфейсами создаются через взаимодействия объектов ядер (обычно тех узлов, в которых расположены интерфейсы). Следовательно, когда ядро создает указатель интерфейса, оно предоставляет инструкции о том, как нужно контактировать для установления связывания с указываемым интерфейсом. В терминах инженерного языка ядро идентифицирует *коммуникационный(е) интерфейс(ы),* через который(е) должно осуществляться взаимодействие ядро—ядро. Идентификация коммуникационного интерфейса может потребовать информацию о поддерживаемых на этом коммуникационном интерфейсе коммуникационных протоколах.

Обычно ядро назначает один из своих собственных коммуникационных интерфейсов для устанавливающих связывание взаимодействий, но допустим и более общий подход. Коммуникационный интерфейс (и его коммуникационный протокол), используемый в устанавливающем связывание взаимодействии, не подразумевает чего-либо о коммуникационном интерфейсе (и коммуникационном протоколе), используемом для обеспечения создаваемого связывания. Это аналогично тому, как телефонный разговор может быть использован для инициализации отправки бумажного документа по почте.

Указатель интерфейса передает очень сложную информацию, которая интерпретируется во всей системе ОРО. Следовательно, подробная структура указателя интерфейса является предметом отдельного стандарта. БМ-ОРО не предписывает, должен ли указатель интерфейса физически содержать описанную выше информацию или он является лишь ключом для доступа к этой информации через взаимодействия с другими инженерными объектами (например, имя должно разрешаться сервером имен).

В качестве примера установления канала рассмотрим установление канала потока между двумя инженерными объектами. Оно осуществляется в несколько шагов.

**Шаг 1**

Один из двух инженерных объектов инициирует конфигурацию канала путем взаимодействия со своим ядром. Синтаксис взаимодействия может иметь следующий формат:

**InitChannel** *(StreamChannel, producer/consumer, IFPC1, result IFrefStreamchannel)*

где *StreamChannel* есть тип *«StreamChannel»*, а *Streamchannel* — тип канала, который должен быть создан; *producer/consumer* указывает, что рассматриваемый инженерный объект будет играть для канала потока роль как создателя, так и потребителя; *IFPC1* — интерфейс инженерного объекта, к которому привязывается канал потока.

Когда происходит такое взаимодействие, ядро создает объект-заглушку, объект-связник и протокольный объект, соответствующие типу канала и роли объекта. Эти инженерные объекты связываются для создания первой части канала потока. Интерфейс уровня представления объекта-заглушки связывается с интерфейсом *IFPC1.* Затем объект-заглушка связывается с объектом-связ-

ником, который, в свою очередь, связывается с протокольным объектом. Результатом этого взаимодействия является указатель интерфейса *(IFrefStreamchannel)*. Этот указатель интерфейса будет передан инженерным объектам, которые захотят привязаться к каналу.

**Шаг 2**

Указатель интерфейса канала передается второму инженерному объекту. Этот объект взаимодействует со своим ядром для привязки к каналу с помощью следующего взаимодействия:

**BindChannel** *(StreamChannel, producer/consumer, IFPC2, IFrefStreamchannel)*

где *StreamChannel* — тип канала; *producer/consumer* указывает, что второй инженерный объект будет играть для канала потока роль как создателя, так и потребителя; *IFPC2* — интерфейс второго инженерного объекта, который привязывается к каналу потока.

Ядро по указателю интерфейса *IFrefStreamchannel* определяет тип канала и положение протокольных объектов других участников канала потока. Ядро создает объект-заглушку, объект-связник и протокольный объект, соответствующие типу канала и роли объекта. Эти объекты привязываются к уже установленной части канала потока и к второму инженерному объекту. Затем связники в канале взаимодействуют друг с другом для обеспечения коммуникации по всему каналу.

**Шаг 3**

Другие объекты могут привязаться к существующему каналу, используя то же самое взаимодействие **BindChannel** ( ).

8.5.6 Интерфейсы административного управления

Только сам объект может изменить свое поведение. Объект может отвечать на запросы модификации его поведения от приложения административного управления и может в последующем делегировать ответственность за некоторую часть управления этому приложению.

Для административного управления ресурсом требуется, чтобы было возможно вызывать операции административного управления для отдельных услуг, содержащих их инженерных объектов, содержащих эти объекты кластеров, содержащих эти кластеры капсул и обеспечивающих эти капсулы узлов.

8.5.7 Пересечения

Пересечения устанавливаются для удовлетворения идентифицированных в предпринимательской спецификации требований к федерации технологических или административных областей. Пересечения соответствуют понятиям «шлюз», «агент» или «монитор», которые стоят между двумя областями и допускают или осуществляют взаимодействия на основе устанавливающего федерацию контракта между администрациями.

В пределах технологической области объекты ядра имеют идентичные представления данных и функциональные возможности протоколов, наименований и адресации. Когда встречаются две технологические области, имеется возможность слияния областей, т. е. вычислительные объекты в каждой области расширяются так, что могут использовать как свою собственную технологию, так и технологию соседней области (или они переключаются на более общую технологию). Когда это происходит, технологическая граница растворяется и оказывается достаточно прозрачности доступа. Пересечения охватывают случай, когда невозможно изменить технологию одной или обеих областей и, следовательно, на границе должно быть пересечение, обеспечивающее преобразование протокола и перевод имен.

Передача протокола и данных через технологические границы осуществляется пересечением в оперативном режиме, и пересечение участвует во всех взаимодействиях объектов через границу. Для большей эффективности на каждой технологической границе следует использовать только одно пересечение, работающее в оперативном режиме (см. рисунок 7).

Пересечения административных границ целиком находятся в пределах административной области и осуществляют защиту предмета ответственности администрации. Например, пересечение административных границ может быть использовано при передаче информации безопасности, содержащей разрешения. Такое пересечение может использоваться до взаимодействия инфраструктуры вычислительных объектов. Административное пересечение может связаться с аналогичным пересечением в другой области для обмена такой информацией, как шифровальные ключи, и проверки административной информации до ее преобразования (см. рисунок 8). Преобразования, осуществляемые административными пересечениями, могут использоваться несколькими последовательными взаимодействиями между вычислительными объектами без дополнительного использования пересечений.




Рисунок 7 — Работающее в оперативном режиме пересечение — технологическая границы

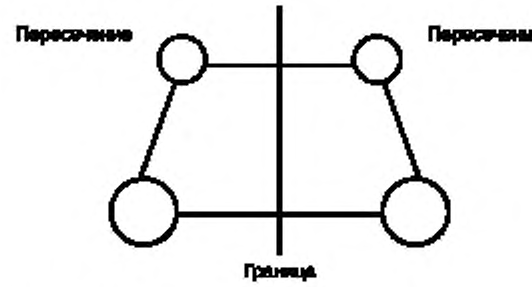



Рисунок 8 — Расщепленное пересечение — административная граница

Когда административная и технологическая границы совпадают, то могут быть использованы оба вида пересечений (комбинация рисунков 7 и 8). В качестве альтернативного варианта работающее в оперативном режиме пересечение может быть расщеплено так, чтобы для поддержки взаимосвязей в каждой административной области было по одному такому пересечению (см. рисунок 9).

В общем случае границы являются N-сторонними, так как в одной точке могут встретиться более двух подсистем; однако логически N-сторонняя граница всегда может моделироваться как N двусторонних границ и, следовательно, в архитектуре нужно рассматривать только двусторонние пересечения.




Рисунок 9 — Расщепленное пересечение — совпадение технологической и административной границ

Специальные условия возникают тогда, когда федерация требует, чтобы были связаны участники торга каждой области. Если участник торга доступен через оперативное пересечение, то он может быть использован для достижения последующих участников, но пересечение должно обеспечить специальные функции инициализации, чтобы позволить доступ в обоих направлениях по крайней мере к начальному участнику торга через него самого. Такую инициализацию можно осуществить, имея пересечение федеративных участников торга по обе стороны границы или имея пересечение, ставшее представителем участника торга, или создав участника торга в самом пересечении.

8.5.8 Точки соответствия

Построение структуры инженерной спецификации в виде кластеров, капсул и узлов и поддержка взаимодействий структурированными каналами позволяют ввести большое число интерфейсов, любой из которых может быть выбран в качестве точки соответствия, допускающей наблюдение и тестирование соответствия.

Разные интерфейсы относятся к разным видам соответствия. Интерфейс между протокольными объектами является точкой соответствия взаимодействия, предоставленной для обычных методов (аналогичных тестированию ВОС), основанных на наблюдении коммуникационного поведения. Большинство других интерфейсов являются внутренними для узла и представляют границы между программными модулями; они являются *программируемыми опорными точками* и допускают тестирование совместимости и переносимости программного обеспечения. Некоторые интерфейсы базовых инженерных объектов могут допускать другие виды тестирования соответствия — соответствия обмена или восприятия (т. е. правильность взаимодействия с реальным миром); они могут быть точками соответствия, в которых делаются утверждения о поведении для согласованности с требованиями предпринимательской и информационной спецификации.

**8.6 Технологический язык**

Технологическая спецификация описывает реализацию системы ОРО в терминах конфигурации технологических объектов, представляющих компоненты технического и программного обеспечения реализации. Она ограничена стоимостью и доступностью технологических объектов (технических и программных продуктов), удовлетворяющих данной спецификации. Эти объекты могут соответствовать реализуемым стандартам, которые являются эффективными шаблонами для технологических объектов. Таким образом, технологическая точка зрения обеспечивает связь между набором спецификаций с различных точек зрения и фактической реализацией, перечисляя стандарты,

используемые для предоставления необходимых базовых операций в спецификациях с других точек зрения, а цель технологической спецификации состоит в предоставлении дополнительной информации, необходимой для реализации и тестирования, путем выбора стандартных решений для базовых компонентов и средств коммуникации. Такой выбор необходим для полной спецификации системы, но он существенно удален от остального процесса проектирования.

Однако технологический выбор имеет некоторые последствия. Одной областью, в которой выбор технологической спецификации возвращает нас к другим аспектам проектирования системы, является обеспечение конкретного качества услуг. Выборы с технологической точки зрения определяют стоимость взаимодействий и тем самым, косвенно, качество услуг, которое может быть достигнуто поведением, определенным в спецификациях с других точек зрения.

Технологическая спецификация играет основную роль в процессе тестирования соответствия. Она идентифицирует точки соответствия в реальной системе, в которых тестер может проводить наблюдения поведения, и предоставляет информацию, необходимую для интерпретации наблюдений тестера в терминах словарей и понятий спецификаций системы с других точек зрения. Это позволяет, например, распознавать допустимые взаимодействия так, что их приемлемость может быть проверена относительно некоторого специфицированного поведения технологического объекта. Нужная для этой цели информация называется дополнительной информацией для тестирования реализации (ДИТР).

**8.7 Согласованность между точками зрения**

Спецификации системы с пяти точек зрения связаны утверждениями, определяющими отношения между их основными терминами и устанавливающими, что:

- спецификации относятся к одной системе и не являются независимыми;
- спецификации самосогласованные;
- наблюдаемое поведение в точках соответствия технологической спецификации может быть соотнесено с требованиями в спецификациях с других точек зрения.

Многие необходимые связи представлены неявно используемыми нотациями, являясь результатом соответствия между именами. Однако некоторые ключевые ограничения необходимо устанавливать явно. В архитектуре ограничения накладываются на отношения между терминами в самих языках точек зрения, устанавливая некоторые пределы отображений, которые могут быть введены. Большинство ограничений устанавливается для отношений между терминами в вычислительном и инженерном языках; они определяются так, чтобы создать согласованные интерпретации, когда компоненты системы (такие, как поддерживающие функции ОРО) специфицируются по отдельности.

Четкие отображения между точками зрения необходимы, если процессы идентификации интерфейсов и обеспечения прозрачностей должны обеспечиваться автоматически средствами разработки. Например, вычислительный объект может быть реализован как набор связанных инженерных объектов, но один инженерный объект не может представлять несколько вычислительных объектов; вычислительный интерфейс не может быть разделен на отдельные инженерные интерфейсы за исключением случая, когда они связаны функцией дублирования; вычислительные интерфейсы недвусмысленно идентифицируются инженерными идентификаторами. Ограничения такого вида позволяют гарантировать, что для обеспечения всего диапазона возможного вычислительного поведения будут применимы общие инженерные средства.

Для иллюстрации характера возможных соответствий предположим, например, что некоторая спецификация системы может быть представлена как набор взаимодействующих объектов точки зрения 1 (V1), как показано на рисунке 10. Та же самая система может быть описана с другой точки зрения (V2) как другой набор других взаимодействующих объектов V2, что показано на рисунке 11.

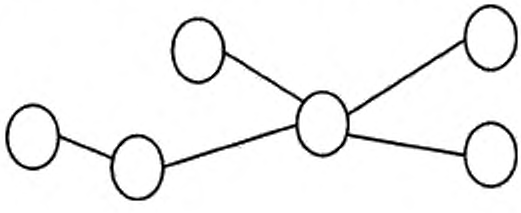


Рисунок 10 — Вид V1 системы

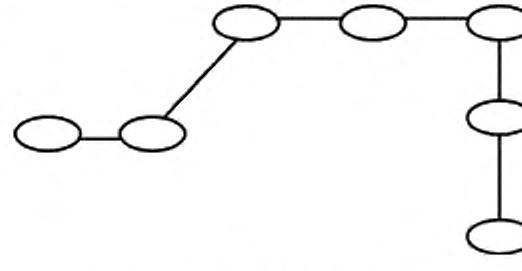


Рисунок 11 — Вид V2 системы

Так как имеется два описания одной и той же системы, то можно сгруппировать объекты двух представленных рисунков для того, чтобы проверить, что удовлетворяются правила соответствия между группами. Это показано на рисунке 12.

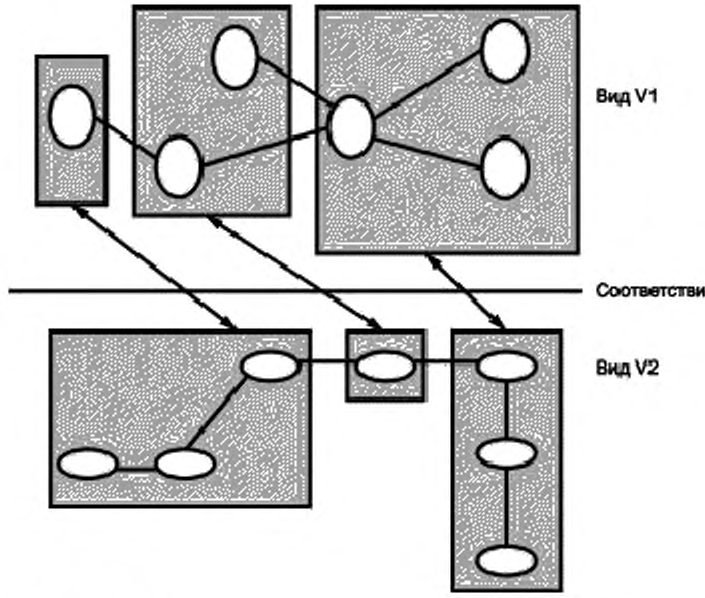



Рисунок 12 — Соответствие между разными точками зрения на систему

Если такое соответствие не может быть установлено, то эти две точки зрения не согласуются и должны уточняться до тех пор, пока будет продемонстрировано соответствие.

В общем случае, конфигурации сравниваемых объектов (конфигурации, заключенные в прямоугольники на рисунке 12) определены только для обнаружения соответствия между двумя спецификациями. Другими словами, для сравнения спецификации SpecA, написанной на данном языке точки зрения L1, с другой спецификацией SpecC той же самой системы, написанной на языке другой точки зрения L2, в общем случае необходимо:

а) преобразовать спецификацию SpecA в другую спецификацию на L2. Назовем ее SpecB. БМ-ОРО **не определяет** какие-либо алгоритмы преобразования.

Примечание — Иногда может оказаться удобным для осуществления указанного преобразования вывести из SpecA другую спецификацию той же самой системы, так же написанную на языке L1, которая эквивалентна SpecA, но лучше подходит для проверки соответствий. Это может быть сделано, например, объединением групп объектов, определенных в системе на рисунке 12, в другие объекты (прямоугольники);

б) проверить, что между SpecB и SpecC нет конфликтов.

Соответствия устанавливаются между спецификациями, выраженными на языках разных точек зрения, но не между терминами этих языков. Другими словами, нет непосредственного перевода с языка одной точки зрения на язык другой.

8.7.1 Согласованность предпринимательской точки зрения с другими

Предпринимательский язык должен служить в качестве основы для спецификации предпринимательских целей, которые прямо или косвенно должны быть отражены в спецификациях со всех точек зрения. Предпринимательская точка зрения описывает явным образом цели системы в контексте организации в терминах членов, ролей, действий, назначений, испоользования и политики.

Следовательно, спецификации с информиационной, вычислительной, инженерной и технологической точек зрения согласуются с предпринимательской спецификацией, правильно отражают все описанные в ней роли, деятельности и политики. Например, определенная в информацион-

ной спецификации динамическая схема должна подчиняться политике, описанной в предпринимательской спецификации. Различные роли, идентифицированные в предпринимательской спецификации, могут поддерживаться разными вычислительными объектами, имеющими разные требования к прозрачности. Таким образом, прозрачность, необходимая для каждой роли в предпринимательской спецификации, должна отражаться используемыми средствами прозрачности в инженерной спецификации. Требование гибкости или политика в предпринимательской спецификации могут привести к выбору конкретных технологий для реализации распределенной системы.

8.7.2 Соответствия между вычислительной и инженерной спецификациями

Существует соответствие между вычислительной и инженерной спецификацией, которое должно соблюдаться. Инженерная спецификация демонстрирует поведение, описанное в вычислительной спецификации. Пример такого соответствия показан на рисунке 13.

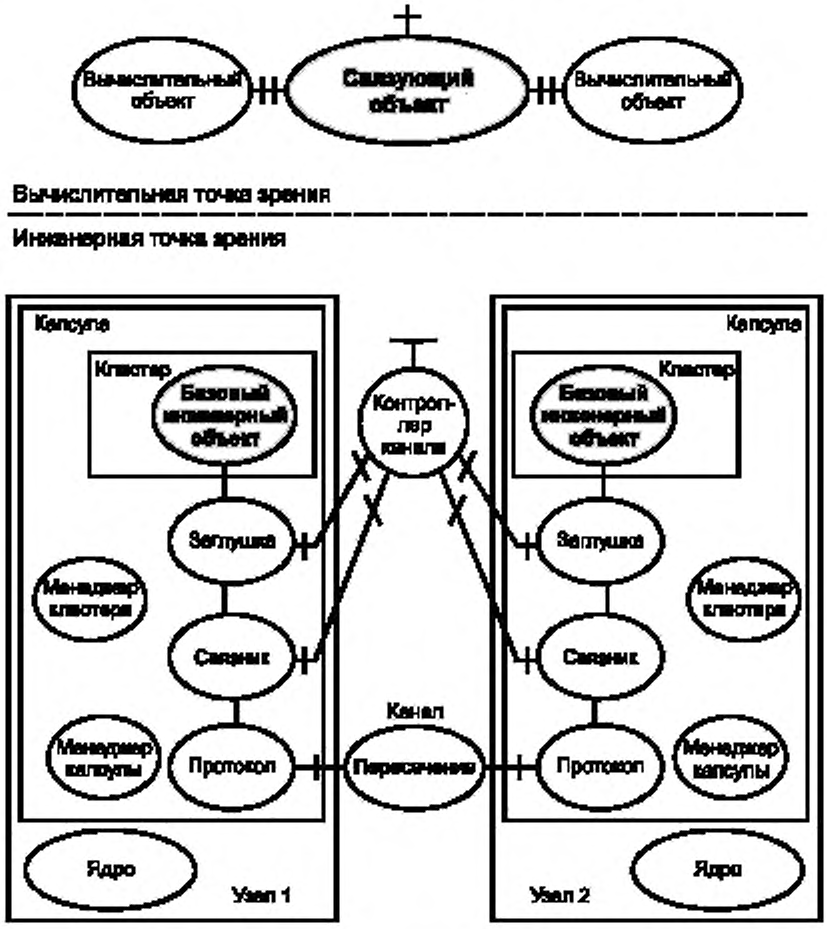



Рисунок 13 — Пример соответствия между вычислительной и инженерной точками зрения

Базовые инженерные объекты соответствуют вычислительным объектам. Базовые инженерные объекты группируются в кластеры, которые могут представлять, например, куски выполнимого кода С или С++. Кластеры организованы в капсулы, которые могут представлять, например, процесс UNIX. Капсула привязана к объекту ядра (представляющему, например, конкретную операционную систему), который принадлежит узлу (представляющему, например, рабочую станцию). Базо-

вые инженерные объекты поддерживаются дополнительными инженерными объектами, как показано на рисунке 13.

*Уточнение* вычислительных шаблонов в инженерных шаблонах соответствует представлению о *компиляции* программ для создания объектного кода. Уточнение инженерных шаблонов *в шаблонах кластеров* соответствует *понятию* связывания модулей для создания выполнимой программы. Понятие капсулы соответствует в большинстве операционных систем понятию *адресного пространства* или *процесса*.

Связующий объект в вычислительной спецификации, показанный на рисунке 13, соответствует конфигурации канала в инженерной спецификации. Ограничения среды, специфичные для связываемых интерфейсов (например, безопасность, КУ), учитываются при установлении канала между рассматриваемыми базовыми инженерными объектами.

Управляющий интерфейс связующего объекта в вычислительной спецификации соответствует интерфейсам к заглушкам и связникам в разных узлах в инженерном представлении. Может быть введен контроллер канала, отвечающий за отправку управляющий операций. Коммуникация между контроллером канала и заглушками, и связниками осуществляется через установленные для этой цели каналы (не показанные на рисунке 13). Поддерживающие инженерные объекты (например, объекты синхронизации) могут быть созданы для управления и контроля за набором взаимодействующих каналов.

Вычислительная спецификация описывает функциональные возможности системы в виде набора взаимодействующих вычислительных объектов. Инженерная спецификация ограничивается вычислительной в том плане, что она должна относиться к вычислительным объектам и их интерфейсам. Каждый вычислительный объект должен быть представлен как один базовый инженерный объект или группа таких объектов. В простейшем случае один инженерный объект эквивалентен одному вычислительному. Например, откомпилированная прикладная программа соответствует инженерному объекту, который после редактирования связей с другими инженерными объектами (заглушками, связниками и пр.) должен быть загружен как модуль в кластер. Такая конфигурация изображена на рисунке 14.

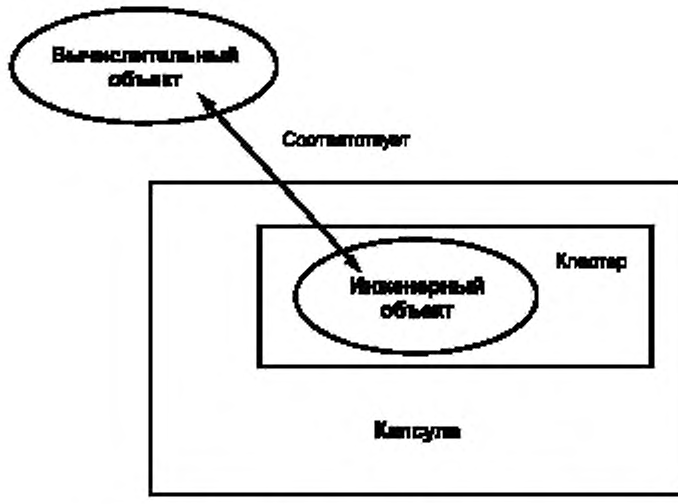



Рисунок 14 — Соответствие один-к-одному

Для каждого вычислительного интерфейса должен быть один соответствующий инженерный интерфейс за исключением случаев, когда задействованы прозрачности, дублирующие объекты. В последнем случае с разными идентификаторами инженерных интерфейсов может быть ассоциирован некоторый дополнительный интерфейс, допускающий дублирование, например, по соображениям производительности. Вычислительный интерфейс ассоциируется с набором *указателей инженерных интерфейсов*, соответствующих разным инженерным объектам. Деятельность этих инженерных объектов должна координироваться дублирующими объектами для того, чтобы гарантировать, что система поддерживает согласованное глобальное состояние. Пример такого случая приведен на

рисунке 15, где показаны два базовых инженерных объекта, расположенные в двух разных узлах, которые дублируют функциональные возможности, предоставляемые вычислительным интерфейсом. Так как инженерные объекты находятся в разных узлах, координация дублирования обеспечивается двумя дублирующими объектами, по одному в каждом узле, которые взаимодействуют через канал.

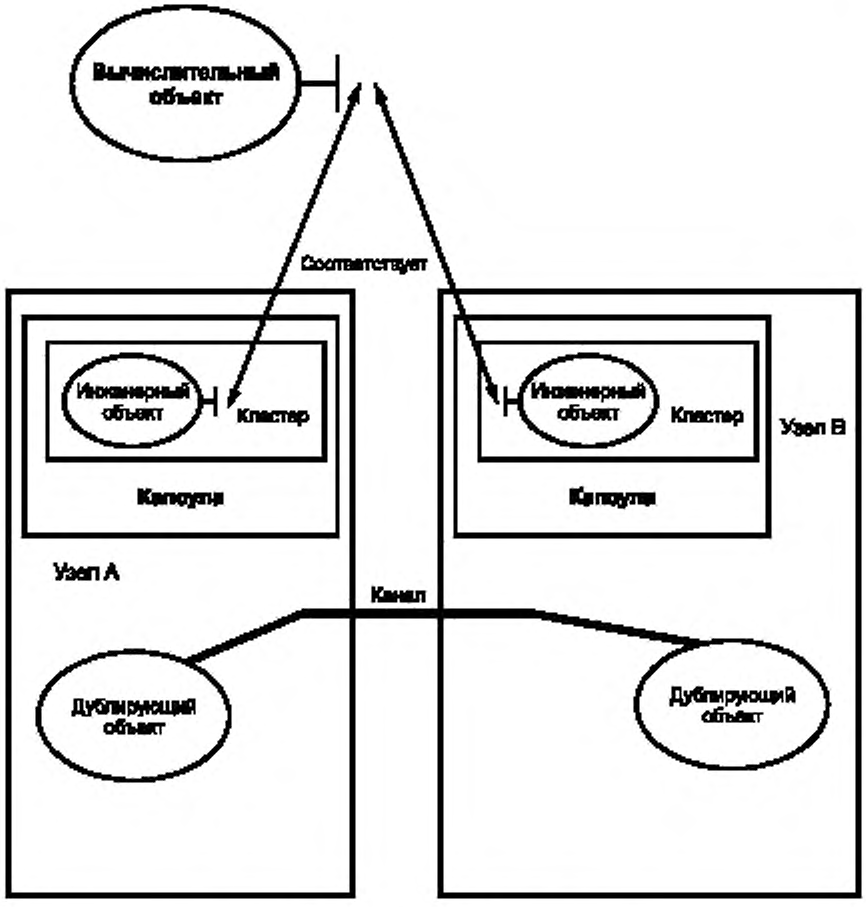



Рисунок 15 — Соответствие многие-одному

Процесс уточнения вычислительной спецификации до инженерной может состоять просто из идентификации подходящих поддерживающих объектов для наполнения канала, который представляет связующий объект вычислительной спецификации. В других случаях может происходить существенное преобразование шаблонов самих вычислительных объектов, при котором декларативные утверждения о поведении (например, ограничения синхронизации) заменяются точно выраженными функциями ОРО (например, функцией транзакции).

При достаточном знании конфигурации системы может быть проведена конкретная оптимизация для случаев, когда не требуется совершенно общая инженерная структура. Так разработчик может «закоротить» взаимодействия между базовыми инженерными объектами в одном и том же узле (например, используя, для повышения производительности, вызов локальных процедур) при условии, что это не повлияет на взаимодействие через интерфейсы, видимые объектам в других узлах. Таким способом архитектура справляется с распределением без требования предоставлять различные методы для локальных и удаленных взаимодействий и без снижения эффективности выполнения.

**8.8 Функции ОРО**

Общее описание набора функций ОРО приведено в ГОСТ Р ИСО/МЭК 10746-3. Эти функции являются либо основными, либо широко применимыми в конструкциях систем ОРО. Подробная спецификация этих функций будет предметом отдельных стандартов, которые можно будет комбинировать для создания спецификаций компонентов систем ОРО.

Полный набор функций ОРО подразделяется на четыре группы:

а) функции управления,

б) функции координации,

в) функции хранилища,

г) функции безопасности.

8.8.1 Функции управления

К функциям управления относятся:

- функция управления узлом,
- функция управления объектом,
- функция управления кластером,
- функция управления капсулой.

Функция управления узлом предоставляется ядром узла и направлена на управление функциями обработки, хранения и коммуникации в узле. Она обеспечивает:

- административное управление связками обработки;
- доступ к часам и административное управление таймером;
- создание канала и обработку указателей инженерных интерфейсов;
- реализацию шаблона капсулы и удаление капсулы.

Функция управления объектом предоставляется по мере необходимости любым объектом и позволяет получать контрольные точки и удалять объект.

Функция управления кластером предоставляется менеджером кластера и позволяет получать контрольные точки, восстанавливать, осуществлять миграцию, деактивировать или удалять кластер.

Функция управления капсулой предоставляется менеджером капсулы и позволяет реализовывать (включая восстановление и реактивацию), получать контрольные точки, деактивировать и удалять все кластеры в капсуле и удалять саму капсулу.

8.8.2 Функции координации

К функциям координации относятся:

- функция уведомления о событии,
- функция создания контрольной точки и восстановления,
- функция деактивации и реактивации,
- функция группирования,
- функция дублирования,
- функция миграции,
- функция транзакции,
- функция отслеживания указателей инженерных интерфейсов.

Функция уведомления о событии записывает и делает доступной историю событий. Событие создает взаимодействие с этой функцией для создания истории событий, а функция извещает зарегистрированных потребителей событий о доступности истории событий.

Функция создания контрольной точки и восстановления координирует создание контрольных точек и восстановление отказавших кластеров из этих *контрольных точек*. Она реализует политику, управляющую тем, когда должны создаваться контрольные точки кластеров, где должны храниться контрольные точки, когда и где кластеры должны восстанавливаться, какая контрольная точка восстанавливается.

Функция деактивации и реактивации координирует деактивацию и реактивацию кластеров. Она реализует политику, управляющую тем, когда кластеры должны деактивироваться, где должны сохраняться контрольные точки, связанные с деактивацией, когда и где должны реактивироваться кластеры, какая контрольная точка должна реактивироваться.

Функция группирования предоставляет необходимые механизмы для координации взаимодействий объектов в многостороннем связывании.

Функция дублирования имеет дело с частным случаем группы, члены которой поведенчески совместимы. Она обеспечивает необходимые средства для гарантии того, что группа воспринимает-

ся другими объектами как единственный объект, и позволяет добавлять и удалять членов группы. Эта функция может быть использована на уровне кластера вместе с функцией группирования для создания скоординированного набора групп копий, объекты каждого кластера образуют группу копий.

Функция миграции координирует миграцию кластера из одной капсулы в другую. Она может работать, либо дублируя кластер с помощью функции дублирования, либо деактивуя кластер и реактивируя его в другой капсуле с помощью функции деактивации и реактивации.

Функция транзакции координирует и контролирует множество транзакций для достижения заданного уровня видимости и неизменности. Какие действия учитываются транзакцией, является вопросом политики. Транзакция ACID является частным случаем общей функции транзакции, имеющей свойства элементарности (A-atomic), согласованности (C-consistent), изолированности (I-isolated) и устойчивости (D-durable).

Функция отслеживания указателей инженерных интерфейсов осуществляет мониторинг передачи указателей инженерных интерфейсов между инженерными объектами в разных кластерах для определения того, когда поддерживающая инфраструктура указателя уже более не требуется потому, что в других кластерах нет объектов, связанных с указанным интерфейсом.

8.8.3 Функции хранилища

К функциям хранилища относятся:

- функция сохранения,
- функция организации информации,
- функция перемещения,
- функция хранилища типов,
- функция торга.

Функция сохранения сохраняет данные.

Функция организации информации управляет хранилищем информации, описываемой информационной схемой, и позволяет изменять и обновлять как схему, так и хранилище, и обращаться к хранилищу с запросом.

Функция перемещения управляет хранилищем положений интерфейсов и функциями управления для кластеров, поддерживающих эти интерфейсы.

Функция хранилища типов управляет хранилищем типов спецификаций и типов связей.

Функция торга обеспечивает экспорт *поставщиком* предоставляемых услуг в виде информации об интерфейсе, через который предоставляются услуги, и импорт пользователем предоставляемых услуг, соответствующих конкретным требованиям.

8.8.4 Функции безопасности

Функции безопасности ориентированы на удовлетворение требований конфиденциальности, целостности, доступности и учета. К ним относятся:

- функция управления доступом,
- функция проверки безопасности,
- функция аутентификации,
- функция целостности,
- функция конфиденциальности,
- функция неопровержения,
- функция управления ключом.

Функция управления доступом предотвращает неавторизированные взаимодействия с объектом.

Функция проверки безопасности обеспечивает мониторинг и сбор информации об относящихся к безопасности действиях и последующий анализ информации с целью обзора политики безопасности, контроля и процедур.

Функция аутентификации предоставляет гарантии заявленной идентичности объекта.

Функция целостности выявляет и/или предотвращает неавторизованное создание, изменение или удаление данных.

Функция конфиденциальности предотвращает неавторизованное раскрытие информации.

Функция неопровержения предотвращает отрицание одним из объектов, участвовавших во взаимодействии, факта участия в этом взаимодействии.

Функция управления ключом предоставляет возможности для управления криптографическими ключами.

Эти функции предоставляют услуги, которые могут применяться как к самим объектам, так и к взаимодействиям между ними. Методы обеспечения услуг безопасности сами требуют защиты, так как для среды, в которой требуется обеспечение безопасности, характерны хитроумные и злонамеренные угрозы. Инженерная инкапсуляция может помочь обеспечить такую защиту. Во многих случаях услуги безопасности могут быть предоставлены без обязательного указания в вычислительных спецификациях.

**8.9 Прозрачности распределения ОРО**

Как описано в 8.5, инженерные объекты взаимодействуют друг с другом через заглушки, связники, протокольные объекты, пересечения и ядра. Инженерные объекты кооперируются для обеспечения прозрачности путем внесения однородности в некоторые аспекты распределения базовых инженерных объектов. Например, чтобы замаскировать различия в кодировании данных, они могут преобразовывать вызовы в обмен сообщениями, использующими общий формат данных.

Некоторым видам прозрачности требуются поддерживающие функции. Например, если инженерные объекты могут перемещаться с места на место, то требуется средство для записи и обнаружения текущего положения компонентов (функция перемещения).

Поддерживающие функции сами могут иметь требования прозрачности. Например, функция перемещения может дублироваться для повышения ее доступности.

Поддерживающие функции моделируются как инженерные объекты так, что архитектура предоставляет максимальную степень гибкости конфигурации и повторного использования архитектурных понятий при определении распределения этих функций. В реализации поддерживающие функции могут быть, например, размещены совместно друг с другом для эффективного взаимодействия или дублированы для доступности.

Прозрачность распределения является способностью скрывать свойства распределения от конечного пользователя и специфицируется на предпринимательском, информационном и вычислительном языках. Стандарты на «композиции компонентов» должны содержать точные указания по использованию функций и других базовых компонентов для обеспечения прозрачностей.

8.9.1 Прозрачность доступа

Прозрачность доступа допускает взаимодействия через неоднородные архитектуры компьютеров и языки программирования.

Прозрачность доступа является критической при построении распределенных систем, использующих неоднородные компьютерные архитектуры, языки программирования и пр.

8.9.2 Прозрачность отказа

Прозрачность отказа *скрывает* от вычислительного объекта отказ и возможное восстановление других объектов и его самого, делая возможной устойчивость относительно неисправностей. Она может быть обеспечена соответствующей инфраструктурой. В противном случае, она поддерживается функцией создания контрольной точки и восстановления или функцией дублирования вместе с функцией перемещения.

Услуга, которая основывается на контрольной точке для прозрачности отказа, должна предоставлять клиенту, как часть этой услуги, средства для выявления того, что был осуществлен повторный старт с контрольной точки и что состояние полученной клиентом информации об услуге может не относится к указанному моменту времени. Если имеется требование согласованности между несколькими вычислительными объектами, то должна быть задана прозрачность транзакции.

8.9.3 Прозрачность положения

Прозрачность положения скрывает от вычислительного объекта положения в пространстве, в которых находятся взаимодействующие с ним вычислительные объекты. Тем самым подразумевается, что интерфейсы могут быть идентифицированы и достигнуты без спецификации их положения в пространстве.

8.9.4 Прозрачность миграции

Прозрачность миграции скрывает от вычислительного объекта факт его перемещения. Она обеспечивается функцией миграции.

Прозрачность миграции может быть скомбинирована с прозрачностями постоянства или отказа таким образом, что кластер не реактивируется в его исходном положении, а шаблон кластера передается непосредственно в новое положение и кластер реактивируется там.

8.9.5 Прозрачность постоянства

Прозрачность постоянства скрывает от вычислительного объекта выделение и изъятие ресурсов у кластеров или их шаблонов и обеспечивает совместное использование ресурсов. Она поддерживается функциями реактивации и деактивации.

Базовый инженерный объект в кластере может взаимодействовать с другими объектами. Когда кластер деактивируется, базовый инженерный объект сохраняется с замороженными деятельностями в шаблоне кластера и не может взаимодействовать с другими объектами. Прозрачность постоянства скрывает деактивацию и реактивацию шаблонов кластеров и таким образом базовые инженерные объекты всегда предоставляются доступными для взаимодействия. Это подразумевает, что объект имеет свое собственное время жизни, не зависящее от обеспечивающей его среды.

8.9.6 Прозрачность перемещения

Прозрачность перемещения скрывает от вычислительного объекта тот факт, что интерфейсы, с которыми связан этот объект, изменили свое положение. Это подразумевает возможность переустановки связывания в случае необходимости. Прозрачность перемещения поддерживается функцией перемещения.

8.9.7 Прозрачность дублирования

Прозрачность дублирования маскирует использование группы вычислительных объектов для обеспечения одного вычислительного интерфейса. Она поддерживается функциями дублирования и перемещения.

8.9.8 Прозрачность транзакции

Координация транзакций включает в себя составление расписания, мониторинг и возврат действий, рассматриваемых в этих транзакциях. Для достижения такого контроля требуются взаимодействия между вычислительными объектами, участвующими в выполнении рассматриваемых действий, и вычислительными объектами, реализующими функцию транзакции. В общем случае невозможно координировать рассматриваемые действия, просто конфигурируя объекты в каналах для перехвата этих действий. В частности, такие внутренние действия, как начало и завершение транзакции, не выявляются при мониторинге взаимодействий между вычислительными объектами.

Следовательно, мониторинг и управление транзакциями обычно будут требовать вычислительно видимых взаимодействий между вычислительными объектами, представляющими информационные объекты или динамические схемы (т. е. прикладные объекты), с одной стороны, и вычислительными объектами, предоставляющими функцию транзакции, с другой. Однако обработка транзакции является очень сложной, и нежелательно усложнять спецификацию прикладных функциональных возможностей дополнительными сложными взаимодействиями, необходимыми для управления транзакциями.

Прозрачность транзакции предоставляет автоматический процесс, который позволяет превратить вычислительную спецификацию без управления транзакциями в спецификацию с таким управлением.

Характер нужных для этого уточнений (и степень участия разработчика спецификации в этом процессе) зависит от конкретного используемого механизма транзакций. Обычно будут необходимы дополнительные вычислительные интерфейсы для связывания с вычислительными объектами, участвующими в функции транзакции, и взаимодействия через эти вычислительные интерфейсы для координации составления расписаний, мониторинга и возврата действий, рассматриваемых функцией транзакции. Необходимо будет расширить поведение вычислительных объектов для добавления возвращающих действий, а интерфейсы, предоставляющие доступ к обычным функциональным возможностям, могут быть расширены или заменены.

## 9 Оценка соответствия

### 9.1 Оценка соответствия и процесс разработки

Разработка продукта начинается от начальной реализации одного или нескольких требований к системе ОРО до окончательного создания экземпляра системы ОРО, удовлетворяющего этим требованиям. Процесс разработки потенциально включает в себя создание ряда спецификаций. Одна спецификация может отвечать за создание нескольких последующих спецификаций, использующих по одному из нескольких видов шагов («преобразований»), включая в себя:

- перевод и
- уточнение.

Спецификация выражена на каком-то естественном или формальном языке. Перевод создает спецификацию с тем же самым смыслом (обычно, на другом языке). С другой стороны, уточнение создает спецификацию с новыми подробностями, которые предназначены для более детальной спецификации продукта. Если спецификация характеризуется множеством потенциальных продук-

тов, которые могут ей удовлетворять, то перевод оставляет это множество неизменным, а уточнение приводит к его подмножеству.

После установки и приемки система ОРО переходит в стадию операционного использования, при котором встречаются потребности, выраженные в устанавливающих эти требования документах. Способность продукта удовлетворить эти требования зависит от ряда приемов, использованных при разработке. Стандарты по обеспечению «качества» описывают согласованный набор таких приемов, относящихся к организации людей, документации и стадий жизненного цикла, на которых они применяются. Обычно какие-то измерения качества осуществляются на каждой стадии, и проводятся соответствующие изменения, если обнаруживается недостаточное качество. Оценка соответствия обеспечивает измерение качества обычно на той фазе, на которой реализуется спецификация. Однако она может быть использована или применена и на других стадиях.

**9.2 Оценка соответствия — рассматриваемые взаимосвязи**

Относящиеся к соответствию взаимосвязи между спецификациями и фактическими реализациями подразделяются на две группы:

- взаимосвязи между спецификациями и фактическими реализациями (соответствие) и
- взаимосвязи между самими спецификациями (согласованность, уточнение, последовательность и внутренняя корректность).

Соответствие является взаимосвязью между спецификацией и фактической реализацией. Оно имеет место, когда конкретные требования спецификации (требования соответствия) удовлетворяются реализацией. Оценка соответствия — процесс, при котором эта взаимосвязь определяется.

Согласованность является отношением между двумя спецификациями А и В, которое имеет место, когда спецификация А устанавливает требования, удовлетворяемые спецификацией В (когда В согласуется с А).

В процессе разработки продукта соответствие фактической реализации не всегда оценивается относительно спецификации «нижнего уровня» (т. е. реализуемой). Для этого можно использовать спецификацию «верхнего уровня» (например, ту, уточнение которой привело к реализуемой спецификации). В любом случае важно соответствие между последовательными спецификациями. Такое соответствие определяется правилами уточнения.

Две спецификации связаны отношением уточнения, когда одна из них является уточнением другой и все продукты, которые могут удовлетворять уточнению, также соответствуют спецификации, из которой оно было получено. Это гарантирует, что все ограничения более общей спецификации присутствуют в уточненной спецификации.

Спецификации не всегда взаимосвязаны тем, что были выведены из одного и того же набора требований. Иногда они могут разрабатываться с двух и более независимых точек зрения (например, спецификации на языках разных точек зрения). В таком случае последовательность спецификаций может оказаться под вопросом: важно, чтобы требования одной спецификации не противоречили требованиям другой. Последовательность является отношением между двумя спецификациями, которое имеет место, когда может существовать по крайней мере один пример продукта, соответствующего обеим спецификациям.

Спецификация является корректной, когда нет противоречия между установленными ею свойствами и свойствами, которые она требует неявно (например, можно ожидать, что спецификация протокола будет свободна от взаимной блокировки), и когда может существовать по крайней мере один пример продукта, который может ей соответствовать (т. е. она не является внутренне противоречивой).

Оценка соответствия является определением этих взаимосвязей либо путем тестирования (соответствие), либо путем проверки спецификаций (согласованность, проверка уточнения, последовательности и внутренней корректности).

**9.3 Точки соответствия и относящиеся к ним понятия**

Когда, используя аттестационное тестирование, оценивается соответствие реализации спецификации ОРО, ее поведение проверяется (путем воздействий и наблюдений за результирующими событиями) в конкретных точках (взаимодействия). Используемые точки называются «точками соответствия» и обычно выбираются из числа тех точек, положение которых специфицировано в архитектуре БМ-ОРО. Эти потенциальные точки соответствия называются опорными точками.

Для согласованности с БМ-ОРО стандарты ОРО обязаны содержать утверждения о соответствии, которые должны (в числе прочего) устанавливать, какие опорные точки должны использоваться при аттестационном тестировании. Подразумевается, что каждая установленная в спецификации точка соответствия является определенной в архитектуре опорной точкой.

Кроме приведенных выше понятий точки соответствия, опорной точки и точки взаимодействия, в методологии аттестационного тестирования ВОС (ГОСТ Р ИСО/МЭК 9646-1) определено понятие пункта контроля и наблюдения (ПКН), который является не точкой, где определено соответствующее поведение (т. е. точкой соответствия), а точкой, где контролируется и наблюдается поведение в точке соответствия.

Точка соответствия устанавливается в спецификации ОРО, о соответствии которой должно быть заявлено. Пункт контроля и наблюдения устанавливается при документировании конкретного средства тестирования реализации. Разные методы могут привести к идентификации разных наборов ПКН.

Система обработки, используемая для осуществления тестирования, может быть описана в терминах ОРО. Следует учитывать, что ПКН тестируемого приложения являются, по существу, точками соответствия тестирующей системы. Опорные точки тестирующей системы являются потенциальными ПКН, которые могли быть предварительно установлены для рассмотрения при определении метода тестирования.

Основы и архитектура ОРО в ГОСТ Р ИСО/МЭК 10746-2 и ГОСТ Р ИСО/МЭК 10746-3 не определяют ПКН или потенциальные ПКН. Вместо них могут быть использованы понятия точек соответствия и опорных точек тестирующей системы. Размещение потенциальных ПКН только в опорных точках гарантирует, что тестирующие системы сами являются системами ОРО и ограничивают диапазон методов тестирования, которые может потребовать спецификация. По мере распространения ОРО термин ПКН может стать излишним, так как практически нет различия между опорными точками и потенциальными ПКН.

**9.4 Спецификации соответствия ОРО**

9.4.1 Уровень абстракции

При тестировании реализации некоторой спецификации ОРО тестеру может потребоваться дополнительная информация. Такая информация называется дополнительной информацией для тестирования реализации (ДИТР) и содержит сведения, которые требуются для связи понятий заявки о соответствии реализации (ЗСР) ОРО с их реализацией.

Уровень абстракции, на котором специфицирована точка соответствия, влияет на объем работы, необходимой для обеспечения отображения между ДИТР, ограничениями, установленными в процессе реализации, и степенью независимости от реализации.

Существует требование к дополнительной информации о спецификации для отображения реализации на высоком уровне абстракции. Так как предоставление этой информации и процесс реализации включают в себя связь терминов спецификации с реализуемыми объектами, то предоставление отображения ДИТР может потребовать усилий, пропорциональных тем, которые требовались в процессе реализации. Для точек соответствия, определенных на высоком уровне абстракции, это может потребовать неразумно высокую цену и сделать тестирование неприемлемым.

Все ПКН должны иметь явную и приемлемую реализацию в фактической реализации — характеристика, не обязательно совместимая с другими аспектами спецификации. Следовательно, спецификация точек соответствия (для которых должны быть выделены ПКН) представляет собой ограничение на процесс реализации, которое будет очень сильным, если определено большое количество точек. Менее привязанная к реализации спецификация точек соответствия (т. е. более высокая степень их абстракции) приводит к большей несогласованности при обеспечении их явного представления в реализации.

Использование в спецификации низкого уровня абстракции точек соответствия подразумевает абстрагирование от меньшего количества деталей реализации и, следовательно, спецификация оказывается в меньшей степени независящей от реализации. Большее количество подробностей также подразумевает, что в процессе оценки соответствия могут потребоваться большие усилия для оценки каждой детали (хотя в ДИТР потребуется меньше интерпретации).

9.4.2 Использование кратных опорных точек

Определение в спецификации многих точек соответствия может потребоваться в целях, отличных от простого увеличения числа предписанных аспектов реализации.

Спецификация объекта может определить несколько различных типов интерфейсов со средой. Например, могут быть интерфейсы с людьми, другими объектами, средствами коммуникации и хранения. Каждый из них может характеризоваться опорными точками разных классов (воспринимаемыми, программируемыми, взаимодействия или обмена), каждый из которых имеет разные аттестационные тестирования. Это подразумевает, что должны быть идентифицированы разные точки соответствия, когда присутствуют интерфейсы разных типов.

Даже если рассматривать только один тип интерфейса, реализация объекта может быть предназначена для доступа в физически и логически разных точках. Связь с ними разных точек соответствия учитывает эти детали реализации и, таким образом, понижает уровень абстракции, на котором установлены точки соответствия (так как предписывается больше деталей реализации), что приводит к описанным в 9.4.1 последствиям. Раздельные точки соответствия также приводят к предписанию «проектировать» раздельные точки доступа (и тем сам гарантировать их реализацию), а не к описанию доступа.

Когда объект специфицируется в терминах нескольких компонентов с разными интерфейсами между ними, то не всегда понятно, служат ли внутренние подробности для описания внешнего поведения реализации или они являются предписанием (т. е. обязательным способом построения объекта). Однако, когда компоненты рассматриваются как предписываемые детали, то необходимо определить дополнительные точки соответствия (т. е. для каждого интерфейса между компонентами).

**9.5 Влияние языков точек зрения на соответствие**

Использование языков точек зрения в спецификациях ОРО имеет ряд последствий:

- тестер соответствия проводит оценку в терминах инженерной спецификации тестируемой реализации, что вместе с другими факторами требует от клиента испытательной лаборатории предоставления дополнительной информации для тестирования, включая:

1) ДИТР + ЗСР, относящиеся к реализации понятий и структур предпринимательской спецификации в инженерной спецификации;

2) ДИТР + ЗСР, относящиеся к реализации понятий и структур информационной спецификации в инженерной спецификации;

3) ДИТР + ЗСР, относящиеся к реализации понятий и структур вычислительной спецификации в инженерной спецификации;

4) ДИТР + ЗСР, относящиеся к реализации понятий и структур инженерной спецификации в технологической спецификации (это должно предоставляться как часть технологической спецификации);

- каждый язык точки зрения допускает раздельную спецификацию разных типов требований, например предпринимательские цели, системный проект и используемая технология могут рассматриваться по отдельности — этим подразумевается возможность параллельной разработки отдельных спецификаций;

- проверка параллельной разработки спецификаций требует время от времени оценки последовательности спецификаций относительно друг друга в процессе разработки.

Так как спецификации с разных точек зрения могут быть независимыми и не связанными уточнением, то проверка уточнения не обеспечивает в ходе разработки полной проверки параллельных развитий разных спецификаций. Тем не менее такая проверка весьма желательна. Это требование удовлетворяется проверкой последовательности (см. 8.7). Объекты, описанные в спецификациях разных точек зрения, часто образуют единое рассматриваемое пространство: например, информационный объект в информационной спецификации может появиться как аргумент функции в интерфейсе вычислительной спецификации; точки соответствия предпринимательской спецификации могут появиться как точки соответствия в вычислительной и инженерной спецификациях, и т. д.

**9.6 Деятельность по оценке соответствия**

Процесс оценки соответствия включает в себя следующие виды деятельности:

- проверка уточнения (между спецификациями),
- проверка внутренней корректности одной спецификации,
- проверка последовательности нескольких спецификаций,
- тестирование реализации (соответствие реализации и спецификации).

## 10 Административное управление системами ОРО

Система ОРО включает в себя ряд приложений ОРО и поддерживающие услуги. Последние включают в себя такие услуги, как обработка, хранение файлов, доступ пользователей и коммуникация, которые предоставляются традиционными операционными системами в централизованных системах, но они включают в себя и такие, необходимые для распределенной системы услуги, как справочник, административное управление именами, торг и услуги, обеспечивающие прозрачность

распределения. Всем этим услугам и приложениям ОРО требуется административное управление; характер нужных функций административного управления будет зависеть от рассматриваемых услуг и приложений.

Само административное управление системой ОРО будет осуществляться одним или несколькими приложениями административного управления, взаимодействующими с системными услугами и приложениями ОРО через интерфейсы, которые предоставляют уровень административного управления с подробностью, определенной спецификацией рассматриваемого приложения административного управления. При необходимости это приложение будет взаимодействовать с функциями ОРО через определенные для него функции административного управления.

Приложения административного управления в системе ОРО будут отражать взаимоотношения между функциями административного управления; например, тот факт, что может потребоваться переконфигурация для поддержки необходимого КУ, демонстрирует, что функции административного управления являются взаимосвязанными. Обычно в число приложений административного управления включается учет, отражая тот факт, что большинство услуг нуждается во внутреннем учете, даже если пользователи не платят за услуги.

Функции безопасности не включают в число функций административного управления, хотя имеется в виду, что функции административного управления будут предметом политики контроля, например аутентификации, а управление доступом очень важно для защиты от управляющих действий над компонентами услуг или приложений со стороны неавторизованных пользователей или управляющих. Таким образом, все услуги в среде ОРО должны использовать услуги безопасности, если им нужно использовать услуги коммуникации, хотя административное управление безопасностью не является услугой административного управления, необходимой для управления другими услугами.

**10.1 Области административного управления**

Для того чтобы справиться со сложностью административного управления и проблемами масштаба (особенно в больших системах ОРО), важно обеспечить общий подход к делению на части всего административного управления. В среде ОРО имеется несколько сосуществующих точек зрения на административное управление и границ ответственности, основанных на разных структурных критериях.

Области административного управления предоставляют гибкие и удобные средства спецификации границ административной ответственности и уполномоченных, отражающие эти различные точки зрения. Область идентифицирует множество объектов, каждый из которых связан характеризующим отношением с контролирующим область объектом. Объекты-члены могут быть ресурсами, рабочими станциями, модемами, процессами и пр. в зависимости от целей, для которых определена конкретная область административного управления. Контролирующий объект имеет атрибут, идентифицирующий объекты-члены. Объект рассматривается как член этого множества, если его идентичность известна контролирующему объекту области административного управления.

Область административного управления позволяет проводить общую политику контроля за множеством управляемых объектов, предоставляя основу для борьбы со сложностью крупномасштабных систем ОРО. Эти области упрощают деятельность административного управления, так как политика и множество управляемых объектов могут быть изменены через взаимодействия с единственным объектом — контролирующим объектом области административного управления, а не заставлять управляющих взаимодействовать индивидуально с каждым из многочисленных управляемых объектов в среде ОРО.

Области административного управления не инкапсулирует объекты-члены: внешние объекты могут взаимодействовать непосредственно с объектом в области. Области административного управления являются постоянно присутствующими, даже если в некоторый момент времени они не содержат объектов, так как должна быть возможность создания пустой области административного управления и последующего включения в нее объектов.

**10.2 Политика административного управления**

В предпринимательских терминах характеризующее отношение области воплощает связанную с областью политику. Таким образом, области административного управления предоставляют средства для спецификации политики административного управления для группы управляемых объектов, а не для каждого объекта в отдельности. Глобальные цели административного управления и внешние ограничения, связанные с законодательством, регулированием или политикой более высоко-

го уровня, являются двумя сторонами политики для области административного управления. Это показывает, что при формальной спецификации некоторой конкретной политики могут возникнуть трудности.

Внутренние причины устанавливают ограничения на операции, которые могут осуществляться над объектом в области административного управления. Они могут быть выражены декларативно в терминах обязательств потенциальных членов области.

Важным аспектом политики административного управления является спецификация того, какие управляющие операции может осуществлять менеджер над управляемыми им объектами. Правило доступа является взаимоотношением полномочий, которое устанавливает набор допустимых взаимодействий между областью управляющих и областью управляемых объектов. Например, все члены области «СистемныеПрограммисты» могут стартовать и останавливать объекты в области «УслугиОтдела». Допустимые взаимодействия могут быть подмножеством взаимодействий административного управления, определенных интерфейсами к объектам области.

**10.3 Моделирующие структуры административного управления**

Взаимосвязи области могут быть использованы для моделирования структур административного управления. По определению, две области административного управления являются разделенными, если они не имеют общих объектов. Две области административного управления перекрываются, если существуют объекты, являющиеся членами обеих областей. Примером разделения административного управления является шлюз, связывающий две сети и находящийся под управлением центров обеих сетей. Это может быть осуществлено указанием объекта из обеих областей административного управления.

Может иметь место неявное перекрытие двух областей административного управления, содержащих управляемые объекты разных типов, но относящихся к одним и тем же категориям реального мира. Примером являются области планирования и сопровождения, в которых исключение рабочей станции из услуг в области сопровождения делает ее недоступной в области планирования. Неявное перекрытие происходит тогда, когда имеется функциональное разделение административного управления на разные области.

## 11 Использвание стандартов в системах ОРО

Цель настоящего раздела — на простых примерах показать использование стандартов, отличных от специфичных стандартов ОРО, в системах ОРО и там, где такое использование определяется в спецификациях систем ОРО. Для этого приведен пример, описывающий, в общих чертах, содержание спецификаций с каждой точки зрения, и для каждой спецификации обсуждаются типы стандартов, относящихся к встретившимся требованиям.

Пример системы показан на рисунке 16. Она представляет собой систему сервера с локальным оператором, связанную средствами телекоммуникации с рабочей станцией системы клиента с двумя локальными операторами. В описании системы рассматривается одно приложение рабочей станции, которое предоставляет локальным операторам доступ к приложению предоставления картинок на системе сервера. Предполагается, что как система сервера, так и система рабочей станции предоставляют и другие услуги, и эти системы принадлежат разным организациям, т. е. подчиняются разным административным политикам.

Описания с разных точек зрения идентифицируют требования к общим спецификациям. Эти общие спецификации будут, в общем случае, соответствовать стандартам разных уровней.

**11.1 Предпринимательская точка зрения**

11.1.1 Предпринимательская спецификация

На рисунке 17 показана модель объектов для предпринимательской спецификации системы, представленной на рисунке 16, где:

- пользователь картинки *n* — предпринимательский объект, представляющий человека и играющий роль «Пользователь картинки *n*»;
- поставщик картинки — предпринимательский объект, представляющий систему ИТ и играющий роль «Поставщик картинки»;
- соучастник картинки — предпринимательский объект, представляющий систему ИТ и играющий роль «Соучастник картинки».

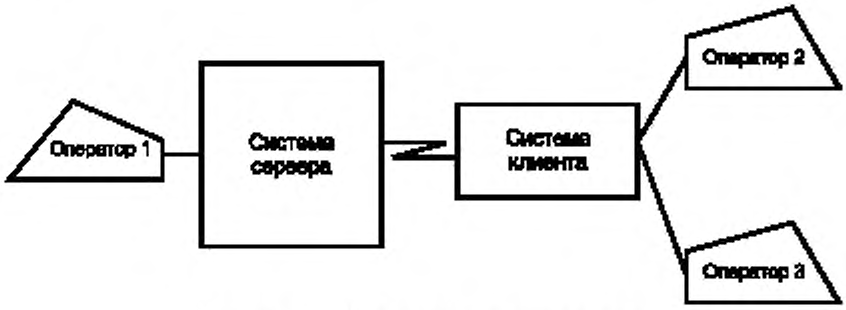



Рисунок 16 — Пример конфигурации системы

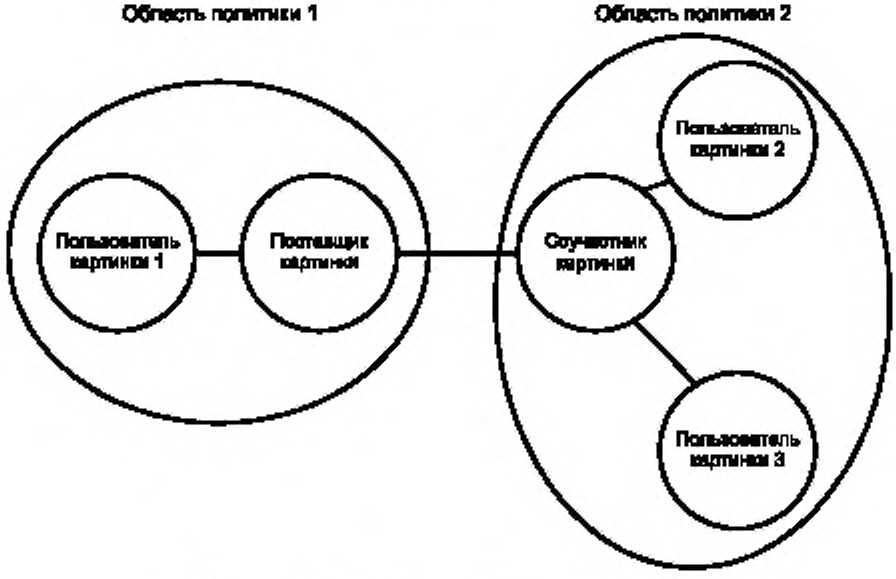



Рисунок 17 — Предпринимательская точка зрения

Предпринимательская цель, представленная конфигурацией предпринимательских объектов, управляет характером поведения, связанного с выполняемыми этими объектами ролями. Например, цель может потребовать, чтобы пользователям картинки 2 и 3 отображалась фиксированная картинка или чтобы соучастник картинки поддерживал работу с информацией картинки, добавление такой информации и возврат ее поставщику картинки.

Таким образом, предпринимательская спецификация определяет цели конфигурации предпринимательских объектов и, следовательно:

- информацию картинки, необходимую пользователям 1—3;
- взаимодействия пользователя картинки 1 с поставщиком картинки;
- взаимодействия пользователей картинки 2 и 3 с соучастником картинки;
- взаимодействия между соучастником и поставщиком картинки;
- политику (включая безопасность), управляющую взаимодействиями между предпринимательскими объектами;
- требования КУ для пользователей картинки 1—3.

11.1.2 Применение стандартов

В предпринимательском описании могут быть разработаны общие спецификации для специфических деловых и производственных функций, таких, как *политика безопасности,* которая может быть выражена в терминах управления доступом и степени защиты. Такие стандарты должны относиться к рассматриваемой прикладной области.

**11.2 Информационная точка зрения**

11.2.1 Информационная спецификация

С информационной точки зрения система представляется в терминах:

- классов информационных объектов, задействованных в приложении;
- информационных деятельностей (изменений состояний информационных объектов), которые образуют приложение;
- ограничений на изменения состояний информационных объектов.

Информационная спецификация включает в себя:

1) спецификацию самих классов информационных объектов, например:

- шаблоны информационных объектов «картинка», которые определяют доступные классы информационных объектов «картинка»,
- шаблоны объектов отображения «картинки», которые определяют доступные классы объектов отображения «картинки». Шаблон объектов отображения «картинки» определяют композицию информационных объектов «картинка»;
- шаблон информационных объектов «запрос», где информационный объект «шаблон» содержит информацию, необходимую для запроса отображения, включая управляющую информацию (например, безопасность),
- *информационный* объект *«управление доступом»,* содержащий информацию, необходимую для подтверждения информационного объекта «запрос»;

2) ограничения на конфигурации информационных объектов, например:

- множество классов шаблонов информационных объектов «картинка» для соучастника то же самое, что и для поставщика картинки;
- множество классов шаблонов объектов отображения «картинки» для пользователя картинки 1 является подмножеством множества для поставщика картинки;
- множество классов шаблонов объектов отображения «картинки» для пользователей картинки 1 и 2 является подмножеством множества для соучастника картинки.

Для рассматриваемой системы (упрощенным) примером информационной деятельности могло бы быть:

1) создание информационного объекта «запрос» со статусом «недействительный»,

2) взаимодействие информационного объекта «запрос» с информационным объектом «управление доступом» и изменение статуса информационного объекта «запрос» на «действительный»,

3) взаимодействие информационного объекта «запрос» с информационными объектами «картинка» и создание объектов отображения «картинок».

11.2.2 Применение стандартов

В информационном описании могут быть разработаны общие спецификации информационных объектов. Стандарты будут относится к рассматриваемой прикладной области.

**11.3 Вычислительная точка зрения**

11.3.1 Вычислительная спецификация

На рисунке 18 показана модель объектов для вычислительной спецификации системы, представленной на рисунке 16. Модель объектов включает в себя конфигурацию:

1) вычислительные объекты «пользователь 1—3», «отображение картинок 1 и 2», «составление картинки» и «базу данных картинок»;

2) элементарные связывания, для которых нет явных утверждений о контрактах среды, между интерфейсами:

- «пользователь 1» и «отображение картинки 1» ($по_1$),
- «пользователи 2 и 3» и «отображение картинки 2» ($по_2$ и $по_3$),
- «отображение картинки 1» и «составление картинки» (ос),
- «база данных картинок» и «составление картинки» (бс);

3) составное связывание, включающее в себя связующий объект («связывание») между интерфейсами «отображение картинки 2» (со) и «составление картинки» (сс), с утверждением о контрактах среды;

4) деятельности, установленные для конфигурации вычислительных объектов, которые реализуют требования, установленные в предпринимательской и информационной спецификациях. Интерфейсы являются опорными точками для соответствия в терминах ОРО.

С вычислительной точки зрения система представляется в терминах:

1) спецификаций вычислительных объектов в терминах абстрактных спецификаций операций и поведения, которые объекты поддерживают на своих интерфейсах;

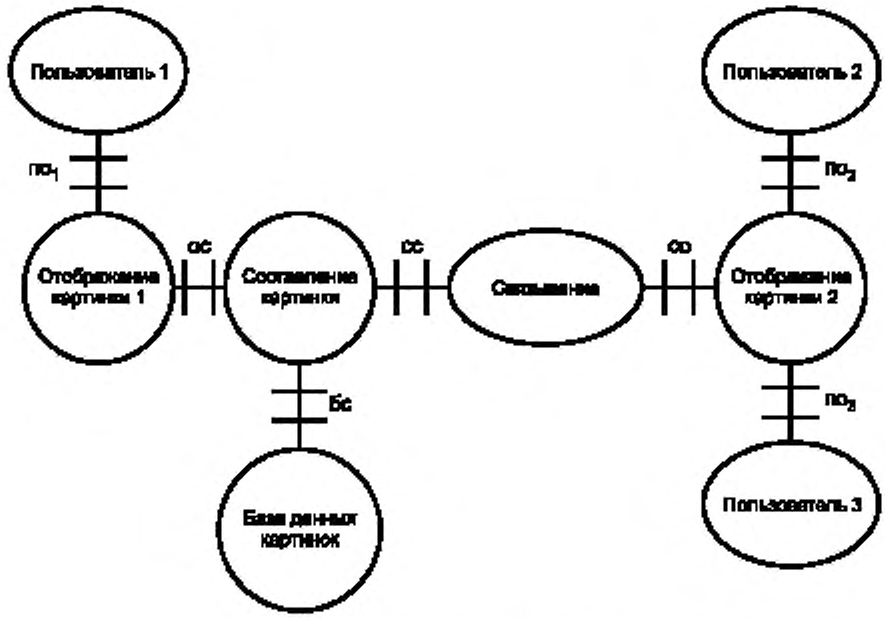



Рисунок 18 — Вычислительная точка зрения

2) спецификации шаблона связующего объекта, которая включает в себя:

- абстрактную спецификацию операций интерфейсов,
- спецификацию контрактов среды, согласованную с предпринимательскими требованиями к КУ;

3) спецификации абстрактных типов данных, которые соответствуют информационным объектам, идентифицированным с информационной точки зрения;

4) спецификаций деятельностей, которые могут происходить для обеспечения приложения.

11.3.2 Применение стандартов

В вычислительном описании могут быть разработаны общие спецификации для интерфейсов, идентифицированных на рисунке 18, в терминах абстрактных операций и абстрактных типов данных (соответствующих информационным объектам в информационном описании). Стандарты для абстрактных операций могут быть общими для нескольких прикладных областей; стандарты для абстрактных типов данных могут быть определены для рассматриваемых прикладных областей. Например, спецификация, основанная на OMG/CORBA и описанная на языке OMG/CORBA IDL (OMG IDL[2]), является вычислительной спецификацией.

**11.4 Инженерная точка зрения**

11.4.1 Инженерная спецификация

На рисунке 19 показана (частично) модель объектов для инженерной спецификации системы, представленной на рисунке 16. В этой модели:

1) базовый инженерный объект «пользователь 1» представляет «оператора 1»;

2) базовые инженерные объекты «отображение картинки», «составление картинки» и «база данных картинок» соответствуют вычислительным объектам «отображение картинки», «составление картинки» и «база данных картинок»;

3) объекты «заглушка», «связник» и «протокол» образуют часть канала, соответствующего связующему объекту в вычислительном описании;

4) «узел» соответствует серверу на рисунке 16 (но на рисунке 19 показана только часть полной конфигурации капсул, кластеров и каналов узла).

С инженерной точки зрения система представляется в терминах:

1) спецификаций поведения инженерных объектов в канале, соответствующем связующему объекту в вычислительной точке зрения, включая в себя:

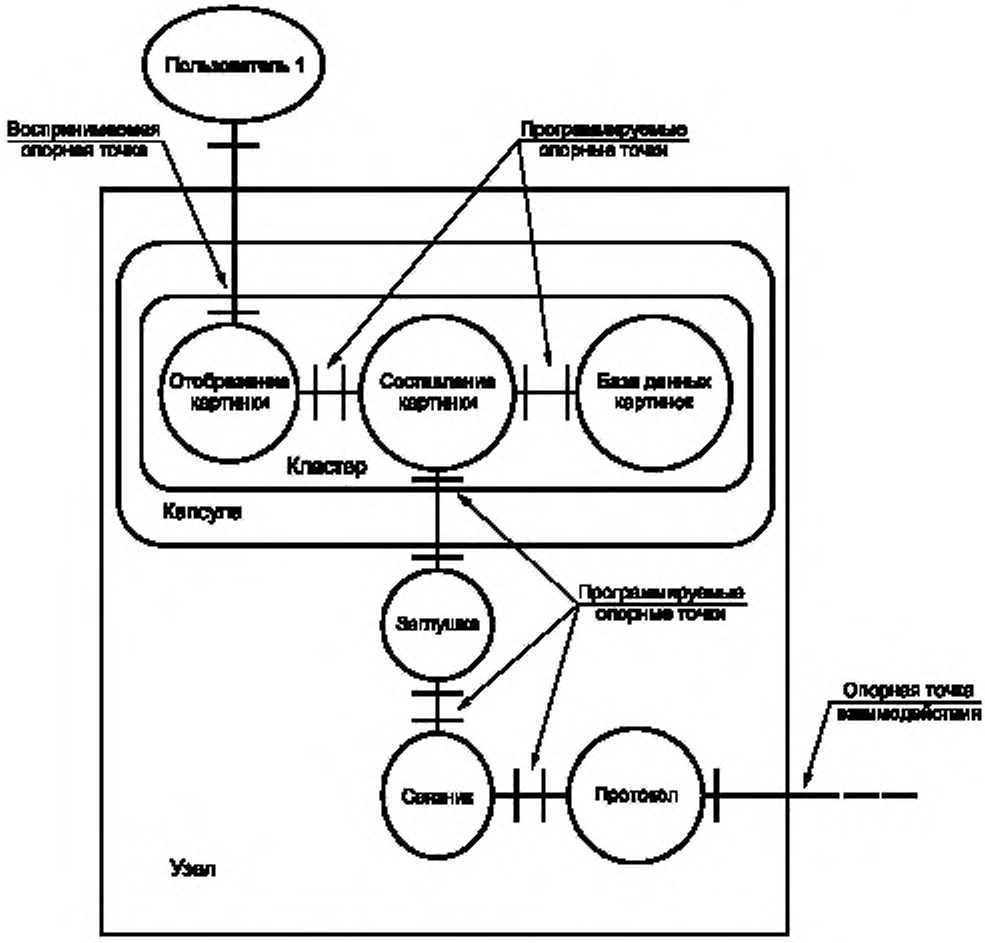



Рисунок 19 — Инженерная точка зрения (часть)

- спецификацию протоколов в *опорной точке взаимодействия* между протокольными объектами (или между протокольным объектом и пересечением),

- спецификацию конкретного представления абстрактных типов данных, идентифицированных в вычислительном описании,

- требования КУ;

2) для каждой опорной точки — спецификации синтаксиса в тех терминах, в которых выражено поведение в этой опорной точке;

3) ограничений на поведение в опорных точках, которые отражают установленные вычислительные деятельности.

11.4.2 Применение стандартов

В инженерном описании могут разработаны общие спецификации для приложений в опорных точках, идентифицированных на рисунке 19:

1) для спецификаций протоколов, абстрактных и конкретных синтаксисов передачи, которые применяются в опорной точке взаимодействия в канале (соответствующем связующему объекту в вычислительном описании), например:

- для протоколов, абстрактных и конкретных синтаксисов передачи могут применяться профили ВОС (А-профиль УДБД, соответствующие Т- и F-профили) при установлении и сопровождении канала,

- для рассматриваемой прикладной области должны быть определены стандарты и F-профили для абстрактных и конкретных синтаксисов, соответствующих абстрактным типам данных в вычислительном описании.

Примечание — Протоколы и конкретные синтаксисы в опорной точке взаимодействия содержат то, что необходимо для установления и сопровождения канала, и то, что соответствует абстрактным операциям и абстрактным типам данных, специфицированным для связующего объекта;

2) для спецификаций ППИ (включая спецификации абстрактного синтаксиса для данных), которые применяются в программируемых опорных точках, например:

- в опорной точке между «составлением картинки» и «базой данных картинок» может применяться профиль SQL,

- в опорной точке между «составлением картинки» и «заглушкой» может применяться профиль ППИ для услуг, поддерживаемых профилем УДБД,

- в опорной точке между «составлением картинки» и «отображением картинки» может применяться профиль ППИ для услуг создания окон — он может быть включен в графические стандарты;

3) для спецификаций ИЧК, которые применяются в *воспринимаемой опорной точке,* например, стандарт ГИП.

**11.5 Технологическая точка зрения**

11.5.1 Технологическая спецификация

С технологической точки зрения (см. рисунок 20) система представляется в терминах заявлений поставщика о соответствии его системы. Оно может быть выражено в виде:

- идентификации точек соответствия для поставляемой системы, которые соответствуют опорным точкам инженерного описания;

- установления дополнительной информации для тестирования соответствия системы поведению, специфицированному для ее точек соответствия.

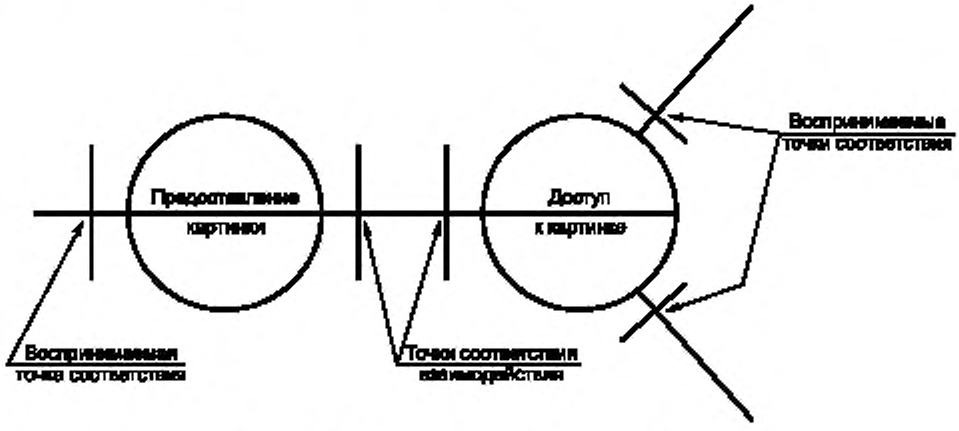



Рисунок 20 — Технологическая точка зрения (часть)

Поставщик не обязан сделать все опорные точки видимыми в качестве точек соответствия.

11.5.2 Применение стандартов

В технологическом описании:

- реализация конкретного протокола ВОС является примером применения стандарта ВОС;

- точки соответствия идентифицируют места, в которых можно наблюдать за поведением системы;

- стандарты, которые применяются в этих точках соответствия, специфицируют синтаксис и порядок обмена в тех терминах, в которых выражено поведение.

Кроме того, имеются дополнительные ограничения на поведение в точках соответствия вследствие требования, что поведение системы согласуется со стандартами, специфицированными в инженерном описании для других опорных точек.

## 12 Примеры спецификаций ОРО

В данном разделе приведено несколько примеров использования в спецификации системы понятий и правил, описанных в ГОСТ Р ИСО/МЭК 10746-2 и ГОСТ Р ИСО/МЭК 10746-3. Примеры являются упрощенными и неполными описаниями реальных систем и служат введением в использование ОРО; они иллюстрируют применение ключевых понятий каждой точки зрения и взаимосвязи между описаниями с разных точек зрения.

В примере 12.1 используются понятия и правила БМ-ОРО для проектирования **мультимедийной системы конференций** (ММСК) с пяти точек зрения. ММСК позволяет осуществлять в реальном времени взаимодействие нескольких пользователей, используя мультимедийную информацию (текст, видео и звук).

В примере 12.2 специфицирован **многосторонний аудио/видеообмен** в распределенных системах — отдельный компонент ММСК. В качестве основы для многостороннего обмена аудио- и видеопотоками использовано понятие «связывание потоков» и представлена спецификация с пяти точек зрения ОРО. Особое внимание обращается на соответствия между спецификациями с пяти точек зрения для обеспечения соответствия между ними.

В примере 12.3 показано место **административного управления** в БМ-ОРО.

В примере 12.4 дан обзор спецификации **распределенной базы данных.**

В примерах спецификаций с разных точек зрения проиллюстрированы следующие понятия:

- **предпринимательские:**
  коммуникация/федерация,
  связь ролей с предпринимательскими объектами,
  контракт, шаблон, политика;
- **информационные:**
  статическая схема,
  динамическая схема,
  инвариантная схема;
- **вычислительные:**
  спецификация вычислительного объекта, включая контракт среды и поведение,
  спецификация интерфейса операций и потоков,
  понятие связывания,
  указатели интерфейсов и правила взаимодействия,
  прозрачность;
- **инженерные:**
  установление канала (протокольный объект, связник, заглушка),
  использование правил и функций построения инженерной структуры и для спецификации инфраструктуры, согласующейся с предпринимательской, информационной и вычислительной спецификациями;
- **технологические:**
  выбор конкретных программных и технических компонентов, согласующихся со спецификациями с других точек зрения,
  идентификация точек соответствия, подразумеваемых технологическим выбором.

### 12.1 Мультимедийная система конференций

12.1.1 Введение

Мультимедийная система конференций (ММСК) позволяет в реальном времени взаимодействовать нескольким пользователям, используя мультмедийную информацию (текст, звук, видео). Услуга позволяет группе лиц, физически удаленных друг от друга, совместно работать над (мультимедийным) документом и взаимодействовать друг с другом. В течение сеанса к конференции могут присоединяться новые участники или текущие могут покидать ее.

Более того, услуга предоставляет пользователю контроль за несколькими атрибутами услуги, например желательный(е) тип(ы) информации, качество типа информации и пр. ММСК предоставляет фундамент для различных приложений, требующих кооперации.

Такие приложения, как видео/аудиоконференции, совместное редактирование и электронная рассылка должны быть интегрированы в перспективе. Для достижения открытости должны быть возможны расширения ММСК.

Конфигурация ММСК на рисунке 21 может состоять из рабочих станций для представления изображения, звука и текста. Дополнительные камеры, микрофоны и мультимедийные базы данных (возможно, распределенные) присоединяются к глобальной сети.

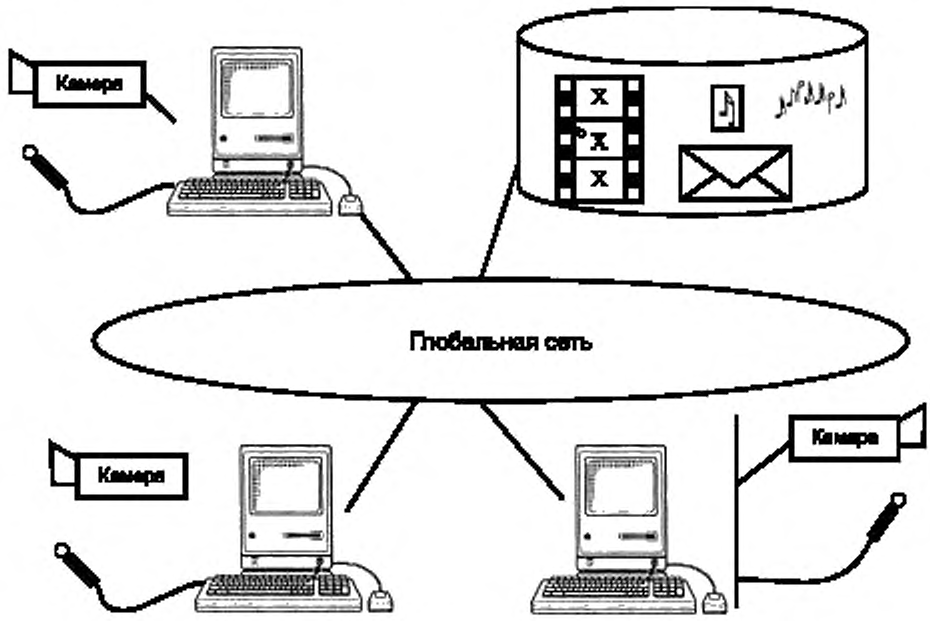



Рисунок 21 — Конфигурация мультимедийной системы конференций

Для спецификации этого примера ММСК будут применены соответствующие понятия и правила каждой из точек зрения ОРО. Цель данного примера — описать все аспекты ММСК, а не проиллюстрировать языки точек зрения для спецификации открытых распределенных приложений или систем. Для выражения предпринимательской и информационной спецификаций ОРО применены методы объектно ориентированного анализа ММО [1].

12.1.2 Предпринимательская спецификация

Предпринимательская спецификация описывает цели, политику и требования, относящиеся к ММСК. Требования и политика услуги ММСК вытекают из задач участвующих сторон, которые могут быть классифицированы в соответствии с их ролями:

- пользователь — человек или машина, использующий(ая) услуги для удовлетворения каких-то коммуникационных потребностей;
- покупатель или подписчик — лицо или организация, которому(ой) по контакту предоставляются услуги поставщиком;
- поставщик услуги — организация, которая на коммерческих условиях управляет предоставлением услуг потребителю в соответствии с условиями контракта.

Для структурирования требований конкретной услуги в соответствии с ролями может потребоваться специализация введенных выше общих ролей. Например, для ММСК можно различать два следующих типа пользователей: участник сеанса и руководитель сеанса.

Тогда возникает вопрос: «Что должно быть описано для каждой роли, участвующей в услуге?» БМ-ОРО дает указания относительно того, что охватывается крупномасштабной политикой, и правила определения того, что представляет интерес для описания распределенных услуг. Представляющими интерес общими правилами и политикой являются: использование ресурсов, правила областей, правила разрешения конфликтов, организационные правила, деловые правила, правила передачи, правила безопасности, правила КУ и правила административного управления.

На рисунке 22 показана простая предпринимательская спецификация, использующая предпринимательские понятия ОРО и графические обозначения ММО.

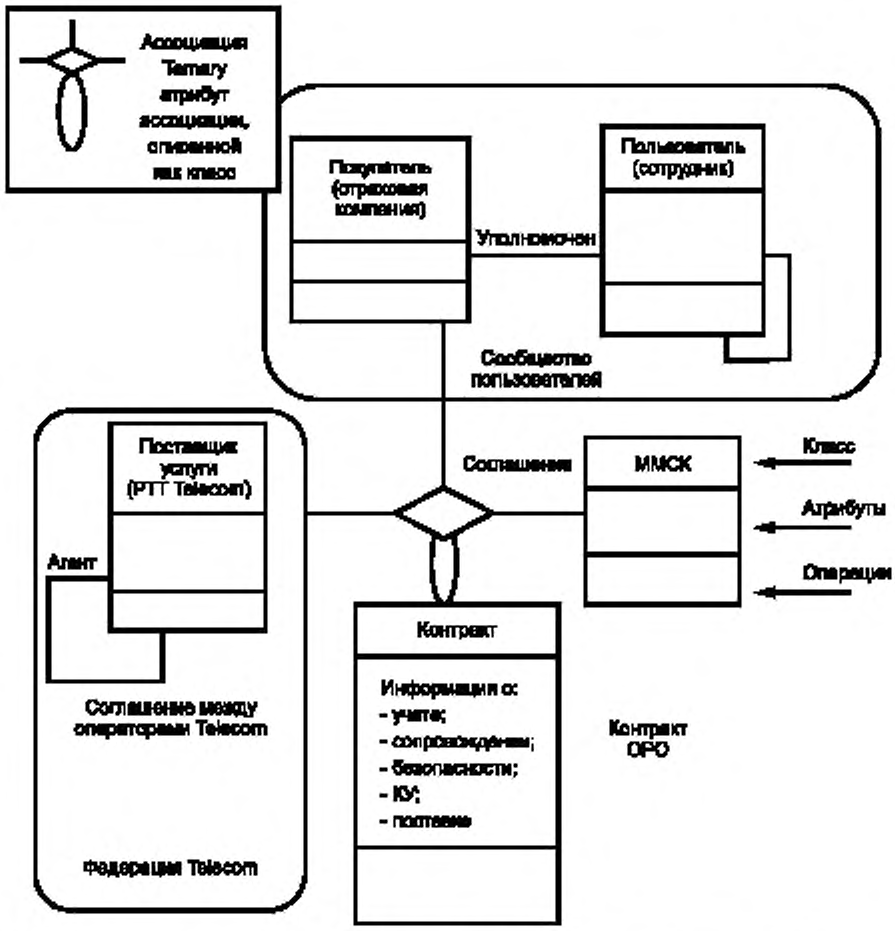



Рисунок 22 — Предпринимательская спецификация ОРО, использующая графические обозначения ММО

Услуга мультимедийной конференции оказывается в нескольких странах. Ее можно интерпретировать как нескольких пользователей, взаимодействующих через местных операторов Telecom (поставщика услуги). Для выражения ММСК в терминах ОРО это можно представить, используя понятия сообщества и федерации. Можно утверждать, что пользователи существуют в сообществе пользователей, а операторы Telecom входят в федерацию. Взаимоотношения и взаимодействия между сообществом и федерацией могут быть рассмотрены и, относительно ММСК, согласованно установлены с помощью контракта.

Пользователь, поставщик услуги и ММСК могут быть описаны в терминах ОРО как предпринимательские объекты с соответствующими ролями. Например, пользователь может осуществлять действия своей роли с ММСК через покупателя и ММСК осуществляет действия своей роли с пользователем через покупателя. Эти взаимодействия изменяют так называемые отношения обязательства, разрешения и запрещения между пользователем и ММСК. Группа пользователей взаимодействует друг с другом для образования сообщества ОРО, так как пользователи имеют общий контракт обязательств между ролями, выполняемый предпринимательскими объектами и множеством деятельностей.

12.1.3 Информационная спецификация

В информационной спецификации устанавливаются семантика и требования к обработке информации услуги. Это делается с помощью понятия *схемы*. Для каждой роли пользователя определе-

на локальная схема, представляющая информацию, которая относится к каждой роли пользователя, и информацию, относящуюся к услуге. Такая информационная спецификация является примером статической схемы, а не динамической или инвариантной.

Ограничиваясь классом ММСК, идентифицированном в предпринимательской спецификации, статическая схема для системы в момент, когда существует сеанс, представляется информационной спецификацией ММО, показанной на рисунке 23.

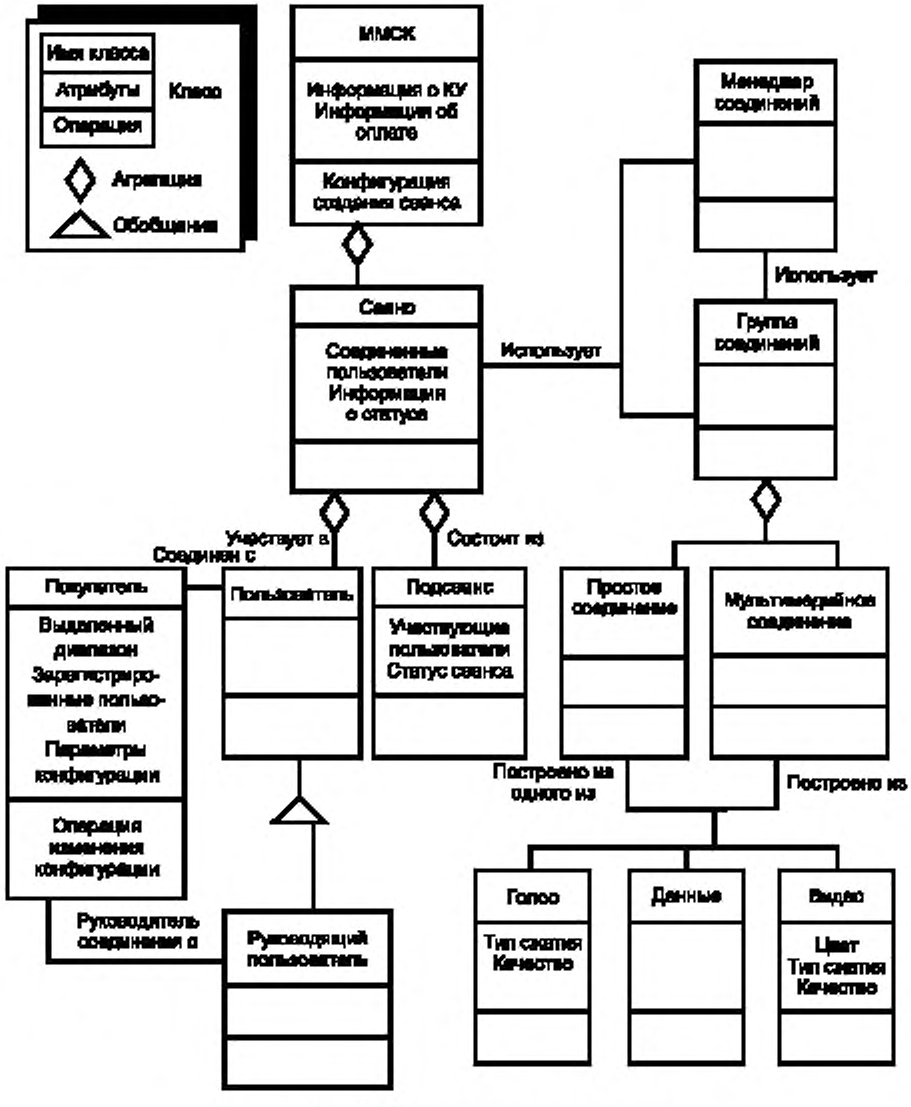



Рисунок 23 — Информационная спецификация

Для покупателя приведены параметры конфигурации ММСК; там же показана информация о выделенном диапазоне, списке зарегистрированных пользователей, которым можно начинать телеконференцию, списке доступных конечным пользователям опций и т. п. Такая информация покупателя может рассматриваться как его атрибуты в модели объекта ММО. Информационный объект «пользователь» связан с информационным объектом «покупатель» в том смысле, что он может

участвовать в конференции только в том случае, если в предпринимательской спецификации соответствующий конечный пользователь авторизован соответствующим покупателем. Руководящий пользователь является пользователем, контролирующим конференцию.

12.1.4 Вычислительная спецификация

Вычислительная спецификация должна быть последовательной относительно предпринимательской и информационной спецификаций. Отображение между информационными и вычислительными объектами не обязательно должно быть один-к-одному. Информационная спецификация существенно отличается от вычислительной спецификации, в частности, вычислительные объекты устанавливаются в терминах интерфейсов, а информационные — нет.

Соответствие между информационной и вычислительной спецификациями в каждом случае должно быть установлено так, чтобы могла быть проверена последовательность спецификаций. Эта важная задача должна быть решена проектировщиком услуги. На рисунке 24 показано отображение для ММСК.

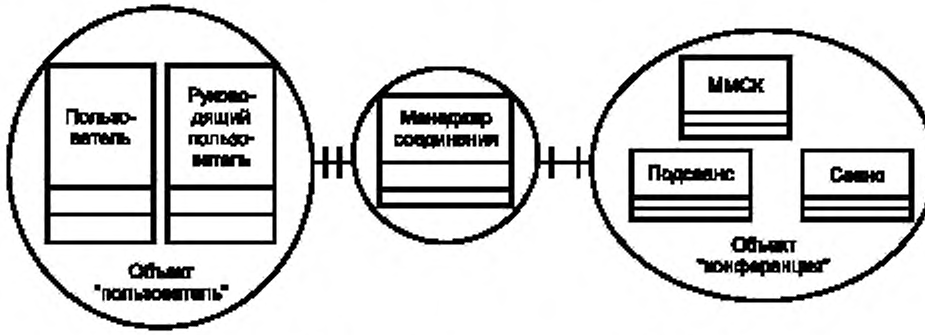



Рисунок 24 — Вычислительная конфигурация ММСК

Для идентификации и проектирования вычислительных объектов используется анализ ММО. При идентификации вычислительных объектов возможна перегруппировка тех элементов, которые являются функционально связанными. Анализ ММО так же позволяет сделать некоторый выбор для инженерной конфигурации и технологического обеспечения.

Группировка классов в вычислительные объекты является решением проектировщика услуги безотносительно к вопросам распределения:

- при проектировании вычислительного объекта «пользователь» принимаются во внимание объекты ММО, классы и ассоциации, относящиеся к пользователю и руководящему пользователю;
- при проектировании вычислительного объекта «конференция» принимаются во внимание объекты ММО, классы и ассоциации, относящиеся к конференции (например, ММСК, сеанс, подсеанс);
- при проектировании вычислительных связываний принимаются во внимание объекты ММО, классы и ассоциации, относящиеся к соединению (например, менеджер соединения), и они будут полезны при выборе инженерной конфигурации (например, многоцелевые средства).

Для каждого полученного в результате анализа идентифицированного вычислительного объекта необходимо определить интерфейсы для взаимодействий с другими вычислительными объектами (например, операции конференции или операции отправки аудио/видеопотоков). В общем случае объекты и интерфейсы могут быть графически представлены так, как показано на рисунке 25.

Если ограничиться взаимодействиями, которые возможны через связывающие объекты, то они устанавливаются явными связывающими действиями, позволяющими пользователю специфицировать требуемое связывание между вычислительными объектами. Получающийся в результате явного связывания связующий объект может обеспечить обмен операциями между вычислительными объектами или поток, если обмен относится к непрерывному информационному потоку.

На рисунке 26 графически показано представление явного связывания в случае потока. Связующий объект контролирует и управляет взаимодействиями между интерфейсами окружающих его вычислительных объектов. Контролирующие операции осуществляются через контролирующий интерфейс.




Рисунок 25 — Вычислительное представление объектов и интерфейсов

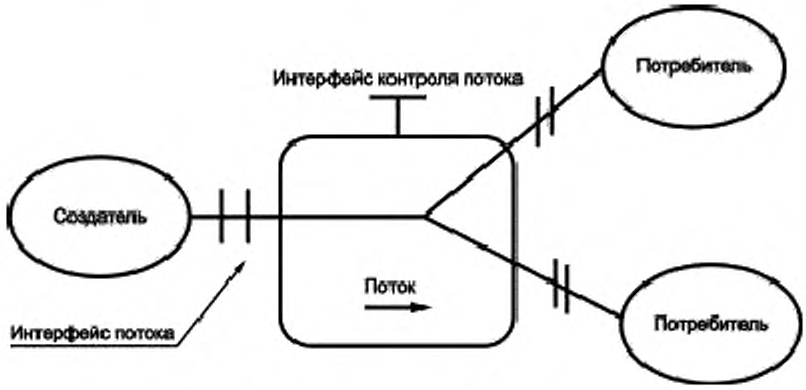



Рисунок 26 — Представление явного связывания (для потока)

В вычислительном представлении вычислительных объектов участвует аудио/видеообмен в ММСК, как показано на рисунке 27. Идентифицированы два главных объекта: «пользователь» и «конференция».

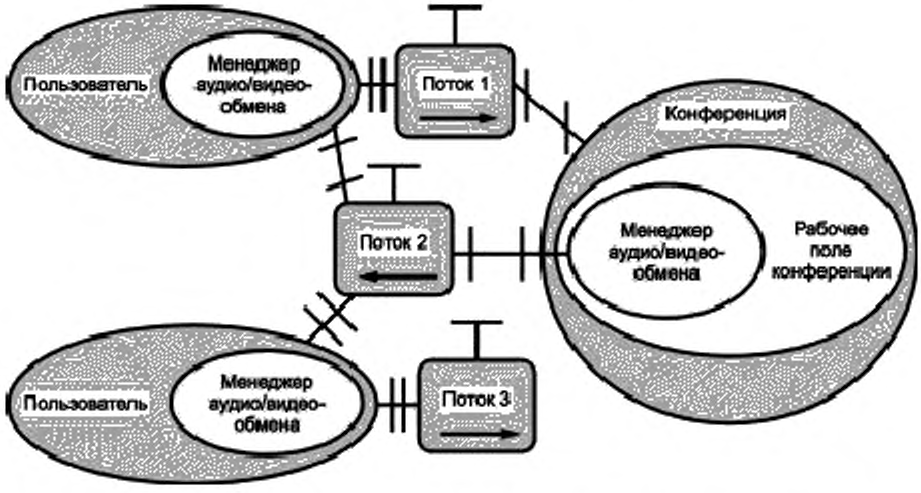



Рисунок 27 — Конфигурация объектов ММСК, участвующих в аудио/видеообмене

Объект «пользователь позволяет конечным пользователям присоединиться к конференции и предоставляет им средства, необходимые для взаимодействия друг с другом через объект «конференция» или другие объекты «пользователь». Объект «пользователь» предоставляет конечным пользователям такие операции, как присоединение к конференции, средства редактирования, средства обмена аудио- и видеоданными.

Объект «конференция» содержит функциональные возможности, необходимые для мультимедийной конференции. В частности, он содержит совместно используемое рабочее поле конференции, которое отвечает за пересылку аудио- и видеопотоков между объектами «пользователь».

Менеджеры аудио/видеообмена отвечают за получение и отправку аудио/видеоданных во время конференции.

Объекты «поток 1» и «поток 3» представляют аудио/видеопотоки от объектов «пользователь» к совместно используемому рабочему полю. «Поток 2» представляет многоцелевые аудио/видеопотоки ко всем объектам «пользователь».

Различные контролирующие действия, такие как динамический контроль КУ и синхронизация аудио- и видеопотоков, осуществляются через контролирующие интерфейсы потоков.

12.1.5 Инженерная спецификация

Инженерный язык ОРО позволяет моделировать машину услуг, которая обеспечивает выполнение вычислительной спецификации. На рисунке 28 показана упрощенная инженерная архитектура. Главным элементом этой архитектуры является ядро, которое контролирует использование ресурсов и обеспечивает возможность коммуникации между различными инженерными объектами. Распределенным приложениям доступны некоторые функции, называемые функциями ОРО, такие как функции торга или административного управления, которые являются общими для самых разных распределенных услуг.

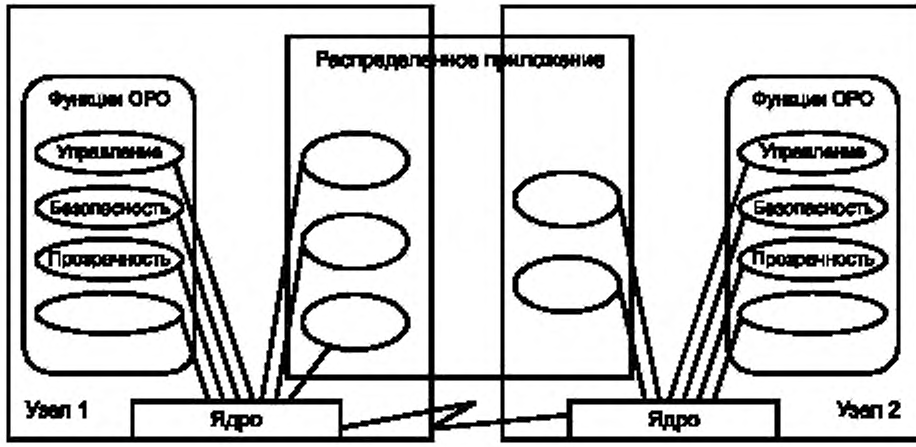



Рисунок 28 — Инженерное обеспечение

Распределенная услуга составляется из базовых инженерных объектов, которые являются представлениями времени выполнения (например, часть исполняемого кода C++) вычислительной спецификации. Связывание между объектами, расположенными в разных ядрах, отображается с помощью каналов между этими объектами. Если ограничиться аудио/видеоменеджерами и потоком 2 на рисунке 27, то соответствующее инженерное обеспечение представлено на рисунке 29.

Многоконечный канал (инженерное представление потока) установлен между менеджерами аудио/видеообмена. Этот канал привязан к рассматриваемым ядрам.

Заглушки обеспечивают функции адаптации для поддержания прозрачности распределения (например, преобразования формата данных из одного кодирования видеосигнала в другое).

Связующие объекты проверяют совместимость связываемых интерфейсов и обеспечивают целостность связывания между менеджерами аудио/видеообмена.

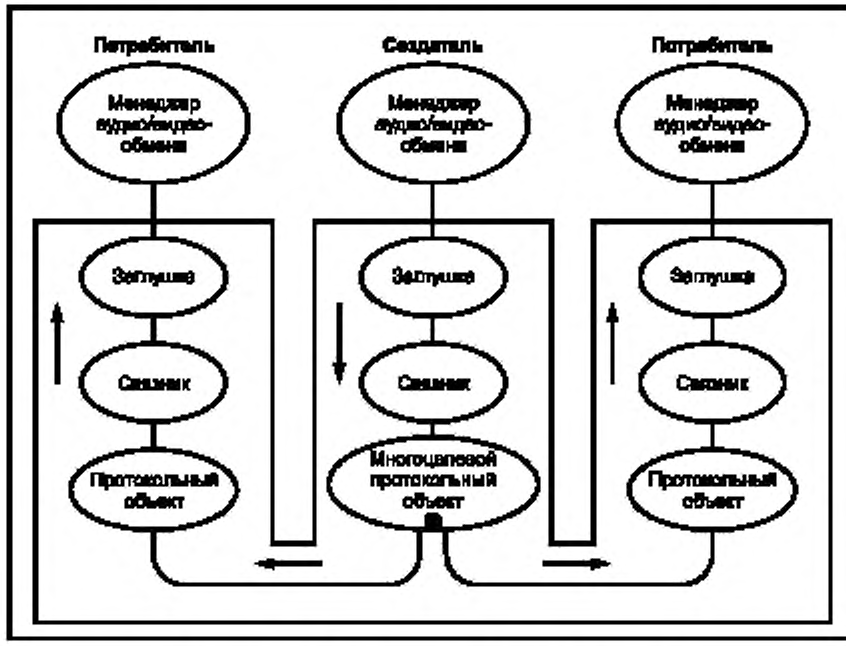



Рисунок 29 — Многоконечный канал для потока 2

Протокольные объекты взаимодействуют друг с другом, передавая необходимую информацию для обеспечения прозрачности распределения взаимодействий между менеджерами аудио/видеообмена. Протокольные объекты выбраны так, чтобы соответствовать ограничениям КУ (например, реального времени или безопасности).

12.1.6 Технологическая спецификация

Технологическая спецификация ММСК определяет реализацию системы в терминах компонентов технического и программного обеспечения. Идентифицированные в спецификациях с других точек зрения опорные точки определяются как точки соответствия, в которых может быть проверено поведение ММСК.

Технологическая спецификация так же выражает требования к адекватным для представления видео- и аудиоданных, а также текстовых данных рабочим станциям. Для обеспечения коммуникационных потребностей путем предоставления достаточно широких полос для доставки аудио/видеоинформации, а также текстовой информации расположенным в разных местах участникам требуются высокоскоростные и глобальные сети передачи данных.

**12.2 Многостороннее связывание аудио/видеопотока**

В этом примере основное внимание обращено на связывание многостороннего потока, используемого в предыдущем примере.

В примере использован следующий подход: очертить проблемную область в вычислительных терминах и дать соответствующие пять спецификаций с разных точек зрения.

Прежде всего определены некоторые дополнительные понятия и правила, применяемые к рассматриваемой проблеме. В предпринимательской спецификации введены роли *функционеров*[*] (пользователь, покупатель, поставщик) для более подробного структурирования проблемной области. В качестве конкретных нотаций для выражения информационной и вычислительной спецификаций использованы ММО [1] и IDL [2], соответственно.

В информационной спецификации введено несколько отношений между классами инвариантной схемы. Эти отношения выведены из набора базовых отношений ММО, и обычно в качестве параметров они имеют текст, уточняющий их смысл.

[*] Функционер — понятие, обозначающее организацию или лицо, имеющее коммерческие и управленческие интересы в телекоммуникационных услугах.

В вычислительной спецификации введены аудио/видеоконтроллер и диспетчер для обработки связывания многостороннего аудио/видеопотока.

В инженерной спецификации введен специализированный канал потока для передачи непрерывного потока. Для обеспечения многостороннего связывания аудио/видеопотока имеется конфигурация инженерных объектов, включающая в себя объекты для контроля и координации кратных каналов потока. Можно было бы специфицировать контроллер и диспетчер другим способом. Однако в данном примере не введены дополнительные усложнения.

Реализация, описанная с технологической точки зрения, используется для проверки процесса моделирования с разных точек зрения ОРО. В операционной среде, вероятно, можно встретить более специфическое техническое и программное обеспечения, например точно выполняющие требования мультимедийных приложений.

12.2.1 Общее описание

Обмен непрерывными носителями в распределенных мультимедийных приложениях является сложным. Например, в мультимедийном приложении конференции реального времени участники разделены географически и связываются путем обмена видео- и аудиоинформацией реального времени. Аудио/видеообмен должен быть как можно более естественным и гибким. Это подразумевает, что при спецификации мультимедийного приложения конференции нужно учитывать такие требования, как синхронизация краев и отображений.

Для выполнения этих требований мультимедийное приложение имеет строгие требования к производительности сети и синхронизации при обмене аудио- и видеопотоками. Более того, некоторые потоки обмена и соответствующее качество могут изменяться за время жизни конференции. Это вытекает из того обстоятельства, что приложение предоставляет управляющие операции для присоединения и выхода из конференции и для изменения КУ потоков.

Для работы с такими сложными функциональными возможностями БМ-ОРО определяет в вычислительном языке понятие связывания объектов. БМ-ОРО предоставляет теоретическое вычислительное понятие без уточнения конкретных требований данной проблемной области. В настоящем примере связующий объект специфицирован в терминах пяти точек зрения ОРО, т. е. введен многосторонний аудио/видеосвязующий объект (Gay 95[3]). Этот объект управляет интерфейсами потоков, которые используются для многосторонних аудио/видеовзаимодействий реального времени. Над многосторонним связующим объектом также могут осуществляться операции управления.

На рисунке 30 показано вычислительное представление многостороннего аудио/видеосвязующего объекта и его среда.

Прямоугольник в центре обозначает многосторонний аудио/видеосвязующий объект. Его среда (серые области) состоит из прикладных и системных частей, а также обеспечивающей сетевой инфраструктуры. Символ † обозначает интерфейсы потоков, через которые обмениваются создатели и потребители аудио/видеоинформации (1). Многосторонний аудио/видеосвязующий объект управляет взаимодействиями между входящими в него интерфейсами потоков. Он инкапсулирует используемые для этого методы и абстрагируется от вопросов распределения. Символ † сверху прямоугольника обозначает интерфейс управления потоком связующего объекта. Через этот интерфейс многосторонний аудио/видеосвязующий объект предоставляет среде операции (3 и 4), которые управляют его функционированием.

12.2.2 Предпринимательская спецификация

Предпринимательская спецификация дает описание требований и целей, которые предприниматель ставит перед проектируемой системой. Они удовлетворяются проектом системы. Для описания многостороннего аудио/видеосвязывания используется понятие предпринимательских объектов, играющих роли *выполнимых действий*.

Роли, которые функционеры могут играть в телекоммуникации отиосительно услуги (т. е. пользователь, покупатель, поставщик), используются в спецификации связующего объекта. Для прикладных и обеспечивающих систем это приводит к идентификации объектов создателей и потребителей аудио/видеоинформации (пользователей) и обектов, управляющих этими пользователями (покупателей). Для обеспечивающей сетевой инфраструктуры управляющий объект (поставщик) выделяется тем, что управляет связующим объектом.

12.2.2.1 Пользователь: создатель/потребитель аудио/видеопотоков

Пользователь — это предпринимательский объект в прикладной или обеспечивающей системе, играющий роль, целью которой является создание и (или) потребление потоков через интерфейс потока (см. рисунок 30, поз.1). Примерами пользователей являются микрофоны, докладчики,

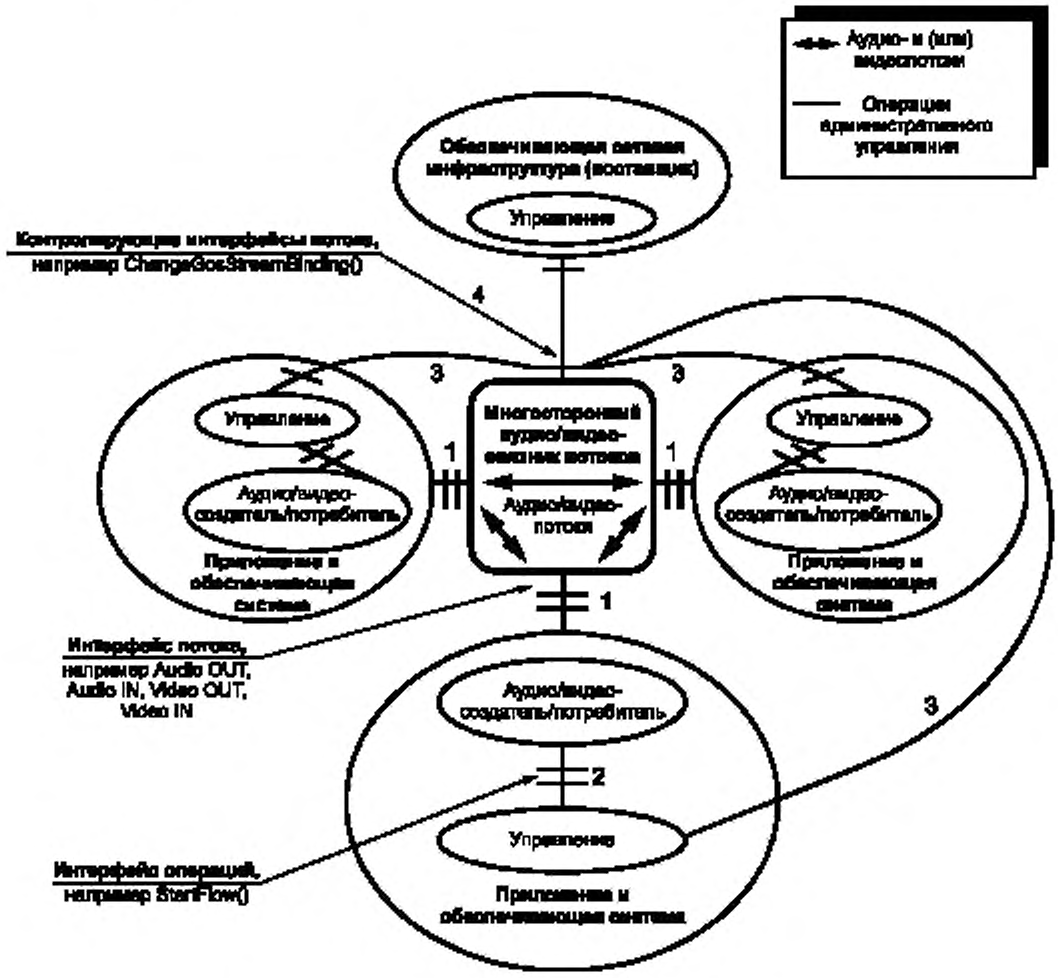



Рисунок 30 — Многостороннее связывание аудио/видеопотоков

камеры, мониторы. Пользователь указывает типы потоков, которые он может обрабатывать, и требуемые форматы кодирования. Более того, он определяет нужные ему значения параметров КУ. Эти параметры специфицируют качество аудио/видеоинформации в терминах качества широкополосного телевидения, HDTV, качества телефонии, Hi-Fi или CD. Кроме того, определены требования КУ между потоками, которые описывают, например, что требуется синхронизация краев для аудио- и видеопотоков и что желательна одновременная доставка аудио/видеоинформации нескольким пользователям.

12.2.2.2 Покупатель: локальные прикладная и обеспечивающая системы

Покупатель является предпринимательским объектом с ролью, целью которой является административное управление и обеспечение создателя/потребителя аудио/видеоинформации. Он учитывает политику пользователей, которых он должен обеспечивать и которыми должен управлять. Покупатели могут вызывать выполнимые действия (см. рисунок 30, поз. 2) для создания и удаления пользователей или для исправления политики существующих пользователей.

Покупатель осуществляет выполнимые действия, относящиеся к административному управлению конфигурацией и ресурсами. Например, он должен управлять приоритетами потоков. В случае перегрузки сети или локальных проблем с ресурсами поток с наинизшим приоритетом удаляется

или задерживается. Кроме того, покупатель осуществляет сквозное согласование с другими покупателями для определения доступных, предпочтительных и недоступных значений параметров КУ. Последнее отражается в «связывающем контракте» (см. рисунок 31). Связывающий контракт описывает согласованные договоренности между пользователем, покупателем и поставщиком, которым нужно следовать во время существования многостороннего связующего объекта. Покупатель может осуществлять те выполнимые действия (см. рисунок 30, поз. 3) над связующим объектом, которые относятся к административному управлению многосторонним связыванием потока. Эти действия относятся, например, к установлению, удалению и исправлению связывания потока в соответствии с согласованными сквозными требованиями КУ. Выполнимые действия могут быть инициированы связующим объектом, например для указания того, что он не может обеспечить согласованные значения КУ.

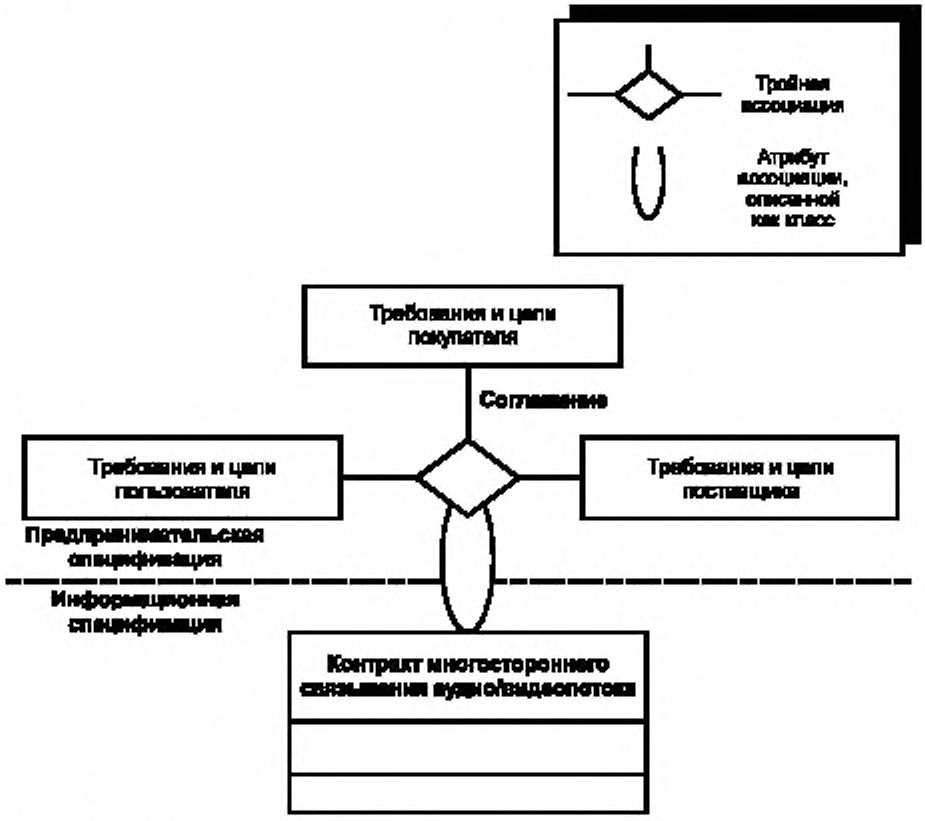



Рисунок 31 — Связывающий контракт между пользователями, покупателями и поставщиками

12.2.2.3 Связник и поставщик

Связник является предпринимательским объектом, который позволяет обмениваться аудио- и видеопотоками между географически разделенными пользователями. Поставщик является предпринимательским объектом, который предоставляет инфраструктуру, используемую связником, и административно управляет связником в соответствии со статусом инфраструктуры.

Поставщик отвечает за топологию потока, вопросы оплаты и безопасности, неисправности административного управления и обеспечение КУ нижележащими сетями и ресурсами. Поставщик занимается административным управлением сетью для обеспечения связывающего контракта. Например, он выбирает подходящие процедуры для аудио/видеоканала и резервирует на каждом узле ресурсы под эти процедуры. Если поставщик не в состоянии далее гарантировать выполнение связывающего контракта, то он осуществляет действия (см. рисунок 30, поз. 4), которые влияют на связник в отношении согласования этого контракта.

12.2.3 Информационная спецификация

Информационная спецификация многостороннего связывания аудио/видеопотока описывает информацию, относящуюся к функционерам связывания. В ней учитываются требования и цели, очерченные в предпринимательской спецификации.

12.2.3.1 Переход от предпринимательской к информационной спецификации

Информация, относящаяся к пользователям, покупателям и поставщикам, специфицирована в связывающем контракте. Этот контракт является результатом соглашения между пользователями, покупателями и поставщиками. Он удовлетворяет их требованиям и целям, очерченным в предпринимательской спецификации. На рисунке 31 показано взаимоотношение между предпринимательской и информационной спецификациями в терминах ММО.

12.2.3.2 Инвариантная схема связывающего контракта

Связывающий контракт, показанный на рисунке 31, рассматривается функционерами как атрибут и описывается как один класс. Однако при более подробном рассмотрении информационная спецификация оказывается более сложной. На рисунке 32 показана инвариантная схема контракта многостороннего связывания аудио/видеопотока.

Общая структура каждого контракта между покупателями, пользователями и поставщиком специфицируется как инвариантная схема. Для связывающего контракта она содержит информацию о пользователях связывания (информационные объекты «пользователь»), интерфейсах потоков, участвующих в связывании (информационные объекты «интерфейс потока») и операциях, которые могут вызывать функционеры. Далее, контракт специфицирует многостороннее связывание аудио/видеопотока между интерфейсами потоков. Информация в связывании моделируется с помощью информационного объекта «многосторонний связник аудио/видеопотока». Инвариантная схема для этого информационного объекта специфицирует все операции, которые могут вызывать функционеры.

Пользователь может иметь один или несколько интерфейсов потоков, чем подразумевается, что каждый информационный объект «пользователь» состоит из одного или нескольких информационных объектов «интерфейс потока». Интерфейс потока состоит из одного или нескольких потоков; в результате информационный объект «интерфейс потока» состоит из одного или нескольких информационных объектов «поток» (аудио-, видео- или составной поток). Информационный объект «поток» состоит из атрибутов, указывающих, кроме всего прочего, направления потоков и параметры КУ. Параметры КУ, определенные в информационной спецификации, будут обрабатываться в спецификации вычислительной.

Информация, относящаяся к связыванию, заключена в информационном объекте «многосторонний связник аудио/видеопотока». Этот объект соединен с двумя или более информационными объектами «интерфейс потока». Он содержит информацию о топологии потока и описывает КУ, которое должно обеспечиваться при обмене аудио/видеопотоками между интерфейсами.

12.2.3.3 Статическая схема связывающего контракта

Конкретный связывающий контракт, который существует между функционерами в некоторый момент времени, специфицируется как статическая схема. Связывающий контракт должен удовлетворять инвариантной схеме и состоит из информационных объектов «пользователи», «интерфейсы потоков», «потоки» и «многосторонние связники аудио/видеопотоков». Частью контракта могут быть и другие информационные объекты, если это не противоречит инвариантной схеме.

12.2.3.4 Динамическая схема связывающего контракта

Динамическая схема для связывающего контракта имеет два аспекта. Во-первых, влияние, которое оказывают на контракт вызываемые операции, во-вторых, условия, при которых эти операции могут быть вызваны функционерами. Ниже обсуждаются оба эти аспекта.

Влияние операций на информацию связывающего контракта существенно зависит от выбора реализации. В общем случае результаты могут быть классифицированы по трем категориям: сообщающие, согласующие результаты и отсутствие результата.

Операция, имеющая сообщающий результат, является операцией, с помощью которой функционер информирует связывающий контракт о вновь созданном вычислительном объекте. Влияние на контракт заключается в том, что добавляется новый вычислительный объект. Например, операция AddNewUser (добавить нового пользователя) имеет в качестве результата новый информационный объект «пользователь», один или несколько информационных объектов «интерфейс потока» и один или несколько информационных объектов «поток».

Операция, имеющая согласующий результат, является операцией, с помощью которой функционер договаривается со связующим объектом и с другими функционерами об изменении в связывающем контракте. Например, успешная операция RemoveUser (удалить пользователя) имеет в

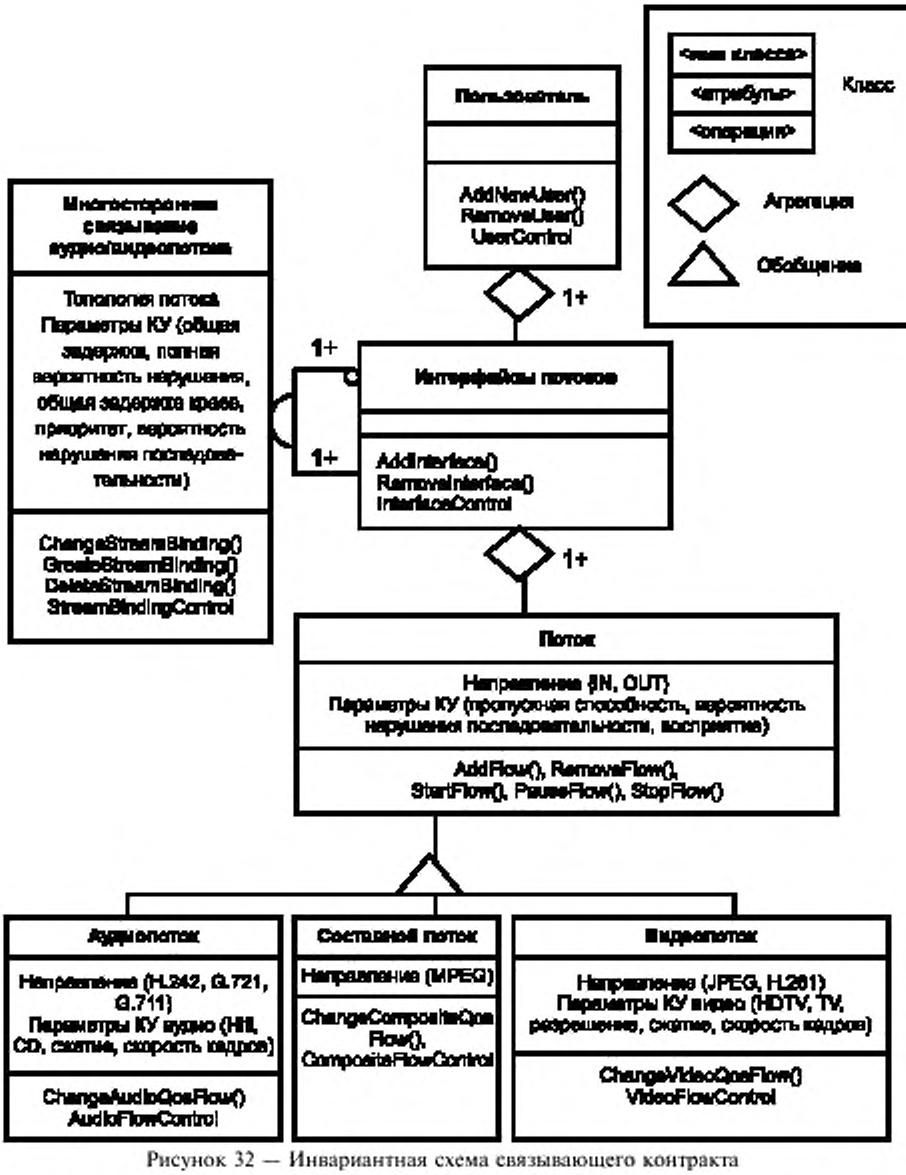



Рисунок 32 — Инвариантная схема связывающего контракта

качестве результата удаление информационного объекта «пользователь» и связанных с ним информационных объектов «интерфейс потока» и «поток». Кроме того, будет изменен или удален связующий информационный объект, который связывает интерфейсы удаленного пользователя с интерфейсами других пользователей.

Операция, не имеющая результата, является операцией, которая не влияет на информационные объекты в связывающем контракте. Это, главным образом, контролирующие операции. Например, операция PauseFlow (приостановить поток) имеет в качестве результата приостановку рассматриваемого потока. Это не имеет последствий для информации связывающего контракта.

Динамическая схема также описывает условия для вызова операций над связывающим контрактом. Инвариантная и статическая схемы не устанавливают конкретного порядка операций, которые могут вызывать функционеры. Однако для получения осмысленного связывающего контракта необходимо определить условия для вызовов операций. Вот некоторые примеры:

- покупатель может вызвать операцию AddNewUser только над существующим связником потоков;

- покупатели могут вызывать операции ChangeAudioQOSFlow и Remove Flow только над существующими потоками;

-покупатели и поставщики могут вызвать операции ChangeStreamBinding и DeleteStreamBinding только над существующими связываниями.

12.2.4 Вычислительная спецификация

В настоящем подразделе специфицирована сигнатура контролирующего интерфейса связника потока. Для полного описания интерфейса должны быть описаны поведение и контракты среды.

12.2.4.1 Переход от информационной к вычислительной спецификации

Отображение между информационными и вычислительными объектами не обязательно должно быть одно-к-одному. Соответствие между информационной и вычислительной спецификациями в каждом случае должно устанавливаться таким образом, чтобы могла быть подтверждена их последовательность. Решение о группировании классов в объекты принимает проектировщик услуги. На этой стадии вопросы распределения не принимают во внимание.

На рисунке 33 показано отображение нескольких информационных классов в вычислительные интерфейсы. Для контролирующего интерфейса (см. рисунок 33, поз. 3 и 4) связника многостороннего аудио/видеопотока учтены операции класса связника многостороннего аудио/видеопотока.

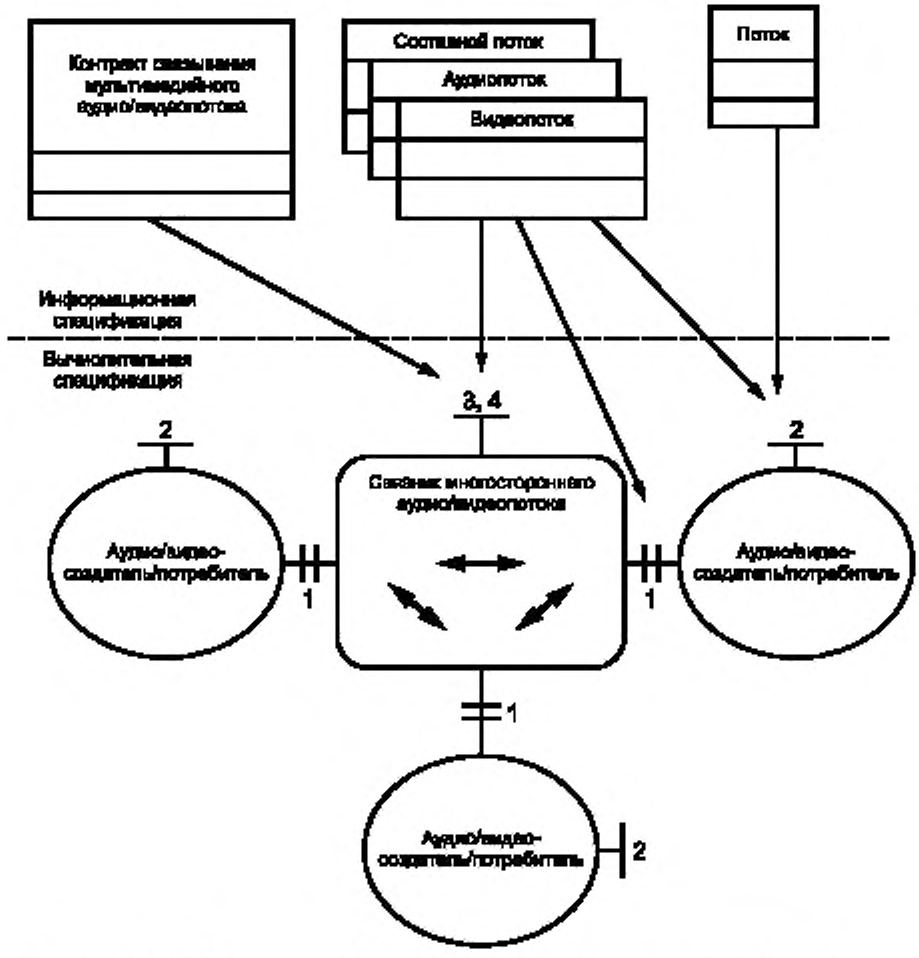



Рисунок 33 — Отображение информационной спецификации в вычислительную

Интерфейс операций создателя/потребителя аудио/видеопотока (2) будет отражен в операциях, определенных в классе «поток». Интерфейс потока (1) имеет характеристики атрибутов подкласса «потока». Атрибуты, определенные в информационной спецификации, будут представлены параметрами в вычислительных операциях. Имена операций и параметров в обеих спецификациях не зависят друг от друга.

12.2.4.2 Спецификация на IDL

Примечание — Сказанное ниже тесно связано с IDL [2].

БМ-ОРО описывает вычислительную модель, применимую для распределенных приложений, но не предоставляет конкретный язык спецификаций для вычислительных объектов и интерфейсов. Поэтому для вычислительной спецификации контролирующего интерфейса связника потока здесь использован дополнительный язык спецификаций IDL (см. таблицу 1). IDL предоставляет средства для выражения вычислительной спецификации, ориентированной на телекоммуникации и мультимедийность. Полученная спецификация IDL основана на информационной спецификации.

Таблица 1 — Спецификация на IDL контролирующего интерфейса связника потока

```
interface template StreamBindingControl Interface; /★(3), (4) тип интерфейса операций ★/
typedef sequence <Flow>StreamInterface;
operations
void ChangeQosStreamBinding (in StreamBindingId Binding,
        in QOS RequestedQos, out QOS ProvidedQos);
void RemoveStreamBinding (in StreamBindingId Binding,
        out StreamBindingId RemainingBindings);
void AddNewUser (in A VuserIdNewuser, in StreamInterface NewPlows,
        in QOS RequestedQos, out QOS ProvidedQos, out ResultReport StatusBinding);
/★ В административном управлении приложением, системой и сетью могут быть определены дополнительные управляющие операции связника потока ★/
/★ behaviour «Экземпляр этого шаблона интерфейса обеспечивает другим вычислительным объектам осуществление контролирующих операций над объектом 'многосторонний связник.'» ★/
```

12.2.4.3 Вычислительный выбор конфигурации многостороннего аудио/видеообмена

Связник многостороннего аудио/видеопотока может быть уточнен для получения инженерной конфигурации. Возможно несколько решений, но во многих распределенных мультимедийных системах используется один выбор. Несколько реализаций распределенных многосторонних систем имеют функциональный компонент, называемый аудио/видеоконтроллером и диспетчером, который управляет аудио/видеопотоками. Он получает все аудио/видеопотоки создателей и отправляет их (возможно, после переработки) потребителям. Этот подход адаптирован для связника многостороннего аудио/видеопотока на рисунке 34.

Аудио/видеоконтроллер и диспетчер отвечает за перенаправления контролирующие операции потоков (3 и 4) к каждому из связников подпотоков (5). Он также имеет дело с установлением, контролем и освобождением аудио/видеосвязываний между создателями и потребителями. Он согласовывает требования и цели, идентифицированные в предпринимательской спецификации (Например, алгоритмы кодирования и сжатия, скорость кадров). Связник аудио/видеопотока соединяет каждого создателя/потребителя аудио/видеопотока с аудио/видеоконтроллером и диспетчером.

12.2.5 Инженерная спецификация

В настоящем подразделе приведена инженерная спецификация многостороннего связующего аудио/видеообъекта, определенного в вычислительной спецификации.

12.2.5.1 Переход от вычислительной к инженерной спецификации

Должно быть выполнено отображение вычислительной спецификации в инженерную. Эта инженерная спецификация сохраняет поведение, описанное в вычислительной спецификации.

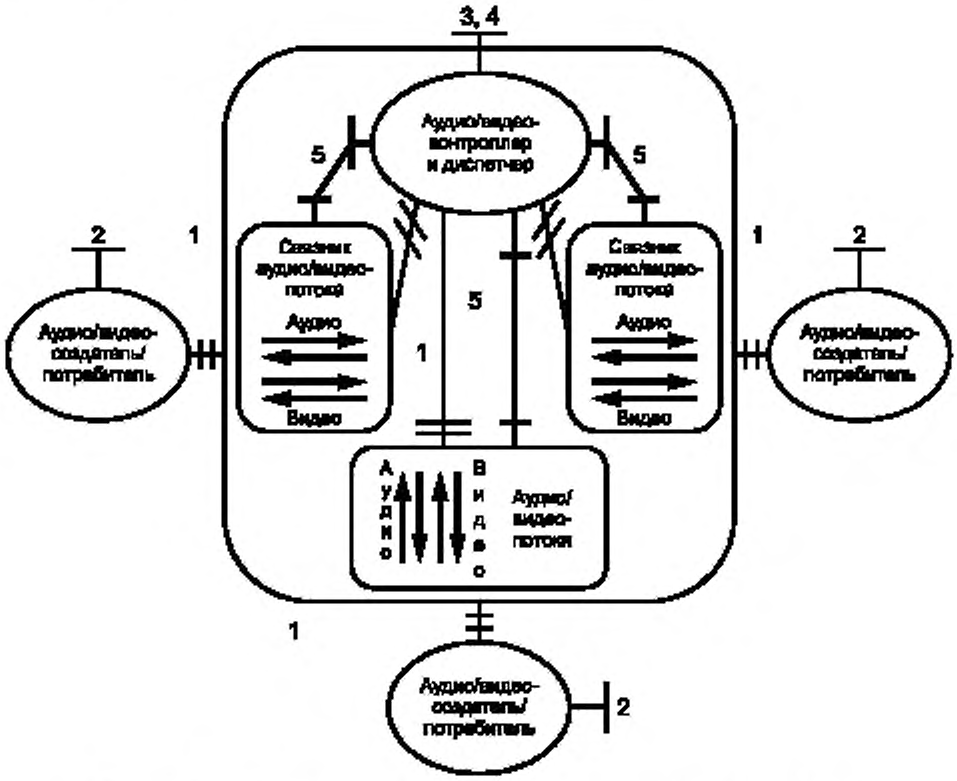



Рисунок 34 — Объекты, участвующие в многостороннем аудио/видеообмене

Инженерное преобразование интерфейса потока (см. рисунок 35, поз. 1) приводит к созданию канала потока создателя/потребителя, специализированного для непрерывных потоков. Параметры КУ, связанные с аудио/видеопотоком и определенные в вычислительной спецификации, влияют на выбор компонентов канала потока.

Интерфейс операций (см. рисунок 35, поз. 2) отражается в инженерной спецификации как конфигурация канала клиент/сервер, как определено в БМ-ОРО. Специфичные для интерфейсов контракты среды (например, безопасность), принимаются во внимание при установлении канала между рассматриваемыми вычислительными объектами.

Вычислительный *контролирующий интерфейс связника потока* (см. рисунок 35, поз. 3—5) в инженерном представлении располагается в разных узлах. Коммуникация между ними (не показанная на рисунке 35) осуществляется через контролирующие каналы, имеющие такую же структуру, как и каналы операций. Более того, могут быть созданы обеспечивающие объекты (например, объект синхронизации) для административного управления и контроля за набором взаимосвязанных каналов потоков.

*Аудио/видеосоздатели/потребители,* как и *аудио/видеоконтроллер и диспетчер,* преобразуются в базовые инженерные объекты. Если эти объекты распределены по разным узлам, то должна быть проведена дальнейшая декомпозиция и должны быть созданы несколько инженерных объектов и каналов. На рисунке 35 показано отображение двух вычислительных аудио/видеосоздателей/потребителей в инженерную спецификацию.

12.2.5.2 Инженерный канал потока

Понятие канала ОРО предоставляет инженерные средства для обеспечения прозрачности взаимодействия между базовыми инженерными объектами. Канал состоит из трех инженерных объектов: протокольного, связника и заглушки. Через канал передаются два разных типа информации. Во-первых, операции контроля, позволяющие, например, согласовать КУ. Этим операциям нужна

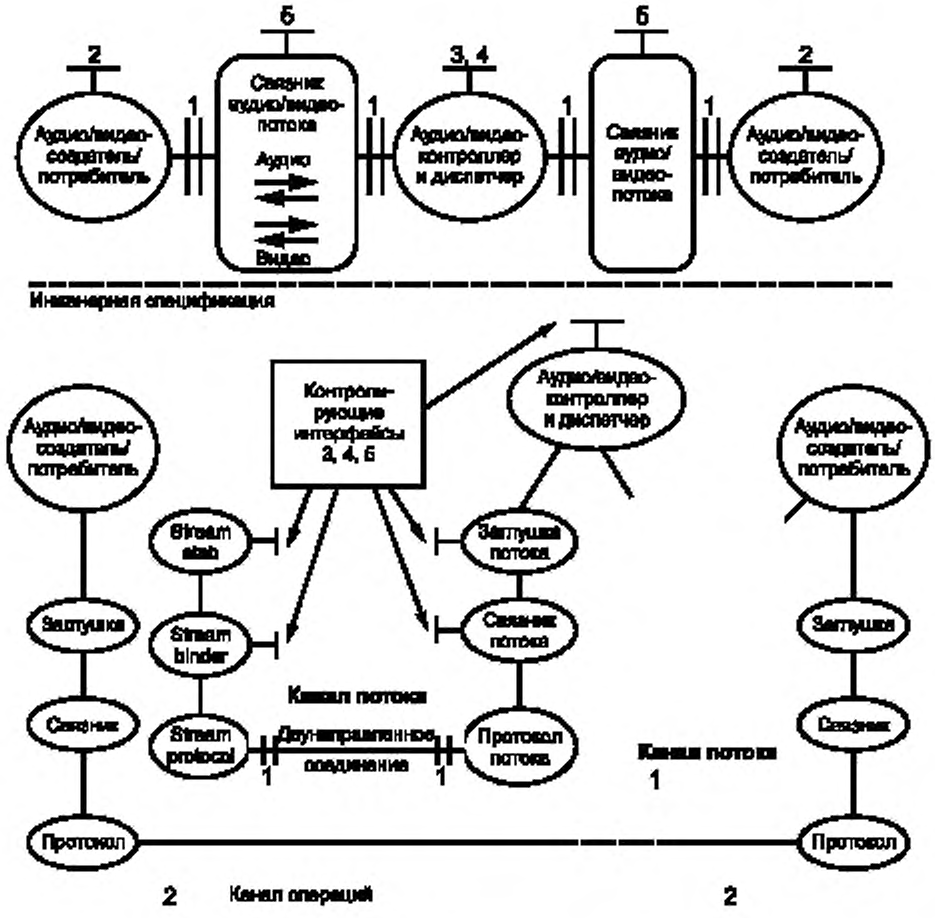



Рисунок 35 — Отображение вычислительной спецификации в инженерную

узкая полоса, но они требовательны к надежности. Во-вторых, взаимодействия реального времени, такие как обмен звуковой и визуальной информацией, для которого нужна широкая полоса, но низкая надежность. Следовательно, канал разделяется на каналы операций и потока, каждый из которых имеет свои характеристики. В данном подразделе сначала вводятся понятия заглушек, связников и протокольных объектов, а затем описывается, как устанавливается конкретный канал потока.

**Заглушка** предоставляет функции адаптации для обеспечения взаимодействия между интерфейсами базовых инженерных объектов в разных узлах. Для вызовов операций заглушка обеспечивает маршализацию/размаршализацию параметров операции для обеспечения прозрачности доступа взаимодействий. Потоки требуют от заглушек других функциональных возможностей из-за другого характера передаваемой информации. Заглушка должна предоставлять средства для кодирования и декодирования аудио/видеоинформации. Более того, данные, доступные создателю или потребителю аудио/видеопотока, должны сопровождаться сообщениями, и заглушка потока предоставляет контролирующие операции над локальными ресурсами (например, увеличение размера буфера) и сообщениями о событиях, относящихся к потоку (например, изменение КУ, в буфере нет доступного места, потеря данных и т. п.). Заглушка имеет интерфейс представления для использования привязанным к каналу объектом и контролирующий интерфейс, например для управления КУ.

**Связники** взаимодействуют друг с другом для обеспечения целостности связывания. Они хранят информацию о канале. Связники также отвечают за правильность указателя интерфейса и за взаимодействие с объектом «релокатор» для восстановления информации о положении интерфейса после ошибки связывания. В случае потоков они хранят информацию о требуемом КУ. Связник имеет контролирующий интерфейс, который позволяет изменять конфигурацию канала и уничтожать весь канал или его часть.

**Протокольные объекты** обеспечивают, чтобы вычислительные объекты могли взаимодействовать друг с другом на расстоянии. Протокольные объекты необходимы, если связываемые вычислительные объекты расположены в разных узлах. В общем случае используется метод УВП, когда операция отправляется в виде сообщения удаленному протокольному объекту, который способен его получить. Объект, которому предназначен вызов, выполняет процедуру и оправляет обратно ответное сообщение. Для многостороннего связывания видео/аудиопотока протокольные объекты типа УВП подходят для вычислительных объектов, которые вызывают операции, например, для каналов административного управления и контроля (см. рисунок 35, поз. 3—5). Однако для обмена непрерывными потоками требуется другой специализированный протокол (называемый протоколом потока) без методов УВП. УВП требует, чтобы каждый передаваемый элемент буфферизованных данных был связан с отдельным действием, т. е. в этом случае нет взаимосвязи между предыдущим и последующим вызовом УВП. Непрерывному потоку нужны взаимосвязи между вызовами, поэтому применяется протокол потока, который создает виртуальный канал между двумя протокольными объектами для длительного обмена аудио/видеопотоком. В таком случае взаимосвязи между аудио- и видеоданными могут быть определены особо. Протокольные объекты могут взаимодействовать с объектами вне канала для получения необходимой им информации.

12.2.6 Технологическая спецификация

В данном подразделе описывается начальная реализация связывания потока с использованием системы ANSA, которая основана на архитектуре, аналогичной архитектуре OPO. ANSA используется как распределенная платформа, которая предоставляет то преимущество, что реализует несколько инженерных методов (например, интерфейсы операций поддерживаются ANSA). Представленная здесь технологическая спецификация соответствует реализации многосторонней видеотелефонной услуги, позволяющей конечным пользователям обмениваться аудио- и видеоинформацией через настольные компьютеры. Пользователи могут динамически добавляться или удаляться из текущего сеанса.

Технологическая спецификация определяется спецификациями с других точек зрения, но реализация основывается главным образом на инженерной спецификации. Технологическая спецификация состоит из описания технического и программного обеспечения, которое может реализовать инженерную спецификацию при учете дополнительных предпринимательских требований (например, какое программное и техническое обеспечение доступно).

На рисунке 36 показано отображение между инженерными объектами и технологическими решениями. Приведенная на рисунке схема ограничена реализацией каналов потока и операций; показано, как инженерные объекты реализуются в технических и программных компонентах.

**12.3 Пример административного управления — метрический объект**

ИСО/МЭК 10164-11 (метрические объекты и атрибуты) специфицирует смысл метрического объекта, который сканирует значение одного и того же атрибута из идентифицированного наблюдаемого объекта через регулярные интервалы времени (период дробления — ПД), обновляет оценку среднего наблюдаемого значения атрибута и применяет к этой оценке пороговый метод, который создает сообщение при превышении порога.

12.3.1 Предпринимательская спецификация

**Сообщества**

Сообщество метрических операций включает в себя следующие роли:

- сканера — роль объекта-клиента, который отвечает за инициализацию сканирования ассоциированного с ним наблюдаемого объекта, играющего роль наблюдаемого;

- наблюдаемого — роль объекта-сервера, который отвечает на сканирование значением наблюдаемого атрибута.

Сообщество метрического контроля включает в себя следующие роли:

- метрического контроллера — роль объекта-клиента для управляющего объекта, ответственного за инициализацию операций метрического контроля на одном или нескольких метрических объектах;

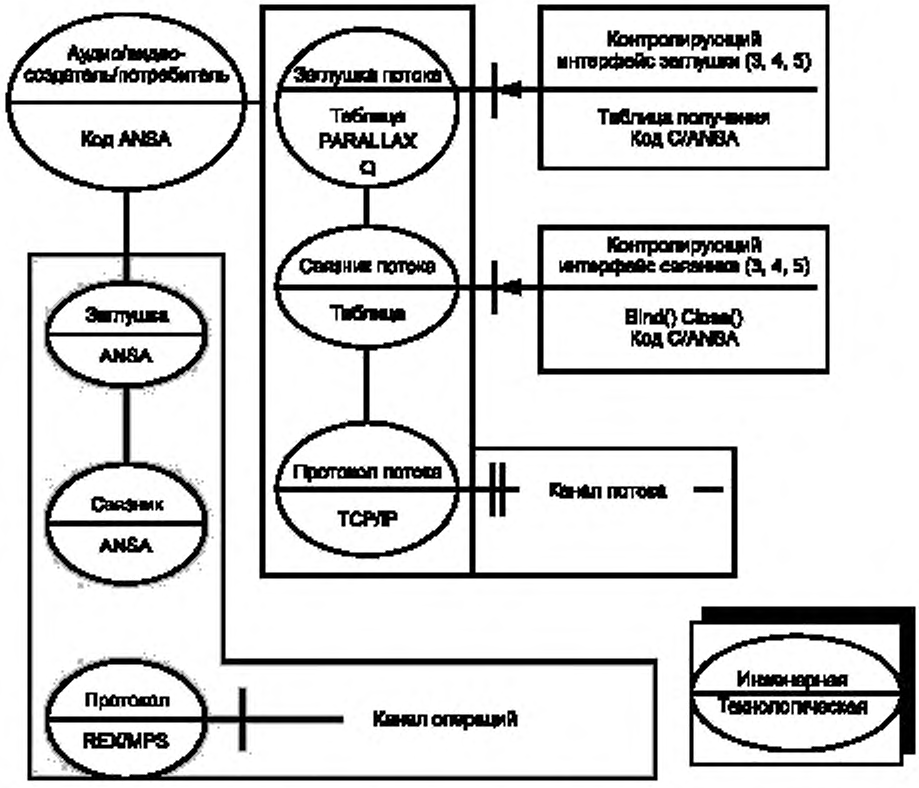



Рисунок 36 — Отображение инженерной спецификации в технологическую

- метрического контролера — роль объекта-сервера для метрического объекта, который отвечает на контролирующие операции изменением состояния метрического объекта в части, относящейся к параметрам алгоритма, связанного с метрическим поведением.

Сообщество метрических сообщений включает в себя следующие роли:

- метрического уведомителя — роль объекта-клиента для метрического объекта, который создает сообщения, относящиеся к метрическому алгоритму, действующему на сканируемые значения наблюдаемого атрибута;

- распространителя сообщений — роль объекта-сервера, который получает созданные сообщения для последующего распространения.

**Политика**

Политика сканирования — объект-сканер должен инициировать сканирование через регулярные интервалы времени, установленные управляющим атрибутом «период дробления».

Политика наблюдения — при получении запроса скана наблюдаемый объект должен отвечать наблюдаемым значениям. Если он не может ответить, то возникает ошибка, и скан является недопустимым для этого наблюдения.

Политика метрического контроля — метрический объект должен отражать изменения параметров алгоритма, выполненные в роли метрического контролера в ходе обрабатывающего поведения его метрического алгоритма усреднения. Рекомендуется контроль за следующими параметрами алгоритма: период дробления, период передвижения, указатель взаимосвязей экземпляра наблюдаемого объекта, идентификатор наблюдаемого атрибута, порог запуска сообщения, переустановка порога.

Политика метрических сообщений — метрический объект должен создавать сообщение, когда среднее наблюдаемое значение пересечет порог запуска сообщения. Метод гистерезиса обеспечивает переустановку порога: когда порог пересекается в обратном направлении, то он становится вновь готовым для последующих сообщений.

12.3.2 Информационная спецификация

Рассматриваемыми информационными объектами являются метрический и наблюдаемый объекты.

**meanMonitorMetricObject InfoDescription {**

**Атрибуты:**

**ид наблюдаемого атрибута — идентификатор, используемый для получения значения наблюдаемого атрибута.**

**период дробления — время между последовательными сканированиями значения наблюдаемого атрибута.**

**период передвижения — эффективный промежуто времени, через которое вычисляется среднее значение наблюдаемого атрибута.**

**полученная оценка — текущее значение среднего, полученное метрическим объектом с помощью сканирования и поведения алгоритма обновления.**

**порог запуска сообщения — значение, с которым должна сравниваться полученная оценка в конце каждого алгоритма обновления; сообщение создается, когда запуск взведен и значение полученной оценки больше значения порога. После сообщения запуск не взведен до последующей переустановки с помощью гистерезиса.**

**переустановка порога — значение, с которым должна сравниваться полученная оценка в конце каждого алгоритма обновления; запуск взводится, когда значение полученной оценки меньше, чем данное значение.**

**Состояния:**

**операционное состояние — отражает, находится ли метрический объект в рабочем состоянии.**

**состояние сканирования — значения:**

**ожидает (следующего старта через интервал дробления),**

**сканирует (ожидает от наблюдаемого объекта результата сканирования).**

**состояние запуска — значения: взведен, не взведен.**

**Инварианты:**

**период передвижения должен быть больше периода дробления.**

**значение переустановки порога должно быть меньше, чем значение порога запуска сообщения.**

**Переходы между состояниями:**

**В рабочем операционном состоянии могут происходить следующие переходы информационных состояний:**

**ожидает −> сканирует — происходит по достижении метрическим объектом следующего периода дробления.**

**сканирует −> ожидает — происходит, когда:**

**а) значение наблюдаемого атрибута получено, алгоритм обновил полученную оценку и применил обработку порога, или**

**б) скан является недопустимым.**

**взведен −> не взведен — происходит, когда обработка порога выявила пересечение порога запуска сообщения (что привело к созданию сообщения),**

**не взведен −> взведен — происходит, когда обработка порога выявила пересечение переустановки порога.**

**Взаимосвязи:**

**Экземпляр наблюдаемого объекта — указатель на наблюдаемый объект, который имеет идентифицированный наблюдаемый атрибут, сканируемый в каждый период дробления.**

**}**

«Наблюдаемым объектом» является любой объект, который имеет идентифицируемый атрибут со сканируемым значением. Единственным интересующим состоянием наблюдаемого объекта является текущее значение его наблюдаемых атрибутов.

**observedObject InfoDescription {**

**Атрибуты:**

**наблюдаемый атрибут — любой идентифицированный атрибут, имеющий значение целочисленного или вещественного типа.**

**}**

Следующий информационный объект моделирует содержимое объекта записи тревоги от качества услуги, связанный событием пересечения в положительном направлении порога запуска сообщения:

**qualityOfServiceAlarmRecord InfoDescription {**
**Атрибуты:**
**идентификатор экземпляра метрического объекта**
**идентификатор экземпляра наблюдаемого объекта**
**идентификатор наблюдаемого атрибута**
**время сообщения**
**значение полученной оценки**
**}**

12.3.3 Вычислительная спецификация

Размещение метрического объекта в системе, отличной от системы наблюдаемого объекта, находится вне области применения стандарта по функциям административного управления метрическими объектами.

Приведенные ниже вычислительный объект и шаблоны интерфейсов показывают декомпозицию вычислительных объектов, позволяющую проводить распределенное сканирование наблюдаемого объекта метрическим объектом.

**meanMonitorMetric ComputationalObject {**
**Интерфейсы сервера:**
**meanMonitorControl_Server**
**Интерфейсы клиента:**
**scanObservedObjectValue_Client**
**qualityOfServiceAlarm_Client**
**Поведение — см. ИСО/МЭК 10164-11**
**}**
**Шаблоны интерфейсов:**
**meanMonitorControl Interface {**
**Операции (атрибуты, имеющие методы get и replace):**
**экземпляр наблюдаемого объекта**
**идентификатор наблюдаемого атрибута**
**период дробления**
**период передвижения**
**полученная оценка**
**порог запуска сообщения**
**переустановка порога**
**операционное состояние**
**Контракт среды:**
**контролирующие операции должны воздействовать на параметризованное поведение метрического алгоритма сканирования, возникающего возле установки атрибутов.**
**}**
**scanObservedObjectValue Interface {**
**Операции:**
**для получения текущего значения наблюдаемого атрибута клиент вызывает операцию get над экземпляром наблюдаемого объекта.**
**Контракт среды:**
**если операция сканирования не привела к возврату значения наблюдаемого объекта до начала следующего периода дробления, то это сканирование является недействительным и не используется для процесса обновления метрического алгоритма.**
**}**
**qualityOfServiceAlarm Interface {**
**Операции:**
**«ISO/IEC 10164-4»:qualityOfServiceAlarmNotification**
**Контракт среды:**
**значения сообщения сигнала тревоги от качества услуги устанавливаются равными значениям, приведенным в спецификации информационного объекта для записи сигнала тревоги.**
**}**

**12.4 Пример базы данных**

Этот пример основан на (очень упрощенной) системе обработки заказов, которая обеспечивает деловую актитвность торговой организации (или части более сложной организации), имеющей дело с поставками покупателям продукции некоторых видов.

Торговая организация имеет ряд товарных складов, используемых для хранения запасов продукции. Эта продукция продается покупателям, являющимся другими организациями, которые были признаны кредитоспособными торговыми партнерами. Продажа включает в себя представление покупателем заказа на указанное количество одного или нескольких видов продукции; требуемая продукция поставляется покупателю со склада; покупателю сообщается стоимость продукции; наконец, записывается получение платежа.

12.4.1 Предпринимательская спецификация

Для этой системы могут быть идентифицированы следующие предпринимательские объекты:

- торговая организация — сообщество, которое уполномочено торговать;
- взаимодействие с покупателем — категория в торговой организации, позволяющая покупателям размещать заказы;
- административное управление складом — категория в торговой организации, отвечающая за операции на складе;
- счетчик — категория в торговой организации, отвечающая за финансовые вопросы;
- обработка заказов — категория в торговой организации, отвечающая за обработку и сопровождение записей заказов;
- покупатель — организация, которая торгует (размещает заказы) с торговой организацией.

Примеры этих предпринимательских объектов показаны на рисунке 37.

12.4.2 Информационная спецификация

Возможная инвариантная схема для данного примера, выраженная с помощью нотации ММО, приведена на рисунке 38 с примерами атрибутов и операций. Она представляет состояние, поведение и взаимосвязи между информационными объектами, относящимися к торговой организации. В частности, информационный объект «покупатель» представляет информацию о предпринимательском объекте «покупатель», которая требуется торговой организации для обеспечения торговли с ним.

12.4.3 Вычислительная спецификация

Вычислительная спецификация для этого примера имеет две существенных характеристики:

- имеется несколько пользователей, каждый из которых соответствует одному из предпринимательских объектов (взаимодействие с покупателем, административное управление складом, счетчик), и требуемые им услуги относятся к некоторой части информационной схемы;
- все экземпляры информационной схемы требуют совместно используемого, постоянного представления.

Эти два требования могут быть удовлетворены системой базы данных; возможная вычислительная модель для данного примера показана на рисунке 39.

В этой вычислительной модели каждый из объектов «пользователь» имеет тип, соответствующий взаимодействию с покупателем, административному управлению складом или счетчику. Каждый пользовательский процесс предоставляет соответствующие типу его клиента услуги, которым нужны некоторые (вряд ли все) объекты информационной схемы. Все экземпляры (но не операции) информационной схемы содержатся в объектах «база данных», а их политика распределения по разным базам данных может быть связана, например, с:

- распределением объектов по типам: все объекты «покупатель» и «заказ» — в одной базе данных, а «склад» и «продукция» — в другой;
- распределение объектов по некоторым свойствам: одна база данных хранит экземпляры всех типов объектов для одного склада (например, электрическая продукция), а другая — для другого (например, строительные материалы);
- обе указанные выше политики могут быть скомбинированы с дублированием некоторых объектов для обеспечения различных видов требований обработки (а возможна и дальнейшая декомпозиция объектов).

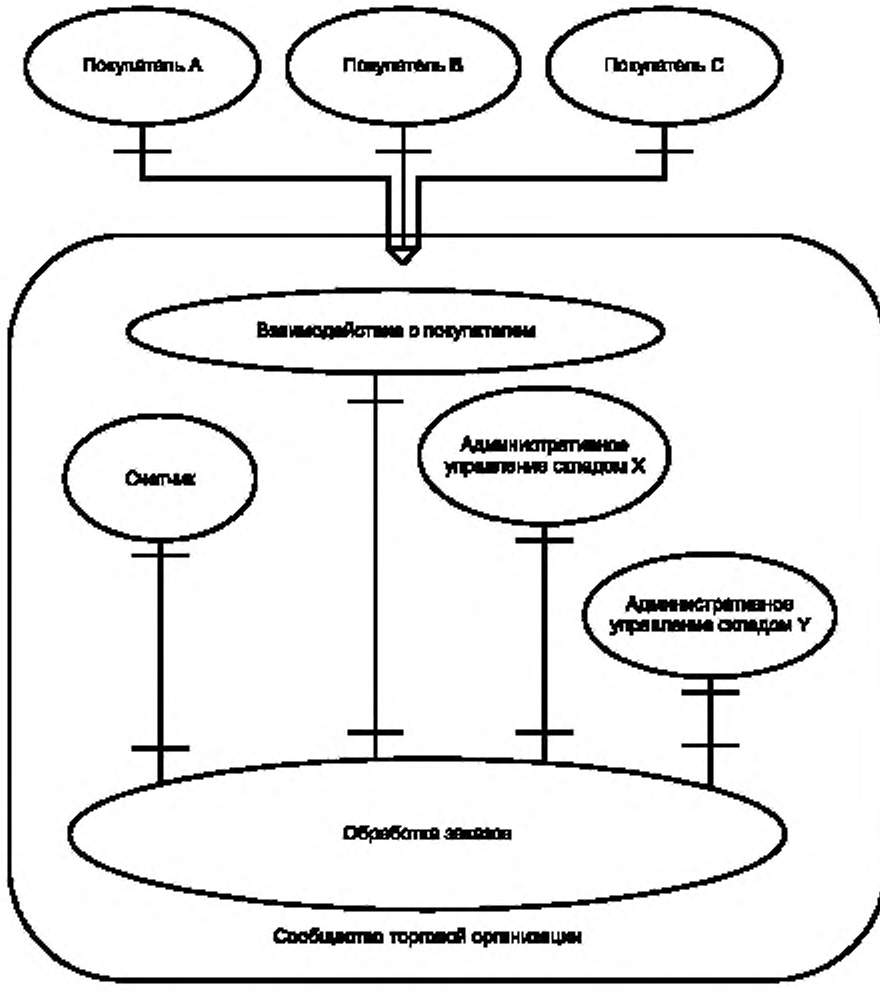



Рисунок 37 — Система обработки заказов — предпринимательская спецификация

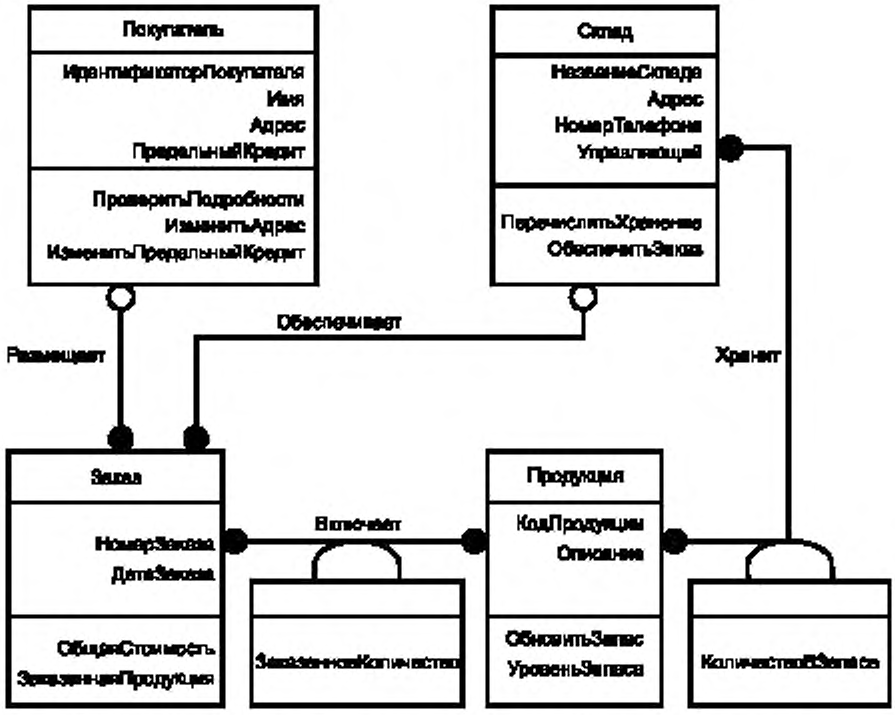



Рисунок 38 — Система обработки заказов — информационная спецификация

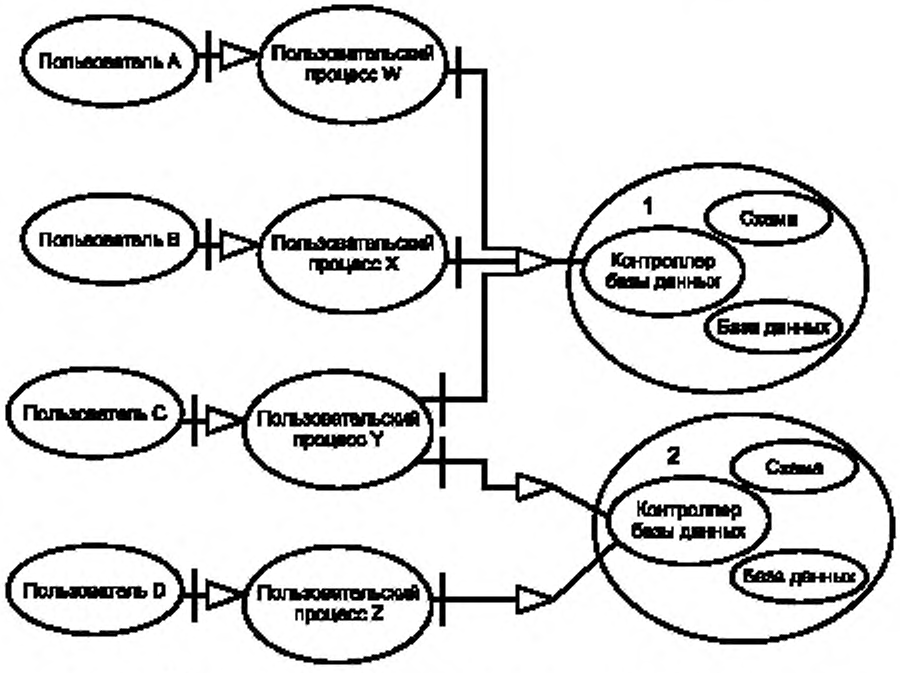



Рисунок 39 — Система обработки заказов — вычислительная спецификация

ПРИЛОЖЕНИЕ А
(справочное)

## Библиография

[1](Rumbaugh 91) J. RUMBAUGH et al.: Object oriented modelling and design, Prentice Hall, 1991

[2](OMG IDL) The Common Object Request Broker: Architecture and Specification, OMG Document Nomber 91.12.1, Draft edition, December 1991

[3](Gay 95) V.GAY, P.LEYDEKKERS and R.HUIS IN T VELD: Specification of Multiparty Audio and Video Interaction Based on the Reference Model of Open Distributed Processing, Computer Networks and ISDN Systems. — Special issue on RM-ODP, 1995

УДК 681.324:006.354 ОКС 35.080 П85 ОКСТУ 4002

Ключевые слова: взаимосвязь открытых систем, управление данными, открытая распределенная обработка, архитектура, базовая модель

Редактор *В. П. Огурцов*
Технический редактор *О. Н. Власова*
Корректор *Н. И. Гаврищук*
Компьютерная верстка *Т. Ф. Кузнецовой*

Изд. лиц. № 02354 от 14.07.2000. Сдано в набор 12.02.2004. Подписано в печать 20.04.2004. Усл. печ. л. 9,30. Уч.-изд. л. 9,20. Тираж 300 экз. С 1981. Зак. 501.

ИПК Издательство стандартов, 107076 Москва, Колодезный пер., 14.
http://www.standards.ru e-mail: info@standards.ru
Набрано и отпечатано в Калужской типографии стандартов, 248021 Калуга, ул. Московская, 256.
ПЛР № 040138

gost.e-ecolog.ru