ГОСУДАРСТВЕННЫЙ СТАНДАРТ РОССИЙСКОЙ ФЕДЕРАЦИИ

Информационная технология

Взаимосвязь открытых систем

УПРАВЛЕНИЕ ДАННЫМИ И ОТКРЫТАЯ РАСПРЕДЕЛЕННАЯ ОБРАБОТКА

Часть 3

Архитектура

Издание официальное

БЗ 3—2001/46

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 ПРИНЯТ И ВВЕДЕН В ДЕЙСТВИЕ Постановлением Госстандарта России от 20 ноября 2001 г. № 467-ст

3 Настоящий стандарт содержит полный аутентичный текст международного стандарта ИСО/МЭК 10746-3—96 «Информационная технология. Взаимосвязь открытых систем. Управление данными и открытая распределенная обработка. Часть 3. Архитектура»

4 ВВЕДЕН ВПЕРВЫЕ

© ИПК Издательство стандартов, 2002

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 . . . 1
3.1 Описательные определения . . . 1
3.2 Сокращения . . . 3
4 Общие положения . . . 3
4.1 Точки зрения . . . 3
4.1.1 Понятия . . . 3
4.1.2 Использование точек зрения . . . 3
4.2 Языки точек зрения ОРО . . . 4
4.2.1 Понятие . . . 4
4.2.2 Использование языков точек зрения . . . 4
4.3 Функции ОРО . . . 4
4.4 Прозрачность распределения ОРО . . . 4
4.4.1 Понятия . . . 4
4.4.2 Использование прозрачности распределения . . . 5
4.5 Стандарты, вытекающие из общей схемы . . . 5
4.6 Соответствие . . . 6
5 Предпринимательский язык . . . 6
5.1 Понятия . . . 6
5.2 Структурирующие правила . . . 6
5.3 Соответствие и опорные точки . . . 7
6 Информационный язык . . . 7
6.1 Понятия . . . 8
6.2 Структурирующие правила . . . 8
6.3 Соответствие и опорные точки . . . 9
7 Вычислительный язык . . . 9
7.1 Понятия . . . 9
7.2 Структурирующие правила . . . 10
7.2.1 Правила наименования . . . 10
7.2.2 Правила взаимодействия . . . 11
7.2.2.1 Правила взаимодействия для сигналов . . . 11
7.2.2.2 Правила взаимодействия для потоков . . . 11
7.2.2.3 Правила взаимодействия для операций . . . 12
7.2.2.4 Правила для параметров . . . 12
7.2.2.5 Потоки, операции и сигналы . . . 12
7.2.3 Правила связывания . . . 13
7.2.3.1 Правила неявного связывания для интерфейсов операций сервера . . . 13
7.2.3.2 Правила элементарного связывания . . . 13
7.2.3.3 Правила составного связывания . . . 13
7.2.4 Правила для типов . . . 14
7.2.4.1 Правила образования подтипов сигнатур для интерфейсов сигналов . . . 14
7.2.4.2 Правила образования подтипов сигнатур для интерфейсов потоков . . . 14
7.2.4.3 Правила образования подтипов сигнатур для интерфейсов операций . . . 14
7.2.5 Правила для шаблонов . . . 15

7.2.5.1 Правила для шаблонов вычислительных объектов . . . 15
7.2.5.2 Реализация вычислительного интерфейса . . . 15
7.2.5.3 Реализация шаблона вычислительного объекта . . . 15
7.2.6 Правила для отказов . . . 15
7.2.7 Правила переносимости . . . 16
7.3 Соответствие и опорные точки . . . 16
8 Инженерный язык . . . 17
8.1 Понятия . . . 17
8.2 Структурирующие правила . . . 18
8.2.1 Правила для каналов . . . 19
8.2.1.1 Заглушки . . . 20
8.2.1.2 Связники . . . 21
8.2.1.3 Протокольные объекты . . . 21
8.2.1.4 Пересечения . . . 21
8.2.2 Правила для указателей интерфейсов . . . 21
8.2.3 Правила для распределенного связывания . . . 23
8.2.4 Правила перемещения . . . 23
8.2.5 Правила для кластеров . . . 24
8.2.6 Правила для капсулы . . . 25
8.2.7 Правила для узла . . . 26
8.2.8 Правила прикладного управления . . . 27
8.2.9 Правила для отказов . . . 28
8.3 Соответствие и опорные точки . . . 28
9 Технологический язык . . . 28
9.1 Понятия . . . 28
9.2 Структурирующие правила . . . 29
9.3 Соответствие и опорные точки . . . 29
10 Правила согласования . . . 29
10.1 Соответствие вычислительной и информационной спецификаций . . . 30
10.2 Соответствие инженерной и вычислительной спецификаций . . . 30
11 Функции ОРО . . . 31
12 Функции управления . . . 32
12.1 Функция управления узлом . . . 32
12.1.1 Управление связками . . . 32
12.1.2 Доступ к часам и управление таймером . . . 32
12.1.3 Создание каналов и обнаружение интерфейсов . . . 32
12.1.4 Реализация шаблона капсулы и удаление капсулы . . . 32
12.2 Функция управления объектом . . . 33
12.3 Функция управления кластером . . . 33
12.3.1 Создание контрольной точки кластера . . . 33
12.3.2 Удаление, деактивация и отказ кластера . . . 33
12.3.3 Реактивация и восстановление кластера . . . 34
12.3.4 Миграция кластера . . . 34
12.4 Функция управления капсулой . . . 34
12.4.1 Реализация шаблона кластера . . . 34
12.4.2 Удаление капсулы . . . 34
13 Функции координации . . . 35

13.1 Функция уведомления о событии . . . 35
13.1.1 Понятия . . . 35
13.1.2 Правила . . . 35
13.2 Функция создания контрольной точки и восстановления . . . 35
13.2.1 Создание контрольной точки . . . 35
13.2.2 Восстановление . . . 36
13.3 Функция деактивации и реактивации . . . 36
13.3.1 Деактивация . . . 36
13.3.2 Реактивация . . . 36
13.4 Функция группирования . . . 36
13.4.1 Понятия . . . 36
13.4.2 Правила . . . 36
13.5 Функция дублирования . . . 37
13.6 Функция миграции . . . 37
13.6.1 Дублирование . . . 37
13.6.2 Деактивация и реактивация . . . 37
13.7 Функция транзакции . . . 37
13.7.1 Понятия . . . 37
13.7.2 Правила . . . 37
13.8 Функция транзакции ACID . . . 38
13.9 Функция отслеживания указателей инженерных интерфейсов . . . 38
14 Функции хранилища . . . 38
14.1 Функция сохранения . . . 38
14.1.1 Понятия . . . 38
14.1.2 Правила . . . 38
14.2 Функция организации информации . . . 39
14.3 Функция перемещения . . . 39
14.3.1 Понятия . . . 39
14.3.2 Правила . . . 39
14.4 Функция хранилища типов . . . 40
14.4.1 Правила . . . 40
14.5 Функция торга . . . 40
14.5.1 Понятия . . . 40
14.5.2 Правила . . . 40
15 Функции безопасности . . . 41
15.1 Понятия . . . 41
15.2 Функция управления доступом . . . 41
15.3 Функция проверки безопасности . . . 41
15.4 Функция аутентификации . . . 41
15.5 Функция целостности . . . 42
15.6 Функция конфиденциальности . . . 42
15.7 Функция неопровержения . . . 42
15.8 Функция управления ключом . . . 43
16 Прозрачность распределения ОРО . . . 43
16.1 Прозрачность доступа . . . 44
16.2 Прозрачность отказа . . . 44
16.2.1 Понятие . . . 44

16.2.2 Правила . . . 44
16.2.2.1 Дублирование . . . 44
16.2.2.2 Создание контрольных точек и восстановление . . . 44
16.3 Прозрачность положения . . . 44
16.4 Прозрачность миграции . . . 44
16.4.1 Понятие . . . 44
16.4.2 Правила . . . 45
16.5 Прозрачность постоянства . . . 45
16.5.1 Понятие . . . 45
16.5.2 Правила . . . 45
16.6 Прозрачность перемещения . . . 45
16.7 Прозрачность дублирования . . . 45
16.7.1 Понятие . . . 45
16.7.2 Правила . . . 45
16.8 Прозрачность транзакции . . . 46
16.8.1 Понятие . . . 46
16.8.2 Правила . . . 46
Приложение А Формальные правила для вычислительных супертипов/подтипов . . . 47
А.1 Обозначения и соглашения . . . 47
А.2 Система типов . . . 47
А.2.1 Правила типов . . . 47
А.2.2 Определения типов . . . 48
А.2.3 Алгоритм проверки типа . . . 49
А.3 Типы сигнатур интерфейсов сигналов . . . 50
А.4 Типы сигнатур интерфейсов операций . . . 50
А.5 Типы интерфейсов потоков . . . 51
А.6 Пример . . . 51

ГОСУДАРСТВЕННЫЙ СТАНДАРТ РОССИЙСКОЙ ФЕДЕРАЦИИ

Информационная технология. Взаимосвязь открытых систем

УПРАВЛЕНИЕ ДАННЫМИ И ОТКРЫТАЯ РАСПРЕДЕЛЕННАЯ ОБРАБОТКА

Часть 3

Архитектура

Information technology. Open Systems Interconnection. Data control and Open Distributed Processing. Reference model. Part 3. Architecture

Дата введения 2003—01—01

1 Область применения

В настоящем стандарте:

- определено, как специфицируются системы открытой распределенной обработки (ОРО) с использованием понятий, введенных в ГОСТ Р ИСО/МЭК 10746-2;
- идентифицированы характеристики, по которым системы относятся к системам ОРО.

В стандарте установлен каркас для координации разработки стандартов по системам ОРО.

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 10746-2—2000 Информационная технология. Открытая распределенная обработка. Часть 2. Базовая модель

ИСО/МЭК 10181-2—96* Информационная технология. Взаимосвязь открытых систем. Основы защиты информации для открытых систем. Часть 2. Основы аутентификации

ИСО/МЭК 10181-3—96* Информационная технология. Взаимосвязь открытых систем. Основы защиты информации для открытых систем. Часть 3. Управление доступом

ИСО/МЭК ПМС 10181-4—97* Информационная технология. Взаимосвязь открытых систем. Основы защиты информации для открытых систем. Часть 4. Неопровержимое получение

ИСО/МЭК 10181-5—96* Информационная технология. Взаимосвязь открытых систем. Основы защиты информации для открытых систем. Часть 5. Конфиденциальность

ИСО/МЭК 10181-6—96* Информационная технология. Взаимосвязь открытых систем. Основы защиты информации для открытых систем. Часть 6. Целостность

ИСО/МЭК 10181-7—96* Информационная технология. Взаимосвязь открытых систем. Основы защиты информации для открытых систем. Часть 7. Основы проверки защиты

ИСО/МЭК 11170-1—96* Информационная технология. Методы безопасности. Административное управление ключом. Часть 1. Основы проверки защиты

3 Определения

В настоящем стандарте используют следующие определения.

3.1 Описательные определения

В настоящем стандарте используют следующие термины, определенные в:

*Оригиналы и проекты стандартов ИСО/МЭК — во ВНИИКИ Госстандарта России.

Издание официальное

1) ГОСТ Р ИСО/МЭК 7498-1
- **синтаксис передачи;**

2) ИСО/МЭК 10181-2:
- **заявитель,**
- **обменная информация аутентификации,**
- **основной,**
- **доверительная третья сторона;**

3) ИСО/МЭК 10181-3:
- **информация управления доступом,**
- **функция решения о доступе,**
- **функция принудительного доступа,**
- **инициатор,**
- **цель;**

4) ИСО/МЭК ПМС 10181-4:
- **генератор доказательства;**
- **пользователь доказательства;**
- **верификатор доказательства,**
- **источник (неопровержимых данных),**
- **получатель (неопровержимых данных),**
- **неопровержимое доказательство,**
- **запросчик неопровержимой услуги,**
- **нотариус;**

5) ИСО/МЭК 10181-5:
- **конфиденциально защищенные данные,**
- **сокрытие,**
- **источник,**
- **получатель,**
- **вскрытие;**

6) ИСО/МЭК 10181-6:
- **целостно-защищенные данные,**
- **источник,**
- **получатель,**
- **защищать,**
- **подтверждать;**

7) ИСО/МЭК 10181-7:
- **функция сборщика сигналов тревоги,**
- **функция проверяющего сигналы тревоги,**
- **функция анализатора проверки следов,**
- **функция архивиста проверки следов,**
- **функция записывающего проверки,**
- **функция исследователя проверок следов,**
- **функция сборщика проверок следов;**

8) ИСО/МЭК 11170-1:
- **создание ключа,**
- **регистрация ключа,**
- **сертификация ключа,**
- **дерегистрация ключа,**
- **распространение ключа,**
- **сохранение ключа,**
- **архивация ключа,**
- **удаление ключа.**

Термины, определенные в ГОСТ Р ИСО/МЭК 10746-2, приведены на рисунке 1.

абстракция	объект-клиент	связка
архитектура	объект-сервер	связывание
введение	обязательство	связь
взаимодействие	опорная точка	сигнатура интерфейса
воспринимаемая опорная точка	опорная точка взаимодействия	система
группа <X>	опорная точка обмена	система ОРО
данные		создание
действие	отказ	состояние
деятельность	открытая распределенная обработка	среда
декомпозиция	ошибка	стандарты ОРО
запрещение	поведение	супертип
идентификатор	подобласть	термин
имя	подтип	тип
инвариант	политика	торг
инициализирующий объект	положение во времени	точка зрения
интерфейс	положение в пространстве	точка соответствия
информация	порождающее действие	уведомление
категория	постоянство	удаление
качество услуги	потребляющий объект	управление коммуникацией
класс	программируемая опорная точка	устанавливающее поведение
композиция	прозрачность распределения	устойчивость
контекст наименования	производящий объект	уточнение
контракт	разрешение	шаблон <x>
конфигурация	распределенная обработка	экземпляр
неисправность	реализация	элементарная
область наименования	роль	
область <x>		
объект		

Рисунок 1 — Термины, определенные в ГОСТ Р ИСО/МЭК 10746-2.

3.2 Сокращения

В настоящем стандарте используют следующие сокращения:

ДИТР — дополнительная информация для тестирования реализации;

ОРО — открытая распределенная обработка;

ЯОИ — язык определения интерфейсов.

4 Общие положения

4.1 Точки зрения

4.1.1 Понятия

4.1.1.1 Предпринимательская точка зрения — точка зрения на систему ОРО и ее среда, которая сфокусирована на назначении, области применения и политике этой системы.

4.1.1.2 Информационная точка зрения — точка зрения на систему ОРО и ее среда, которая сфокусирована на семантике и обработке информации.

4.1.1.3 Вычислительная точка зрения — точка зрения на систему ОРО и ее среда, которая позволяет ее распределить путем функциональной декомпозиции на объекты, взаимодействующие через интерфейсы.

4.1.1.4 Инженерная точка зрения — точка зрения на систему ОРО и ее среда, которая сфокусирована на механизмах и функциях, требуемых для обеспечения распределенного взаимодействия между объектами в системе.

4.1.1.5 Технологическая точка зрения — точка зрения на систему ОРО и ее среду, которая сфокусирована на выборе технологии в этой системе.

4.1.2 Использование точек зрения

Перечисленные выше точки зрения были выбраны как необходимый и достаточный набор для удовлетворения потребностей стандартов ОРО. С этих точек зрения может рассматриваться на соответствующем уровне абстракции вся система ОРО, и в этом случае среда определяет контекст, в котором работает система ОРО. С этих же точек зрения могут рассматриваться и отдельные

компоненты системы ОРО; в этом случае среда компонента включает в себя некоторую абстракцию как среды системы, так и остальных компонентов.

Примечание — Процесс абстрагирования может быть таким, что среда системы и оставшиеся компоненты объединяются в единый объект.

4.2 Языки точек зрения ОРО

4.2.1 Понятие

4.2.1.1 Язык <точки зрения> — определения понятий и правил для спецификации системы ОРО с данной точки зрения; например, инженерный язык — определения понятий и правил для спецификации системы ОРО с инженерной точки зрения.

4.2.2 Использование языков точек зрения

В настоящей базовой модели определен набор из пяти языков, каждый из которых соответствует одной из определенных в 4.1.1 точек зрения. Каждый язык используют для спецификации системы ОРО с соответствующей точки зрения. Этими языками являются:

- предпринимательский язык (определенный в разделе 5);
- информационный язык (определенный в разделе 6);
- вычислительный язык (определенный в разделе 7);
- инженерный язык (определенный в разделе 8);
- технологический язык (определенный в разделе 9).

Каждый язык использует понятия из ГОСТ Р ИСО/МЭК 10746-2 и вводит уточнения этих понятий, предписывающие правила и дополнительные, специфические для данной точки зрения, понятия, относящиеся к природе рассматриваемых спецификаций. Указанные дополнительные понятия, в свою очередь, определяются с использованием понятий ГОСТ Р ИСО/МЭК 10746-2.

Спецификация системы включает в себя спецификации одной или нескольких точек зрения. Эти спецификации должны быть взаимно согласованными. Правила для согласования структур спецификаций точек зрения приведены в разделе 10. Спецификатор должен продемонстрировать каким-либо образом, что термины в спецификациях используются согласованно. Спецификация системы, использующая спецификации нескольких точек зрения, часто ограничивает реализаторов в большей степени, чем спецификация, использующая меньше точек зрения. Объекты, идентифицированные с одной точки зрения, могут быть специфицированы с использованием языка этой или другой точки зрения. Для достижения взаимной согласованности набора спецификаций точек зрения не обязательно полностью специфицировать объект с каждой точки зрения.

Примечания

1 Список терминов, заимствованных из ГОСТ Р ИСО/МЭК 10746-2, приведен на рисунке 1.

2 Квалификация термина из ГОСТ Р ИСО/МЭК 10746-2 названием точки зрения (например, «вычислительный объект») интерпретируется как использование термина из ГОСТ Р ИСО/МЭК 10746-2, дополнительные положения для которого заданы в языке указанной точки зрения.

3 Использование неквалифицированного термина из ГОСТ Р ИСО/МЭК 10746-2 в спецификации точки зрения (например, «интерфейс») интерпретируется как термин, квалифицированный названием точки зрения (например, «вычислительный интерфейс»), если соответствующий язык точки зрения устанавливает дополнительные ограничения для этого термина.

4.3 Функции ОРО

4.3.1 Функция ОРО — функция, необходимая для обеспечения открытой распределенной обработки.

4.3.2 Использование функций ОРО

В настоящей базовой модели, в разделах 11—15, специфицированы функции, требуемые для обеспечения ОРО.

Описание каждой функции ОРО содержит:

- объяснение использования функции для открытой распределительной обработки;
- предписывающие утверждения о структуре и поведении функции, достаточные для того, чтобы гарантировать общую целостность базовой модели;
- утверждение о других функциях ОРО, от которых зависит данная.

4.4 Прозрачность распределения ОРО

4.4.1 Понятия

4.4.1.1. Прозрачность доступа — прозрачность распределения, которая маскирует различия в представлении данных и методах вызова для того, чтобы сделать возможным взаимодействие между объектами.

4.4.1.2 Прозрачность отказа — прозрачность распределения, которая маскирует (от объекта) отказ и возможное восстановление других объектов (или самого этого объекта) для того, чтобы обеспечить устойчивость к неисправностям.

4.4.1.3 Прозрачность положения — прозрачность распределения, которая маскирует использование информации о положении в пространстве при идентификации и связывании.

4.4.1.4 Прозрачность миграции — прозрачность распределения, которая маскирует (от объекта) способность системы изменить положение этого объекта. Миграцию часто используют для достижения сбалансированной загрузки и уменьшения задержки.

4.4.1.5 Прозрачность перемещения — прозрачность распределения, которая маскирует перемещение интерфейса от других связанных с ним интерфейсов.

4.4.1.6 Прозрачность дублирования — прозрачность распределения, которая маскирует использование для обеспечения интерфейса группы поведенчески взаимно совместимых объектов. Дублирование часто используют для повышения производительности и доступности.

4.4.1.7 Прозрачность постоянства — прозрачность распределения, которая маскирует (от объекта) деактивацию и реактивацию других объектов (или самого этого объекта). Деактивацию и реактивацию часто используют для обеспечения постоянного присутствия объекта, когда система не в состоянии предоставлять ему функции обработки, хранения и коммуникации непрерывно.

4.4.1.8 Прозрачность транзакции — прозрачность распределения, которая маскирует координацию действий объектов для достижения согласованности.

4.4.2 Использование прозрачности распределения

Прозрачность распределения является важным требованием конечного пользователя распределенной системы. В настоящей базовой модели определен ряд видов прозрачности распределения, что позволяет создавать системы ОРО с прозрачным, с точки зрения пользователей, распределением. Прозрачность распределения селективна; базовая модель включает в себя правила для выбора и комбинирования видов прозрачности распределения в системах ОРО.

Для прозрачности распределения каждого вида, определенного в 4.4.1, данная базовая модель содержит определения:

- схемы для выражения требований к конкретному виду прозрачности;
- процесса уточнения для преобразования спецификации, содержащей требования к конкретному виду прозрачности распределения, в спецификацию, явно выражающую маскировку, подразумеваемую этой прозрачностью.

Примечания

1 В некоторых случаях (например, прозрачности доступа) схема пуста; в других случаях (например, прозрачности транзакции) схема содержит один или несколько параметров, предписывающих точную форму требуемой прозрачности.

2 Процесс уточнения обычно требует введения дополнительного поведения, включая использование одной или нескольких функций ОРО в спецификации.

Спецификации процесса уточнения в разделе 16 являются предписывающими до уровня требований для того, чтобы гарантировать общую целостность базовой модели.

4.5 Стандарты, вытекающие из общей схемы

Настоящая базовая модель дает общую схему для определения новых стандартов и использования существующих в качестве стандартов ОРО.

Стандартами ОРО являются стандарты:

- компонентов систем ОРО;
- сборки компонентов системы ОРО;
- моделирования и спецификации систем ОРО.

Стандарты ОРО используют:

- предпринимательский язык для спецификации политики;
- информационный язык для спецификации согласованного использования и интерпретации информации в стандартах (и между ними);
- вычислительный язык для спецификации конфигурации и поведения интерфейсов;
- инженерный язык для спецификации требуемых инфраструктур;
- технологический язык для задания соответствия международным, частным и согласованным спецификациям.

Стандарты по методологии, моделированию, программированию, реализации и тестированию систем ОРО используют общую схему в целом.

Стандарты ОРО могут основываться на подмножестве данной базовой модели (например,

исключающие некоторые формы взаимодействия, конкретные функции или виды прозрачности). Такие стандарты могут быть расширены за пределы базовой модели при условии, что вводимые расширения не изменяют положения этой модели и не противоречат им. Расширения должны связывать новые термины с терминами, определенными в настоящей базовой модели: например, путем введения новых типов или правил.

Стандарты ОРО должны подчиняться всем предписывающим утверждениям настоящей базовой модели.

4.6 Соответствие

Предпринимательский, информационный, вычислительный и инженерный языки используют для спецификации требований соответствия для систем ОРО. Технологический язык может использоваться в системах ОРО для выражения соответствия стандартам ОРО. Каждый интерфейс, определенный как точка соответствия, имеет информационную спецификацию, допускающую интерпретацию взаимодействий этого интерфейса. Правила для идентификации точек соответствия выражают на вычислительном и инженерном языках.

Система ОРО соответствует стандарту ОРО, если она удовлетворяет требованиям соответствия этого стандарта.

5 Предпринимательский язык

Предпринимательский язык включает в себя понятия, правила и структуры для спецификации системы ОРО с предпринимательской точки зрения.

Предпринимательская спецификация определяет назначение, область применения и политику системы ОРО.

В данной базовой модели предписание предпринимательской точки зрения ограничено небольшим основным набором понятий и правил, относящихся к области применения и характеру предпринимательских спецификаций.

5.1 Понятия

Предпринимательский язык содержит понятия ГОСТ Р ИСО/МЭК 10746-2 и настоящего стандарта, подчиняющиеся правилам 5.2.

5.1.1 Общность — конфигурация объектов, образованная для достижения цели. Цель формулируется в виде контракта, который задает, как цель может быть достигнута.

5.1.2 <X> федерация — общность <x> областей.

5.2 Структурирующие правила

Предпринимательская спецификация определяет, а предпринимательский язык должен позволять выразить цель, область применения и политику системы ОРО в терминах каждого из следующих элементов:

- ролей, исполняемых системой;
- деятельностей, предпринимаемых системой;
- утверждений о политике системы, включая относящиеся к контрактам среды.

В предпринимательской спецификации система ОРО и среда, в которой она работает, представляются как общность. На некотором уровне описания система ОРО представляется как предпринимательский объект в общности. Цели и область применения системы ОРО определяются в терминах ролей, исполняемых ею в общности, частью которой она является, и утверждений о политике относительно этих ролей. Общность определяется в терминах каждого из следующих элементов:

- предпринимательских объектов, входящих в общность;
- ролей, исполняемых каждым из этих объектов;
- политики, управляющей взаимодействием между предпринимательскими объектами;
- политики, управляющей созданием, использованием и удалением ресурсов предпринимательскими объектами;
- политики, управляющей конфигурацией предпринимательских объектов и назначением им ролей;
- политики, относящейся к контрактам среды, управляющим системой.

Роль определяется в терминах разрешений, обязательств, запрещений и поведения предпринимательского объекта, играющего эту роль. Предпринимательский объект может исполнять в общности одну или несколько ролей, а роли, которые он может играть, определяются контрактом,

на котором основана общность. Хотя он является частью некоторой общности, предпринимательский объект может исполнять роли и в других общностях; этот вопрос должен быть оговорен в контрактах участвующих общностей. Предпринимательский объект может играть разные роли в разных общностях. Взаимодействия между предпринимательскими объектами, играющими соответствующие роли в разных общностях, могут рассматриваться как взаимодействия между этими общностями.

Примечания

1 Примеры ролей: администратор по политике, президент, поставщик услуг, владелец, управляющий, акционер, потребитель.

2 Примеры контрактов среды в предпринимательских спецификациях: требования безопасности, нормативные требования, практические указания.

3 В предпринимательской спецификации термин «<x> объект», где <x> — роль, интерпретируется как «предпринимательский объект, играющий роль <x>». Когда объект играет несколько ролей, имена могут быть сцепленными, например «объект-владелец-драйвер».

При выполнении роли объект становится субъектом для разрешений, обязательств и запрещений путем их делегирования или передачи. В некоторых ролях объектам разрешается изменять политику. Имеются пять основных типов действий относительно контракта:

- объект устанавливает обязательство для другого объекта (в это время ему должно быть разрешено устанавливать обязательства);
- объект подчиняется обязательствам другого объекта;
- объект отказывается от обязательств другого объекта;
- объект приобретает разрешение от другого объекта осуществить некоторое действие, которое ранее было для него запрещено;
- объекту запрещается осуществлять некоторое действие, которое ранее было ему разрешено.

Примечание 4 — Важным частным случаем приобретения является разрешение действия, являющегося производящим, т. е. когда объекту в подчиненной роли позволяется создавать последующие запрещения или обязательства для поведения объекта, играющего старшую роль. Это приводит к понятию о посредничестве или делегировании.

Обязательства включают в себя подсчет и ответственность за использование ресурсов. Биллинг и оплату моделируют как перераспределение ресурсов между объектами в соответствии с исполняемыми ролями.

Ресурс является либо потребляемым, либо непотребляемым. Потребляемый ресурс исчерпывается после использования некоторого его количества. В <x> федерации цель определяет ресурсы каждой <x> области, совместно используемые с другими членами федерации. Цель может сохранить за каждой областью определенную степень автономии в использовании ее собственных ресурсов. Устанавливающее поведение для <x> федерации может допускать автономию для каждой участвующей <x> области в решении, становиться или нет частью федерации.

5.3 Соответствие и опорные точки

Утверждения соответствия в предпринимательском языке требуют, чтобы поведение системы ОРО соответствовало конкретному набору целей и политики.

Реализатор, декларирующий соответствие, должен идентифицировать инженерные опорные точки, которые дают доступ к системе, и инженерные, вычислительные и информационные спецификации, которые в этих точках применимы. Тем самым идентифицированные опорные точки становятся точками соответствия. Взаимодействия в этих точках соответствия могут быть затем интерпретированы в терминах предпринимательского языка для проверки того, что предпринимательская спецификация не была нарушена.

Предпринимательские спецификации могут применяться для всех четырех классов опорных точек (программируемых, воспринимаемых, взаимодействия и обмена), идентифицированных в ГОСТ Р ИСО/МЭК 10746-2.

6 Информационный язык

Информационный язык охватывает понятия, правила и структуры для спецификации системы ОРО с информационной точки зрения.

Информационная спецификация определяет семантику информации и семантику обработки информации в системе ОРО.

В настоящей базовой модели предписание с информационной точки зрения ограничено небольшим основным набором понятий и правил, ориентированных на область применения и характер информационных спецификаций.

6.1 Понятия

Информационный язык содержит понятия ГОСТ Р ИСО/МЭК 10746-2 и настоящего стандарта, подчиняющиеся правилам 6.2.

6.1.1 Инвариантная схема — набор предикатов об одном или нескольких информационных объектах, которые всегда должны быть истинны. Предикаты ограничивают возможные состояния и изменения состояний объектов, к которым они применяются.

Примечание — Таким образом, инвариантная схема является спецификацией типов одного или нескольких информационных объектов, которая всегда должна удовлетворять при любом поведении объектов.

6.1.2 Статическая схема — спецификация состояния (в некоторый момент времени) одного или нескольких информационных объектов, удовлетворяющих ограничениям всех инвариантных схем.

Примечание — Таким образом, статическая схема является спецификацией типов одного или нескольких информационных объектов в некоторый момент времени. Эти типы являются подтипами заданных инвариантов схемы типов.

6.1.3 Динамическая схема — спецификация допустимых изменений состояний одного или нескольких информационных объектов, удовлетворяющих ограничениям всех инвариантных схем.

Примечания

1 Поведение в информационной системе может моделироваться как переход от одной статической схемы к другой, т. е. как изменение классов экземпляров с одного типа на другой.

2 В информационном языке изменение состояния, в котором участвует несколько объектов, может рассматриваться как взаимодействие между этими объектами. Не все объекты, участвующие во взаимодействии, обязательно изменяют состояние; некоторые объекты могут быть вовлечены во взаимодействие «только для чтения».

6.2 Структурирующие правила

Информационная спецификация определяет семантику информации и семантику обработки информации в системе ОРО в терминах конфигурации информационных объектов, поведения этих объектов и контрактов среды для системы.

Шаблон информационного объекта ссылается на статическую, инвариантную и динамическую схемы. Взаимосвязи между информационными объектами могут моделироваться как часть состояний этих информационных объектов. Информационные объекты являются либо элементарными, либо представлены как композиция других информационных объектов. Состояние составного объекта представляется комбинацией состояний его компонентов. Шаблон элементарного информационного объекта представляет понятие, для которого нет модели на некотором конкретном уровне абстракции. Составной информационный объект представляет производное понятие, выраженное в терминах других понятий. Так как композиция объектов включает в себя их обособление, то информационный объект, являющийся компонентом одного составного объекта, не может быть компонентом другого. Следовательно, информационные объекты, получающиеся в результате реализации шаблона составного информационного объекта, существуют только как часть реализованного составного объекта и не имеют смысла вне его.

Допустимые изменения состояний, заданные динамической схемой, могут включать в себя создание новых информационных объектов и удаление информационных объектов, задействованных в динамической схеме. Допустимые изменения состояний могут быть предметом упорядочения и временных ограничений.

Примечание 1 — Результат достижения состояния одним или несколькими информационными объектами может моделироваться как создание нового информационного объекта.

В информационной спецификации конфигурации информационных объектов и их поведение не обязательно должны быть пригодны для распределения (т. е. не обязательно должна существовать некоторая концепция отказа или размещения информационных взаимодействий).

Примечание 2 — Если информационная нотация использует понятие интерфейса, то ни один из определенных интерфейсов не может сам являться опорной точкой; таким образом, нет привязки к интерфейсам, появляющимся в реализации.

6.3 Соответствие и опорные точки

Утверждения соответствия в информационных спецификациях требуют, чтобы поведение системы ОРО соответствовало конкретному набору инвариантных, статических и динамических схем.

Реализатор, декларирующий соответствие, должен перечислить инженерные опорные точки, которые дают доступ к системе, и инженерные и вычислительные спецификации, которые в этих точках применимы. Тем самым идентифицированные опорные точки становятся точками соответствия. Взаимодействия в этих точках соответствия могут быть затем интерпретированы в терминах информационного языка для проверки того, что они согласуются с инвариантными, статическими и динамическими схемами.

Информационные спецификации могут применяться для всех четырех классов опорных точек (программируемых, воспринимаемых, взаимодействия и обмена), идентифицированных в ГОСТ Р ИСО/МЭК 10746-2.

7 Вычислительный язык

Вычислительный язык охватывает понятия, правила и структуры для спецификации системы ОРО с вычислительной точки зрения.

Вычислительная спецификация определяет функциональную декомпозицию системы ОРО на объекты, взаимодействующие через интерфейсы.

С вычислительной точки зрения приложения и функции ОРО состоят из конфигураций взаимодействующих вычислительных объектов.

7.1 Понятия

Вычислительный язык содержит понятия ГОСТ Р ИСО/МЭК 10746-2 и понятия настоящего стандарта, подчиняющиеся правилам 7.2.

7.1.1 Сигнал — элементарное совместно используемое действие, приводящее к односторонней коммуникации от инициирующего объекта к отвечающему.

Примечание — Сигнал является взаимодействием.

7.1.2 Операция — взаимодействие между объектом-клиентом и объектом-сервером, которое является либо запросом, либо сообщением.

7.1.3 Сообщение — взаимодействие (вызов), инициированное объектом-клиентом; оно приводит к передаче от этого объекта-клиента к объекту-серверу информации, запрашивающей выполнение функции этим объектом-сервером.

7.1.4 Запрос — взаимодействие, состоящее из:

- первого взаимодействия (вызова), инициированного объектом-клиентом; оно приводит к передаче от этого объекта-клиента к объекту-серверу информации, запрашивающей выполнение функции этим объектом-сервером, за которым следует;
- второго взаимодействия (завершения), инициированного объектом-сервером; оно приводит к передаче информации от объекта-сервера к объекту-клиенту в ответ на вызов.

Примечание — В запросах всегда есть пара вызов-завершение. Сообщение не имеет завершения. Таким образом, не возможна операция, состоящая из одного вызова и последовательности завершений.

7.1.5 Поток — абстракция последовательности взаимодействий, приводящих к переносу информации от объекта-производителя к объекту-потребителю.

Примечание — Поток может использоваться для абстрагирования, например, от точной структуры последовательности взаимодействий или от непрерывного взаимодействия, включая специальный случай аналогового информационного потока.

7.1.6 Интерфейс сигналов — интерфейс, в котором все взаимодействия являются сигналами.

7.1.7 Интерфейс операций — интерфейс, в котором все взаимодействия являются операциями.

7.1.8 Интерфейс потоков — интерфейс, в котором все взаимодействия являются потоками.

7.1.9 Шаблон вычислительного объекта — шаблон объекта, включающий в себя набор шаблонов вычислительных интерфейсов (которые объект может реализовывать), спецификации поведения и контракта среды.

7.1.10 Шаблон вычислительного интерфейса — шаблон интерфейса для интерфейса либо сигналов, либо потоков, либо операций. Шаблон вычислительного интерфейса включает в себя сигнатуру интерфейса (сигналов, потоков или операций), спецификации поведения и контракт среды.

7.1.11 Сигнатура интерфейса сигналов — сигнатура интерфейса для интерфейса сигналов. Сигнатура интерфейса сигналов включает в себя конечный набор шаблонов действий, по одному для каждого типа сигналов в интерфейсе. Каждый шаблон действия, в свою очередь, включает в себя имя сигнала, количество, имена и типы параметров сигнала и указание причинности (инициирующий или ответный, но не оба одновременно) относительно объекта, реализующего шаблон.

7.1.12 Сигнатура интерфейса операций — сигнатура интерфейса для интерфейса операций. Сигнатура интерфейса операций включает в себя набор сигнатур сообщений и запросов, по одной для каждого типа операций в интерфейсе, и указание причинности (клиент или сервер, но не оба одновременно) для интерфейса в целом относительно объекта, реализующего шаблон.

Каждая сигнатура сообщения является шаблоном действия, содержащего имя вызова, количество, имена и типы его параметров.

Каждая сигнатура запроса включает в себя шаблон действия со следующими элементами:

- имя вызова;
- количество, имена и типы его параметров;
- конечный и не пустой набор шаблонов действий, по одному на каждый возможный тип завершения вызова, каждый из которых содержит имя завершения, количество, имена и типы его параметров.

7.1.13 Сигнатура интерфейса потоков — сигнатура интерфейса для интерфейса потоков. Сигнатура интерфейса потоков включает в себя конечный набор шаблонов действий, по одному для каждого типа потока в интерфейсе. Каждый шаблон действия для потока, в свою очередь, включает в себя имя потока, его информационный тип и указание причинности (производитель или потребитель, но не оба одновременно) относительно объекта, реализующего шаблон.

Примечания

1 Фраза «сигнатура интерфейса, дополнительная к X», где X сам является сигнатурой интерфейса, описывает сигнатуру интерфейса, идентичного X во всех отношениях, кроме причинности, которая является противоположной причинности X.

2 Многие языки определения интерфейсов (ЯОИ) охватывают только шаблоны действий сигнатуры и зависят от контекста, в котором ЯОИ используют для определения применяемой причинности.

7.1.14 Связующий объект — вычислительный объект, обеспечивающий связывание между другими вычислительными объектами.

Примечание — Связующие объекты занимают особое положение (см. 7.2.3).

7.2 Структурирующие правила

Вычислительная спецификация в терминах прозрачности распределения описывает функциональную декомпозицию системы ОРО как:

- конфигурацию вычислительных объектов (включая связующие объекты);
- внутренние действия этих объектов;
- взаимодействия между этими объектами;
- контракты среды для этих объектов и их интерфейсов.

Вычислительная спецификация ограничена правилами вычислительного языка. Последние включают в себя:

- правила взаимодействия (см. 7.2.2), связывания (см. 7.2.3) и типа (см. 7.2.4), которые обеспечивают прозрачность распределения;
- правила шаблона (см. 7.2.5), которые применяют для всех вычислительных объектов;
- правила отказа (см. 7.2.6), которые применяют для всех вычислительных объектов и идентифицируют потенциальные точки отказа при вычислительной деятельности.

Правила переносимости (см. 7.2.7) дают руководство для разработчиков стандартов по переносимости ОРО.

Вычислительная спецификация определяет начальный набор вычислительных объектов и их поведение. Конфигурация будет изменяться по мере того, как вычислительные объекты будут:

- реализовывать последующие вычислительные объекты;
- реализовывать последующие вычислительные интерфейсы;
- осуществлять связующие действия;
- выполнять управляющие функции на связующих объектах;
- удалять вычислительные интерфейсы;
- удалять вычислительные объекты.

7.2.1 Правила наименования

Каждый вид имени, определенный в вычислительном языке, имеет соответствующий контекст, а именно:

- имя сигнала в сигнатуре интерфейса сигналов является идентификатором в контексте этой сигнатуры;
- имя потока в сигнатуре интерфейса потоков является идентификатором в контексте этой сигнатуры;
- имя вызова в сигнатуре интерфейса операций является идентификатором в контексте этой сигнатуры;
- имя завершения в сигнатуре интерфейса операций является идентификатором в контексте шаблона операции, в котором оно появилось;
- имя параметра в шаблоне сигнала является идентификатором в контексте этого шаблона;
- имя параметра в шаблоне вызова в сигнатуре интерфейса операций является идентификатором в контексте этого шаблона;
- имя параметра в шаблоне завершения в сигнатуре интерфейса операций является идентификатором в контексте этого шаблона;
- имя параметра в шаблоне сигнала в сигнатуре интерфейса сигналов является идентификатором в контексте этого шаблона.

Примечание 1 — Таким образом, имена сигналов различны в любой сигнатуре интерфейса сигналов, но сигналы в разных сигнатурах могут иметь одинаковые имена, и т. д.

Идентификатор вычислительного интерфейса является недвусмысленным в пределах своего контекста (т. е. не может быть связан с более чем одним вычислительным интерфейсом в этом контексте). Выбор контекстов для идентификаторов вычислительных интерфейсов является вопросом языка проектирования и, следовательно, находится вне области применения настоящей базовой модели. Таким образом, базовая модель не устанавливает ограничений на области действия контекстов для идентификаторов вычислительных интерфейсов. Следовательно, нельзя надеяться на:

- область действия контекстов наименования для идентификаторов вычислительных интерфейсов (например, какие-либо предположения о них связаны с такими структурами инженерного языка, как узлы или области коммуникации);
- единственность идентификаторов вычислительных интерфейсов (т. е. допустимы синонимы);
- то, что идентификатор вычислительного интерфейса обозначает один и тот же вычислительный интерфейс всюду, где он появляется (т. е. имена не обязательно являются «глобальными»).

Примечание 2 — Конкретная вычислительная нотация может не иметь явных терминов, обозначающих вычислительные идентификаторы; следовательно, в такой нотации идентификаторы вычислительных интерфейсов являются неявными; однако они подчиняются приведенным выше правилам.

7.2.2 Правила взаимодействия

Каждое взаимодействие вычислительного объекта происходит через один из его вычислительных интерфейсов. Вычислительный язык устанавливает ограничения на поведение, допустимое в вычислительном интерфейсе. Взаимодействие в несвязанном интерфейсе отвергается. Правила связывания (см. 7.2.3) устанавливают ограничения на то, как должен связываться интерфейс.

Описывающая взаимодействия часть вычислительного языка поддерживает три модели взаимодействия, каждая из которых имеет соответствующий вид вычислительного интерфейса:

- сигналы и интерфейсы сигналов;
- потоки и интерфейсы потоков;
- операции и интерфейсы операций.

В дополнение к различным видам поддерживаемых интерфейсов модели взаимодействий различаются свойствами относительно отказов. Стороны, участвующие в потоке или операции, могут иметь несогласованные точки зрения на взаимодействие в разные моменты времени, особенно если имел место отказ. В противоположность потокам и операциям, нет понятия частичного отказа сигнала: сигнал одинаково удачен или неудачен для обоих участников взаимодействия.

7.2.2.1 *Правила взаимодействия для сигналов*

Вычислительный объект, предоставляющий интерфейс сигналов данного типа:

- инициирует сигналы, которые имеют инициирующую причинность в сигнатуре интерфейса;
- отвечает на сигналы, которые имеют ответную причинность в сигнатуре интерфейса.

7.2.2.2 *Правила взаимодействия для потоков*

Вычислительный объект, предоставляющий интерфейс потоков:

- инициирует потоки, которые имеют причинность производителя в сигнатуре интерфейса;

- принимает потоки, которые имеют причинность потребителя в сигнатуре интерфейса.

7.2.2.3 *Правила взаимодействия для операций*

Объект-клиент, используя интерфейс операций, вызывает операции, названные в сигнатуре интерфейса. Объект-сервер, предоставляющий интерфейс операций, ожидает какую-либо из операций, названных в сигнатуре интерфейса. В случае запроса сервер отвечает на вызов инициированием одного из завершений, указанного для операции в сигнатуре интерфейса сервера. Клиент ожидает какое-либо из завершений, указанных для операции в сигнатуре интерфейса клиента. Продолжительность операции произвольна, если только другое не требуется контрактами среды, применяемыми к этим объектам и интерфейсу.

Примечание — Если клиент вызывает цепочку запросов, то взаимное соответствие запросов и завершений гарантирует, что сервер будет отвечать на операции в том же самом порядке, в каком их инициировал клиент. Если клиент вызывает цепочку сообщений (или цепочку, содержащую сообщения и запросы), то нет способа гарантировать порядок, в котором сервер будет отвечать на сообщения, если только это не подразумевается контрактами среды, применяемыми к взаимодействию. Нет гарантий порядка ни для запросов, ни для сообщений, которые находятся в разных дочерних деятельностях ранее разделенного действия.

7.2.2.4 *Правила для параметров*

В число параметров для сигналов, вызовов и завершений могут входить идентификаторы для вычислительных интерфейсов и типов сигнатур вычислительных интерфейсов.

Примечание 1 — Возможности использования параметров сигнатуры вычислительного интерфейса делают типы вычислительных сигнатур более упорядоченными. Явное представление типов сигнатур требуется, например, при торге, когда параметры операций импорта и экспорта включают в себя типы сигнатур:

trader.import (T: Type, ...) : (servise: T) - > failed (reason: String) trader.export (T: Type, service: T) : (...) -> failed (reason: String)

Таким образом появляется необходимость динамической проверки подтипа сигнатуры (см. 7.2.5.1).

Формальный параметр, который является идентификатором для вычислительного интерфейса, уточняется типом сигнатуры вычислительного интерфейса. Соответствующий фактический параметр должен указывать интерфейс с этим типом сигнатуры (или с одним из ее подтипов). Фактический параметр может использоваться, если он указывает вычислительный интерфейс с тем же самым типом сигнатуры, что и формальный параметр (или с ее подтипом). После взаимодействия инициатор и ответчик могут ссылаться на идентифицированный интерфейс, хотя, возможно, и с разными его идентификаторами.

Примечание 2 — Это правило препятствует пользователю интерфейса, указанного фактическим параметром, осуществлять дополнительные взаимодействия через интерфейс с типом сигнатуры формального параметра, даже если интерфейс, указанный фактическим параметром, является подтипом интерфейса, связанного с формальным параметром.

7.2.2.5 *Потоки, операции и сигналы*

Потоки и операции могут быть определены в терминах сигналов. Это позволяет использовать интерфейсы сигналов как основу для объяснения многостороннего связывания, сквозных характеристик качества услуг и составного связывания между разными типами интерфейсов (например, связывание интерфейсов потоков и операций).

Определение потоков в терминах сигналов зависит от деталей взаимодействий, абстрагированных в спецификации рассматриваемого интерфейса потоков, и, следовательно, находится вне сферы действия настоящей базовой модели.

Операции могут моделироваться как сигналы путем введения соответствующих интерфейсов сигналов для интерфейсов операций клиента и сервера:

- в интерфейсе сигналов, удовлетворяющем интерфейсу операций клиента, имеются сигналы (подача вызова), соответствующие всем вызовам с теми же самыми параметрами, а в случае интерфейса, содержащего запросы, сигналы (доставка завершения), — всем возможным завершениям с теми же самыми параметрами;

- в интерфейсе сигналов, удовлетворяющем интерфейсу операций сервера, имеются сигналы (доставка вызова), соответствующие всем вызовам с теми же самыми параметрами, а в случае интерфейса, содержащего запросы, сигналы (подача завершения), — всем возможным завершениям с теми же самыми параметрами.

Полученный таким образом набор сигналов эквивалентен набору вызовов и завершений в описываемом интерфейсе операций.

7.2.3 Правила связывания

В настоящей базовой модели связывание определяется через ссылку на связывающие действия. Использование таких действий называется явным связыванием. Имеются два вида связывающих действий: элементарные и составные.

Элементарные связывающие действия непосредственно связывают два вычислительных объекта. Составное связывающее действие может быть выражено в терминах элементарных связывающих действий, связывающих два или несколько вычислительных объектов через связующий объект. Присутствие связующего объекта в вычислительном связывании придает смысл выражению «управление конфигурацией и качеством услуги» (см. 7.2.3.3).

В нотациях, в которых нет терминов для выражения связывающих действий, связывание является неявным. Неявное связывание для интерфейсов, отличных от интерфейса операций сервера, в базовой модели не определяется, так как в этих случаях несамоочевидно, где должна размещаться инициатива связывания относительно последующего взаимодействия. Дополнительная информация может быть предоставлена при явном связывающем действии.

7.2.3.1 *Правила неявного связывания для интерфейсов операций сервера*

Если вызов объектом-клиентом указывает на интерфейс операций сервера, с которым клиент не связан, то требуется неявное связывание. Установление неявного связывания осуществляется по следующей процедуре, если не существует нужного интерфейса операций клиента, связанного с сервером:

- создается интерфейс операций клиента с типом сигнатуры, дополнительным интерфейсу сервера;
- связывается интерфейс операций клиента с интерфейсом операций сервера;
- вызывается объект-сервер через интерфейс операций клиента;
- (факультативно) по завершении операции удаляется интерфейс клиента.

7.2.3.2 *Правила элементарного связывания*

Элементарное связывающее действие позволяет связать интерфейс объекта, инициировавшего действие, с другим интерфейсом (другого или того же самого объекта). Параметрами связывающего действия являются два идентификатора, по одному на каждый участвующий интерфейс. Предусловиями для элементарного связывающего действия являются следующие: оба участвующих интерфейса должны быть одного вида (а именно сигналов, потоков или операций), быть причинно дополнительными, и их типы сигнатуры должны быть дополнительными.

Элементарное связывание или устанавливает связь между двумя рассматриваемыми интерфейсами, или завершается неудачно.

Удаление интерфейса, который был связан с другим интерфейсом с помощью элементарного связывающего действия, удаляет так же и связь.

7.2.3.3 *Правила составного связывания*

Составные связывающие действия позволяют связать несколько интерфейсов, используя связующий объект для обеспечения связи. За исключением сказанного в данном разделе, во всех других отношениях связующий объект является обычным вычислительным объектом. В шаблоне связующего объекта спецификация поведения выражается в терминах набора параметров формальных ролей, каждый из которых связан с шаблоном интерфейса.

Составные связывающие действия имеют в качестве параметров шаблон связующего объекта и набор интерфейсов, которые должны быть связаны для взаимодействия.

Предусловиями для составного связывания для каждой формальной роли в шаблоне связующего объекта являются:

- соответствующий параметр интерфейса должен быть того же самого вида (а именно сигналов, потоков или операций), что и шаблон интерфейса, ассоциированный с формальной ролью в шаблоне связующего объекта;
- соответствующий параметр интерфейса должен быть причинно дополнительным шаблону интерфейса, ассоциированному с формальной ролью в шаблоне связующего объекта;
- соответствующий параметр интерфейса должен быть подтипом типа сигнатуры шаблона интерфейса, ассоциированный с формальной ролью в шаблоне связующего объекта.

Составное связывающее действие включает в себя следующие шаги:

- связующий объект реализуется по шаблону связующего объекта;
- реализуется каждый шаблон интерфейса в связующем объекте, ассоциированный с параметром формальной роли в шаблоне связующего объекта;

- связующий объект использует элементарные связывающие действия для связи каждого такового интерфейса с интерфейсом, указанным в соответствующем фактическом параметре;

- реализуется набор управляющих интерфейсов, а их идентификаторы возвращаются в качестве результата связывающего действия (т. е. становятся частью состояния объекта, осуществившего действие; этот объект может передать идентификатор при взаимодействии с другими вычислительными объектами).

Управляющие интерфейсы связующего объекта обеспечивают некоторые или все из следующих функций:

- мониторинг использования связи;
- мониторинг изменений связи;
- авторизацию изменений связи;
- изменение членства в этой связи;
- изменение экземпляров коммуникаций, доступных для связи;
- изменение качества услуги связи;
- удаление всей связи.

Влияние удаления связи на связующий объект определяется его поведением.

7.2.4 Правила для типов

В настоящей базовой модели установлены правила для типов сигнатур вычислительных интерфейсов. Правила образования подтипов сигнатур определяют минимальные требования для того, чтобы один интерфейс мог заменить другой. Правила основаны на семантике взаимодействий вычислительных интерфейсов (а именно сигналов, потоков и операций). Они достаточны, чтобы гарантировать, что подставляемый интерфейс может согласованно интерпретировать структуру любого взаимодействия.

Правила образования подтипов сигнатур для интерфейсов с альтернативными семантиками взаимодействия могут быть определены в терминах сигналов; такие определения могут быть введены в стандартах ОРО.

7.2.4.1 *Правила образования подтипов сигнатур для интерфейсов сигналов*

Определение подтипов сигнатур интерфейсов сигналов приведено в приложении А. Для тех типов интерфейсов сигналов, которые не определены рекурсивно, сводка правил приведена ниже.

Тип сигнатуры интерфейса сигналов X является подтипом сигнатуры интерфейса сигналов Y, если выполнены следующие условия:

- для каждой сигнатуры инициирующего сигнала в Y имеется соответствующая сигнатура инициирующего сигнала в X с тем же самым именем, с тем же самым количеством и именами параметров, и тип каждого параметра в X является подтипом соответствующего типа параметра в Y;

- для каждой сигнатуры ответного сигнала в X имеется соответствующая сигнатура ответного сигнала в Y с тем же самым именем, с тем же самым количеством и именами параметров, и тип каждого параметра в Y является подтипом соответствующего типа параметра в X.

7.2.4.2 Правила образования подтипов сигнатур для интерфейсов потоков

Зависят от деталей взаимодействий, абстрагированных в определении рассматриваемых интерфейсов потоков. В частности, эти детали будут вносить ясность в вопрос, будут или нет правила образования подтипов допускать неполное соответствие между наборами потоков в двух интерфейсах. Следовательно, полные правила образования подтипов для сигнатур потоков находятся вне сферы действия настоящего стандарта. Ограничения на образование подтипов сигнатур потоков приведены в приложении А. Для типов интерфейсов потоков, которые не определены рекурсивно, сводка ограничений приведена ниже.

Интерфейс потоков X является подтипом сигнатуры интерфейса потоков Y, если для всех потоков с идентичными именами выполнены следующие условия:

- если причинность — производитель, то информационный тип в X является подтипом информационного типа в Y;

- если причинность — потребитель, то информационный тип в Y является подтипом информационного типа в X.

7.2.4.3 *Правила образования подтипов сигнатур для интерфейсов операций*

Определение подтипов сигнатур интерфейсов операций приведено в приложении А. Для типов интерфейсов, которые не определены рекурсивно, сводка правил приведена ниже.

Интерфейс операций X является подтипом сигнатуры интерфейса операций Y, если выполнены следующие условия:

- для каждого запроса в Y имеется сигнатура запроса в X (соответствующая сигнатура в X), которая определяет запрос с тем же самым именем;
- для каждой сигнатуры запроса в Y соответствующая сигнатура в X имеет то же самое число параметров с теми же самыми именами;
- для каждой сигнатуры запроса в Y тип каждого параметра является подтипом соответствующего типа параметра соответствующей сигнатуры запроса в X;
- набор имен завершений сигнатуры запроса в Y содержит набор имен завершений соответствующей сигнатуры запроса в X;
- для каждой сигнатуры запроса в Y данное завершение в соответствующей сигнатуре запроса в X имеет то же самое число результирующих параметров с теми же самыми именами, что и одноименное завершение в сигнатуре запроса в Y;
- для каждой сигнатуры запроса в Y каждый тип результата, связанный с данным завершением в соответствующей сигнатуре запроса в X, является подтипом типа результата (с тем же именем) в одноименном завершении в Y;
- для каждой сигнатуры сообщения в Y имеется сигнатура сообщения в X (соответствующая сигнатура в X), которая определяет сообщение с тем же самым именем;
- для каждой сигнатуры сообщения в Y соответствующая сигнатура сообщения в X имеет то же самое число параметров с теми же самыми именами;
- для каждой сигнатуры сообщения в Y тип каждого параметра является подтипом типа соответствующего параметра в соответствующей сигнатуре сообщения в X.

7.2.5 Правила для шаблонов

7.2.5.1 *Правила для шаблонов вычислительных объектов*

Вычислительный объект (включая частный случай связующего объекта) может:
- инициировать или отвечать на сигналы;
- создавать или потреблять потоки;
- инициировать вызовы операций;
- отвечать на вызовы операций;
- инициировать завершения операций;
- отвечать на завершения операций;
- реализовывать шаблоны интерфейсов;
- реализовывать шаблоны объектов;
- связывать интерфейсы;
- предоставлять доступ и изменять свое состояние;
- удалять один или несколько из своих интерфейсов;
- удалять самого себя;
- порождать, разветвлять и объединять деятельности;
- получать идентификатор вычислительного интерфейса для экземпляра функции торга;
- проверять, является ли сигнатура вычислительного интерфейса подтипом другой сигнатуры.

Любое из этих действий может привести к отказу.

7.2.5.2 *Реализация вычислительного интерфейса*

Устанавливает один или несколько идентификаторов для нового вычислительного интерфейса в объекте, осуществляющем реализацию.

7.2.5.3 *Реализация шаблона вычислительного объекта*

Выражение поведения в шаблоне вычислительного объекта включает в себя описание поведения, которое должно происходить при реализации шаблона (реализующего поведения). Спецификация контракта среды описывает контракт, который должен быть установлен между реализуемым объектом и его средой при реализации шаблона. Когда реализующее поведение включает в себя реализации интерфейсов, то реализация устанавливает идентификаторы для этих интерфейсов в объекте, который инициировал реализацию.

7.2.6 Правила для отказов

Видимые объекту режимы отказа определяются спецификациями его поведения и контракта среды.

Любые вычислительные действия в 7.2.5.1 могут привести к отказу и этот отказ может наблюдаться объектом, осуществляющим действие. Взаимодействие может быть разорвано из-за отказа участвующих объектов, или из-за их связывания, или из-за того и другого сразу. В случае сигналов отказ идентичен (и видим) для всех участников взаимодействия. В случае потоков и операций отказ

не обязательно должен происходить для всех участников, а может случаться в разное время с разными параметрами для каждого отказавшего участника.

Примечание — Примерами отказов взаимодействия являются отказы безопасности, коммуникации и ресурса.

Для операций отказ вычислений сервера для ответа на вызов или для инициирования завершения может быть наблюдаем участвующим вычислительным объектом-клиентом.

Реализация шаблона объекта или шаблона вычислительного интерфейса приводит к отказу, если не может быть удовлетворен контракт среды. Связывающее действие может привести к отказу, если не может быть удовлетворен какой-либо из контрактов среды в связываемых интерфейсах.

7.2.7 Правила переносимости

Стандарты по переносимости в системах ОРО специфицируют шаблоны действий, описанных в 7.2.5.1. Спецификация таких шаблонов зависит от языка проектирования и, следовательно, находится вне сферы действия данной базовой модели. В дополнение к синтаксическим понятиям стандарты по переносимости должны охватывать специфические семантические вопросы, включая:

- правила композиции для шаблонов действий, включая шаблоны для разветвляющих и объединяющих действий, для того, чтобы допускать параллельность и синхронизацию;
- термины, применяемые в спецификациях шаблонов объектов и интерфейсов, и правила их композиции;
- упорядочение и гарантии доставки для сообщений.

Стандарт по переносимости может представлять допустимые действия непосредственно (например, как библиотечные функции) или косвенно через синтаксические структуры. Могут существовать альтернативные стандарты по переносимости как в терминах стиля (например, модели обработки, основанной на событиях, и модели, основанной на связках), так и содержания (например, по числу поддерживаемых вычислительных действий). В данной модели идентифицированы два вида таких стандартов.

Базовым по переносимости является такой стандарт, который содержит по крайней мере:

- опросы;
- неявное связывание;
- реализацию вычислительного объекта;
- реализацию вычислительного интерфейса;
- доступ и изменение состояния;
- поддержку связок с порождающими, разветвляющими и объединяющими действиями;
- получение идентификатора для вычислительного интерфейса, при котором обеспечивается функция торга (допускающая последующее связывание и использование функций);
- проверку подтипа сигнатуры интерфейса.

Расширенным по переносимости является такой стандарт, который содержит все действия, описанные в 7.2.5.1.

7.3 Соответствие и опорные точки

В вычислительном языке существуют опорные точки для любого интерфейса объекта. Каждая опорная точка может стать программируемой, воспринимаемой точкой соответствия, точкой соответствия взаимодействия или обмена в зависимости от требований, установленных при назначении опорной точки в качестве точки соответствия конкретным стандартом или спецификацией системы.

В вычислительном языке эти требования задаются в терминах шаблонов интерфейсов и объектов, которые определяют интерфейс соответствующего объекта.

Реализатор, заявляющий о соответствии вычислительной спецификации, должен перечислить инженерные опорные точки, которые соответствуют требуемым вычислительным опорным точкам, и констатировать, какие в них применены прозрачности и инженерные структуры. Тем самым идентифицированные опорные точки становятся точками соответствия. Набор взаимодействий в этих опорных точках может быть интерпретирован в терминах вычислительного языка для определения того, что вычислительная спецификация не нарушена.

Соответствие объекта в программируемой точке соответствия может быть проверено в терминах стандартизированного языка спецификаций интерфейса и связывающего языка, который удовлетворяет правилам переносимости. Соответствие объекта в точке соответствия взаимодействия может быть проверено в терминах видимых взаимодействий в коммуникационных протоколах.

8 Инженерный язык

Инженерный язык охватывает понятия, правила и структуры для спецификации системы ОРО с инженерной точки зрения.

Инженерная спецификация определяет методы и функции, требующиеся для обеспечения распределенного взаимодействия между объектами в системе ОРО.

8.1 Понятия

Инженерный язык содержит понятия ГОСТ Р ИСО/МЭК 10746-2 и настоящего стандарта, подчиняющиеся правилам 8.2.

8.1.1 Базовый инженерный объект — инженерный объект, который требует обеспечения распределенной инфраструктуры.

8.1.2 Кластер — конфигурация базовых инженерных объектов, образующих единое целое для задач деактивации, создания контрольных точек, реактивации, восстановления и миграции.

Примечание — Примером кластера является сегмент виртуальной памяти, содержащий объекты.

8.1.3 Менеджер кластера — инженерный объект, который управляет базовыми инженерными объектами в кластере.

8.1.4 Капсула — конфигурация инженерных объектов, образующих единое целое для задач инкапсуляции обработки и хранения.

Примечание — Примером капсулы является виртуальная машина (например, процесс).

8.1.5 Менеджер капсулы — инженерный объект, который управляет инженерными объектами в капсуле.

8.1.6 Ядро — инженерный объект, координирующий функции обработки, хранения и коммуникации для использования другими инженерными объектами в пределах узла, к которому он относится.

Примечание — Примером ядра является операционная система.

8.1.7 Узел — конфигурация инженерных объектов, образующих единое целое для задач локализации в пространстве, которая включает в себя набор функций обработки, хранения и коммуникации.

Примечания

1 Примером узла является компьютер с программным обеспечением (операционной системой и приложениями).

2 Узел может иметь внутреннюю структуру, которая не рассматривается в инженерной спецификации. Например, узел может быть с параллельными компьютерами под управлением единой операционной системы.

8.1.8 Канал — конфигурация заглушек, связников, протокольных объектов и пересечений, обеспечивающая связывание набора интерфейсов с базовыми инженерными объектами, через которые может осуществляться взаимодействие.

Примечание — Связывания, требующие каналов, в инженерном языке называются распределенными связываниями; связывания между инженерными объектами, не требующие каналов (например, между инженерными объектами в одном кластере), называются локальными связываниями.

8.1.9 Заглушка — инженерный объект в канале, который интерпретирует взаимодействия, переносимые каналом, и осуществляет необходимые преобразования или мониторинг на основе этой интерпретации.

Примечание — Например, заглушка может осуществлять погружение параметров в коммуникационные буфера или записывать деятельности для процессов аудита.

8.1.10 Связник — инженерный объект в канале, который обеспечивает распределенное связывание между взаимодействующими базовыми инженерными объектами.

8.1.11 <X> пересечение — инженерный объект в канале, расположенный на границе между <x> областями. <X> пересечение осуществляет:

- проверки для осуществления или мониторинга политики допустимых взаимодействий между базовыми инженерными объектами в разных областях;
- преобразования для маскировки различий в интерпретации данных базовыми инженерными объектами в разных областях.

8.1.12 Протокольный объект — инженерный объект в канале, который осуществляет коммуникацию с другими протокольными объектами в том же самом канале для достижения взаимодействия между базовыми инженерными объектами (возможно в разных кластерах, капсулах или узлах).

8.1.13 Коммуникационная область — набор протокольных объектов, способных взаимодействовать.

8.1.14 Коммуникационный интерфейс — интерфейс протокольного объекта, который может быть связан с интерфейсом пересечения или другого протокольного объекта в опорной точке взаимодействия.

8.1.15 Оконечный идентификатор связывания — идентификатор (в контексте наименований капсулы), используемый базовым инженерным объектом для выбора в целях взаимодействия одного из связываний, в котором он участвует.

Примечания

1 Примером оконечного идентификатора связывания является адрес в памяти (структуры данных, представляющей инженерный интерфейс).

2 Может использоваться одна и та же форма оконечного идентификатора связывания для локальных и распределенных связываний.

8.1.16 Указатель инженерного интерфейса — идентификатор (в контексте области управления указателями инженерных интерфейсов) интерфейса инженерного объекта, который доступен для распределенного связывания.

Примечание — Указатель инженерного интерфейса необходим для установления распределенного связывания и отличается от оконечного идентификатора связывания, используемого базовым инженерным объектом в целях взаимодействия.

8.1.17 Область управления указателями инженерных интерфейсов — множество узлов, образующих область наименования с целью присвоения указателей инженерных интерфейсов.

8.1.18 Политика управления указателями инженерных интерфейсов — набор разрешений и запрещений, действующих для федерации областей управления указателями инженерных интерфейсов.

8.1.19 Шаблон кластера — шаблон объекта для конфигурации объектов и любой деятельности, требуемой для реализации этих объектов и установления начальных связываний.

8.1.20 Контрольная точка — шаблон объекта, полученный из состояния и структуры инженерного объекта, который может быть использован для реализации другого инженерного объекта, согласующегося с состоянием исходного объекта в момент создания контрольной точки.

8.1.21 Создание контрольной точки — контрольные точки могут быть созданы, только когда участвующий инженерный объект удовлетворяет предусловию, установленному политикой создания контрольных точке.

8.1.22 Кластерная контрольная точка — шаблон кластера, содержащий контрольные точки базовых инженерных объектов в кластере.

8.1.23 Деактивация — создание контрольной точки кластера с последующим удалением самого кластера.

8.1.24 Клонирование — реализация кластера из кластерной контрольной точки.

8.1.25 Восстановление — клонирование кластера после отказа или удаления кластера.

8.1.26 Реактивация — клонирование кластера после его деактивации.

8.1.27 Миграция — перемещение кластера в другую капсулу.

8.2 Структурирующие правила

Инженерная спецификация определяет инфраструктуру, необходимую для обеспечения функционального распределения системы ОРО, идентифицируя:

- функции ОРО, требуемые для управления физическим распределением, коммуникацией, обработкой и хранением;
- роли различных инженерных объектов, обеспечивающих функции ОРО (например, ядро).

Инженерная спецификация выражается в терминах:

- конфигурации инженерных объектов, структурированных на кластеры, капсулы и узлы;
- деятельности, осуществляемой в этих инженерных объектах;
- взаимодействий этих инженерных объектов.

Инженерная спецификация ограничена правилами инженерного языка. К ним относятся:

- правила для каналов (см. 8.2.1), указателей интерфейсов (см. 8.2.2), распределенного связы-

вания (см. 8.2.3) и перемещения (см. 8.2.4) с целью обеспечения прозрачности распределения взаимодействия среди инженерных объектов;

- правила для кластеров (см. 8.2.5), капсул (см. 8.2.6) и узлов (см. 8.2.7), управляющие конфигурацией инженерных объектов;
- правила для отказов (см. 8.2.9).

8.2.1 Правила для каналов

Каналы обеспечивают прозрачность распределения взаимодействия среди инженерных объектов. Обеспечение включает в себя:

- выполнение операций между объектом клиента и объектом сервера;
- группирование объектов, играющих несколько ролей для других групп объектов;
- взаимодействие потоков, в которых участвуют несколько объектов-производителей и объектов-потребителей.

Взаимодействие между инженерными объектами вызывает передачу одного или всего из следующего:

- указатели инженерных интерфейсов;
- шаблоны кластеров;
- данные.

Канал является конфигурацией заглушек, связников, протокольных объектов и пересечений, объединяющей набор инженерных объектов. Конфигурация является ациклическим графом с заглушками в самых внешних вершинах, как показано на рисунках 2 и 3. Каждый путь по графу между заглушками объектов состоит (последовательно) из:

- связника, протокольного объекта, протокольного объекта и связника, или
- связника, протокольного объекта, пересечения, протокольного объекта и связника.

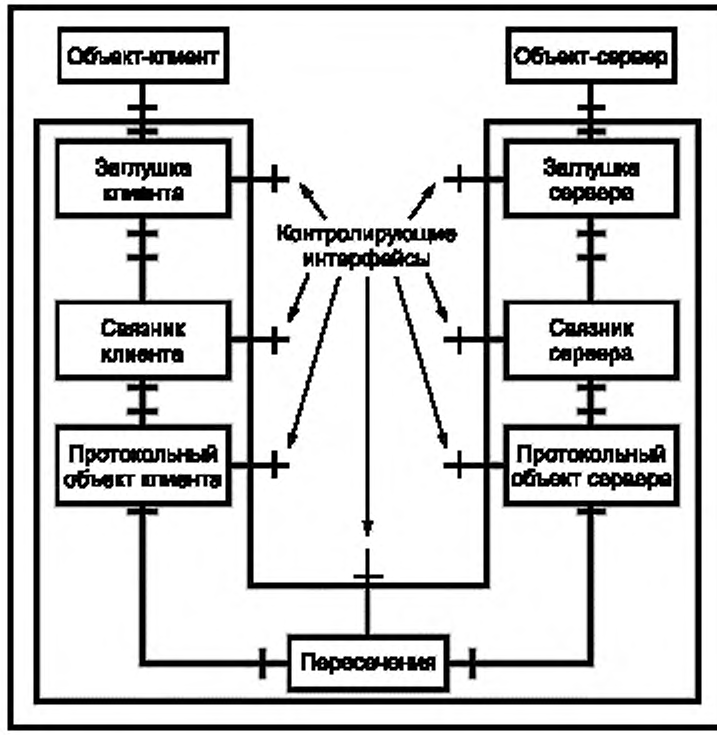

Рисунок 2 — Пример базового канала клиент/сервер

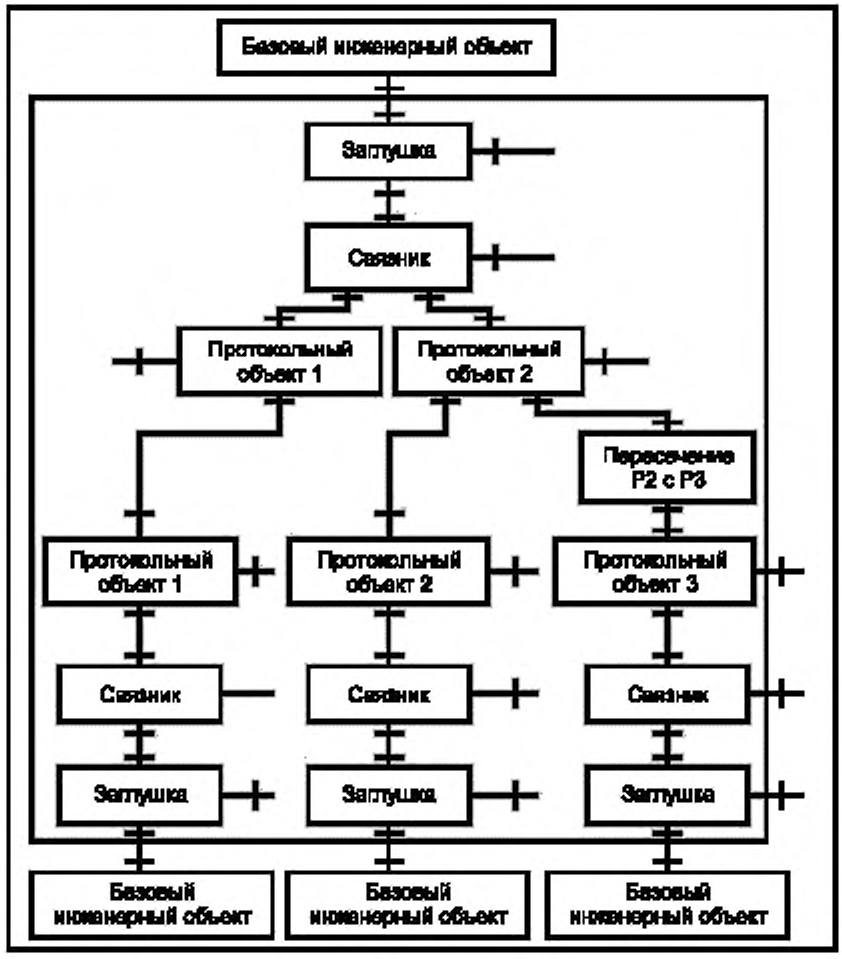

Рисунок 3 — Пример канала с несколькими оконечными точками

Поведение канала относительно конфигурации канала или управления качеством услуги контролируется через управляющие интерфейсы заглушек, связников, протокольных объектов и пересечений. Эти управляющие интерфейсы являются факультативными.

Примечания

1 Заглушки, связники, протокольные объекты и пересечения в канале могут иметь (локальные или распределенные) связывания с другими инженерными объектами вне канала, обеспечивающими, например, функции перемещения или координации.

2 В зависимости от типа задействованных преобразований пересечение может быть разложено на заглушки, связники, протокольные и базовые инженерные объекты, отображающие структуру канала.

Объекты в канале сами могут быть базовыми инженерными объектами, поддерживаемыми каналами.

8.2.1.1 *Заглушки*

Базовые инженерные объекты, взаимодействующие через каналы, локально связаны с заглушками. В канале заглушки обеспечивают преобразование данных, передаваемых при взаимодействии. Заглушки могут использовать управление и сохранение записей (например, для безопасности и учета). Если нужно, заглушки могут взаимодействовать с инженерными объектами вне канала (например, для функций безопасности). Заглушка в канале имеет интерфейсы для поддерживаемого

ею базового инженерного объекта и для взаимодействия со связником. Она может иметь и управляющий интерфейс.

Когда взаимодействующие заглушки используют разные синтаксисы передачи, на пути между ними должно быть пересечение, преобразующее данные из одного синтаксиса в другой.

Заглушка может иметь одну из следующих форм:

а) специфичную для экземпляра интерфейса базового инженерного объекта, с которым она связана;

б) специфичную для типа интерфейса базового инженерного объекта, с которым она связана (следовательно, заглушка может совместно использоваться несколькими каналами одного и того же типа);

в) родовую (т. е. не специфичную для конкретного типа интерфейса); такие заглушки могут использоваться несколькими каналами разных типов.

Примечания

1 В случае а) взаимодействия между инженерным объектом и заглушкой переносят только данные взаимодействия (например, в случае вызова — имена операций и параметры). Заглушка действует как локальный уполномоченный для других базовых инженерных объектов, привязанных к этому каналу.

2 В случае б) взаимодействия должны дополнительно включать идентификатор канала, который должен использоваться.

3 В случае в) взаимодействия должны дополнительно включать идентификатор и тип канала, который должен использоваться, чтобы позволить заглушке гарантировать, что данные взаимодействия совместимы с типом канала.

8.2.1.2 *Связники*

Связники в канале управляют сквозной целостностью этого канала. Когда требуется, связники обеспечивают прозрачность перемещения, осуществляя мониторинг коммуникационных отказов и вновь устанавливая разорванные распределенные связывания. Связники в канале могут взаимодействовать с инженерными объектами вне канала для получения дополнительных данных, необходимых для осуществления их функций (например, для получения сведений о размещении данных). Связник в канале имеет по меньшей мере один интерфейс для взаимодействия с заглушкой и один или несколько интерфейсов для взаимодействия с протокольными объектами. Связник может иметь управляющий интерфейс. Управляющий интерфейс, если он есть, позволяет изменять конфигурацию канала и уничтожать его.

8.2.1.3 *Протокольные объекты*

Обеспечивают коммуникационные функции. Они могут взаимодействовать с инженерными объектами вне канала (например, с функциями справочника) для получения необходимой информации. Протокольный объект имеет интерфейс для взаимодействия со связником и, по крайней мере, один коммуникационный интерфейс для взаимодействия с другими протокольными объектами (если нужно — через пересечение). Протокольный объект может иметь управляющий интерфейс. Когда протокольные объекты в канале имеют разные типы, им требуется пересечение, обеспечивающее преобразование протоколов. Все протокольные объекты в коммуникационной области могут взаимосвязываться непосредственно, используя возможности коммуникационной области (которые находятся вне сферы действия настоящей базовой модели).

При любом заданном положении во времени протокольный объект идентифицируется своим положением в пространстве, но разные протокольные объекты могут занимать одно и то же положение в пространстве в разные моменты времени (т. е. сетевой адрес может использоваться повторно). Когда протокольные объекты находятся в канале одного и того же типа, но в разделенных коммуникационных областях, возможны конфликты наименований (например, имена коммуникационных интерфейсов могут быть двусмысленными). В таких случаях требуется пересечение для преобразования передаваемых имен в процессе установления и поддержания целостности канала.

8.2.1.4 *Пересечения*

<X> пересечение в канале устанавливает границу между <x> областями и обеспечивает проверки и преобразования во взаимодействиях, пересекающих границы <x> областей. В зависимости от пересекаемой границы пересечениям требуется разная информация для выполнения своих функций. Некоторым пересечениям требуется знать типы сигнатур интерфейсов базовых инженерных объектов, связанных с каналом, в котором расположено пересечение, чтобы они могли интерпретировать взаимодействия, обеспечиваемые каналом. Пересечение имеет, по крайней мере, два коммуникационных интерфейса. Оно может иметь управляющий интерфейс.

8.2.2 Правила для указателей интерфейсов

Для целей распределенного связывания инженерные интерфейсы локализуются в пространстве и времени с помощью указателей инженерных интерфейсов. Указатели инженерных интерфейсов определены относительно области управления указателями инженерных интерфейсов, которая устанавливает политику для содержимого, размещения, отслеживания и допустимости инженерных интерфейсов в пределах этой области. Область управления указателями инженерных интерфейсов может быть федерацией, если политики управления указателями инженерных интерфейсов ее составляющих не находятся в противоречии друг с другом.

Указатель инженерного интерфейса содержит информацию, которая позволяет устанавливать связывание с интерфейсами инженерных объектов. Эта информация позволяет объектам ядра создавать каналы, а связникам в каналах — обеспечивать распределенное связывание между взаимодействующими инженерными объектами. Информация в указателе инженерного интерфейса может иметь вид:

- данных;
- идентификаторов интерфейсов, предоставляющих доступ к таким данным;
- комбинации данных и идентификаторов.

Данные, необходимые для связывания, могут включать в себя любой или все из следующих элементов:

- тип указываемого интерфейса;
- шаблон канала, описывающий пересечения, протокольные объекты, связники и заглушки, которые могут быть выбраны при конфигурировании канала для обеспечения распределенного связывания;
- положение в пространстве и времени (например, сетевые адреса) коммуникационных интерфейсов, в которых должен быть инициирован процесс связывания;
- информацию, позволяющую выявлять и восстанавливать распределенные связывания, разрушенные перемещением инженерного объекта.

Область управления указателями интерфейсов может быть разделена на подобласти. В этом случае указатели интерфейсов в области организованы как множество альтернативных наборов информации, по одному на каждую подобласть, в которой допустимо связывание.

Примечания

1 Если ядро, поддерживающее инженерный интерфейс, обеспечивает различные протоколы, процессы связывания и синтаксисы передачи, то указатель инженерного интерфейса должен идентифицировать допустимые комбинации, которые могут быть выбраны в любом конкретном распределенном связывании; разные связывания могут иметь разные выборы.

2 Данная базовая модель не предписывает метод, которым шаблон канала и положение в пространстве и времени соответствующего интерфейса получаются из указателя инженерного интерфейса.

Указатели инженерных интерфейсов распределяются ядрами через интерфейсы, обеспечивающие функции управления узлом (см. 12.1.3). Они отслеживаются функцией отслеживания указателей инженерных интерфейсов (см. 13.9) с целью выявления инженерных интерфейсов, на которые нет указаний. Политика для повторного связывания интерфейсов тех инженерных объектов, которые были перемещены, и обновление соответствующих указателей инженерных интерфейсов записываются функцией перемещения (см. 14.3). Эти три функции (управление узлом, отслеживание указателей инженерных интерфейсов и перемещение) могут быть скоординированы функцией организации совместно используемой информации (см. 14.2).

Указатели инженерных интерфейсов являются недвусмысленными в контексте наименования области управления указателями инженерных интерфейсов. Для достижения недвусмысленности узлы в области управления указателями инженерных интерфейсов должны скоординировано распределять указатели инженерных интерфейсов. Инженерные интерфейсы должны распределяться таким образом, чтобы предотвратить ссылки указателей инженерных интерфейсов на неправильные интерфейсы даже при наличии отказов и перемещений интерфейсов. В самом крайнем случае указатель интерфейса может ссылаться на несуществующий интерфейс (например, непосредственно после отказа инженерного объекта, обеспечивающего интерфейс).

Примечание 3 — В системах ОРО, в которых большинство интерфейсов не изменяют положения, управление указателями интерфейсов может быть оптимизировано: ядро может распределять указатели инженерных интерфейсов автономно; тип канала и идентификатор коммуникационного интерфейса, связанного с данным интерфейсом, могут быть сохранены и переданы в указатель инженерного интерфейса; функция перемещения может быть использована для проверки достоверности и обновления указателей тех инженерных интерфейсов, которые были перемещены.

Перед созданием указателя инженерного интерфейса ядро строит шаблон канала, определяющий конфигурацию заглушки, связника и протокольных объектов, подходящую для обеспечения взаимодействий в интерфейсе. Кроме того, ядро устанавливает локальную структуру, достаточную для того, чтобы сделать возможным связывание с интерфейсом, и связывает шаблон и локальную структуру с коммуникационным интерфейсом. Инженерный интерфейс делает эту информацию доступной.

Пересечение, которое устанавливает границу между областями управления указателями инженерных интерфейсов, обеспечивает отображение между указателями инженерных интерфейсов в этих областях. Когда указатель инженерного интерфейса или шаблон кластера, содержащий указатели инженерных интерфейсов, пересекает границу области указателей инженерных интерфейсов, последние должны быть преобразованы так, чтобы быть допустимыми в новой области.

Обмен указателями инженерных интерфейсов между областями управления указателями инженерных интерфейсов возможен только тогда, когда определена процедура отображения указателей, предотвращающая их двусмысленность.

8.2.3 Правила для распределенного связывания

Установление канала требует создания соответствующих заглушек, связников, протокольных объектов и пересечений. Установление канала может быть инициировано любым инженерным объектом. Оно обеспечивается каждым ядром как функция его интерфейса управления узлом. Распределенное связывание приводит к взаимодействию с ядрами узлов, с интерфейсами которых нужно связаться. Установление канала параметризовано шаблоном канала и набором указателей интерфейсов, каждому из которых присвоена конкретная роль в шаблоне канала. Шаблон канала должен быть совместим с типами каналов, названными указателями инженерных интерфейсов для тех интерфейсов, с которыми должна быть установлена связь. Для каждого объекта, который должен быть связан, ядро создает в своем узле конфигурацию заглушек, связников и протокольных объектов для обеспечения интерфейсов тех объектов, которые должны быть связаны. Эта конфигурация включает в себя и управляющие интерфейсы. Протокольные объекты, поддерживающие канал, соединяются (возможно, через пересечения) в своих коммуникационных интерфейсах. Выбор и конфигурация заглушек, связников, протокольных объектов и пересечений определены шаблоном и типами каналов задействованных указателей интерфейсов. Каждому базовому инженерному объекту, связанному каналом, присвоен идентификатор оконечной точки связывания для каждого интерфейса, который объект имеет в этом канале. Базовые инженерные объекты используют идентификаторы оконечных точек связывания для назначения интерфейса, через который осуществляется распределенное взаимодействие.

Примечания

1 Каналы могут быть установлены любым инженерным объектом, независимо от того, имеет ли объект интерфейс, который должен быть связан каналом.

2 Базовый инженерный объект, инициирующий распределенное связывание, требует набор указателей интерфейсов. Они могут быть получены любым из следующих способов:

а) при инициализации объекта;

б) через взаимодействие инициирующего объекта с ядром, как часть реализации интерфейсов инициирующего объекта;

в) через некоторую цепочку взаимодействий с другими рассматриваемыми объектами (например, при передаче параметров или при торге).

3 Шаблон канала может содержать альтернативные конфигурации, которые должны применяться при выбранных обстоятельствах. Например, если коммуникационные линии ненадежны, могут потребоваться кодирующие заглушки.

8.2.4 Правила перемещения

Инженерные объекты могут быть перемещены в результате:

- реактивации и деактивации;
- создания контрольной точки и восстановления;
- миграции;
- функций управления областью коммуникации (например, изменение идентификатора коммуникационного интерфейса).

Примечания

1 Идентификатор коммуникационного интерфейса может измениться в результате изменений сетевого адреса узла.

2 При переустановке каналов могут использоваться заглушки, связники и протокольные объекты,

отличные от используемых до перемещения. Таким образом, идентификатора коммуникационного интерфейса может оказаться недостаточно для идентификации интерфейса инженерного объекта.

Перемещение может привести к отказу канала и сделать недопустимыми указатели инженерных интерфейсов. Отказавшие каналы могут быть соединены повторно, если деятельность, изменившая положение интерфейса инженерного объекта, известила соответствующую функцию перемещения (см. 14.3).

Связники в канале детектируют, когда перемещение разрушило канал. Либо связники сотрудничают для исправления отображения между указателями инженерных интерфейсов и структуры канала (т. е. требуется прозрачность перемещения — см. 16.6), либо канал отказывает. Когда требуется прозрачность перемещения, информация, доступная через указатель инженерного интерфейса, позволяет связникам использовать функцию перемещения для определения новых положений задействованных базовых инженерных объектов.

8.2.5 Правила для кластеров

Кластер содержит набор базовых инженерных объектов, связанных с менеджером кластера. Каждый член кластера может иметь интерфейс, поддерживающий функцию управления объектом. Каждый такой интерфейс управления объектом связан с менеджером кластера. Базовый инженерный объект в кластере всегда связан с его ядром через интерфейс, обеспечивающий функцию управления узлом, и с менеджером кластера. Кроме того, базовый инженерный объект в кластере может быть связан с другими базовыми инженерными объектами в том же или в других кластерах. Каждый менеджер кластера в капсуле связан с менеджером капсулы. Эта структура показана на рисунке 4.

Кластер всегда содержится в единственной капсуле. Кластер отвечает за свою собственную безопасность, но ему могут помогать функции безопасности. Любая функция безопасности может либо обеспечиваться объектом в той же самой капсуле, что и кластер, либо быть доступной через взаимодействия безопасности, если она находится вне капсулы. Инженерные объекты в одном и том же кластере могут взаимодействовать, используя локальное связывание в пределах кластера или распределенное связывание, обеспечиваемое каналом. Инженерные объекты в разных кластерах взаимодействуют, используя распределенные связывания, обеспечиваемые каналами.

Примечания

1 Взаимодействия в воспринимаемых опорных точках или опорных точках обмена не устранимы.

2 Хотя нет необходимости в канале для обеспечения локального связывания между объектами в одном и том же кластере, идентификатор, используемый для вызовов, — того же самого вида, что и используемый для инженерных объектов в разных кластерах, и также называется идентификатором оконечной точки связывания.

Реализация кластера (включая клонирование как специальный случай) осуществляется менеджером капсулы.

Если шаблон является кластерной контрольной точкой, то реализация (т. е. клонирование) позволяет действовать новому кластеру в качестве замены исходного кластера, из которого был получен шаблон. Когда требуется (по соображениям прозрачности распределения), процесс клонирования включает в себя переустановление всех распределенных связываний, которые имел исходный кластер.

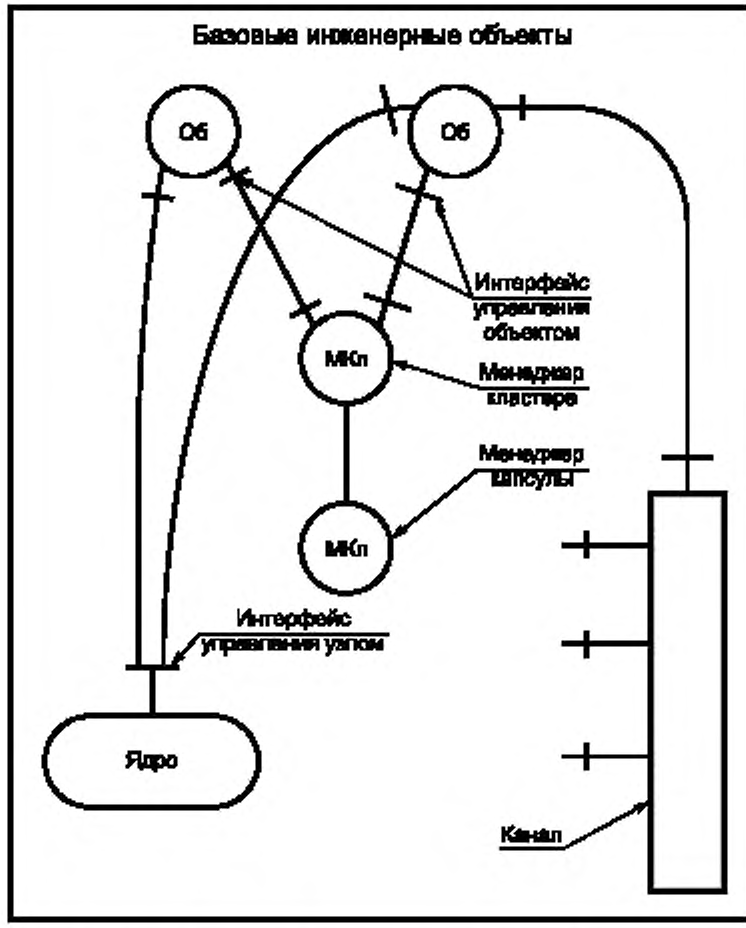

Рисунок 4 — Пример структуры, поддерживающий базовый инженерный объект

Кластер имеет связанного с ним менеджера кластера. Менеджер кластера обеспечивает функцию управления кластером. Менеджер кластера осуществляет политику управления для инженерных объектов в этом кластере. Политика управления кластером может привести менеджера кластера к взаимодействию с другими функциями ОРО для завершения деятельностей управления кластером.

Менеджер кластера помогает своему менеджеру капсулы в управлении указателями инженерных интерфейсов объектов. Это может потребовать доступа к функции отслеживания указателей инженерных интерфейсов.

8.2.6 Правила для капсулы

Капсула состоит из:

- кластеров;
- менеджеров кластеров, по одному на каждый кластер в капсуле;
- менеджера капсулы, с которым связан каждый менеджер кластера в капсуле.

Заглушки, связники и протокольные объекты канала, привязанного к интерфейсу базового инженерного объекта в кластере в капсуле, могут быть включены в эту капсулу. Все инженерные объекты в капсуле привязаны к одному и тому же интерфейсу управления узлом. Инженерные объекты в других капсулах привязаны к разным интерфейсам управления узлом. Капсула содержится в узле. Капсула имеет менеджера капсулы. Менеджер капсулы привязан (через интерфейс, предоставляющий функцию управления кластером) к каждому менеджеру кластера в капсуле. Эта структура показана на рисунке 5 (рисунок абстрагирован от подробностей ядра).

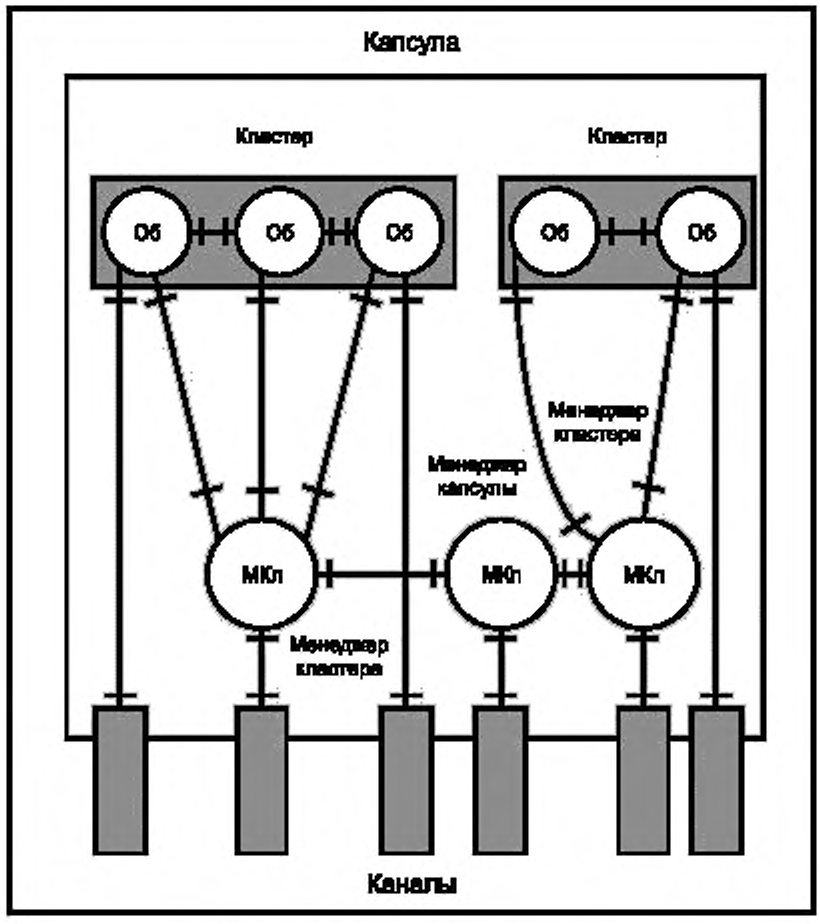

Рисунок 5 — Пример структуры капсулы

Реализация капсулы осуществляется ядром с использованием шаблона капсулы, который задает начальную конфигурацию инженерных объектов в капсуле, включая кластеры, менеджеров кластеров, заглушки, связники, протоколы и менеджера капсулы.

Капсула является контекстом наименования для идентификаторов оконечных точек связывания. В базовой модели не требуется, чтобы эти идентификаторы были допустимыми в каком-либо более широком контексте. Для передачи между капсулами сведений об интерфейсах инженерных объектов (в целях связывания) используют указатели инженерных интерфейсов.

Менеджер капсулы проводит политику управления кластерами в этой капсуле. Политика управления капсулой может привести менеджера капсулы к взаимодействию с другими функциями для завершения деятельностей управления капсулой. Менеджер капсулы имеет интерфейс, обеспечивающий функцию управления капсулой. Структуры, обеспечивающие взаимодействие между менеджерами кластеров, менеджерами капсул и ядрами в узле, относятся к деталям реализации и находятся вне сферы действия настоящей базовой модели.

8.2.7 Правила для узла

Узел состоит из одного ядра и набора капсул. Все инженерные объекты в узле совместно используют общие функции обработки, хранения и коммуникации.

Узел является членом одной или нескольких областей управления указателями инженерных интерфейсов.

Ядро обеспечивает набор интерфейсов управления узлом, по одному на каждую капсулу в узле.

Структура узла показана на рисунке 6.

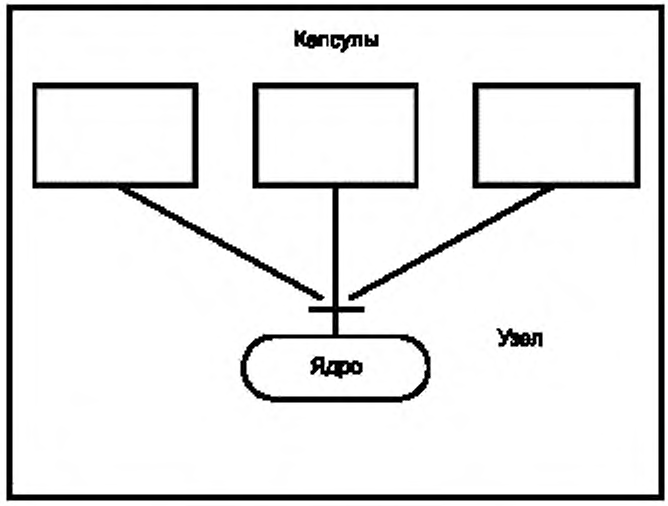

Рисунок 6 — Пример структуры узла

Процедура реализации ядра находится вне области действия настоящей базовой модели; процедура должна привести к:

- введению ядра узла и связанных с ним функций обработки, хранения и коммуникации, включая введение функций управления узлом, делающих возможным распределенное связывание указателей инженерных интерфейсов;
- введению функции торга, необходимой для процесса реализации;
- реализации всех каналов, требуемых как часть начальной конфигурации узла (например, для обеспечения таких инженерных объектов как релокатор).

Набор протокольных объектов, вводимых при реализации узла, определяет начальный набор коммуникационных областей, к которым относится узел. Ядро предоставляет функцию управления узлом и осуществляет политику управления узлом. Эта политика может привести ядро к взаимодействию с другими функциями ОРО для завершения деятельностей по управлению узлом. Капсула является основной единицей для применения политики управления узлом; хотя отдельный объект может использовать функцию управления узлом, этот вопрос относится к политике, применяемой его капсулой. Различные капсулы могут быть субъектом различных политик управления узлом.

8.2.8 Правила прикладного управления

Управление жизненным циклом (созданием, миграцией, деактивацией, реактивацией, взятием контрольной точки, отказами, восстановлением и удалением) скоординированных наборов кластеров проводят специфичной для приложений политикой управления. Прикладная политика управления может применяться к отдельным кластерам или к скоординированным наборам кластеров. Набор управляемых кластеров образует область прикладного управления. Политика прикладного управления может быть реализована специфичными для приложения функциями управления, которые вызывают изменения, используя механизмы, предоставляемые функциями координации и управления, определенными в данной базовой модели, например функцией управления кластером.

Когда это возможно (т. е. в зависимости от того, какие используются прозрачности распределения), специфичные для приложения функции управления могут получать уведомления о существенных событиях, влияющих на управляемые ими кластеры, и предпринимать действия в ответ на эти уведомления. Например, сообщения об отказе связывания могут приводит к реактивации кластера, а сообщения об избыточной нагрузке — к миграции кластеров. Запросы и уведомления, относящиеся к одному кластеру в прикладной области управления, могут приводить к специфичным для приложения функциям управления, инициализирующим действия жизненного цикла для других кластеров в области.

Конкретные прикладные области управления находятся вне сферы действия настоящей базовой модели.

8.2.9 Правила для отказов

Отказы могут быть классифицированы как вовлекающие кластеры, капсулы, узлы или области коммуникации. Анализ отказов основывается на том факте, что:

- отказ, локализованный в кластере, может быть обнаружен менеджером этого кластера;
- отказ, локализованный в капсуле, может быть обнаружен менеджером этой капсулы;
- отказ узла может быть обнаружен протокольными объектами других узлов, с которыми данный узел взаимодействует;
- отказ области коммуникации может быть обнаружен протокольными объектами в других областях коммуникации, с которыми данная область взаимодействует.

Примечание — В коммуникационных системах имеется внутренняя неоднозначность, которая может помешать протокольному объекту отличить коммуникационный отказ от отказа удаленного узла.

8.3 Соответствие и опорные точки

В точке взаимодействия между менеджером кластера и базовым инженерным объектом имеется программируемая опорная точка.

В точке взаимодействия между базовыми инженерными объектами имеется программируемая опорная точка.

В интерфейсе базового инженерного объекта может быть воспринимаемая или обменная опорная точка.

Когда взаимодействие между базовыми инженерными объектами обеспечивается каналом, тогда для каждого участвующего базового инженерного объекта имеются программируемые опорные точки в точках взаимодействия между этими объектами и соответствующие заглушки в канале.

В конфигурации объектов, образующих канал, имеется по программируемой опорной точке в каждой из следующих точек взаимодействия в канале:

- между заглушками (абстрагированная от связников, протокольных объектов и пересечений в канале между заглушками);
- между заглушками и связниками;
- между связниками (абстрагированная от протокольных объектов и пересечений в канале между связниками);
- между связниками и протокольными объектами;
- между протокольными объектами и другими протокольными объектами в том же самом узле (абстрагированная от пересечений, если они есть, между протокольными объектами), и имеется по опорной точке взаимодействия в каждой точке взаимодействия между протокольными объектами и другими протокольными объектами или пересечениями в других узлах.

Управляющие интерфейсы заглушек, связников, протокольных объектов и пересечений являются программируемыми опорными точками.

Когда инженерный объект в канале взаимодействует с другими инженерными объектами (либо в канале, либо вне этого канала) через интерфейсы, которые не находятся в этом канале, тогда опорные точки, применимые к таким интерфейсам, определяются рекурсивным применением настоящих правил.

Взаимодействие систем становится возможным благодаря определению соответствия в опорных точках взаимодействия.

Переносимость инженерных объектов между системами становится возможной благодаря определению соответствия в программируемых опорных точках.

Соответствие отдельных инженерных объектов в программируемых опорных точках само по себе не гарантирует, что инженерный объект будет переносим во все системы или будет взаимодействовать с подходящими инженерными объектами, связанными с другими ядрами.

9 Технологический язык

Технологическая спецификация определяет выбор технологии для системы ОРО.

9.1 Понятия

Технологический язык содержит понятия ГОСТ Р ИСО/МЭК 10746-2 и настоящего стандарта, подчиняющиеся правилам 9.2.

9.1.1 Реализуемый стандарт — шаблон для технологического объекта.

9.1.2 Реализация — процесс реализации, правильность которого может быть предметом тестирования.

9.1.3 ДИТР — дополнительная информация для тестирования реализации.

9.2 Структурирующие правила

Технологическая спецификация определяет выбранную технологию ОРО в терминах:

- конфигурации технологических объектов и
- интерфейсов между ними.

Технологическая спецификация:

- выражает то, как реализуются спецификации системы ОРО;
- идентифицирует для технологии спецификации, относящиеся к конструкции систем ОРО;
- предоставляет таксономию для таких спецификаций;
- специфицирует информацию, требуемую от реализаторов для обеспечения тестирования.

При применении спецификации, написанной на языке другой точки зрения, технологическая спецификация строится так, чтобы дать интерпретацию элементарных терминов спецификаций других точек зрения.

Технологическая спецификация для функции ОРО может ссылаться на спецификации для других функций ОРО.

Технологическая спецификация состоит из утверждений о том, что технологические объекты являются экземплярами названных реализуемых стандартов.

Технологическая спецификация дает форму ДИТР, перечисляя требуемый набор шаблонов и описательных имен для всех необходимых опорных точек.

Все реализуемые стандарты вводятся через ссылки на другие спецификации. Технологический язык не определяет каких-либо других правил, ограничивающих поведение технологических объектов или создающих реализуемые стандарты.

9.3 Соответствие и опорные точки

Технологический язык используют для утверждений, что технологические объекты являются экземплярами реализуемых стандартов; эти стандарты, в общем случае, содержат требования соответствия.

10 Правила согласования

Набор спецификаций системы ОРО, написанных на языках различных точек зрения, не должен содержать взаимно противоречивых утверждений (см. 4.2.2), т.е. они должны быть взаимно согласованными. Таким образом, полная спецификация системы включает в себя утверждения соответствия между терминами и языками, образующими связь спецификаций одной точки зрения со спецификацией другой точки зрения и показывающими, что требования согласованности удовлетворены. Минимальные требования согласованности набора спецификаций для системы ОРО состоят в том, что они должны демонстрировать соответствия, определенные в настоящей базовой модели и в самом наборе спецификаций. Настоящая базовая модель не декларирует родовых соответствий между всеми парами языков точек зрения. В данном разделе ограничиваются спецификации соответствия между вычислительной и информационной спецификациями и между вычислительной и инженерной спецификациями. В любом случае соответствия выражаются как интерпретации взаимосвязей терминов в языке одной точки зрения с терминами в языке другой. Набор спецификаций, основанных на настоящей модели, должен, в общем случае, связывать спецификации со всех точек зрения.

Ключом к согласованности является идея соответствия между спецификациями, т.е., утверждение, что некоторые термины или структуры в одной спецификации соответствуют другим терминам или структурам во второй. Соответствие может быть установлено между двумя разными спецификациями на одном языке или на двух разных языках. Утверждения о соответствии двух языков подразумевают эквивалентные соответствия между парой спецификаций, выражающих эти языки.

Анализ согласованности зависит от применения конкретных методов согласования. Большинство из них основано на проверке конкретных видов несогласованности и, таким образом, не могут служить доказательством абсолютной согласованности. Один вид согласования привлекает набор правил соответствия для управления преобразованием одного языка в другой. Так, для заданных спецификации S_1 на языке точки зрения L_1 и спецификации S_2 на языке точки зрения L_2, где S_1 и S_2 специфицируют одну и ту же систему, к S_1 может быть применено преобразование T, приводящее к новой спецификации $T(S_1)$ на языке точки зрения L_2, которая может непосредственно сравниваться

с S_2 для проверки, например, поведенческой совместимости якобы эквивалентных объектов или конфигураций объектов.

10.1 Соответствие вычислительной и информационной спецификаций

Настоящая базовая модель не предписывает точного соответствия информационных и вычислительных объектов. В частности, не все состояния вычислительной спецификации обязательно должны соответствовать состояниям информационной спецификации. Могут существовать переходные вычислительные состояния с фрагментами вычислительного поведения, которые абстрагированы как элементарные переходы в информационной спецификации.

Когда информационный объект соответствует набору вычислительных объектов, статическая и инвариантная схемы информационного объекта соответствуют возможным состояниям вычислительных объектов. Каждое изменение в состоянии информационного объекта соответствует либо некоторому набору взаимодействий между вычислительными объектами, либо внутреннему действию вычислительного объекта. Инвариантная и динамическая схемы информационного объекта соответствуют поведению и контракту среды вычислительных объектов.

Примечание — Если в информационной спецификации используют понятие информационного интерфейса, то не обязательно должно быть соответствие между информационным интерфейсом и каким-либо вычислительным интерфейсом.

10.2 Соответствие инженерной и вычислительной спецификаций

Каждый вычислительный объект, который не является связующим, соответствует набору из одного или нескольких базовых инженерных объектов (и любых связывающих их каналов). Все базовые инженерные объекты в наборе соответствуют только этому вычислительному объекту.

За исключением случаев, когда участвуют прозрачности, дублирующие объекты, каждый вычислительный интерфейс соответствует только одному инженерному интерфейсу, а этот инженерный интерфейс соответствует только этому вычислительному интерфейсу.

Примечание 1 — Инженерный интерфейс поддерживается одним из базовых инженерных объектов, который соответствует вычислительному объекту, поддерживающему вычислительный интерфейс.

Когда участвуют прозрачности, дублирующие объекты, каждый вычислительный интерфейс дублируемого объекта соответствует набору инженерных интерфейсов, по одному на каждый базовый инженерный объект, получившийся при дублировании. Каждый из этих инженерных интерфейсов соответствует только исходному вычислительному интерфейсу.

Каждый вычислительный интерфейс идентифицируется любым членом набора из одного или нескольких идентификаторов вычислительных интерфейсов. Каждый инженерный интерфейс идентифицируется любым членом набора из одного или нескольких указателей инженерных интерфейсов. Следовательно, так как вычислительный интерфейс соответствует инженерному интерфейсу, идентификатор вычислительного интерфейса может быть недвусмысленно представлен указателем инженерного интерфейса из соответствующего набора.

Каждое вычислительное связывания (элементарное или составное с соответствующими связующими объектами) соответствует либо инженерному локальному связыванию, либо инженерному каналу. Это инженерное локальное связывание или инженерный канал соответствуют только этому вычислительному связыванию. Если вычислительное связывание обеспечивает операции, то инженерное локальное связывание или инженерный канал должны обеспечивать, по крайней мере, обмен:

- именами вычислительных сигнатур;
- именами вычислительных операций;
- именами вычислительных завершений;
- параметрами вызовов и завершений (включая идентификаторы и сигнатуры вычислительных интерфейсов).

За исключением случаев, когда участвуют прозрачности, дублирующие объекты, каждый управляющий интерфейс вычислительного связующего объекта имеет соответствующий инженерный интерфейс и существует цепочка инженерных взаимодействий, связывающая этот интерфейс с некоторыми заглушками, связниками, протокольными объектами или пересечениями, которыми нужно управлять для обеспечения вычислительного связывания.

Примечание 2 — Набор участвующих управляющих интерфейсов зависит от типа связующего объекта.

Каждое вычислительное взаимодействие соответствует некоторой цепочке инженерных взаимодействий, начинающейся и заканчивающейся взаимодействием, которое вовлекает один или

несколько базовых инженерных объектов, соответствующих взаимодействующим вычислительным объектам.

Каждый вычислительный сигнал соответствует либо взаимодействию в инженерном локальном связывании, либо цепочке инженерных взаимодействий, которая обеспечивает необходимый согласованный вид вычислительного взаимодействия.

Требования прозрачности в разделе 16 специфицируют дополнительные соответствия.

Примечания

3 Базовые инженерные объекты, соответствующие разным вычислительным объектам, могут быть членами одного и того же кластера.

4 В полностью объектно-ориентированном вычислительном языке данные представляются как абстрактные типы данных (т.е., интерфейсы к вычислительным объектам).

5 Параметры вычислительного интерфейса (включая параметры для абстрактных типов данных) могут быть переданы по ссылке; такие параметры соответствуют указателям инженерных интерфейсов.

6 Параметры вычислительного интерфейса (включая параметры для абстрактных типов данных) могут быть переданы путем миграции или дублирования объекта, поддерживающего интерфейс. В случае миграции такие параметры соответствуют шаблонам кластеров.

7 Если абстрактное состояние вычислительного объекта, обеспечивающего параметр интерфейса, инвариантно, объект может не мигрировать, а быть клонирован.

8 Шаблоны кластеров могут быть представлены как абстрактные типы данных; следовательно, достаточно строгих соответствий между вычислительными параметрами и указателями инженерных интерфейсов. Использование шаблонов кластеров или данных существенно для инженерных оптимизаций и, следовательно, не исключается.

11 Функции ОРО

В настоящем стандарте определены функции ОРО, которые являются либо фундаментальными, либо широко применимыми для построения систем ОРО.

Спецификации отдельных функций ОРО могут быть скомбинированы для образования спецификаций компонентов систем ОРО. Идентификация таких компонентов является вопросом последующей стандартизации и не предписывается настоящей базовой моделью: она содержит только общее описание функций на основе понятий ГОТС Р ИСО/МЭК 10746-2.

Некоторые описания функций в настоящей базовой модели вводят объекты как упрощающие модельные конструкции. За исключением явных ограничений на распределение этих объектов, они не устанавливают обязательную структуру реализации.

Ниже перечислены функции, определенные в настоящей базовой модели. Те из них, которые являются частью вычислительного языка, помечены символом «*», а являющиеся частью инженерного языка — знаком «+».

а) Функции управления:
   1) функция управления узлом+,
   2) функция управления объектом+,
   3) функция управления кластером+,
   4) функция управления капсулой+.

б) Функции координации:
   1) функция уведомления о событии,
   2) функция создания контрольной точки и восстановления,
   3) функция деактивации и реактивации,
   4) функция группирования,
   5) функция дублирования,
   6) функция миграции,
   7) функция отслеживания указателей инженерных интерфейсов+,
   8) функция транзакции.

в) Функции хранилища:
   1) функция сохранения,
   2) функция организации информации,
   3) функция перемещения,
   4) функция хранилища типов,
   5) функция торга+*.

г) Функции безопасности:
1) функция управления доступом,
2) функция проверки безопасности,
3) функция аутентификации,
4) функция целостности,
5) функция конфиденциальности,
6) функция неопровержения,
7) функция управления ключом.

12 Функции управления

12.1 Функция управления узлом

Контролирует функции обработки, сохранение и коммуникации в узле.

Она предоставляется каждым ядром через один или несколько интерфейсов управления узлом.

Каждая капсула использует интерфейс управления узлом, отличный от интерфейсов управления узлом, используемых другими капсулами в том же самом узле.

Функция управления узлом:

- управляет связками;
- получает доступ к часам и управляет таймерами;
- создает каналы и обнаруживает интерфейсы.

В архитектуре, установленной настоящей базовой моделью, функция управления узлом используется всеми другими функциями.

12.1.1 Управление связками

Интерфейсы управления узлом обеспечивают функции для порождения и разветвления связок в пределах капсулы и для объединения, удаления и синхронизации связок в пределах капсулы.

12.1.2 Доступ к часам и управление таймером

Интерфейсы управления узлом обеспечивают функции для определения текущего времени в заданной области управления часами и для начала, мониторинга и завершения таймеров.

12.1.3 Создание каналов и обнаружение интерфейсов

Интерфейсы управления узлом обеспечивают функции для:

а) возможности связывания инженерного объекта в капсуле с экземпляром функции торга;

б) обеспечения доступности инженерного интерфейса для связывания с объектами в других капсулах;

в) установления связывания инженерного объекта в капсуле с набором других инженерных объектов (идентифицированных указателями инженерных интерфейсов);

г) определения (из заданного указателя инженерного интерфейса) типа канала и предполагаемого им коммуникационного интерфейса.

Примечания

1 Использование взаимодействий б) и в) объяснено в 8.2.3.

2 Взаимодействие г) выявляет информационное содержимое указателя интерфейса, позволяя сохранить или преобразовать его для использования в другой области управления указателями интерфейсов.

Предоставление возможности для связывания интерфейса с объектами в других капсулах — взаимодействие б) — состоит из следующих действий:

- присвоение указателя инженерного интерфейса в заданной области управления указателями инженерных интерфейсов;
- присвоение коммуникационного интерфейса, через который могут быть установлены связывания с рассматриваемым интерфейсом;
- присвоение интерфейсу типа канала.

12.1.4 Реализация шаблона капсулы и удаление капсулы

Интерфейсы управления узлом обеспечивают функции для реализации шаблонов капсул и удаления капсул.

Реализация шаблона капсулы состоит из следующих шагов:

- размещение функций обработки, сохранения и коммуникации для новой капсулы в том же самом узле, что и ядро, обеспечивающее интерфейс управления узлом;
- создание менеджера капсулы для новой капсулы;
- создание интерфейса управления капсулой в новом менеджере капсулы;

- создание идентификатора для интерфейса управления капсулой;
- создание контролирующего интерфейса капсулы для новой капсулы в ядре;
- создание идентификатора контролирующего интерфейса капсулы.

Контролирующий интерфейс капсулы, созданный при реализации шаблона капсулы, позволяет удалять капсулу (например, при отказе менеджера).

Удаление капсулы удаляет все объекты в капсуле.

12.2 Функция управления объектом

Создает контрольные точки и удаляет объекты.

Когда объект относится к кластеру, который может быть деактивирован, для которого может быть создана контрольная точка или который может мигрировать, объект должен иметь интерфейс управления объектом, обеспечивающий одну или несколько из следующих функций:

- создание контрольной точки объекта;
- удаление объекта.

Примечания

1 Таким образом разные интерфейсы управления объектами могут иметь разные типы интерфейсов в зависимости от того, какие функции они поддерживают.

2 Создание контрольной точки объекта дает информации, подходящую для ввода в контрольную точку кластера.

3 При удалении объекта заглушки, связники, протокольные объекты и пересечения, поддерживающие его связывания, могут быть удалены.

Функция управления объектом используется функцией управления кластером.

12.3 Функция управления кластером

Создает контрольные точки, восстанавливает, мигрирует, деактивирует и удаляет кластеры и предоставляется каждым менеджером кластера через интерфейс управления кластером, охватывающий одну или несколько из следующих функций относительно управляемого кластера:

- изменение политики управления кластером (например, в отношении размещения контрольных точек этого кластера, использования функции перемещения для переключения реактивации или восстановления кластера);
- деактивация кластера;
- создание контрольной точки кластера;
- замена кластера новым, реализованным из контрольной точки кластера (т.е., удаление с последующим восстановлением);
- миграция кластера в другую капсулу (используя функцию миграции);
- удаление кластера.

Примечание — Таким образом разные интерфейсы управления кластерами могут иметь разные типы интерфейсов в зависимости от того, какие функции они поддерживают.

Поведение менеджера кластера ограничено политикой управления для этого кластера. Создание контрольной точки кластера и деактивация возможны только если объекты в кластере имеют интерфейсы управления объектами, поддерживающие функцию создания контрольной точки объекта. Деактивация и удаление кластера требуют, чтобы объекты в кластере поддерживали удаление объектов.

В архитектуре, определенной в настоящей базовой модели, функция управления кластером:

- используется функциями управления капсулой, деактивации и реактивации, создания контрольной точки и восстановления, миграции и управления указателями инженерных интерфейсов;
- использует функцию сохранения для хранения контрольных точек.

12.3.1 Создание контрольной точки кластера

Контрольная точка кластера содержит информацию, необходимую для переустановки кластера, и включает в себя следующие элементы:

а) контрольные точки объектов в кластере;

б) конфигурацию объектов в кластере;

в) информацию, достаточную для переустановки распределенных связываний, в которых участвуют объекты кластера.

Примечание — Указатели инженерных интерфейсов являются существенной частью информации, требуемой для установления связываний. Контрольная точка кластера должна содержать все указатели инженерных интерфейсов, вытекающие из перечислений а) и в).

12.3.2 Удаление, деактивация и отказ кластера

Удаление кластера представляет собой удаление всех объектов в кластере, менеджера кластера

и любых объектов, поддерживающих кластер или его менеджера (например, заглушки и связники). Деактивация кластера координируется функцией деактивации и реактивации; при деактивации создается контрольная точка кластера, затем удаляются кластер и поддерживающие его структуры. Отказ кластера вызывает уничтожение всех объектов в кластере и, в некоторых случаях, приводит к уничтожению поддерживающих кластер структур.

12.3.3 Реактивация и восстановление кластера

Деактивированный кластер может быть реактивирован из одной из его контрольных точек. Реактивация кластера является обязанностью функции управления капсулой, так как менеджер кластера был удален в процессе деактивации кластера. Кластер может быть восстановлен из одной из его контрольных точек. Если соответствующий менеджер кластера не был удален, то он может инициировать восстановление; в противном случае восстановление является функцией управления капсулой и включает в себя создание нового менеджера кластера.

12.3.4 Миграция кластера

Состоит в клонировании исходного кластера в целевую капсулу с последующим удалением исходного кластера. Она координируется функцией миграции и параметризуется интерфейсом управления целевой капсулой.

12.4 Функция управления капсулой

Реализует кластеры (включая восстановление и реактивацию), создает контрольные точки всех кластеров капсулы, деактивирует все кластеры капсулы и удаляет капсулу. Она предоставляется каждым менеджером капсулы через интерфейс управления капсулой, включающий в себя одну или несколько из следующих функций относительно управляемой капсулы:

- реализацию (в капсуле) шаблона кластера.

Примечание 1 — Реализация включает в себя реактивацию и восстановление;

- деактивацию капсулы путем деактивации всех кластеров в ней (используя функцию управления кластером);
- создание контрольной точки капсулы путем создания контрольных точек всех кластеров в капсуле (используя функцию управления кластером);
- удаление капсулы путем удаления всех кластеров в ней с последующим удалением менеджера этой капсулы.

Примечание 2 — Таким образом разные интерфейсы управления капсулами могут иметь разные типы интерфейсов в зависимости от того, какие функции они поддерживают.

Поведение менеджера капсулы ограничено политикой управления капсулой.

В архитектуре, определенной в настоящей базовой модели, функция управления капсулой используется функциями деактивации и реактивации, создания контрольной точки и восстановления, а также миграции.

12.4.1 Реализация шаблона кластера

Параметризована шаблоном кластера и состоит из:

- реализации кластера из шаблона;
- введения менеджера для нового кластера;
- создания идентификатора для интерфейса управления кластером в новом менеджере кластера;
- связывания нового кластера с другими объектами в соответствии с правилами инженерного языка и связующей информации в шаблоне кластера.

Примечание — Реактивация и восстановление кластера являются частными случаями реализации кластера, в которых шаблоном кластера является контрольная точка кластера.

Шаблон кластера может содержать информацию, специфичную для области. Если шаблон должен быть реализован в другой области, то эта информация должна быть преобразована. В частности, должны быть преобразованы указатели инженерных интерфейсов, содержащиеся в кластере, если кластер должен быть реализован в другой области управления указателями инженерных интерфейсов.

12.4.2 Удаление капсулы

Влечет за собой удаление менеджера капсулы и может привести к удалению заглушек, связников, протокольных объектов или пересечений, которые поддерживают объекты капсулы или ее менеджера.

13 Функции координации

13.1 Функция уведомления о событии

Записывает и делает доступной историю событий.

13.1.1 Понятия

История событий — объект, представляющий существенные действия.

13.1.2 Правила

Событие создает взаимодействие с функцией уведомления о событии для создания истории событий. Функция уведомления о событии извещает объект-потребитель событий о доступности истории событий.

Функция уведомления о событии поддерживает один или несколько типов истории событий и имеет политику уведомлений о событиях, которая определяет поведение функции, в частности:

- какие объекты могут создавать истории событий;
- какие объекты извещаются о создании новой истории событий;
- способы, которыми осуществляются такие извещения;
- требования постоянства и устойчивости для историй событий;
- отношения упорядочения между взаимодействиями с объектами-производителями и объектами

-потребителями событий.

Потребитель событий взаимодействует с функцией уведомления о событии для регистрации уведомлений о новых историях событий. В зависимости от политики извещения о событиях взаимодействие может:

- установить связывания с текущими доступными историями событий;
- позволить коммуникации с историями событий, созданными после взаимодействия.

Примечание — Истории событий со строгими требованиями постоянства и стабильности могут обеспечиваться использованием функций транзакций и дублирования. Упорядочение и многоцелевые уведомления могут обеспечиваться использованием функции группирования.

13.2 Функция создания контрольной точки и восстановления

Координирует создание контрольных точек и восстановление отказавших кластеров.

Она реализует политику управления:

- когда должны создаваться контрольные точки кластеров;
- когда кластеры должны восстанавливаться;
- где кластеры должны восстанавливаться;
- где должны храниться контрольные точки;
- какая контрольная точка восстанавливается.

Создание контрольной точки и восстановление кластеров является вопросом политики безопасности, связанной с этими кластерами, в частности, в вопросах о том, где должны храниться контрольные точки и где должны восстанавливаться кластеры.

В архитектуре, определенной в настоящей базовой модели, функция создания контрольной точки и восстановления использует функции управления кластером и капсулой.

13.2.1 Создание контрольной точки

За создание контрольной точки отвечают функции управления объектом и кластером. Создание контрольной точки кластера координируется менеджером этого кластера: сначала менеджер кластера использует функцию управления объектом для получения контрольных точек всех объектов в кластере; из этих контрольных точек объектов менеджер кластера строит контрольную точку кластера, которая затем (с помощью функции сохранения) делается постоянной.

В зависимости от политики контрольных точек создание контрольной точки кластера может привести к созданию контрольных точек других кластеров, участвующих в общих деятельностях с рассматриваемым кластером, подчиняясь следующим правилам согласованности:

- начальный кластер должен быть в согласованном состоянии перед тем, как будет создана его контрольная точка;
- должна быть согласованность между всеми совместно полученными контрольными точками различных кластеров (например, все контрольные точки отражают один и тот же набор взаимодействий, происходивших между кластерами);
- когда создается контрольная точка кластера, должны быть созданы контрольные точки всех других кластеров, имеющих ограничения по контрольным точкам с рассматриваемым кластером.

Это правило должно применяться рекурсивно до получения замкнутого множества кластеров, для которых согласованно могут быть созданы контрольные точки.

13.2.2 Восстановление

Кластер может быть восстановлен:

- либо в капсуле, в которой ранее была создана его контрольная точка,
- либо в другой капсуле (например, когда капсула, в которой была создана контрольная точка, в дальнейшем отказала).

За восстановление кластера отвечает менеджер этого кластера или, при его отсутствии, менеджер капсулы. Функция создания контрольной точки и восстановления взаимодействует с менеджером кластера или капсулы для реализации контрольной точки кластера. До восстановления кластера функция создания контрольной точки и восстановления должна гарантировать, что кластер уже убран (например, вследствие отказа). Объекты, привязанные к восстанавливаемому кластеру, должны быть способны обнаружить, что кластер был восстановлен из контрольной точки (например, чтобы они могли повторить взаимодействия, произошедшие после создания контрольной точки).

Восстановление одного кластера может привести к восстановлению других кластеров, например образующих единую контрольную точку с восстанавливаемым кластером.

13.3 Функция деактивации и реактивации

Координирует деактивацию и реактивацию кластеров. Она реализует политику управления:

- когда кластеры должны деактивироваться;
- где должны сохраняться контрольные точки, связанные с деактивацией;
- когда должны реактивироваться кластеры;
- какая контрольная точка должна реактивироваться (например, самая последняя);
- где должны реактивироваться кластеры.

Деактивация и реактивация кластеров являются вопросом политики безопасности, связанной с этим кластером, в частности, в архитектуре, определенной в настоящей базовой модели, функция деактивации и реактивации использует функции управления объектом, кластером и капсулой. Функция деактивации и реактивации используется функцией миграции.

13.3.1 Деактивация

Деактивация кластера является функцией управления кластером и состоит из следующего:

- менеджер рассматриваемого кластера взаимодействует с каждым объектом в этом кластере для получения контрольных точек, которые могут быть использованы для создания контрольной точки кластера;
- менеджер кластера (используя функцию сохранения) делает контрольную точку кластера постоянной;
- менеджер кластера удаляет кластер (и может быть удален сам).

13.3.2 Реактивация

Функция деактивации и реактивации реактивирует кластер, используя функцию управления капсулой для реализации контрольной точки кластера в целевой капсуле (включая создание менеджера кластера). Целевой может быть либо капсула, в которой кластер был ранее деактивирован, либо другая капсула (например, для балансировки инфраструктуры, загруженной в несколько узлов).

13.4 Функция группирования

Предоставляет необходимые механизмы для координации взаимодействий объектов в многостороннем связывании.

13.4.1 Понятия

Группа взаимодействия — подмножество объектов, участвующих в связывании, управляемом функцией группирования.

13.4.2 Правила

Для каждого множества объектов, связанных в группу взаимодействия, функция группирования управляет:

- взаимодействием — решая, какие члены группы в каких взаимодействиях участвуют, в соответствии с политикой взаимодействия;
- сортировкой — получением согласованного вида взаимодействий (включая отказавшие взаимодействия) в соответствии с политикой сортировки;
- упорядочением — гарантируя, что взаимодействия между членами группы корректно упорядочены относительно политики упорядочения;
- членством — работая с отказами и восстановлениями членов группы, а также — с добавлениями и исключениями членов в соответствии с политикой членства.

Примечание — Поведение связующего объекта, связывающего членов группы, определяет, как должно работать взаимодействие.

13.5 Функция дублирования

Является частным случаем функции группирования, когда члены группы поведенчески совместимы (например, потому, что они являются копиями с одного и того же шаблона объекта). Она обеспечивает восприятие группы другими объектами в качестве единственного объекта, гарантируя, что все члены участвуют во всех взаимодействиях и при этом в одном и том же порядке.

Политика членства для группы копий может допускать увеличение или уменьшение числа членов этой группы. Увеличение размера группы копий приводит к тому же результату, что и клонирование члена группы с последующим добавлением к группе в едином элементарном действии.

Для функции дублирования, примененной к кластеру, объекты, входящие в кластер, дублируются и конфигурируются в множество идентичных кластеров. Соответствующие объекты в каждом дубликате кластера образуют группу копий. Таким образом копия кластера является скоординированным множеством групп копий.

Функция дублирования используется функцией миграции.

13.6 Функция миграции

Координирует миграцию кластера из одной капсулы в другую. Она использует функции управления кластером и капсулой и реализует политику, управляющую мигрированием и размещением.

Двумя возможными способами миграции являются:

- дублирование;
- деактивация в одной капсуле с последующей реактивацией в другой.

13.6.1 Дублирование

Миграция кластера с использованием функции дублирования состоит из следующей последовательности действий:

- старый кластер рассматривается как кластер группы копий размера 1;
- создается копия исходного кластера (вместе с менеджером кластера) в капсуле назначения;
- объекты в обоих кластерах собираются в группы копий (размера 2);
- объекты в старом кластере удаляются из групп объектов (возвращая группу к размеру 1);
- старый кластер (и его менеджер) удаляются.

13.6.2 Деактивация и реактивация

Миграция кластера путем деактивации и реактивации координируется менеджером кластера и состоит из деактивации кластера в старом положении с последующей реактивацией — в новом.

13.7 Функция транзакции

13.7.1 Понятия

13.7.1.1 Транзакция — действие, которое приводит к множеству изменений состояния объекта, согласующихся с динамической схемой (и с ограничивающей ее инвариантной схемой).

13.7.1.2 Рассматриваемое действие — действие в транзакции, которое приводит к изменению состояния, существенному для транзакции.

13.7.1.3 Видимость — степень, с которой транзакция может получать доступ к состоянию объекта, конкурируя с другими транзакциями.

13.7.1.4 Восстанавливаемость — степень, с которой отменяются изменения состояния объекта, получившиеся в результате отказавших транзакций.

13.7.1.5 Неизменность — степень, в которой отказы могут повлиять на изменения состояния объекта, вызванные завершенными транзакциями.

13.7.2 Правила

Функция транзакции координирует и контролирует множество транзакций для достижения заданного уровня видимости, восстанавливаемости и неизменности.

Функция транзакции:

- взаимодействует с объектами для мониторинга появления рассматриваемых действий, отмены влияния рассматриваемых действий и причинности рассматриваемых действий;
- решает, находятся ли рассматриваемые действия в конфликте;
- взаимодействует с объектами для составления расписания рассматриваемых действий с целью предотвращения их конфликтов;
- взаимодействует с объектами для отмены влияния рассматриваемых действий, которые уже произошли, с целью разрешения конфликтов.

Вопрос политики выражается в определении следующего:

- какие действия являются рассматриваемыми;

- что является конфликтом рассматриваемых действий;
- какие рассматриваемые действия должны быть отменены для разрешения конфликтов.

13.8 Функция транзакции ACID

Является частным случаем общей функции транзакции, в котором:
- видимость задана как изолированность транзакций друг от друга;
- восстанавливаемость задана как требование элементарности транзакций;
- неизменность задана как требование, чтобы изменения состояний, получающиеся при транзакциях, были устойчивыми (т.е. стабильными);
- согласованность достигается корректным выполнением транзакций со свойствами элементарности, изолированности и устойчивости в соответствии с динамической и инвариантной схемами.

Рассматриваемыми действиями для функции транзакции ACID являются:
- начало транзакции;
- фиксирование транзакции;
- сброс транзакции;
- доступ к состоянию;
- изменение состояния;
- отмена любого из перечисленных действий.

Политика транзакции выражается как последовательные правила для транзакций.

13.9 Функция отслеживания указателей инженерных интерфейсов

Осуществляет мониторинг передачи указателей инженерных интерфейсов между инженерными объектами в разных кластерах для определения того, когда инфраструктура, связанная с инженерными интерфейсами, уже более не требуется (т.е. когда в других кластерах нет объектов, связанных с указанным интерфейсом). Она обеспечивает в пределах своей компетенции:
- информацию об обладании указателями инженерных интерфейсов;
- информацию о существовании интерфейсов.

Она также:
- извещается заглушками, когда указатель инженерного интерфейса передается между кластерами;
- извещается менеджерами кластеров, когда все копии указателя инженерного интерфейса удаляются из этого кластера;
- выявляет отсутствие копий указателя инженерного интерфейса, хранящихся вне кластера, поддерживающего указываемый инженерный интерфейс, и извещает об этом соответствующий менеджер кластера;
- извещает менеджеров капсул, являющихся владельцами указателей инженерных интерфейсов, об отказавших или удаленных интерфейсах.

Примечание — Уведомление о событиях отслеживания указателей инженерных интерфейсов может быть получено путем использования функции уведомления о событиях.

Функция отслеживания указателей инженерных интерфейсов ограничена политикой управления указателями инженерных интерфейсов той области управления указателями инженерных интерфейсов, в которой она применяется.

14 Функции хранилища

14.1 Функция сохранения

Запоминает данные.

14.1.1 Понятия

14.1.1.1 Хранилище данных — объект, предоставляющий функцию сохранения.

14.1.1.2 Интерфейс контейнера — интерфейс хранилища данных, открывающий доступ к данным.

14.1.2 Правила

Хранилище данных запоминает наборы данных. Каждый набор данных ассоциируется с интерфейсом контейнера, созданным при сохранении данных. Интерфейс контейнера предоставляет функции для:
- получения копии данных, сохраненных интерфейсом;
- изменения данных, ассоциированных с интерфейсом;
- удаления интерфейса контейнера и ассоциированных данных.

Примечание — Объекты включают в себя как действия, так и состояния (т.е. процессы и данные); они внутренне постоянны. Функция создания контрольной точки и восстановления или функция дублирования могут быть использованы в инфраструктуре хранилища для обеспечения стабильности. Функции могут использовать другие экземпляры функции сохранения в качестве хранилища для контрольных точек кластеров.

14.2 Функция организации информации

Управляет хранилищем информации, описываемой информационной схемой, и включает в себя некоторые или все из следующих элементов:

- изменение и обновление информационной схемы;
- запрос хранилища, используя язык запросов;
- изменение и обновление хранилища.

Форма и характер запросов и ответов на них зависят от языка запросов. Функция организации информации не допускает изменений или обновлений информационного хранилища, которые не согласуются с его схемой.

Функция организации информации может моделироваться как хранилище объектов (с вычислительными интерфейсами), построенное в соответствии с сущностями и связями системы ОРО. Эти объекты поддерживают операции:

- определения атрибутов, свойств и связей для объектов в хранилище;
- добавления и удаления объектов;
- присвоения и удаления атрибутов, свойств и связей для выбранных объектов в хранилище;
- выбора объектов, которые удовлетворяют предикату, заданному в терминах атрибутов, свойств и отношений.

Примечания

1 Функция организации информации может выводить дополнительные связи из функций, реализованных непосредственно.

2 Функция организации информации может использоваться для обеспечения связей между кластером и его контрольной точкой (т.е. интерфейс сохранения, через который контрольная точка может быть создана или достигнута).

3 Для того чтобы допустить возможность восстановления в ответ на взаимодействия с отказавшим кластером, функция организации информации может использоваться для обеспечения связи между кластером и интерфейсом, через который может быть запрошено восстановление кластера.

4 Для того чтобы допустить возможность реактивации в ответ на взаимодействия с деактивированным кластером, функция организации информации может использоваться для обеспечения связи между деактивированным кластером и интерфейсом, через который может быть запрошена реактивация кластера.

5 Информация, необходимая релокатору для подтверждения или обновления указателя интерфейса, может обеспечиваться функцией организации информации.

14.3 Функция перемещения

Управляет хранилищем положений интерфейсов, включая положения функций управления для кластеров поддерживающих эти интерфейсы.

14.3.1 Понятия

Релокатор — объект, предоставляющий функцию перемещения.

14.3.2 Правила

Релокатор имеет справочник размещений интерфейсов, которые изменили свое положение в результате деятельностей либо управления областью коммуникации (например, изменения сетевого адреса узла), либо управления кластером, таких как деактивация, миграция, дублирование или восстановление.

Интерфейс может быть ассоциирован с релокатором; релокаторы могут быть специфическими для отдельных интерфейсов или общими для нескольких. Если область действия релокатора охватывает несколько областей управления указателями инженерных интерфейсов, то методы доступа к релокатору должны быть понятны для этих областей.

Когда релокатор ассоциирован с интерфейсом деятельности, которые изменяют положение интерфейса, должны информировать релокатор о новом положении, в частности, при перемещении кластера менеджер кластера должен известить релокатор о каждом интерфейсе каждого объекта в кластере. Должны записываться сведения только о тех интерфейсах объектов, которые изменили свое положение. Если инженерный интерфейс должен оставаться допустимым даже когда поддерживающий его объект находится в деактивированном или отказавшем кластере, для которого ранее была создана контрольная точка, то релокатор должен осуществлять политику, использующую

функции координации для реставрации кластера путем реактивации или восстановления, когда другой объект пытается получить доступ к допустимому указателю.

Релокатор поддерживает взаимодействия для того, чтобы:

- записывать изменения положения интерфейса, идентифицированного указателем инженерного интерфейса;
- подтверждать правильность интерфейса, идентифицированного указателем инженерного интерфейса (включая, при необходимости, реставрацию кластера, содержащего объект, который поддерживает интерфейс);
- устанавливать политику взаимодействия с функциями координации (например, функцией деактивации и реактивации) при подтверждении правильности положения интерфейса, идентифицированного указателем инженерного интерфейса.

Примечание — Объект, осуществляющий подтверждение правильности указателя инженерного интерфейса, может возвращать результаты подтверждения так, чтобы оптимизировать последующее использование указателя.

14.4 Функция хранилища типов

Управляет хранилищем типов спецификаций и типов связей. Она имеет интерфейс для каждого хранящегося типа спецификации.

14.4.1 Правила

Функция хранилища типов может быть информирована о типах связей помимо тех, которые могут быть получены из сравнения типов спецификаций. Хранилище типов не допускает установление несогласованных связей.

Типы спецификаций являются неизменными.

Функция хранилища типов включает в себя созданные типы и ассоциированные с ними типы интерфейсов.

Интерфейс хранилища типов для конкретного типа предоставляет функции для:

- запроса спецификации типа;
- установления связей между этим типом и другими типами;
- запроса связей между этим типом и другими типами.

Примечание — Взаимоотношение (связь) между типом и подтипом для типов вычислительных сигнатур определяется правилами образования подтипов сигнатур в 7.2.4. Не требуется, чтобы хранилище типов было в состоянии вычислять связи подтипов сигнатур. Когда типы сигнатур включаются в хранилище типов, хранилищу не разрешается устанавливать дополнительные правила или утверждения для типов сигнатур, противоречащих положениям 7.2.4.

14.5 Функция торга

Обеспечивает объявление и открытие интерфейсов.

14.5.1 Понятия

14.5.1.1 Предложение услуги — информация об интерфейсе, включая как идентификатор интерфейса, так и тип сигнатуры вычислительного интерфейса.

Примечания

1 Идентификатор позволяет привязаться к интерфейсу.

2 Вычислительная сигнатура позволяет торгу обеспечить выбор импорта услуги из предложений услуг, который будет взаимодействовать так, как ожидает импортирующий объект.

3 Дополнительная информация в предложении услуги может быть использована для предоставления более подробных различий, чем содержится в сигнатурах интерфейсов.

14.5.1.2 Экспорт услуги — взаимодействие с функцией торга, в котором объявляется предложение услуги путем его добавления к идентифицированному набору предложений услуг.

14.5.1.3 Импорт услуги — взаимодействие с функцией торга, при котором осуществляется поиск по идентифицированному набору предложений услуг для обнаружения интерфейса, предоставляющего услугу, удовлетворяющую заданному типу.

14.5.2 Правила

Функция торга предоставляет импорт и экспорт услуг, а ее поведением управляет политика торга, которая устанавливает правила, как наборы, идентифицированные в экспорте услуг, соотносятся с наборами в импорте услуг. При импорте услуги функция торга должна выбрать только те предложения, которые удовлетворяют политике самой функции торга, политике экспортера предложения услуг и политике импортера предложения услуг. Импорт услуги вовлекает проверку подтипа/супертипа сигнатуры вычислительного интерфейса. Дополнительно он может вовлекать

последующие уровни проверки, включая проверку поведенческой совместимости и ограничений среды.

15 Функции безопасности

15.1 Понятия

Следующие понятия являются общими для всех функций безопасности.

15.1.1 Политика безопасности — набор правил, которые ограничивают один или несколько наборов деятельностей одного или нескольких наборов объектов.

15.1.2 Уполномоченный по безопасности — администратор, ответственный за реализацию политики безопасности.

15.1.3 Область безопасности — область, члены которой обязаны следовать политике безопасности, установленной и администрируемой уполномоченным по безопасности.

Примечание — Уполномоченный по безопасности является контролирующим объектом для области безопасности.

15.1.4 Политика безопасности взаимодействия — те аспекты безопасности разных областей, которые являются необходимыми для взаимодействий между этими областями безопасности.

15.2 Функция управления доступом

Предотвращает неавторизированные взаимодействия с объектом. Она включает в себя как функцию решения управления доступом, так и функцию исполнения управления доступом. В контексте управления доступом объект играет роль либо цели, либо инициатора. Функция запрашивает информацию управления доступом о цели, инициаторе и взаимодействии.

Инициатор запрашивает у функции управления доступом взаимодействие с целью. Функция решения управления доступом решает на основе информации управления доступом, разрешен или нет доступ, и решение выполняется функцией исполнения управления доступом.

Примечание — Функции решения и исполнения управления доступом могут предоставляться объектом, который играет роль цели или какими-либо другими объектами.

15.3 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.

Включает в себя каждую из следующих функций:

- сборщика сигналов тревоги;
- проверяющего сигналы тревоги;
- анализатора проверки следов;
- архивиста проверки следов;
- записывающего проверки;
- исследователя проверок следов;
- сборщика проверок следов.

15.4 Функция аутентификации

Предоставляет гарантии заявленной идентичности объекта. В контексте аутентификации объект играет одну или несколько из следующих ролей:

- главный;
- заявитель;
- третья доверительная сторона.

Аутентификация требует использования обменной аутентификационной информации.

Примечания

1 Любой идентифицируемый объект в системе ОРО может быть главным в аутентификации, включая объекты, моделирующие людей или компьютерные системы.

2 Объект, инициирующий аутентификацию, не обязательно является заявителем.

Имеется две формы аутентификации:

- парная аутентификация категорий, предоставляющая подтверждение идентичности главного в контексте коммуникационной взаимосвязи;
- аутентификация источника данных, предоставляющая подтверждение идентичности главного, ответственного за конкретный блок данных.

Примечание — Методы аутентификации классифицированы в ИСО/МЭК 10181-2.

При аутентификации, в которой участвуют два объекта, один из них или оба могут играть роль заявителя. Когда оба объекта играют роль заявителя, то аутентификация называется взаимной. Обменная аутентификационная информация передается от инициирующего объекта отвечающему, а затем обменная аутентификационная информация может быть передана в обратном направлении. Может происходить и дополнительный обмен: различные методы аутентификации требуют различное число обменов. Парная аутентификация категорий всегда вовлекает взаимодействие с заявителем. Аутентификация источника данных не обязательно вовлекает взаимодействие с заявителем.

Заявитель обеспечивает операции получения информации, необходимой для запроса аутентификации, и создания обменной аутентификационной информации. Верификатор обеспечивает операции получения информации, необходимой для запроса аутентификации, верификации полученной обменной аутентификационной информации и/или ее создания. Обмен информацией может происходить с сервером аутентификации и либо с заявителем, либо с верификатором, либо до, либо во время аутентификационного обмена.

Функция аутентификации может использовать функцию управления ключом.

15.5 Функция целостности

Выявляет и/или предотвращает неавторизованное создание, изменение или удаление данных. Она включает в себя все следующие функции:

- защиту;
- подтверждение;
- снятие защиты.

В контексте целостности объект играет одну или обе из следующих ролей:

- источник целостно-защищенных данных;
- получатель целостно-защищенных данных.

Целостно-защищенные данные передаются от источника к получателю. Источник целостно-защищенных данных поддерживает интерфейс, предоставляющий функцию защиты. Получатель целостно-защищенных данных поддерживает интерфейс, предоставляющий функции подтверждения и снятия защиты.

Функция целостности может использовать функцию управления ключом.

15.6 Функция конфиденциальности

Предотвращает неавторизованное раскрытие информации.

Она включает в себя функции сокрытия и открытия.

В контексте конфиденциальности объект играет одну или обе из следующих ролей:

- источник конфиденциально-защищенной информации;
- получатель конфиденциально-защищенной информации.

Конфиденциально-защищенная информация передается от источника к получателю. Источник конфиденциально-защищенной информации поддерживает интерфейс, предоставляющий функцию сокрытия. Получатель конфиденциально-защищенной информации поддерживает интерфейс, предоставляющий функцию открытия.

Функция конфиденциальности может использовать функцию управления ключом.

15.7 Функция неопровержения

Предотвращает отрицание одним из объектов, участвовавших во взаимодействии, факта участия полностью или частично в этом взаимодействии.

В контексте неопровержения объект играет одну или несколько из следующих ролей:

- источник (неопровержимых данных);
- получатель (неопровержимых данных);
- генератор доказательства;
- пользователь доказательства;
- верификатор доказательства;
- запросчик неопровержимой услуги;
- нотариус;
- судья.

Функция неопровержения использует неопровержимое доказательство. При неопровержении с проверкой происхождения источник играет роль генератора неопровержимого доказательства и включает это доказательство в признание участия во взаимодействии. Получатель играет роль пользователя доказательства и использует услуги верификатора доказательства (которым может быть сам) для получения уверенности в адекватности доказательства. При неопровержении с проверкой доставки получатель играет роль генератора неопровержимого доказательства и включает это дока-

зательство в признание участия во взаимодействии. Источник играет роль пользователя доказательства и использует услуги верификатора доказательства (которым может быть сам) для получения уверенности в адекватности доказательства.

Нотариус предоставляет функции, требуемые источнику и/или получателю. Они могут включать в себя: нотариальное заверение, отметку времени, мониторинг, сертификацию, создание сертификата, создание подписи, верификацию подписи и доставку по ИСО/МЭК ПМС 10181-4.

Если событие оспаривается, то судья собирает информацию и доказательства от дискутирующих сторон (и, факультативно, от нотариуса) и применяет функцию решения, как описано в ИСО/МЭК ПМС 10181-4.

Функция неопровержения может использовать функцию управления ключом.

15.8 Функция управления ключом

Предоставляет возможности для управления криптографическими ключами и включает в себя следующие элементы:

- создание ключа;
- регистрацию ключа;
- сертификацию ключа;
- дерегистрацию ключа;
- распространение ключа;
- сохранение ключа;
- архивацию ключа;
- удаление ключа.

В контексте управления ключом объект может играть одну или несколько из следующих ролей:

- уполномоченный по сертификации;
- центр распространения ключа;
- центр перевода ключа.

Уполномоченный по сертификации является третьей доверительной стороной, которая создает и присваивает сертификаты, как определено в ИСО/МЭК 11170-1. Центр распространения ключа предоставляет способы безопасного установления информации управления ключом среди объектов, авторизованных для ее получения. Центр перевода ключа является специфической формой центра распространения ключа, который устанавливает информацию управления ключом среди объектов в разных областях безопасности.

16 Прозрачность распределения ОРО

Является селективной для систем ОРО. В настоящей базовой модели описано, как достичь следующих прозрачностей распределения:

- прозрачность доступа;
- прозрачность отказа;
- прозрачность положения;
- прозрачность миграции;
- прозрачность постоянства;
- прозрачность перемещения;
- прозрачность дублирования;
- прозрачность транзакции.

Стандарты ОРО могут определять как уточнения описаний настоящей базовой модели, так и дополнительные требования прозрачностей распределения.

Прозрачности определены как ограничения на отображение из вычислительной спецификации, содержащей схему прозрачности, в спецификацию, которая использует конкретные функции ОРО и инженерные структуры для предоставления требуемых видов маскировки.

Поведение заглушек, связников, протокольных объектов и пересечений в каналах определяется комбинацией прозрачностей распределения (например, прозрачности доступа, прозрачности перемещения), которые применяются для канала.

В некоторых реализуемых стандартах, требующих несколько прозрачностей распределения, могут быть заданы правила композиции для объектов, поддерживающих отдельные прозрачности. В других реализуемых стандартах, требующих несколько прозрачностей распределения, единственный объект может обеспечивать комбинированную прозрачность распределения.

Описания прозрачности в настоящей базовой модели обеспечивают, по крайней мере, следующие комбинации прозрачностей распределения:

а) прозрачности доступа и положения;

б) согласно перечислению а) и прозрачность перемещения;

в) согласно перечислению б) и прозрачность миграции;

г) согласно перечислению б) и прозрачность ресурса;

д) согласно перечислению б) и прозрачность отказа;

е) согласно перечислению а) и прозрачность транзакции;

ж) согласно перечислению б) и прозрачность транзакции.

16.1 Прозрачность доступа

Маскирует различия в представлениях данных и методах вызова, делая возможным взаимодействие между объектами. Она обеспечивается выбором подходящей структуры канала (например, в котором заглушки предоставляют подходящие преобразования, такие как приведение к каноническому представлению данных).

16.2 Прозрачность отказа

Маскирует от объекта отказ и возможное восстановление других объектов и его самого, делая возможной устойчивость относительно неисправностей.

16.2.1 Понятие

Схема устойчивости — спецификация режимов отказов, которые объект не будет проявлять.

16.2.2 Правила

Уточнение прозрачности отказа может удовлетворять схеме устойчивости одним из следующих методов:

- локализуя объект в узле с инфраструктурой, исключающей заданные отказы;
- используя функцию создания контрольной точки и восстановление для того, чтобы сделать объект стабильным;
- используя функцию дублирования для того, чтобы сделать объект стабильным.

16.2.2.1 *Дублирование*

В случае дублирования уточнение прозрачности отказа включает в себя:

- определение интерфейса управления объектом, поддерживающего для вычислительного объекта создание контрольных точек и удаление;
- введение функции дублирования;
- установление политики дублирования для кластеров, содержащих объект;
- привязку релокатора, поддерживающего дублирование, к каждому интерфейсу объекта.

В архитектуре, определенной в настоящей базовой модели, прозрачность отказа, основанная на дублировании кластера, требует прозрачности перемещения.

16.2.2.2 *Создание контрольных точек и восстановление*

В случае создания контрольных точек и восстановления уточнение прозрачности отказа включает в себя:

- определение интерфейса управления объектом, поддерживающего для вычислительного объекта создание контрольных точек и удаление;
- введение функции создания контрольных точек и восстановления;
- установление политики создания контрольных точек и восстановления для кластеров, содержащих объект;
- привязку релокатора, поддерживающего восстановление, к каждому интерфейсу объекта.

В архитектуре, определенной в настоящей базовой модели, прозрачность отказа, основанная на создании контрольных точек и восстановлении кластера, требует прозрачности перемещения.

16.3 Прозрачность положения

Маскирует использование информации о положении в пространстве при идентификации и связывании интерфейсов. Это позволяет объектам получать доступ к интерфейсам без использования информации о положении.

16.4 Прозрачность миграции

Маскирует от объекта способность системы изменять положение этого объекта.

16.4.1 Понятие

Схема мобильности — спецификация, устанавливающая ограничения на мобильность объекта.

Схема мобильности включает в себя:

- ограничения скрытности на взаимодействия с объектом;
- ограничения исполнения на связки объекта;

- ограничения безопасности на положение объекта.

16.4.2 Правила

Прозрачность миграции обеспечивается использованием функции миграции для координации положения объекта с целью удовлетворения схемы мобильности. Уточнение прозрачности миграции включает в себя:

- определение интерфейса управления объектом, поддерживающего для вычислительного объекта создание контрольных точек и удаление;
- введение функции миграции;
- установление политики миграции для кластеров, содержащих объект;
- привязку релокатора к каждому интерфейсу объекта.

В архитектуре, определенной в настоящей базовой модели, прозрачность миграции требует прозрачности перемещения для всех каналов, привязанных к кластеру.

16.5 Прозрачность постоянства

Маскирует от объекта деактивацию и реактивацию других объектов (или его самого).

16.5.1 Понятие

Схема постоянства — спецификация ограничений на использование конкретных функций обработки, сохранения и коммуникации.

16.5.2 Правила

Прозрачность постоянства обеспечивается использованием функции реактивации и деактивации для координации реактивации и деактивации кластеров с целью удовлетворения схемы постоянства. Уточнение прозрачности постоянства включает в себя:

- определение интерфейса управления объектом, поддерживающего для вычислительного объекта создание контрольных точек и удаление;
- введение функции реактивации и деактивации;
- установление политики реактивации и деактивации для кластеров, содержащих объект;
- привязку релокатора, поддерживающего реактивацию, к каждому интерфейсу объекта.

В архитектуре, определенной в настоящей базовой модели, прозрачность постоянства требует прозрачности перемещения для всех каналов, привязанных к кластеру.

16.6 Прозрачность перемещения

Маскирует перемещение интерфейса от других привязанных к нему интерфейсов.

Прозрачность перемещения требует, чтобы:

- с каждым интерфейсом в кластере был связан релокатор;
- информация об изменениях положения объекта передавалась релокаторам (например, менеджерами кластеров);
- имелись связники для обмена дополнительными данными при взаимодействиях через канал для подтверждения действительности связника, поддерживаемого каналом. Эти данные выводятся из положения в пространстве и времени, относящегося к указателю инженерного интерфейса, привязанного к каналу.

При обнаружении связником того, что канал стал недействительным из-за перемещения объекта (например, в результате коммуникационного отказа), связник должен подтвердить правильность указателей инженерных интерфейсов, привязанных к каналу, и, в случае необходимости, переустановить канал. Подтверждение правильности осуществляется функцией перемещения.

Примечание — Если канал стал недействительным в результате деактивации объекта или отказа объекта, для которого ранее была создана контрольная точка, то релокатор должен реализовать как часть подтверждения правильности процедуру восстановления кластера, содержащего объект.

16.7 Прозрачность дублирования

Маскирует использование для обеспечения интерфейса группы поведенчески взаимно совместимых объектов.

16.7.1 Понятие

Схема дублирования — спецификация ограничений на дублирование объекта, включая ограничения как на доступность, так и на исполнение объекта.

16.7.2 Правила

Прозрачность дублирования обеспечивается использованием функции дублирования для координации дублирования объектов с целью удовлетворения схемы дублирования. Уточнение прозрачности дублирования включает в себя:

- определение интерфейса управления объектом, поддерживающего для вычислительного объекта создание контрольных точек и удаление;
- введение функции дублирования;
- установление политики дублирования для кластеров, содержащих объект;
- привязку релокатора к каждому интерфейсу объекта.

В архитектуре, определенной в настоящей базовой модели, прозрачность дублирования требует прозрачности перемещения для кластера.

16.8 Прозрачность транзакции

Маскирует координацию деятельностей в конфигурации объектов для достижения согласованности.

16.8.1 Понятие

Схема транзакции — динамическая и инвариантная, определяющая транзакции и их зависимости.

16.8.2 Правила

Прозрачность транзакции обеспечивается использованием функции транзакции для координации поведения объектов с целью удовлетворения схемы транзакции. Уточнение прозрачности транзакции включает в себя:

- выведение политики функции транзакции из схемы транзакции;
- добавление операций создания контрольной точки и восстановления к состоянию объекта;
- замену связываний объектов функцией транзакции;
- расширение интерфейсов вычислительных объектов.

Расширение интерфейсов объектов включает в себя функции как для уведомления о рассматриваемых действиях, так и для восстановления объекта после отмены транзакции.

ПРИЛОЖЕНИЕ А
(обязательное)

Формальные правила для вычислительных супертипов/подтипов

В настоящем приложении определены формальные правила образования подтипов для сигнатур вычислительных интерфейсов. Типы сигнатур вычислительных интерфейсов могут быть более высоких порядков, как подразумевается 7.2.2.4 в правилах для параметров. В приложении формализовано только подмножество первого порядка правил подтипов. Формализация характеристик более высоких порядков для типов сигнатур вычислительных интерфейсов оставлена для последующего исследования.

В настоящем приложении определена система типов первого порядка, которая состоит из языка простых типов, правил равенства типов и правил подтипов сигнатур. Здесь же описан алгоритм исследования и проверки полноты типа для системы типов. На языке типов определены типы сигнатур интерфейсов сигналов, операций и потоков. Так как образование подтипов для сигнатур интерфейсов потоков определено в 7.2.4.2 лишь частично, то данное приложение формализует только правило подтипов, которое применяется для соответствующих потоков.

А.1 Обозначения и соглашения

В настоящем приложении используют следующие обозначения:

- α, β, γ и т.д. — типы;
- t, s и т.д. — идентификаторы для типов (т.е. переменные типов) и основные типы (т.е. постоянные типов); множество переменных (и постоянных) типов обозначают T_{var};
- a, b, c, a_1, a_2, a_n и т.д. — идентификаторы или метки для элементов структур в языке типов; множество меток обозначают Λ;
- α [β/t] — подстановка β для t в α;
- Nil — предопределенный постоянный тип.

А.2 Система типов

Система типов содержит постоянные типов, функции, декартовы произведения, записи, рекурсивные определения целевых объединений. Язык типов Type задается грамматикой рисунка А.1.

$$
\begin{array}{rl}
a ::= & t \\
| & \perp \\
| & \top \\
| & \alpha \rightarrow \beta \\
| & \alpha_1 \times \ldots \times \alpha_n \\
| & < a_1 : \alpha_1 , \ldots , a_n : \alpha_n > \\
| & [\, c_1 : \gamma_1 , \ldots , c_n : \gamma_n \,] \\
| & \mu t \cdot \alpha
\end{array}
$$

Рисунок А.1 — Абстрактный синтаксис для деклараций типов

Основные типы обозначают $\top$ (верхний) и $\perp$ (нижний). Они играют роли, соответственно, наибольшего и наименьшего элементов в отношении подтипов. Функцию обозначают $\alpha \rightarrow \beta$. Декартово произведение обозначают $\alpha_1 \times \ldots \times \alpha_n$. Объединение обозначают $[c_1 : \gamma_1 , \ldots , c_n : \gamma_n]$. Запись обозначают $< a_1 : \alpha_1, \ldots , a_n : \alpha_n >$.

μ является оператором связывания переменных. Рекурсивные типы могут строиться путем связывания типов с идентификаторами и ссылки на идентификатор для одного типа из другого.

В случае необходимости могут использоваться скобки для установления старшинства. При их отсутствии знак $\rightarrow$ присоединяется направо, а область действия μ простирается направо, насколько это возможно.

Множество свободных переменных, встречающихся в α, обозначают как FV (α).

А.2.1 Правила типов

В настоящем разделе даны правила равенства типов и правила подтипов во введенном языке.

Тип α является заключенным в переменной типа t (обозначают $t \downarrow \alpha$), если либо t не встречается свободным в α, либо α может быть развернут как тип одного из следующих видов:

- $\alpha_1 \rightarrow \alpha_2$;
- $< a_1 : \alpha_1, \ldots , a_m : \alpha_m >$;
- $[c_1 : \gamma_1, \ldots , c_n : \gamma_n]$;
- $\alpha_1 \times \ldots \times \alpha_n$.

Правила равенства типов даны на рисунке А.2. Равенство типов обозначают =.

Правила подтипов дают в виде правил вывода суждений, похожих на программу в Прологе. Суждения имеют вид $\Gamma \vdash \alpha \le \beta$, где Γ — множество предложений об образовании подтипа для переменных типа в виде $\{t_1 \le s_1, \ldots, t_n \le s_n\}$. Типичное правило может иметь следующий вид:

$$\Gamma \vdash \alpha_1 \le \beta_1,\ \Gamma \vdash \alpha_2 \le \beta_2 \Rightarrow \Gamma \vdash \alpha \le \beta.$$

Неформально это означает, что для того, чтобы определить, имеет ли место $\Gamma \vdash \alpha \le \beta$, следует определить, имеют ли место $\Gamma \vdash \alpha_1 \le \beta_1$ и $\Gamma \vdash \alpha_2 \le \beta_2$. Если эти цели достигнуты, то можно сделать вывод, что α является подтипом β.

(E. 1)	$\alpha = \alpha$
(E. 2)	$\alpha = \beta \Rightarrow \beta = \alpha$
(E. 3)	$\alpha = \beta,\ \beta = \gamma \Rightarrow \alpha = \gamma$
(E. 4)	$\alpha_1 = \alpha_2,\ \beta_1 = \beta_2 \Rightarrow \alpha_1 \to \beta_1 = \alpha_2 \to \beta_2$
(E. 5)	$\alpha = \beta \Rightarrow \mu t.\ \alpha = \mu t.\ \beta$
(E. 6)	$\forall j \in \{1, \ldots, n\},\ \alpha_j = \beta_j \Rightarrow \alpha_1 \times \ldots \times \alpha_n = \beta_1 \times \ldots \times \beta_n$
(E. 7)	$\forall j \in \{1, \ldots, n\},\ \alpha_j = \beta_j \Rightarrow \langle a_1 : \alpha_1, \ldots, a_n : \alpha_n \rangle = \langle a_1 : \beta_1, \ldots, a_n : \beta_n \rangle$
(E. 8)	$\forall j \in \{1, \ldots, n\},\ \alpha_j = \beta_j \Rightarrow [a_1 : \alpha_1, \ldots, a_n : \alpha_n] = [a_1 : \beta_1, \ldots, a_n : \beta_n]$
(E. 9)	$\mu t.\ t = \bot$
(E. 10)	$\alpha[\mu t.\ \alpha / t] = \mu t.\ \alpha$
(E. 11)	$\alpha[\beta_1/t] = \beta_1,\ \alpha[\beta_2/t] = \beta_2\ \alpha \downarrow t \Rightarrow \beta_1 = \beta_2$

Рисунок А.2 — Правила равенства типов

Правила подтипов приведены на рисунке А.3. Можно сказать, что α является подтипом β, если $\varnothing \vdash \alpha \le \beta$ может быть выведено с помощью правил подтипов и правил равенства.

А.2.2 Определения типов

Элементы множества Type определяются множеством уравнений взаимозависимости, смоделированным как строго построенная среда. Среда является конечным отображением между переменными типов и типами, относящимися к T, где T — нерекурсивное подмножество Type. Строго построенная среда γ является такой средой, что свободные переменные типа α, связанные с переменной t в области γ, все относятся к области γ. Интуитивно ясно,

(S.1)	$\alpha = \beta \Rightarrow \Gamma \vdash \alpha \le \beta$
(S.2)	$\Gamma \vdash \alpha \le \beta,\ \Gamma \vdash \beta \le \gamma \Rightarrow \Gamma \vdash \alpha \le \gamma,$
(S.3)	$t \le s \in \Gamma \Rightarrow \Gamma \vdash t \le s$
(S.4)	$\Gamma \vdash \bot \le \alpha$
(S.5)	$\Gamma \vdash \le \alpha\ \top$
(S.6)	$\Gamma \vdash \alpha_2 \le \alpha_2,\ \Gamma \vdash \beta_1 \le \beta_2 \Rightarrow \Gamma \vdash \alpha_1 \to \beta_1 \le \alpha_2 \to \beta_2 \le$
(S.7)	$\forall j \in \{1, \ldots, n\},\ \Gamma \vdash \alpha_j \le \beta_j \Rightarrow \Gamma \vdash \langle a_1 : \alpha_1, \ldots, a_m : \alpha_m \rangle \le \langle a_1 : \beta_1, \ldots, a_n : \beta_n \rangle$ для $n \le m$
(S.8)	$\forall j \in \{1, \ldots, n\},\ \Gamma \vdash \alpha_j \le \beta_j \Rightarrow \Gamma \vdash [a_1 : \alpha_1, \ldots, a_n : \alpha_n] \le [a_1 : \beta_1, \ldots, a_n : \beta_n]$ для $n \le m$
(S.9)	$\forall j \in \{1, \ldots, n\},\ \Gamma \vdash \alpha_j \le \beta_j \Rightarrow \Gamma \vdash \alpha_1 \times \ldots \times \alpha_n \le \beta_1 \times \ldots \times \beta_n$
(S.10)	$\Gamma \vee \{t \le s\} \vdash \alpha \le \beta \Rightarrow \Gamma \vdash \mu t.\ \alpha \le \mu s.\ \beta$ для t только в α; s только в β; t, s не в Γ

Рисунок А.3 — Правила подтипов

что каждая переменная типа в среде представляет тип. Связь между переменными типов и элементами Т в среде можно понимать как определяющие взаимную зависимость уравнения для соответствующих типов.

Формально, пусть γ_{wf} — множество строго определенных сред; определим $A \to fB$ как частичную функцию из А в В с конечной областью; FV(α) обозначает множество свободных переменных, встречающихся в α:

$$\gamma_{wf} = \text{def}\{\gamma\colon T_{var} \to f\,\text{Type} \mid \forall\, t, t' \in \text{dom}(\gamma), t' \in FV(\gamma(t)) \Rightarrow t' \in \text{dom}(\gamma)\}$$

Пусть $\gamma = \{t \mid\!\!-\!\!> \alpha, t_1 \mid\!\!-\!\!> \alpha_1 \ldots t_q \mid\!\!-\!\!> \alpha_q\}$. $\gamma\backslash t$ обозначает следующую среду:

$$\gamma\backslash t = \text{def}\ \{t_1 \mid\!\!- \alpha_1 \ldots t_q \mid\!\!-\!\!> \alpha_q\}$$

Типы, связанные с переменной типа t в контексте строго определенной среды γ ($t \in \text{dom}(\gamma)$), по определению, есть Val(t, γ), где Val — функция на типах и средах, рекурсивно определенная на рисунке А.4. Таким образом, любой элемент Type может быть определен как Val(t, γ), где γ — строго определенная среда и $t \in \text{dom}(\gamma)$.

(IT. 1)	$\text{Val}(\bot, \gamma) = \bot$
(IT. 2)	$\text{Val}(T, \gamma) = T$
(IT. 3)	$\text{Val}(\text{Nil}, \gamma) = \text{Nil}$
(IT. 4)	$\text{Val}(\alpha \to \beta, \gamma) = \text{Val}(\alpha, \gamma) \to \text{Val}(\beta, \gamma)$
(IT. 5)	$\text{Val}(<a_1 : \alpha_1, \ldots, a_n : \alpha_n>, \gamma) = <a_1 : \text{Val}(\alpha_1, \gamma), \ldots, a_n : \text{Val}(\alpha_n, \gamma)>$
(IT. 6)	$\text{Val}([a_1 : \alpha_1, \ldots, a_n : \alpha_n], \gamma) = [a_1 : \text{Val}(\alpha_1, \gamma), \ldots, a_n : \text{Val}(\alpha_n, \gamma)]$
(IT. 7)	$\text{Val}(\alpha_1 \times \ldots \times \alpha_n, \gamma) = \text{Val}(\alpha_1, \gamma) \times \ldots \times \text{Val}(\alpha_n, \gamma)$
(IT. 8)	если $t \notin \text{dom}(\gamma)$, то $\text{Val}(t, \gamma) = t$
(IT. 9)	если $t \in \text{dom}(\gamma)$, то $\text{Val}(t, \gamma) = \mu t.\ \text{Val}(\gamma(t), \gamma \setminus t)$

Рисунок А.4 — Семантика определений типов интерфейсов

А.2.3 Алгоритм проверки типа

Осуществляется относительно определенных выше правил равенства и подтипов. В алгоритме участвуют две строго определенные среды ε_1 и ε_2, такие, что $\text{dom}(\varepsilon_1) \cup \text{dom}(\varepsilon_2) = \varnothing$ (сравниваемые типы связаны с двумя переменными, одна из которых в ε_1, другая — в ε_2). Он описан как множество правил вывода, включающих $\varepsilon = \text{def}\varepsilon_1 \cup \varepsilon_2$ и множество Σ вида $\{t_1 \le s_1, \ldots, t_n \le s_n\}$, в котором записаны включения переменных, обнаруженные в процессе выполнения алгоритма. Правило вывода соответствует логической импликации суждений вида $\Sigma, \varepsilon \mid\!\!- \alpha \le \beta$. Суждение интуитивно охватывает справедливость $\alpha \le \beta$ в контексте Σ и ε. Начальные суждения $\{t_1 \le s_1, \ldots, t_n \le s_n\}$ должны быть такими, что $\{t_1, s_1, \ldots, t_n, s_n\} \cup \text{dom}(\varepsilon) = \varnothing$.

Правила вывода приведены на рисунке А.5. На этом рисунке α, β ∈ Type; t, s обозначают произвольные переменные, u обозначает переменную не из dom(ε).

(assmp)	$t \le s \in \Sigma \Rightarrow \Sigma, \varepsilon \mid\!\!- t \le s$
(bot)	$\Sigma, \varepsilon \mid\!\!- \bot \le \beta$
(top)	$\Sigma, \varepsilon \mid\!\!- \alpha \le T$
(var)	$\Sigma, \varepsilon \mid\!\!- u \le u$
(fun)	$\Sigma, \varepsilon \mid\!\!- \alpha_2 \le \alpha_1, \Sigma, \varepsilon \mid\!\!- \beta_1 \le \beta_2 \Rightarrow \Sigma, \varepsilon \mid\!\!- \alpha_1 \to \beta_1 \le \alpha_2 \to \beta_2$
(rcd)	$\forall j \in \{1, \ldots, n\},\ \Sigma, \varepsilon \mid\!\!- \alpha_j \le \beta_j \Rightarrow \Sigma, \varepsilon \mid\!\!- <a_1 : \alpha_1, \ldots, a_n : \alpha_n> \le <a_1 : \beta_1, \ldots, a_m : \beta_m>$ для $n \le m$
(uni)	$\forall j \in \{1, \ldots, n\},\ \Sigma, \varepsilon \mid\!\!- \alpha_j \le \beta_j \Rightarrow \Sigma, \varepsilon \mid\!\!- [a_1 : \alpha_1, \ldots, a_n : \alpha_n] \le [a_1 : \beta_1, \ldots, a_m : \beta_m]$ для $n \le m$
(pro)	$\forall j \in \{1, \ldots, n\},\ \Sigma, \varepsilon \mid\!\!- \alpha_j \le \beta_j \Rightarrow \Sigma, \varepsilon \mid\!\!- \alpha_1 \times \ldots \times \alpha_n \le \beta_1 \times \ldots \times \beta_n$
(rec)	$\Sigma \cup \{t \le s\},\ \varepsilon \mid\!\!- \varepsilon(t) \le \varepsilon(s) \Rightarrow \Sigma, \varepsilon \mid\!\!- t \le s$

Рисунок А.5 — Правила проверки типа интерфейса

Для заданной начальной цели $\Sigma, \varepsilon \vdash \alpha \le \beta$ алгоритм состоит в применении правил вывода в обратном порядке для получения подцелей в случаях (rec), (fun), (rcd), (pro) и (uni). Построенное таким образом дерево целей называется деревом выполнения. Дерево выполнения всегда конечно: если $t \le s$ — положение, которое добавляется к Σ, то t и s являются переменными типов в dom(ε); правила (fun), (pro), (uni) и (rcd) уменьшают размер текущей цели, заменяя ее подвыражениями цели, а каждое применение (rec) увеличивает Σ.

Дерево выполнения успешно, если все листья соответствуют применению одного из правил (assmp), (bot), (top) или (var). Оно неудачно, если по крайней мере один лист является неосуществимой целью (т.е. к ней не может быть применено ни одно из правил). Если дерево выполнения, соответствующее цели $\varnothing\varepsilon \vdash \alpha \le \beta$ успешно, то его обозначают $\vdash A\ \alpha \le \beta$.

Для заданных рекурсивных типов α и β, таких, что $\alpha = Val(t_1, \varepsilon_1)$ и $\beta = Val(t_2, \varepsilon_2)$ (ε_1 и ε_2 определены выше), алгоритмом порождается отношение подтипа ($\le A$), определяемое следующим образом:

$$\alpha \le A\ \beta \Leftrightarrow \vdash A\ t_1 \le t_2$$

Это новое отношение подтипа совпадает с предыдущим, т.е.:
- для данных α, β в T, если $\alpha \le A\ \beta$, то $\alpha \le R\ \beta$;
- для данных α, β в T, если $\alpha \le R\ \beta$, то $\alpha \le A\ \beta$.

А.3 Типы сигнатур интерфейсов сигналов

Формализуются путем их интерпретации на языке Type. Множество типов сигнатур интерфейсов сигналов обозначают $Type_{sig}$. Элементы $Type_{sig}$ абстрактно определяются через использование двух функций: intype : $Type_{sig} \to$ Type и outype : $Type_{sig} \to$ Type. Для заданного типа сигнатуры интерфейса сигналов intype описывает множество начальных сигналов, а outype — ответных.

Элементы Type, связанные с типом сигнатуры интерфейса сигналов функциями intype и outype, определяются строго определенными средами с общим подмножеством Type, определяемом грамматикой рисунка А.6, где метки a_i, $i \in \{1, \ldots, q\}$ предполагаются различными. В результате рисунок А.6 предоставляет абстрактный синтаксис для сигнатур интерфейсов сигналов. Метки a_i соответствуют именам сигналов. Продукция Arg соответствует параметрам сигнала. Продукция Sigsig соответствует отдельным сигнатурам сигналов. Функциональный вид, принятый для отдельных сигнатур сигналов, подчеркивает аналогию с объявлениями сигнатур.

Отношение подтипа для типов интерфейсов сигналов ($\le s$) определяется следующим образом:

$\forall \lambda_1, \lambda_2 \in Type_{sig}, \lambda_1 \le \lambda_2 = \lambda_1.intype \le \lambda_2.intype \wedge \lambda_2.outype \le \lambda_1.outype$

α : : =	$< a_1 : Sigsig, \ldots, a_q : Sigsig>$
Sigsig : : =	$Arg \to Nil$
Arg : : =	$Nil \mid t_1 \times \ldots \times tp$

Рисунок А.6 — Абстрактный синтаксис для типов сигнатур интерфейсов сигналов

А.4 Типы сигнатур интерфейсов операций

Типы сигнатур интерфейсов операционного сервера формализуются путем их интерпретации на языке Type (типы сигнатур интерфейсов операционного клиента могут быть получены непосредственно из них как дополнения). Множество типов интерфейсов операционного сервера обозначают $Type_0(S)$. Элементы $Type_0(S)$ абстрактно определяются через использование функции optype : $Type_0(S) \to$ Type.

Элементы Type, связанные с типом сигнатуры интерфейса операций функцией optype, определяются строго определенными средами с общим подмножеством Type, определяемом грамматикой рисунка А.7, где метки a_i, $i \in \{1, \ldots, q\}$ предполагаются различными, а метки c_i, $i \in \{1, \ldots, q\}$ — различными в контексте продукции Opsig.

α : : =	$< a_1 : Opsig, \ldots, a_q : Opsig>$
Opsig : : =	$Arg \to Term \mid Arg \to Nil$
Term : : =	$[\ c_1 : Arg, \ldots, c_q : Arg]$
Arg : : =	$Nil \mid t_1 \times \ldots \times tp$

Рисунок А.7 — Абстрактный синтаксис для типов сигнатур интерфейсов операций

В результате рисунок А.7 предоставляет абстрактный синтаксис для сигнатур интерфейсов операций. Продукция Opsig соответствует отдельным сигнатурам операций. В частности, продукция Opsig вида $Arg \to Term$ соответствует запросу; продукция Opsig вида $Arg \to Nil$ соответствует сообщению. Продукция Arg соответствует

параметрам вызова. Продукция Term соответствует завершению. Nil в левой части продукции Opsig означает, что данный вызов не имеет параметров. Nil в правой части продукции Opsig (т.е. в сигнатуре сообщения) означает, что завершение не ожидается. Метки a_i соответствуют именам операций. Метки c_i соответствуют именам завершений.

Отношение подтипа для типов интерфейсов операционного сервера ($\leq_o$) определяется следующим образом:

$\forall l_1, l_2 \in Type_o(S), l_1 \leq l_2 = l_1.optype \leq l_2.optype$

А.5 Типы интерфейсов потоков

Полное определение правил подтипов сигнатур для интерфейсов потоков находится вне сферы действия настоящей базовой модели (см. 7.2.4.2). Отметим, однако, что отдельные типы сигнатур потоков могут быть формализованы путем их интерпретации на языке Type.

Элементы Type, связанные с сигнатурой потока, определяются строго определенными средами с общим подмножеством Type, определяемым грамматикой рисунка А.8, где метка a_i соответствует имени потока.

В этом случае правила подтипов в 7.2.4.2, связанные с соответствующими потоками (при условии, что они имеют одинаковую причинность), соответствуют отношению подтипа.

$$\alpha ::= \langle a_j : Flowsig \rangle$$

$$Flowsig ::= Arg \rightarrow Nil$$

$$Arg ::= Nil \mid t$$

Рисунок А.8 — Абстрактный синтаксис для типов сигнатур интерфейсов потоков

А.6 Пример

Рассмотрим следующие определения типа сигнатуры интерфейса операций (т.е. строго определенную среду):

$\gamma = \{ t \mid\!\!-\!\!> \alpha, f_t \mid\!\!-\!\!> \beta \}$,

где α = def : <op: t → [ok : Nil, nok : Nil], factory : Nil → [ok : f_t] >

β = def < new : Nil → [ok : t]>

и t, $f_t \in$ Tvar; op, factory, new, ok, nok ∈ L (op, factory и new — имена операций; ok и nok — имена завершений).

Интуитивно ясно, что среда γ соответствует определению двух типов t и f_t. f_t имеет только одну операцию new, которая не имеет аргументов и возвращает указатель на экземпляр типа t. Можно представить, например, что f_t является типом производителя объектов, который по запросу (т.е. при каждом вызове операции new) создает объекты с интерфейсом типа t. t имеет две операции: op и factory. Операция op получает в качестве аргумента указатель на экземпляр объекта типа t — это первый пример рекурсивного определения. Операция factory не имеет аргументов и возвращает указатель на экземпляр типа f_t. Можно представить, например, что в целях управления каждый объект с интерфейсом типа t может по запросу (т.е. при вызове операции factory) возвращать указатель на создавшего его производителя. Это второй пример рекурсивного определения, так как f_t ссылается на t.

Применяя данное выше определение Val, определяя γ_1 = def{$f_t \mid\!\!-\!\!> \beta$} и используя правило равенства Е.10, получаем:

Val(t, γ) = μt. Val(α, {$f_t \mid\!\!-\!\!> \beta$}

= μt. < op : Val(t, γ_1) → [ok : Nil, nok : Nil]

factory : Nil → [ok : Val(f_t, γ_1)]>

= μt.< op : t → [ok : Nil, nok : Nil]

factory : Nil → [ok : μf_t. Val(β, ∅)] >

= μt. < op: t → [ok : Nil, nok : Nil]

factory : Nil → [ok : μ<f_t < new : Nil → [ok : t]>]>

= μt. < op: t → [ok : Nil, nok : Nil]

factory : Nil → [ok :<new : Nil → [ok : t]>]>

УДК 681.324:006.354 ОКС 35.100.01 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 *Т.И. Кононенко*
Компьютерная верстка *А.Н. Золотаревой*

Изд. лиц. № 02354 от 14.07.2000. Сдано в набор 05.12.2001. Подписано в печать 28.12.2001. Усл.печ.л. 6,51. Уч.-изд.л. 6,70.
Тираж 384 экз. С 3261. Зак. 7.

ИПК Издательство стандартов, 107076, Москва, Колодезный пер., 14.
http://www.standards.ru e-mail: info@standards.ru
Набрано в Издательстве на ПЭВМ
Филиал ИПК Издательство стандартов — тип. "Московский печатник", 103062, Москва, Лялин пер., 6.
Плр № 080102

gost.e-ecolog.ru