ГОСУДАРСТВЕННЫЙ СТАНДАРТ РОССИЙСКОЙ ФЕДЕРАЦИИ

Информационная технология

ВЗАИМОСВЯЗЬ ОТКРЫТЫХ СИСТЕМ

УПРАВЛЕНИЕ ДАННЫМИ И ОТКРЫТАЯ РАСПРЕДЕЛЕННАЯ ОБРАБОТКА

Часть 2

Базовая модель

Издание официальное

ГОССТАНДАРТ РОССИИ
Москва

gost.e-ecolog.ru

Предисловие

1 РАЗРАБОТАН Государственным научно-исследовательским и конструкторско-технологическим институтом «Тест» Государственного комитета Российской Федерации по телекоммуникациям

ВНЕСЕН Государственным комитетом Российской Федерации по телекоммуникациям

2 ПРИНЯТ И ВВЕДЕН В ДЕЙСТВИЕ Постановлением Госстандарта России от 26 декабря 2000 г. № 413-ст

3 Настоящий стандарт содержит полный аутентичный текст международного стандарта ИСО/МЭК 10746-2—96 «Информационная технология. Взаимосвязь открытых систем. Управление данными и открытая распределенная обработка. Часть 2. Базовая модель»

4 ВВЕДЕН ВПЕРВЫЕ

5 ПЕРЕИЗДАНИЕ. Октябрь 2006 г.

© ИПК Издательство стандартов, 2001
© Стандартинформ, 2006

Настоящий стандарт не может быть полностью или частично воспроизведен, тиражирован и распространен в качестве официального издания без разрешения Федерального агентства по техническому регулированию и метрологии

Содержание

1 Область применения 1
2 Нормативные ссылки 1
3 Определения 1
3.1 Определения из других стандартов 1
3.2 Основные определения 1
4 Сокращения 2
5 Классификация понятий 2
6 Основные понятия интерпретации 3
7 Основные лингвистические понятия 3
8 Основные моделирующие понятия 4
9 Специфицирующие понятия 6
10 Организационные понятия 9
11 Свойства систем и объектов 11
11.1 Прозрачность 11
11.2 Понятия политики 11
11.3 Временные свойства 12
12 Понятия наименования 12
13 Понятия для поведения 13
13.1 Структура деятельности 13
13.2 Контрактное поведение 13
13.3 Причинность 14
13.4 Устанавливающие поведения 15
13.5 Надежность 16
14 Понятия управления 16
15 Подход ОРО к соответствию 17
15.1 Соответствие стандартам ОРО 17
15.2 Тестирование и опорные точки 17
15.3 Классы опорных точек 17
15.4 Изменение конфигурации 18
15.5 Процесс тестирования соответствия 18
15.6 Результат тестирования 19
15.7 Отношения между опорными точками 19
Алфавитный указатель терминов 20

ГОСУДАРСТВЕННЫЙ СТАНДАРТ РОССИЙСКОЙ ФЕДЕРАЦИИ

Информационная технология

ВЗАИМОСВЯЗЬ ОТКРЫТЫХ СИСТЕМ

УПРАВЛЕНИЕ ДАННЫМИ И ОТКРЫТАЯ РАСПРЕДЕЛЕННАЯ ОБРАБОТКА

Часть 2

Базовая модель

Information technology. Open Distributed Processing.
Part 2. Reference Model

Дата введения 2002—01—01

1 Область применения

В настоящий стандарт включены понятия, необходимые для осуществления моделирования систем открытой распределенной обработки (ОРО) (см. разделы 5—14), и принципы соответствия системам ОРО (см. раздел 15).

Понятия, определенные в разделах 5—15, используют в описательной модели открытой распределенной обработки для обеспечения определений:

а) структуры семейства стандартов, относящихся к описательной модели;

б) структуры распределенных систем, для которых заявлено о соответствии базовой модели (конфигурации систем);

в) понятий, необходимых для описания совместного использования различных поддерживаемых стандартов;

г) основных понятий, которые должны использоваться в спецификациях различных компонентов, образующих открытую распределенную систему.

2 Нормативные ссылки

В настоящем стандарте использована ссылка на ИСО/МЭК 10746-3—96* Информационная технология. Открытая распределенная обработка. Базовая модель. Часть 3. Архитектура

3 Определения

3.1 Определения из других стандартов

В настоящем стандарте не используют определения из других стандартов, по информационным технологиям.

3.2 Основные определения

3.2.1 **распределенная обработка:** Обработка информации, в которой дискретные компоненты могут быть расположены в различных местах, а связь между компонентами может иметь задержку или оказаться неудачной.

3.2.2 **стандарты ОРО:** Данная базовая модель и те стандарты, которые согласуются с ней прямо или косвенно.

---

* До прямого применения данного документа в качестве государственного стандарта распространение его осуществляет секретариат МТК 22 «Информационная технология». С 1 января 2003 г. действует ГОСТ Р ИСО/МЭК 10746-3—2001.

Издание официальное

3.2.3 **открытая распределенная обработка:** Распределенная обработка, соответствующая стандартам ОРО.

3.2.4 **система ОРО:** Система (см. 6.5), которая соответствует требованиям стандартов ОРО.

3.2.5 **информация:** Любой вид знаний, которыми могут обмениваться пользователи, о предметах, фактах, понятиях и так далее в некотором универсуме.

Хотя информация обязательно имеет форму представления, допускающую ее передачу, интерпретация этого представления (смысл) является как раз тем, что интересует в первую очередь.

3.2.6 **данные:** Формы представления информации, с которой имеют дело информационные системы и их пользователи.

3.2.7 **точка зрения (в системе):** Форма абстракции, достигаемая использованием выбранного множества архитектурных понятий и структурных правил для того, чтобы сосредоточиться на конкретных понятиях в системе.

4 Сокращения

В настоящем стандарте использованы следующие сокращения:
БМ-ОРО — базовая модель открытой распределенной обработки
ВОС — взаимосвязь открытых систем
ВОС/ОТ — отработка транзакций в среде ВОС
ДИРПТ — дополнительная информация о реализации протокола для тестирования
ЗСРП — заявка о соответствии реализации протоколу
ОРО — открытая распределенная обработка
ОТ — обработка транзакций

5 Классификация понятий

Моделирующие понятия, определенные в настоящем стандарте, классифицируют следующим образом:

а) основные понятия интерпретации — для интерпретации моделирующих конструкций любого языка моделирования ОРО. Эти понятия описаны в разделе 6;

б) основные лингвистические понятия — относящиеся к языкам; грамматика любого языка для архитектуры ОРО должна быть описана в терминах этих концепций. Эти понятия описаны в разделе 7;

в) основные моделирующие понятия — для построения архитектуры ОРО; моделирующие конструкции любого языка должны быть основаны на данных понятиях. Эти понятия описаны в разделе 8;

г) специфицирующие понятия — относящиеся к требованиям выбранных языков спецификаций, используемых в ОРО. Эти понятия не являются внутренними для распределения и распределенных систем, но они являются требованиями, которые должны быть рассмотрены в языках спецификаций. Эти понятия описаны в разделе 9;

д) структурирующие понятия — вытекающие из рассмотрения различных вопросов распределения и распределенных систем. Они могут непосредственно поддерживаться языками спецификации, адекватными для работы в проблемной области, а могут и не поддерживаться. Определения объектов и функций, которые непосредственно поддерживают эти понятия, должны быть даны, возможно, с использованием языков выбранных спецификаций. Эти понятия описаны в разделах 10—14;

е) понятия соответствия — необходимые для объяснения нотаций соответствия стандартам ОРО и аттестационного тестирования. Они определены в разделе 15.

В ИСО/МЭК 10746-3 понятия настоящего стандарта использованы для спецификации характеристик распределенной обработки, делающих ее открытой. Они организованы как набор языков точек зрения. Каждый язык точки зрения уточняет понятия из множества, определенного в настоящем стандарте. Необязательно, чтобы все языки точек зрения придерживались одних и тех же нотаций. Для отражения требований данной точки зрения могут быть выбраны различные системы нотаций. Эти нотации могут быть естественными, формальными, текстовыми или графическими. Однако для гарантии полной совместимости необходимо установить соответствие между различными языками.

6 Основные понятия интерпретации

Хотя большая часть архитектуры ОРО относится к определению формальных конструкций, должны быть описаны семантика архитектурной модели и любые используемые языки моделирования. Эти понятия являются главным образом мета-понятиями, то есть понятиями, которые применяются к любому виду деятельности по моделированию. Не предполагается, что эти понятия должны быть формально определены или что они будут использоваться как основа для формального определения других понятий.

Любая деятельность по моделированию идентифицирует:

а) элементы рассматриваемого универсума;

б) один или несколько подходящих уровней абстракции.

Элементами универсума являются категории и утверждения.

6.1 **Категория** — любой рассматриваемый конкретный или абстрактный предмет. Хотя, вообще говоря, слово категория может использоваться для ссылки на что угодно, в контексте моделирования оно зарезервировано для ссылки на предметы в моделируемом универсуме.

6.2 **Утверждение** — наблюдаемый факт или состояние дел, относящееся к одной или нескольким категориям, относительно которых можно принимать или отрицать, что факт или состояние дел имеет место для этих категорий.

6.3 **Абстракция** — процесс отбрасывания несущественных деталей для установления упрощенной модели или результат того процесса.

6.4 **Элементарная** — категория является элементарной на данном уровне абстракции, если она не может быть разбита на более мелкие части на этом уровне абстракции.

Установление данного уровня абстракции может включать в себя идентификацию того, какие элементы являются элементарными.

6.5 **Система** — нечто, рассматриваемое как целое или составленное из частей. Следовательно, система может рассматриваться как категория. Компонент системы сам может быть системой и, в этом случае, он называется подсистемой.

Примечание — Для процесса моделирования, понятие системы понимается в его общем системно-теоретическом смысле. Термин «система» может относиться к системе обработки информации, но он может применяться и более широко.

6.6 **Архитектура** (системы) — набор правил для определения структуры системы и взаимосвязей между ее частями.

7 Основные лингвистические понятия

Любые понятия или семантика языка моделирования для архитектуры ОРО должны быть выражены в некотором синтаксисе, который может включать в себя текстовые или графические соглашения. Принято, что любой подходящий язык должен иметь грамматику, определяющую допустимый набор символов и формально правильные лингвистические конструкции языка. Следующие понятия предоставляют общую структуру для увязки синтаксиса любого языка, используемого для архитектуры ОРО, с интерпретацией понятий.

7.1 **Термин** — лингвистическая конструкция, которая может использоваться для ссылки на категорию.

Ссылка может быть на категорию любого вида, включая модель категории или другую лингвистическую конструкцию.

7.2 **Предложение** — лингвистическая конструкция, содержащая один или несколько терминов и предикатов; предложение может использоваться для выражения утверждения относительно категорий, на которые ссылаются термины.

Предикат в предложении может рассматриваться как ссылка на взаимоотношения между категориями, указанными связанными с ними терминами.

8 Основные моделирующие понятия

Подробная интерпретация понятий, определенных в настоящем разделе, будет зависеть от конкретного языка спецификации, но здесь даны общие понятия независимым от языка способом, чтобы позволить взаимоувязывать утверждения на разных языках.

Основные понятия относятся к существованию и деятельности: выражению того, что существует, где оно находится и что делает.

8.1 **Объект** — модель категории. Объект характеризуется своим поведением (см. 8.6) и, двойственно, своим состоянием (см. 8.7). Объект отличается от любого другого объекта. Объект замкнут, то есть любое изменение его состояния может происходить только в результате внутреннего действия или взаимодействия (см. 8.3) с его средой (см. 8.2).

Объект взаимодействует со своей средой в точках взаимодействия (см. 8.11).

В зависимости от точки зрения акцент может быть сделан на поведение или состояние. Когда акцент сделан на поведение, то неформально говорят, что объект выполняет функции и предлагает услуги (об объекте, который делает функцию доступной, говорят, что он предлагает услугу). Для целей моделирования эти функции и услуги определены в терминах поведения объекта и его интерфейсов (см. 8.4). Объект может выполнять более одной функции. Функция может выполняться совместно несколькими объектами в кооперации.

Примечания

1 Понятия услуги и функции неформально используются для выражения цели части стандартизации. В семействе стандартов ОРО функция и услуга формально выражаются в терминах спецификации поведения объектов и интерфейсов, которые они поддерживают. «Услуга» является частной абстракцией поведения, выражающей гарантии, предлагаемые поставщиком услуг.

2 Выражение «использование функции» является сокращением для выражения «взаимодействие с объектом, выполняющим функцию».

8.2 **Среда** (объекта) — часть модели, которая не является частью данного объекта.

Примечание — Во многих языках спецификации среда может рассматриваться как содержащая по крайней мере один объект, который способен без ограничения участвовать во всех возможных взаимодействиях (см. 8.3), представляя процесс наблюдения.

8.3 **Действие** — нечто, что происходит.

Любое действие, представляющее интерес для целей моделирования, связано по крайней мере с одним объектом.

Набор действий, связанных с объектом, подразделяется на внутренние действия и взаимодействия. Внутреннее действие всегда происходит без участия среды объекта. Взаимодействие происходит с участием среды объекта.

Примечания

1 «Действие» означает «действие имеет место». В зависимости от контекста спецификация может выражать, что действие уже произошло, происходит или может произойти.

2 Подробность действий зависит от проекта. Действие не обязательно должно быть мгновенным. Действия могут перекрываться во времени.

3 Взаимодействия могут быть обозначены в терминах причины и результата взаимоотношений между участвующими объектами. Соответствующие понятия обсуждаются в 13.3.

4 Объект может взаимодействовать сам с собой; в таком случае он рассматривается как исполняющий по крайней мере две роли во взаимодействии, и в таком контексте он может рассматриваться как часть своей собственной среды.

5 Участие среды представляет наблюдаемость. Таким образом, взаимодействия являются наблюдаемыми, а внутренние действия не являются наблюдаемыми потому, что объект замкнут.

8.4 **Интерфейс** — абстракция поведения объекта, которая состоит из подмножества взаимодействий этого объекта и набора ограничений на то, когда они могут происходить.

Каждое взаимодействие объекта относится к единственному интерфейсу. Таким образом, интерфейсы объекта образуют разделение на части взаимодействий данного объекта.

Примечания

1 Интерфейс образует часть поведения объекта, которая получается при рассмотрении только взаимодействий этого интерфейса и сокрытии всех других взаимодействий. Скрытые взаимодействия других интер-

фейсов, вообще говоря, будут вводить неопределенность до тех пор, пока рассматривается только данный интерфейс.

2 Фраза «интерфейс между объектами» используется для указания на связь (см. 13.4.2) между интерфейсами рассматриваемых объектов.

8.5 **Деятельность** — одновершинный направленный ациклический граф действий, где появление каждого действия в графе становится возможным при появлении всех непосредственно предшествующих действий (то есть при появлении всех смежных действий, которые завершаются в этой вершине).

8.6 **Поведение** (объекта) — совокупность действий с набором ограничений на то, когда они могут происходить.

Используемый язык спецификаций определяет ограничения, которые могут быть на нем выражены. Ограничения могут включать в себя, например, ограничения последовательности, неопределенности, параллелизма или реального времени.

Поведение может включать в себя внутренние действия.

Действия, которые фактически имеют место, ограничиваются средой, в которой находится объект.

Примечания

1 Композиция (см. 9.1) совокупности объектов неявно порождает эквивалентный объект, предоставляющий эту композицию. Поведение этого объекта часто называется поведением совокупности объектов.

2 Действие и деятельность являются вырожденными случаями поведения.

3 Вообще говоря, с данным поведением совместимы несколько последовательностей взаимодействий.

8.7 **Состояние** (объекта) — положение объекта в данный момент времени, определяющее набор всех последовательностей действий, в которых объект может принять участие.

Так как, вообще говоря, поведение включает в себя много возможных последовательностей действий, в которых объект может принять участие, знание состояния не обязательно позволяет предсказать последовательность действий, которая произойдет на самом деле.

Изменения состояния осуществляются действиями; следовательно, состояние частично определяется предыдущими действиями, в которых объект принимал участие.

Так как объект замкнут, то его состояние не может быть изменено непосредственно из среды, а только косвенно, в результате взаимодействий, в которых объект принимает участие.

8.8 **Связь** — перенос информации между двумя или несколькими объектами в результате одного или нескольких взаимодействий, привлекающих, возможно, некоторые промежуточные объекты.

Примечания

1 Связь может быть помечена в терминах причины и результата взаимоотношений между участвующими объектами. Соответствующие понятия обсуждаются в 13.3.

2 Каждое взаимодействие является экземпляром связи.

8.9 **Положение в пространстве** — интервал произвольного размера в пространстве, на котором может происходить действие.

8.10 **Положение во времени** — интервал произвольного размера во времени, на котором может происходить действие.

Примечания

1 Протяженность интервала во времени или в пространстве выбирается для отражения требований конкретной задачи спецификации и свойств конкретного языка спецификаций. Единственное положение в одной спецификации может подразделяться на несколько во времени или в пространстве (или и в том, и другом) в других спецификациях. В конкретных спецификациях положение в пространстве или во времени определяется относительно некоторой подходящей системы координат.

2 Обобщая, положение объекта является объединением положений действий, в которых объект может принимать участие.

8.11 **Точка взаимодействия** — положение, в котором существует набор интерфейсов.

В любой данный момент времени точка взаимодействия связывается с положением в пространстве с точностью, допускаемой используемым языком спецификаций. В одном и том же месте могут существовать несколько точек взаимодействия. Точка взаимодействия может быть мобильной.

9 Специфицирующие понятия

9.1 Композиция

а) (объектов) — комбинация двух или нескольких объектов, порождающая новый объект на другом уровне абстракции. Характеристики нового объекта определяются объединяемыми объектами и способом их объединения. Поведение составного объекта является соответствующей композицией поведения компонентов.

б) (поведений) — комбинация двух или нескольких поведений, порождающая новое поведение. Характеристики результирующего поведения определяются объединяемыми поведениями и способом их объединения.

Примечания

1 Примерами методов комбинации являются последовательная композиция, параллельная композиция, чередование, выбор и сокрытие действий. Эти общие определения всегда должны использоваться в конкретном значении, идентифицирующим конкретный смысл комбинации.

2 В некоторых случаях комбинация поведений может порождать вырожденное поведение (например, взаимоблокировку) из-за ограничений на исходные поведения.

9.2 **Составной объект** — объект, выраженный как композиция.

9.3 Декомпозиция

а) (объекта) — спецификация данного объекта как композиции.

b) (поведения) — спецификация данного поведения как композиции.

Композиция и декомпозиция являются двойственными терминами и двойственными спецификациями действий.

9.4 **Поведенческая совместимость** — объект является поведенчески совместимым со вторым объектом относительно набора критериев (см. примечания), если первый объект может заменить второй объект так, что среда не сможет установить различие в поведении объектов на основе набора критериев.

Обычно, критерии устанавливают ограничения на допустимое поведение среды. Если критерии таковы, что среда ведет себя как тестер для исходного объекта, то есть среда определяет наименьшее поведение, которое не ограничивает поведение исходного объекта, то результирующее отношение поведенческой совместимости называется расширением.

Критерии могут допускать замену объектом, который получается при модификации объекта, несовместимого в других отношениях, для того, чтобы замена стала приемлемой. Примером такой модификации может быть сокрытие дополнительных параметров в определенных взаимодействиях. Таким образом, взаимодействие нового объекта может быть сделано похожим на взаимодействие исходного объекта. В таких случаях поведенческая совместимость называется приведенной поведенческой совместимостью. Если модификации не нужны, то поведенческая совместимость называется естественной поведенческой совместимостью.

Понятие поведенческой совместимости, определенное выше для объектов, подходит для поведенческой совместимости шаблонов и типов шаблона.

Поведенческая совместимость рефлексивна, но не обязательно симметрична или транзитивна (хотя может иметь одно или оба из этих свойств).

Примечания

1 Набор критериев зависит от используемого языка и применяемой теории тестирования.

2 Поведенческая совместимость (относительно набора критериев) может быть определена для шаблонов (см. 9.11) и типов шаблонов (см. 9.19), таким образом:

а) если S и T — являются шаблонами объектов, то говорят, что S является поведенчески совместимым с T тогда и только тогда, когда любая S-реализация поведенчески совместима с некоторой T-реализацией (см. 9.13);

б) если U и V являются типами шаблонов объекта, то говорят, что U и V являются поведенчески совместимыми, если соответствующие им шаблоны поведенчески совместимы.

9.5 **Уточнение** — процесс преобразования некоторой спецификации в более подробную. Новая спецификация может рассматриваться как уточнение исходной. Спецификации и их уточнения, обычно, не находятся в одном и том же описании системы. Что именно подразумевается под более подробными спецификациями, зависит от выбранного языка спецификаций.

Для каждой поведенческой совместимости, определенной с некоторым набором критериев (см. 9.4), метод спецификации должен допускать определение уточняющего взаимоотношения. Если шаблон X уточняет шаблон Y, то должна быть возможной замена объекта, реализованного из Y, объектом, реализованным из X, в наборе сред, зависящем от выбранного определения поведенческой совместимости. Уточняющие взаимоотношения не обязательно являются симметричными или транзитивными.

9.6 **Трассировка** — запись взаимодействий объекта от его начального состояния до некоторого другого.

Трассировка объекта, таким образом, является конечной последовательностью взаимодействий. Поведение однозначно определяет множество всех возможных трассировок, но не наоборот. Трассировка не содержит записи внутренних действий объекта.

9.7 **Тип (<X>'а)** — предикат, характеризующий совокупность <X>'ов. <X> имеет тип или соответствует типу, если предикат выполняется для этого <X>. Спецификация определяет, какие используемые ею термины имеют типы, то есть являются <X>'ами. В БМ-ОРО типы необходимы по крайней мере для объектов, интерфейсов и действий.

Назначение типа — классифицировать категории, некоторые из которых могут быть интересны для спецификатора (см. понятие класса в 9.8).

9.8 **Класс (<X>'ов)** — совокупность всех <X>'ов, удовлетворяющих типу (см. 9.7). Элементы множества называются членами класса.

Примечания

1 Класс может не иметь членов.

2 Изменяется ли размер множества со временем, зависит от определения типа.

9.9 **Подтип/супертип** — тип A является подтипом типа B, а B является супертипом A, если каждый <X>, который удовлетворяет A, также удовлетворяет B.

Отношения подтипа и супертипа рефлексивны, транзитивны и антисимметричны.

9.10 **Подкласс/суперкласс** — класс A является подклассом другого класса B, а B является суперклассом A только тогда, когда тип, связанный с A, является подтипом типа, связанного с B.

Примечание — Подкласс, по определению, является подмножеством любого из его суперклассов.

9.11 **Шаблон <X>** — спецификация общих характеристик совокупности <X>' ов, достаточно подробная для того, чтобы <X> мог быть реализован с ее использованием. <X> может быть чем-либо, что имеет тип (см. 9.7).

Шаблон <X> является абстракцией совокупности <X>'ов.

Шаблон может специфицировать параметры, которые должны быть заданы на момент реализации.

Данное здесь определение является родовым; точная форма шаблона зависит от используемого метода спецификации. Типы параметров (когда они применяются) также зависят от используемого метода спецификации.

Шаблоны могут быть скомбинированы в соответствии с некоторым исчислением. Точная форма комбинации шаблонов зависит от используемого языка спецификаций.

9.12 **Сигнатура интерфейса** — набор шаблонов действий, связанных со взаимодействиями интерфейса.

Объект может иметь много интерфейсов с одной и той же сигнатурой.

9.13 **Реализация (шаблона <X>)** — <X>, полученный из данного шаблона <X> и другой необходимой информации. Этот <X> демонстрирует характеристики, специфицированные в шаблоне <X>. <X> может быть чем-нибудь, что имеет тип (см. 9.7).

Данное здесь определение является родовым: как реализовать шаблон <X> зависит от используемого языка спецификаций. Реализация шаблона <X> может включать в себя актуализацию параметров, которая, в свою очередь, может включать в себя реализацию других шаблонов <X> или связывание существующих интерфейсов (см. 12.4).

Примечания

1 Реализация шаблона действия приводит к осуществлению действия. Фраза «реализация шаблона действия» не рекомендуется. Предпочтительнее использовать «осуществление действия».

2 Если <X> является объектом, то он реализуется в своем начальном состоянии. Объект может участвовать во взаимодействиях непосредственно после реализации.

3 Реализации разных шаблонов могут соответствовать одному и тому же типу. Реализации одного и того же шаблона могут соответствовать разным типам.

9.14 **Роль** — идентификатор поведения, который может появляться как параметр в шаблоне составного объекта и который связан с одним из компонентов объектов составного объекта.

Спецификация шаблона как композиции ролей позволяет объяснить процесс реализации как связывание заданных компонентов результирующего составного объекта с соответствующими ролями. Связь компонента с ролью может быть результатом актуализации параметров.

9.15 **Создание (<X>)** — реализация X, которая получается в результате действий объектов в модели. <X> может быть чем-либо, что может быть реализовано, в частности, объектом и интерфейсом.

Если <X> является интерфейсом, то он либо создается как часть создания данного объекта, либо как дополнительный интерфейс к созданному объекту. В результате любой данный интерфейс должен быть частью объекта.

9.16 **Введение (<X>)** — реализация <X>, которая получается не в результате действий объектов в модели.

Примечания

1 <X> может быть реализован либо созданием, либо введением, но не тем и другим одновременно.

2 Введение не применимо к интерфейсам и действиям, так как они всегда поддерживаются объектами.

9.17 **Удаление (<X>)** — действие, разрушающее реализованный <X>. <X> может быть чем-либо, что может быть реализовано, в частности, объектом и интерфейсом.

Если <X> является интерфейсом, то он может быть удален только объектом, с которым связан.

Примечание — Удаление действия не имеет смысла: действие уже случилось.

9.18 **Экземпляр (типа)** — <X>, который соответствует типу.

9.19 **Тип шаблона (<X>'ов)** — предикат, определенный в шаблоне, который справедлив для всех реализаций шаблона и выражает требования реализаций шаблона, которые должны быть выполнены.

Отношение подтип/супертип шаблона объектов не обязательно совпадает с поведенческой совместимостью. Экземпляры типа шаблона не обязательно поведенчески совместимы с реализациями соответствующего шаблона. Они совпадают, если:

а) рассматривается транзитивное отношение поведенческой совместимости, и

б) подтипы шаблона поведенчески совместимы со своими супертипами шаблона.

Примечания

1 Это понятие охватывает понятие замещаемости.

2 Форма предиката, который выражает тип шаблона, зависит от используемого языка спецификаций.

3 «Экземпляры шаблона Т», по определению, являются «экземплярами типа шаблона, связанного с шаблоном Т».

4 На рисунке 1 показаны взаимоотношения между некоторыми из понятий: тип шаблона, класс шаблона и т. д. Набор экземпляров t содержит как набор реализаций t, так и набор всех реализаций подтипов t. Наборы реализаций различных шаблонов всегда разделены.

9.20 **Класс шаблонов (<X>)** — набор всех <X>'ов, удовлетворяющих типу шаблона <X>, то есть набор <X>'ов, которые являются экземплярами шаблона <X>. <X> может быть чем-либо, что имеет тип (см. 9.7).

Каждый шаблон определяет единственный класс шаблона, так что можно ссылаться на экземпляры шаблона как на экземпляры класса-шаблона.

Понятие класса используется для указания родовой классификации <X>'ов. Класс шаблона является более ограничивающим понятием, если рассматривать те члены класса шаблона, которые реализуются из шаблона (или из его подтипов), то есть те <X>, которые удовлетворяют типу шаблона <X>.

Примечание — Задавая тип шаблона, можно сократить утверждение, что «класс шаблонов, связанный с шаблоном А, является подклассом класса шаблонов, связанного с шаблоном В» до «шаблон А является подклассом шаблона В» или «шаблон А является подтипом шаблона В».

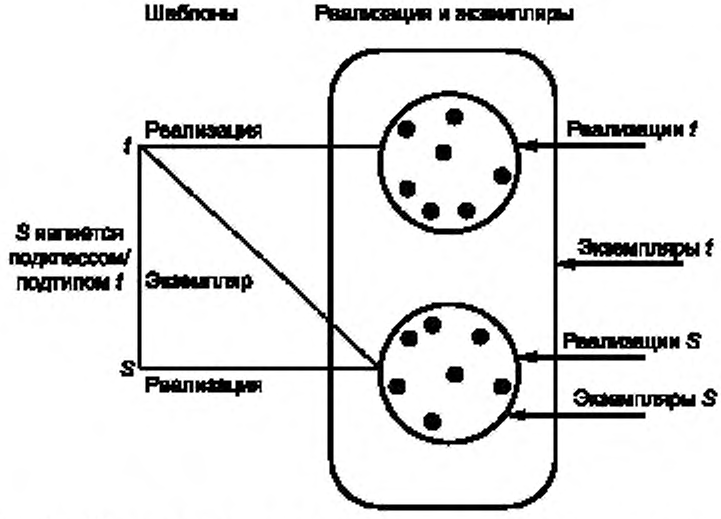

Рисунок 1 — Взаимоотношения между шаблонами, реализациями и экземплярами

9.21 **Производный класс/базовый класс** — если шаблон А является возрастающей модификацией шаблона В, то класс шаблонов СА экземпляров А является производным классом класса шаблонов СВ экземпляров В, а СВ является базовым классом СА.

Критерий того, чтобы рассматривать изменения в качестве возрастающей модификации, зависит от метрик и соглашений, не входящих в область действия настоящего стандарта. Если критерии позволяют, то производный класс может иметь несколько базовых классов.

Возрастающая модификация связанных шаблонов должна гарантировать, что ссылка на самого себя или рекурсия в шаблоне базового класса становится ссылкой на самого себя или рекурсией в шаблоне производного класса.

Возрастающая модификация может, в общем случае, включать в себя добавление или изменение свойств базового шаблона для получения производного шаблона.

Классы могут быть упорядочены в иерархию наследования в соответствии с соотношениями производный класс/базовый класс. Это является интерпретацией наследований в базовой модели ОРО. Если классы могут иметь несколько базовых классов, то говорят, что наследование является кратным. Если критерии запрещают подавление свойств базового класса, то говорят, что наследование является строгим.

Один класс может быть подклассом другого класса, не будучи производным классом, и быть производным классом, не будучи подклассом. Следовательно, иерархия наследования (когда дуги обозначают отношение производного класса) и иерархия типов (когда дуги обозначают отношения подтипов или подклассов) являются логически различными, хотя могут совпадать полностью или частично.

9.22 **Инвариант** — предикат, задающий требования, которые должны быть справедливы на протяжении всего времени жизни множества объектов.

9.23 **Предусловие** — предикат, задающий требования, которые должны быть справедливы для того, чтобы действие осуществилось.

9.24 **Постусловие** — предикат, задающий требования, которые должны быть справедливы непосредственно после осуществления действия.

10 Организационные понятия

10.1 **Группа <X>** — множество объектов с конкретным характеризующим отношением <X>. Отношение <X> характеризует либо структурное взаимоотношение между объектами, либо ожидаемое общее поведение объектов.

Примечание — Примерами специализированных групп являются:

а) группа адресации — множество объектов, к которым адресуются одним и тем же способом;

б) группа неисправностей — множество объектов, которые имеют общую зависимость от неисправностей. Например, может быть принято, что, если компьютер неисправен, то все объекты, существующие на этом компьютере, также неисправны;

в) группа взаимодействий — множество объектов, все объекты которого участвуют в одной и той же последовательности взаимодействий со средой;

г) группа устойчивых ответов на неисправности — группа взаимодействий, назначением которой является обеспечение определенного уровня устойчивости относительно некоторых неисправностей.

10.2 **Конфигурация** (объектов) — совокупность объектов, способных взаимодействовать через интерфейсы. Конфигурация определяет набор объектов, участвующих в каждом взаимодействии.

Спецификация конфигурации может быть статической или выраженной в терминах действия динамического механизма, который изменяет конфигурацию, как операции связывания и развязывания (см. 13.4).

Примечание — Конфигурация может быть выражена в терминах понятий параллельной композиции. Процесс композиции создает объект, эквивалентный конфигурации на другом уровне абстракции.

10.3 **Область <X>** — множество объектов, каждый из которых связан характеризующим отношением <X> с управляющим объектом.

Каждая область имеет управляющий объект, связанный с ней.

Управляющий объект может определять идентичность совокупности объектов, которые образуют соответствующую область. Управляющий объект может взаимодействовать с управляемым объектом динамически или можно считать, что взаимодействие произошло в предшествующую эпоху (см. 10.5) управляющего объекта. Вообще говоря, управляющий объект не является членом связанной с ним области.

Примечания

1 В терминах предпринимателя, управляющим объектом в области может проводиться различная политика.

2 Области могут быть разделенными или перекрывающимися.

3 По определению, область является группой, но не наоборот.

4 Примерами специализированных областей являются:

Область	Член класса	Отношение	Управляющий класс
безопасности	Обрабатывающий объект	Подчиняется политике, установленной	Объект «уполномоченный по защите»
управления	Управляемый объект		Объект «область управления»
адресации	Адресуемый объект	Адрес, выделенный	Объект «уполномоченный по адресации»
наименования	Поименованный объект	Имя, выделенное	Объект «уполномоченный по наименованию»

10.4 **Подобласть** — область, которая является подмножеством данной области.

10.5 **Эпоха** — период времени, в течение которого объект демонстрирует конкретное поведение. Любой объект в любое время находится в единственной эпохе, но взаимодействующие объекты могут находиться в разных эпохах взаимодействия.

Изменение эпохи может быть связано с изменением типа объекта для того, чтобы поддержать эволюцию типа. Напротив, изменение эпохи может быть связано с фазой в поведении объекта постоянного типа.

Для того чтобы распределенные системы функционировали правильно, объекты, образующие ее конфигурацию, должны быть согласованы. Таким образом, по мере того, как система в целом проходит через ряд эпох, отдельные взаимодействующие объекты не должны находиться в эпохах,

в которых их поведения столь существенно различаются, что их параллельная композиция приводит к отказу. Эта концепция обеспечивает формализацию понятий версии и расширяемости.

Примечание — Может потребоваться язык спецификаций для выражения:

а) способа пометки эпох;

б) последовательности эпох и того, должны ли все объекты пройти через всех членов этой последовательности;

в) правил для выведения эпохи композиции из эпох ее объектов, в частности, для конфигураций и составных систем;

г) того, является ли идентификация эпохи объекта необходимой частью состояния того объекта;

д) того, могут ли объекты провести согласование на основе идентификации их текущих эпох;

е) связи эпохи с понятиями локального и глобального времени.

10.6 **Опорная точка** — точка взаимодействия, определенная в архитектуре для выбора в качестве точки соответствия в спецификации, согласующейся с этой архитектурой.

В ОРО идентифицированы важные классы опорных точек; подробности и отношение моделирования к соответствию приведены в разделе 15.

10.7 **Точка соответствия** — опорная точка, в которой поведение может быть наблюдаемо с целью проверки соответствия.

11 Свойства систем и объектов

В данном разделе описаны свойства, которые могут быть у системы ОРО или ее части.

11.1 Прозрачность

11.1.1 **Прозрачность распределения** — свойство сокрытия от конкретного пользователя потенциального поведения некоторых частей распределенной системы.

Примечание — Пользователями могут быть, например, конечные пользователи, прикладные разработчики и реализаторы функций.

11.2 Понятия политики

11.2.1 **Контракт** — соглашение, управляющее частью коллективного поведения набора объектов. Контракт специфицирует обязательства, разрешения и запрещения для участвующих объектов.

Спецификация контракта может содержать:

а) спецификацию различных ролей, которые могут выполнять объекты, участвующие в контракте, и интерфейсы, связанные с этими ролями;

б) атрибуты качества услуги (см. 11.2.2);

в) указания продолжительности или периодов действия контракта;

г) указания поведения, которое нарушает контракт;

д) условия живучести и сохранности.

Примечания

1 Объекты в контракте не обязательно должны быть иерархически связаны, а могут быть связаны по принципу «равный к равному». Требования контракта не обязательно применимы одним и тем же образом ко всем участвующим объектам.

2 Контракт может применяться к заданной опорной точке в системе. В этом случае он задает поведение, которое может ожидаться в этой опорной точке.

3 Шаблон объекта представляет собой простой пример контракта. Шаблон объекта задает поведение, общее для совокупности объектов. Тем самым он определяет, что среда любого такого объекта может зависеть от его поведения. Отметим, что для частичных спецификаций шаблон объекта оставляет неопределенным поведение объекта при определенных условиях среды (например, конкретные взаимодействия); контракт распространяется только на специфицированное поведение.

11.2.2 **Качество услуги** — набор требований качества к коллективному поведению одного или нескольких объектов.

Качество услуги может быть задано в контракте или измерено и сообщено после события.

Качество услуги может быть параметризовано.

Примечание — Качество услуги относится к таким характеристикам, как скорость передачи информации, время ожидания, вероятность разрыва связи, вероятность отказа системы, вероятность отказа памяти и т. д.

11.2.3 **Контракт среды** — контракт между объектом и его средой, включая ограничения качества услуги, ограничения использования и управления. Ограничения качества услуги включают в себя:

- временны́е ограничения (например, предельный срок);
- объемные ограничения (например, пропускную способность);
- ограничения зависимости, охватывающие вопросы доступности, надежности, сопровождаемости, безопасности и сохранности (например, средняя наработка на отказ).

Ограничения использования и управления включают в себя:

- ограничения положения (то есть выбранные положения в пространстве и времени);
- ограничения прозрачности распределения (то есть выбранные прозрачности распределения).

Ограничения качества услуги могут подразумевать ограничения использования и управления. Например, некоторые ограничения качества услуги (такие, как доступность) удовлетворяются предоставлением одной или нескольких прозрачностей распределения (такой, как дублирование).

Ограничения среды могут описывать:

- требования, устанавливаемые для среды объекта для его конкретного поведения;
- ограничения на поведение объекта в конкретной среде.

11.2.4 **Обязательство** — предписание, которым требуется конкретное поведение. Обязательство выполняется осуществлением предписанного поведения.

11.2.5 **Разрешение** — предписание, которое позволяет осуществлять конкретное поведение. Разрешение эквивалентно тому, что нет обязательства не осуществлять это поведение.

11.2.6 **Запрещение** — предписание того, что конкретное поведение не должно осуществляться. Запрещение эквивалентно тому, что есть обязательство не осуществлять это поведение.

11.2.7 **Политика** — набор правил, относящихся к конкретной цели. Правило может быть выражено как обязательство, разрешение или запрещение.

Примечание — Не каждая политика является ограничением. Некоторые политики предоставляют полномочие.

11.3 Временны́е свойства

11.3.1 **Постоянство** — свойство того, что объект продолжает существовать при изменениях контрактного контекста (см. 13.2.3) или эпохи.

11.3.2 **Изохронность** — последовательность действий является изохронной, если каждая соседняя пара действий в последовательности занимает во времени уникальные, смежные интервалы одинакового размера.

12 Понятия наименования

12.1 **Имя** — термин, который в данном контексте наименования указывает на категорию.

12.2 **Идентификатор** — недвусмысленное имя в данном контексте наименования.

12.3 **Пространство имен** — набор терминов, используемых в качестве имен.

12.4 **Контекст наименования** — взаимоотношение между множеством имен и множеством категорий. Множество имен относится к одному пространству имен.

12.5 **Именующее действие** — действие, которое связывает термин из пространства имен с данной категорией.

Все именующие действия относятся к контексту наименования.

12.6 **Область наименования** — подмножество контекста наименования такое, что все именующие действия осуществляются управляющим объектом области (объектом, уполномоченным по наименованию).

Примечание — «Область наименования» является примером понятия области <X> (см. 10.3).

12.7 **Граф наименования** — направленный граф, в котором каждая вершина обозначает контекст наименования, а каждое ребро обозначает связь между:

- именем, появляющимся в исходном контексте наименования, и
- целевым контекстом наименования.

Примечание — Существование ребра между двумя контекстами наименования в графе наименования имен означает, что целевой контекст наименования может быть достигнут (идентифицирован) из исходного контекста наименования.

12.8 **Разрешение имени** — процесс, с помощью которого, задавая исходное имя и исходный контекст наименования, может быть найдена связь между именем и категорией, обозначенной исходным именем.

Примечание — Процесс разрешения имени не обязательно предоставляет достаточно информации для взаимодействия с указанной категорией.

13 Понятия для поведения

13.1 Структура деятельности

13.1.1 **Цепочка** (действий) — последовательность действий в деятельности, когда для каждой смежной пары действий осуществление первого действия является необходимым для осуществления второго действия.

13.1.2 **Связка** — цепочка действий, когда по крайней мере один объект участвует во всех действиях цепочки.

Объект может быть ассоциирован с одной единственной связкой или с несколькими связками одновременно.

13.1.3 **Объединяющее действие** — действие совместно используемое двумя или несколькими цепочками, результатом чего является одна цепочка.

13.1.4 **Разделяющее действие** — действие, которое допустимо для двух или нескольких цепочек.

Имеются два случая разделяющего действия в зависимости от того, требуется ли объединение допустимых цепочек.

13.1.5 **Разветвляющее действие** — разделяющее действие, когда допустимые цепочки должны (в случае отказа) в конце объединиться друг с другом, то есть допустимые цепочки не могут объединяться с другими цепочками и завершаться по отдельности.

13.1.6 **Порождающее действие** — разделяющее действие, когда допустимые цепочки не будут объединяться. Допустимые цепочки могут взаимодействовать и могут завершаться по отдельности.

13.1.7 **Заглавное действие** — действие, которое в данной деятельности не имеет предшественника.

13.1.8 **Поддеятельность** — подграф деятельности, который сам является деятельностью и удовлетворяет следующему условию для любой пары разветвляющих объединяющих действий в порождающей деятельности, если одно из этих действий входит в подграф, то и оба должны входить в подграф.

13.2 Контрактное поведение

Вводимые понятия показаны на рисунке 2.

13.2.1 **Устанавливающее поведение** — поведение, с помощью которого данный контракт вводится в действие между данными объектами. Устанавливающее поведение может быть:

а) явным, получающимся из взаимодействий объектов, которые будут частью контракта, или



Рисунок 2 — Соединение и связанные понятия

б) неявным, осуществляемым внешним агентом (например, объектом третьей стороны, не принимающим участие в контракте) или осуществленным в предыдущую эпоху.

Примечания

1 Согласование является примером частного вида устанавливающего поведения, при котором информация передается в процессе достижения общего взгляда на допустимое будущее поведение.

2 Публикация является примером частного вида устанавливающего поведения, при котором информация распространяется от одного объекта к ряду других.

3 Явное устанавливающее поведение должно включать в себя реализацию шаблона, связанного с контрактом. Оно может следовать за возможным согласованием/публикацией того, какой контракт должен быть установлен, какой шаблон должен быть реализован и с какими параметрами.

13.2.2 **Допустимое поведение** — поведение, характеризующее набор объектов, который становится возможным в результате устанавливающего поведения.

Допустимое поведение не обязательно должно быть одним и тем же для всех объектов.

13.2.3 **Контрактный контекст** — знание того, что конкретный контракт действует и, таким образом, требуется конкретное поведение множества объектов.

Объект может находиться в нескольких контрактных контекстах одновременно; его поведение ограничивается пересечением поведений, предписанных каждым контрактным контекстом.

Примечание — В ВОС понятие контекста представления является примером контрактного контекста, который может быть принят при установлении соединения или позже.

13.2.4 **Соединение** — взаимоотношения между набором объектов, которое является результатом осуществления некоторого устанавливающего поведения; констатация наличия контрактного контекста вообще.

Соединение характеризуется соответствующим допустимым поведением.

Примечания

1 Примерами соединений, которые получаются в результате различных устанавливающих поведений, являются:

а) диалог (как в ВОС-ОТ);

б) связывание (см. 13.4.2);

в) распределенная транзакция (как в ВОС-ОТ);

г) (N)-соединение (как в ВОС);

д) ассоциация между (N)-объектами, позволяющая им участвовать в (N)-передаче без установления соединения (как в ВОС);

е) взаимоотношения между файлами и процессами, которые обращаются к этим файлам.

2 Некоторые поведения могут быть обусловлены установлением кратных взаимосвязанных соединений. Например, распределенная транзакция может зависеть как от соединения между пользователями транзакции, так и от поддерживающей ассоциации. Соединение между пользователями транзакции (распределенная транзакция) может продолжать существовать, но быть неактивным, когда ассоциация разрушена.

3 В соединение могут быть вовлечено более двух объектов. Не обязательно все объекты, вовлеченные в соединение, имеют эквивалентные роли. Так, могут быть соединения для сбора или распределения информации. Число участников и их роли определяются контрактом, который выражен соединением.

4 Имеется двойственность между контрактным контекстом, принятием контрактного обязательства спецификации и допустимым поведением. На практике, структуры могут быть произвольно вложенными и при установлении соединения на одном уровне может быть также согласован контракт, который допускает внутренние уровни соединения.

13.2.5 **Завершающее поведение** — поведение, которое разрушает соединение и аннулирует соответствующие контрактный контекст и контракт.

Завершающее поведение должно быть явно идентифицировано в контракте как таковое, если устанавливающее поведение было явным.

13.3 Причинность

Идентификация причинности позволяет классифицировать роли взаимодействующих объектов. В настоящем разделе приведен основной набор ролей.

Причинность подразумевает ограничение на любое поведение участвующих объектов во время их взаимодействия. Причинность должна быть идентифицирована в определении классов (или подклассов), к которым относятся взаимодействующие объекты, или в уточнении шаблонов для их классов (или подклассов).

13.3.1 **Инициирующий объект** (относительно связи) — объект, вызывающий связь.

Примечание — Идентификация инициирующего объекта относительно связи включает в себя интерпретацию назначения связи.

13.3.2 **Отвечающий объект** — объект, принимающий участие в связи, но не являющийся инициирующим объектом.

13.3.3 **Производящий объект** (относительно связи) — объект, который является источником передаваемой информации.

Использование этого термина не подразумевает какого-либо конкретного метода связи.

13.3.4 **Потребляющий объект** (относительно связи) — объект, который является приемником переданной информации.

Использование этого термина не подразумевает какого-либо конкретного метода связи.

13.3.5 **Объект-клиент** — объект, который запрашивает, чтобы функция была осуществлена другим объектом.

13.3.6 **Объект-сервер** — объект, который осуществляет некоторую функцию в интересах объекта-клиента.

Взаимоотношения клиент/сервер различной природы (или уровня абстракции) могут существовать между объектом и различными композициями объектов, с которыми они взаимодействуют.

Примечание — Неформально говорят, что сервер предоставляет услугу, запрошенную клиентом.

13.4 Устанавливающие поведения

13.4.1 **Связывающее поведение** — устанавливающее поведение между двумя или несколькими интерфейсами (и, следовательно, между поддерживающими их объектами).

Примечание — «Связать» означает «выполнить связывающее поведение».

13.4.2 **Связывание** — контрактный контекст, возникающий из данного устанавливающего поведения.

Устанавливающее поведение, контрактный контекст и допустимое поведение могут вовлекать два или несколько интерфейсов объектов.

Объект, который инициирует устанавливающее поведение, может участвовать или не участвовать в последующем допустимом поведении.

Допустимое поведение (и, аналогично, контрактный контекст) может быть однородным (то есть каждый участвующий объект может выполнять те же самые действия, что и любой другой) или неоднородными (то есть один участвующий объект играет роль, отличную от других ролей, как в случае клиент/сервер).

Соответствие между объектом, который инициирует устанавливающее поведение, и конкретной ролью в неоднородных допустимых поведениях необязательно должно быть однозначным (например, в контрактном контексте клиент-сервер любой объект может инициировать устанавливающее поведение).

13.4.3 **Предусловие связывания** — набор условий, требуемых для успешного выполнения связывающего поведения.

Объекты, осуществляющие связывающее поведение, должны иметь идентификаторы для всех интерфейсов, участвующих в связывании. Могут существовать дополнительные предусловия.

13.4.4 **Развязывающее поведение** — поведение, которое завершает связывание, то есть завершающее поведение для связывания.

13.4.5 **Торг** — взаимодействие между объектами, при котором они обмениваются информацией о новых или потенциальных контрактах через объект третьей стороны. Торг включает в себя:

а) экспортирование — предоставление идентификатора для интерфейса, который объявляется удовлетворяющим некоторым объявленным требованиям (то есть предложение потенциального контракта);

б) импортирование — предоставление идентификатора для интерфейса, который согласуется с данными объявленными требованиями, позволяющими осуществить последующее связывающее поведение (то есть установление контракта).

13.5 Надежность

13.5.1 **Отказ** — нарушение контракта.

Примечания

1 Поведение, заданное в контракте, по определению, является «корректным поведением». Таким образом, отказ является отклонением от согласованности с корректным поведением.

2 Способы, которыми объект может отказывать, называются характером отказа. Различают несколько типов характера отказа:

- произвольные отказы (несоответствие спецификации — наиболее общий характер отказа);
- отказ пропуска (когда ожидаемые взаимодействия не происходят);
- аварийные отказы (постоянно присутствующие отказы пропуска);
- отказы по времени (некорректности связанные с несвоевременным поведением).

3 Отказ может по-разному восприниматься разными объектами. Отказ может быть: непротиворечивым, если все восприятия отказа объектами одинаковы; противоречивым, если объекты в среде могут иметь разные восприятия данного отказа.

13.5.2 **Ошибка** — часть состояния объекта, которая ответственна за возникновение отказов. Проявление неисправности (см. 13.5.3) в объекте.

Примечания

1 Приведет ли ошибка фактически к отказу, зависит от декомпозиции объекта, его внутренней избыточности и поведения объекта. Корректирующее действие может предотвратить возникновение отказа из-за ошибки.

2 Ошибка может быть скрытой (то есть не распознанной как таковая) или обнаруженной. Ошибка может исчезнуть до того, как будет обнаружена.

13.5.3 **Неисправность** — ситуация, которая может вызывать появление ошибок в объекте.

Примечания

1 Неисправности, вызывающие ошибки, могут появляться со времени спецификации объекта до времени его уничтожения. Неисправности в предшествующую эпоху (например, неисправности проекта) могут не привести к отказу в более позднюю эпоху (например, во время выполнения).

2 Неисправность является активной или ожидающей. Неисправность является активной, когда вызывает ошибки. Присутствие активных неисправностей обнаруживается только путем выявления ошибок.

3 Неисправности могут быть:

- случайными (которые появляются или создаются случайно) или преднамеренными (созданными преднамеренно);
- естественными (из-за каких-либо физических процессов) или искусственными (возникающими в результате человеческого поведения);
- внутренними (часть состояния объекта, которая может вызывать ошибки) или внешними (возникающими в результате взаимодействия со средой);
- постоянными или временными.

4 Определения неисправности, ошибки и отказа подразумевают причинную зависимость между ними:

- неисправность может привести к ошибке (она приводит к ошибке, если становится активной);
- ошибка может привести к отказу системы (ошибка приводит к отказу системы, если система не может ее обработать);
- отказ происходит тогда, когда ошибка влияет на корректность услуги, предоставляемой системой (или компонентом системы).

13.5.4 **Устойчивость** — свойство, которое имеет объект относительно данного характера отказов, если он способен не проявлять отказы этого характера.

14 Понятия управления

Управление в ОРО относится к управлению всей системой, включая прикладное управление и управление коммуникацией.

14.1 **Прикладное управление** — управление приложениями в системе ОРО. Некоторые вопросы прикладного управления являются общими для всех приложений и называются независящим от приложения управлением. Те вопросы, которые являются специфическими для данного приложения, называются специфическими для приложения управлением.

14.2 **Управление коммуникацией** — управление объектами, которые обеспечивают коммуникацию между объектами в системе ОРО.

14.3 **Управляющая информация** — относящееся к управлению знание об объектах.

14.4 **Управляемая роль** — вид управляющего интерфейса объекта, которым управляют в системе ОРО.

Примечание — Когда объект обеспечивает услуги взаимосвязи ВОС, управление ВОС относится к управляющему интерфейсу как к управляемому объекту.

14.5 **Управляющая роль** — вид объекта, который выполняет управляющие действия.

14.6 **Уведомление** — взаимодействие, инициированное объектом, играющим управляющую роль.

15 Подход ОРО к соответствию

15.1 Соответствие стандартам ОРО

Соответствие относится к реализации стандарта. Любое утверждение, справедливое в спецификации, должно быть справедливо в его реализации.

Заявка о соответствии является заявлением, идентифицирующим точки соответствия спецификации и поведение, которое должно удовлетворяться в этих точках. Заявки соответствия появляются только в стандартах, предназначенных для ограничения некоторых характеристик реализаций так, что существует, в принципе, возможность их тестирования.

БМ-ОРО идентифицирует в архитектуре некоторые опорные точки как потенциально декларируемые в спецификациях в качестве точек соответствия. А именно, как точки, в которых соответствие может быть проверено и которые, следовательно, обязательно должны быть доступными для тестирования. Однако требование, что конкретная опорная точка должна рассматриваться как точка соответствия, должно быть явно установлено в заявке о соответствии конкретной спецификации.

Требования для необходимой согласованности членов семейства стандартов ОРО друг с другом (как, например, с БМ-ОРО) устанавливаются в процессе стандартизации. Следование этим требованиям называется согласованностью.

Если спецификация согласована, прямо или косвенно, с некоторыми другими стандартами, то утверждения, которые являются справедливыми в этих стандартах, являются справедливыми также и в соответствующей реализации данной спецификации.

15.2 Тестирование и опорные точки

Справедливость заявления реализации может быть определена только путем тестирования и основывается на отображении из терминов в спецификации в наблюдаемые аспекты реализации.

На любом уровне абстракции тест является серией наблюдаемых воздействий и событий, осуществляемых в предписанных точках, называемых опорными точками, и только в этих точках. Эти опорные точки являются доступными интерфейсами. Компонент системы, для которого заявлено о соответствии, рассматривается как «черный ящик», тестируемый только через свои внешние связи. Так, например, соответствие спецификациям протоколов ВОС не зависит от внутренней структуры тестируемой системы.

15.3 Классы опорных точек

Точка соответствия является опорной точкой, в которой тест может быть сделан наблюдаемым, если система удовлетворяет набору критериев соответствия. Заявка о соответствии должна идентифицировать, где находятся точки соответствия и какие критерии удовлетворяются в этих точках. Определено четыре класса опорных точек, в которых может применяться тестирование соответствия.

15.3.1 **Программируемая опорная точка** — опорная точка, в которой для доступа к функции может быть установлен программный интерфейс. Требование программируемого соответствия устанавливается в терминах поведенческой совместимости с целью замены одного объекта другим. Программируемый интерфейс является интерфейсом, который реализуется через связь с языком программирования.

Примечание — Например, программируемая опорная точка может быть установлена в стандарте баз данных для обеспечения связи с языком на некотором уровне абстракции.

15.3.2 **Воспринимаемая опорная точка** — опорная точка, в которой имеется некоторое взаимодействие между системой и физической средой.

Примечания

1 Воспринимаемая опорная точка может быть, например, интерфейсом человек—компьютер или интерфейсом робота (заданным в терминах взаимодействий робота с физической средой).

2 Требования воспринимаемого соответствия интерфейса человек—компьютер устанавливаются в терминах вида информации, предоставляемой человеку, и диалогов, которые могут быть использованы человеком.

3 Воспринимаемая опорная точка, например, может быть установлена в графическом стандарте.

15.3.3 **Опорная точка взаимодействия** — опорная точка, в которой может быть установлен интерфейс, допускающий взаимодействие между двумя или несколькими системами. Требование соответствия взаимодействия устанавливается в терминах обмена информацией между двумя или несколькими системами. Соответствие взаимодействия включает в себя взаимодействие опорных точек.

Примечание — Например, стандарты ВОС основаны на взаимодействии опорных точек взаимодействия (физической среды).

15.3.4 **Опорная точка обмена** — опорная точка, в которой может быть введена в систему внешняя физическая среда хранения. Требование соответствия обмена устанавливается в терминах поведения (методов доступа и форматов) некоторой физической среды так, что информация может быть записана в одной системе, а затем физически перенесена, прямо или косвенно, для использования в другой системе.

Примечание — Например, некоторые стандарты по обмену информацией, основаны на опорных точках обмена.

15.4 Изменение конфигурации

Тестирование соответствия может происходить в единственной опорной точке, но может потребовать некоторого согласования при использовании ряда конфигураций, содержащих несколько опорных точек. Таким образом, может потребоваться тестирование соответствия требованию для компонента:

а) быть работоспособным после некоторого подготовительного процесса для адаптации его к локальной среде;

б) действовать после инициализации в соответствии со спецификацией в конкретной опорной точке;

в) продолжать работать при перемещении в аналогичную среду в ходе функционирования.

Свойства, тестируемые выше, проверяют атрибуты объектов или участвующие интерфейсы в отношении следующих характеристик.

15.4.1 **Переносимость** — свойство опорных точек объекта, позволяющие им адаптироваться к изменению конфигураций.

Примечание — Если опорная точка является программируемой опорной точкой, то результатом может быть переносимость исходного кода или выполнения. Если опорная точка является точкой взаимодействия, то результатом является переносимость оборудования.

15.4.2 **Перемещаемость** — способность изменять конфигурацию, заменяя одну опорную точку объекта на другую во время использования объекта.

15.5 Процесс тестирования соответствия

Соответствие является понятием, которое может применяться на любом уровне абстракции. Например, в стандартах, определяющих символьные шрифты, можно ожидать очень детальное понимание соответствия, а к правилам компоновки экрана применяется более абстрактно понимаемое соответствие.

Чем абстрактней спецификация, тем труднее ее тестировать. Для установления справедливости более абстрактных утверждений о реализации требуется увеличение доли зависящей от реализации интерпретации. Не ясно, возможно ли вообще непосредственное тестирование очень абстрактных спецификаций за разумную цену с использованием существующих или ожидаемых методов.

В процессе тестирования даются ссылки на спецификации. Полная спецификация должна содержать:

а) поведение стандартизуемого объекта и способ достижения его поведения;

б) перечень элементарных терминов, используемых в утверждениях о поведении в спецификации;

в) заявку о соответствии, указывающую точки соответствия, которые должны быть созданы в

реализации, и какую информацию должны предоставить реализаторы (согласно понятий ВОС ЗСРП и ДИРПТ).

В процессе тестирования существуют две роли: реализатор и тестер. Первая роль — реализатор, который создает реализацию на основе спецификации. Реализатор должен предоставить заявление об отображении всех терминов, используемых в спецификации, в предметы или явления реального мира. Таким образом, должны быть указаны интерфейсы, удовлетворяющие точкам соответствия, и даны представления сигналов. Если спецификация является абстрактной, то отображение ее основных терминов в реальном мире само может быть сложным. Например, в спецификации вычислительной точки зрения (см. ИСО/МЭК 10746-3) примитивные термины могут быть набором взаимодействий между объектами. Реализатор, желающий соответствовать спецификации вычислительной точки зрения, должен указать, как обеспечиваются взаимодействия, — либо ссылаясь на инженерную спецификацию, либо предоставляя подробное описание нестандартных методов (хотя этот подход ограничивает область применения реализации системами, в которых имеется соответствие использованному нестандартному методу).

Вторая роль — тестер, который наблюдает поведение тестируемой системы. Тестирование включает в себя некоторое совместное поведение процесса тестирования и тестируемой системы. Если поведение дано в причинных обозначениях, то имеется целый спектр типов тестирования от:

а) пассивного тестирования, при котором все поведение порождается тестируемой системой и регистрируется тестером, до

б) активного тестирования, при котором поведение порождается и регистрируется тестером.

Обычно спецификация тестируемой системы существует в форме интерфейса, как и спецификация тестера и процедур тестирования. Когда происходит тестирование, эти интерфейсы связываются.

Тестер должен интерпретировать свои наблюдения, используя отображение, предоставляемое реализатором, для получения утверждений о реализации, которые затем могут быть проверены для того, чтобы показать, что они справедливы и в спецификации.

15.6 Результат тестирования

Процесс тестирования завершается успешно, если все проверки на соответствие спецификации завершаются успешно. Однако он может завершиться неудачно потому, что:

а) спецификация логически несогласована или неполна, так что утверждения о реализации не могут быть проверены (этого не должно происходить);

б) отображение, данное реализатором, логически неполно, так что оно несогласовано или наблюдения не могут быть связаны с терминами спецификации; в этом случае тестирование невозможно;

в) наблюдаемое поведение не может быть интерпретировано в соответствии с отображением, данным реализатором. Поведение системы не имеет смысла в терминах спецификации; таким образом, тест завершается неудачно;

г) поведение интерпретируется в терминах спецификации, но это происходит так, что получаются утверждения, не справедливые в спецификации; таким образом, тест завершается неудачно.

15.7 Отношения между опорными точками

Поток информации между компонентами моделируемой системы может проходить через несколько опорных точек. Например, распределенная система может вовлекать во взаимодействие два компонента (А и В), но коммуникация между ними может проходить через программируемый интерфейс, точку взаимодействия и следующий программируемый интерфейс.

Уточнение той же самой системы может показать, что взаимодействующие компоненты имеют несколько других компонентов на пути коммуникации между ними.

В любом случае тестирование соответствия может включать в себя:

а) тестирование информационного потока в каждой из этих опорных точек;

б) тестирование согласованности между событиями в парах опорных точек.

В общем случае, тестирование на корректность поведения конфигурации объектов потребует тестирования того, что утверждения о коммуникационных интерфейсах являются справедливыми, а также — наблюдения других интерфейсов этих объектов, таких, что могут быть проверены утверждения о всей композиции.

Общее понятие соответствия учитывает отношение между несколькими точками соответствия. Так как спецификация, относящаяся к данной точке соответствия, может быть дана на различных уровнях абстракции, тестирование в данной точке соответствия будет включать интерпретацию на подходящем уровне абстракции. Таким образом, тестирование глобального поведения требует ско-

ординированного тестирования во всех точках соответствия и использования подходящей интерпретации в каждой точке.

В частности, соответствие шаблона данному программируемому интерфейсу может быть проверено только при рассмотрении связывающего языка, на котором написан шаблон, и соответствии написанного шаблона спецификации связывающего языка, которая сама должна согласовываться со спецификацией абстрактного интерфейса.

АЛФАВИТНЫЙ УКАЗАТЕЛЬ ТЕРМИНОВ

Абстракция	6.3
Архитектура (системы)	6.6
Базовый класс	9.21
Введение (<X>)	9.16
Внутреннее действие	8.3
Взаимодействие	8.3
Воспринимаемая опорная точка	15.3.2
Граф наименования	12.7
Группа <X>	10.1
Данные	3.2.6
Декомпозиция	9.3
Действие	8.3
Деятельность	8.5
Допустимое поведение	13.2.2
Естественная поведенческая совместимость	9.4
Завершающее поведение	13.2.5
Заглавное действие	13.1.7
Запрещение	11.2.6
Идентификатор	12.2
Изохронность	11.3.2
Имя	12.1
Именующее действие	12.5
Инвариант	9.22
Инициализирующий объект	13.3.1
Интерфейс	8.4
Информация	3.2.5
Категория	6.1
Качество услуги	11.2.2
Класс (<X>)	9.8
Класс шаблона (<X>)	9.20
Композиция	9.1
Контекст наименования	12.4
Контракт	11.2.1
Контракт среды	11.2.3
Контрактный контекст	13.2.3
Конфигурация объектов	10.2
Мобильность	15.4.1
Неисправность	13.5.3
Область <X>	10.3
Область наименования	12.6
Объединяющее действие	13.1.3
Объект	8.1
Объект-клиент	13.3.5
Объект-сервер	13.3.6

Обязательство	11.2.4
Опорная точка	10.6
Опорная точка взаимодействия	15.3.3
Опорная точка обмена	15.3.4
Отказ	13.5.1
Отвечающий объект	13.3.2
Открытая распределенная обработка	3.2.3
Ошибка	13.5.2
Перемещаемость	15.4.2
Поведение (объекта)	8.6
Поведенческая совместимость	9.4
Поддеятельность	13.1.8
Подкласс	9.10
Подобласть	10.4
Подтип	9.9
Положение во времени	8.10
Положение в пространстве	8.9
Порождающее действие	13.1.6
Постоянство	11.3.1
Постусловие	9.24
Потребляющий объект	13.3.4
Предложение	7.2
Предусловие	9.23
Предусловие связывания	13.4.3
Приведенная поведенческая совместимость	9.1
Прикладное управление	14.1
Программируемая опорная точка	15.3.1
Прозрачность распределения	11.1.2
Производный класс	9.21
Производящий объект	13.3.3
Пространство имен	12.3
Разветвляющее действие	13.1.5
Развязывающее поведение	13.4.4
Разделяющее действие	13.1.4
Разрешение	11.2.5
Разрешение имени	12.8
Распределенная обработка	3.2.1
Реализация (шаблона <X>)	9.13
Роль	9.14
Связка	13.1.2
Связь	8.8
Связывание	13.4.2
Связывающее поведение	13.4.1
Сигнатура интерфейса	9.12
Система	6.5
Система ОРО	3.2.4
Соединение	13.2.4
Создание (<X>)	9.15
Соответствие	15.1
Составной объект	9.2
Состояние (объекта)	8.7
Среда объекта	8.2
Стандарты ОРО	3.2.2
Суперкласс	9.10
Супертип	9.9

Термин	7.1
Тип (<X>)	9.7
Тип шаблона (<X>)	9.19
Торг	13.4.5
Точка взаимодействия	8.11
Точка зрения (в системе)	3.2.7
Точка соответствия	10.7
Трассировка	9.6
Уведомление	14.6
Удаление (<X>)	9.17
Управляемая роль	14.4
Управления коммуникацией	14.2
Управляющая информация	14.3
Управляющая роль	14.5
Устанавливающее поведение	13.2.1
Устойчивость	13.5.4
Утверждение	6.2
Уточнение	9.5
Цепочка (действий)	13.1.1
Шаблон <X>	9.11
Экземпляр (типа)	9.18
Элементарная	6.4
Эпоха	10.5

УДК 681.3:006.354 ОКС 35.100.70 П85 ОКСТУ 4002

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

Редактор *В.Н. Копысов*
Технический редактор *Л.А. Гусева*
Корректор *Т.И. Кононенко*
Компьютерная верстка *А.Н. Золотаревой*

Подписано в печать 23.10.2006. Формат 60x84$^1/_8$. Бумага офсетная. Гарнитура Таймс. Печать офсетная. Усл. печ. л. 3,26. Уч.-изд. л. 2,85. Тираж 33 экз. Зак. 249. С 3428.

ФГУП «Стандартинформ», 123995 Москва, Гранатный пер., 4.
gost.e-ecolog.ru
Набрано и отпечатано во ФГУП «Стандартинформ»

gost.e-ecolog.ru