ГОСУДАРСТВЕННЫЙ СТАНДАРТ РОССИЙСКОЙ ФЕДЕРАЦИИ

Информационная технология

ВЗАИМОСВЯЗЬ ОТКРЫТЫХ СИСТЕМ

СТРУКТУРА ИНФОРМАЦИИ АДМИНИСТРАТИВНОГО УПРАВЛЕНИЯ

Часть 1

Модель информации административного управления

Издание официальное

БЗ 3—2001/48

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 6 сентября 2001 г. № 376-ст

3 Настоящий стандарт содержит полный аутентичный текст международного стандарта ИСО/МЭК 10165-1—93 «Информационная технология. Взаимосвязь открытых систем. Структура информации административного управления. Часть 1. Модель информации административного управления» с учетом Изменения № 1 (1994 г.) и Дополнения № 1 (1996 г.)

4 ВВЕДЕН ВПЕРВЫЕ

© ИПК Издательство стандартов, 2001

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

Содержание

1 Область применения 1
2 Нормативные ссылки 1
3 Определения 2
3.1 Определения базовой эталонной модели 2
3.2 Определения административного управления 2
3.3 Определения административного управления системы 2
3.4 Определения услуг общей информации административного управления 2
3.5 Определения АСН.1 2
3.6 Определения руководства по определению управляемых объектов 2
3.7 Определения архитектуры безопасности 2
3.8 Дополнительные определения 3
4 Сокращения 4
5 Информационная модель 4
5.1 Понятие управляемого объекта, использующее объектно-ориентированное проектирование 5
5.2 Совместимость и взаимодействие 9
5.3 Операции системного управления 12
5.4 Фильтры 20
5.5 Сообщения 21
6 Принципы вмещения и наименования 21
6.1 Вмещение 21
6.2 Именующее дерево 22
6.3 Структура имени 23
7 Атрибуты высшего класса 26

ГОСУДАРСТВЕННЫЙ СТАНДАРТ РОССИЙСКОЙ ФЕДЕРАЦИИ

Информационная технология

ВЗАИМОСВЯЗЬ ОТКРЫТЫХ СИСТЕМ СТРУКТУРА ИНФОРМАЦИИ АДМИНИСТРАТИВНОГО УПРАВЛЕНИЯ

Часть 1

Модель информации административного управления

Information technology. Open Systems Interconnection. Structure of management information. Management Information Model

Дата введения 2002—07—01

1 Область применения

Настоящий стандарт относится к серии стандартов по услуге информации административного управления ВОС (УИУ). В стандарте определена информационная модель управляемых объектов и их атрибуты, которые соответствуют информационным аспектам модели административного управления системы, установленной в обзоре административного управления системы ГОСТ Р ИСО/МЭК 10040. Таким образом, настоящий стандарт обеспечивает модельные понятия, необходимые для разработки других стандартов по административному управлению системами. В нем также определены принципы наименования управляемых объектов и атрибутов.

Стандарт определяет логическую структуру информации административного управления системы. В соответствии с ГОСТ Р ИСО 7498-4 и ГОСТ Р ИСО/МЭК 10040 информация административного управления структурирована в терминах управляемых объектов, их атрибутов, операций управления, которые могут осуществляться над объектами, и сообщений, которые объекты могут создавать. Набор управляемых объектов в открытой системе вместе с их атрибутами образует информационную базу административного управления (ИБАУ) этой открытой системы.

В настоящем стандарте определено понятие управляемых объектов в информационной модели и установлены принципы наименований управляемых объектов и их атрибутов, которые позволяют идентифицировать эти объекты и атрибуты при доступе через протоколы административного управления. В стандарте также описаны понятия классов управляемых объектов и взаимоотношений, в которые могут вступать управляемые объектов и классы управляемых объектов, включая: наследование, специализацию, алломорфизм и вмещение.

Стандарт применяется для всех определений управляемых объектов и их атрибутов, а также для задач административного управления системами.

Примечание — Хотя настоящий стандарт применяется для административного управления системами, административное управление слоем, когда оно определено, также может использовать этот стандарт.

2 Нормативные ссылки

В настоящем стандарте использованы ссылки на следующие стандарты:

ГОСТ Р ИСО/МЭК 7498-1—99 Информационная технология. Взаимосвязь открытых систем. Базовая эталонная модель. Часть 1. Базовая эталонная модель

ГОСТ Р ИСО/МЭК 7498-2—99 Информационная технология. Взаимосвязь открытых систем. Базовая эталонная модель. Часть 2. Архитектура защиты информации

ГОСТ Р ИСО/МЭК 7498-3—97 Информационная технология. Взаимосвязь открытых систем. Базовая эталонная модель. Часть 3. Присвоение имен и адресация

ГОСТ Р ИСО/МЭК 7498-4—99 Информационная технология. Взаимосвязь открытых систем. Базовая эталонная модель. Часть 4. Основы административного управления

Издание официальное

ГОСТ Р ИСО/МЭК 8824-1—2001 Информационная технология. Абстрактная синтаксическая нотация версии 1 (АСН.1). Часть 1. Спецификация основной нотации

ГОСТ Р ИСО/МЭК 9594-1—98 Информационная технология. Взаимосвязь открытых систем. Справочник. Часть 1. Общее описание принципов, моделей и услуг

ГОСТ Р ИСО/МЭК 9595—99 Информационная технология. Взаимосвязь открытых систем. Определение общих услуг административного управления

ГОСТ Р ИСО/МЭК 10040—99 Информационная технология. Взаимосвязь открытых систем. Основные положения административного управления системы

ГОСТ Р ИСО/МЭК 10165-4—2001 Информационная технология. Взаимосвязь открытых систем. Структура информации административного управления. Часть 4. Руководство по определению управляемых объектов

ИСО/МЭК 10164-4—93* Информационная технология. Взаимосвязь открытых систем. Административное управление системы. Часть 4. Функции уведомления о нештатных ситуациях

ИСО/МЭК 10164-5—93* Информационная технология. Взаимосвязь открытых систем. Административное управление системы. Часть 5. Функции административного управления отчетностью о событиях

3 Определения

3.1 Определения базовой эталонной модели

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

- **(N)-категория;**
- **(N)-уровень;**
- **(N)-протокол;**
- **открытая система;**
- **административное управление системы.**

3.2 Определения административного управления

В настоящем стандарте используют следующие термины, определенные в ГОСТ Р ИСО/МЭК 7498-4:

- **информационная база административного управления;**
- **управляемый объект.**

3.3 Определения административного управления системы

В настоящем стандарте используют следующие термины, определенные в ГОСТ Р ИСО/МЭК 10040:

- **агент;**
- **управляющий;**
- **сообщение;**
- **класс управляемых объектов;**
- **операция (административного управления системы).**

3.4 Определения услуг общей информации административного управления

В настоящем стандарте используют следующие термины, определенные в ГОСТ Р ИСО/МЭК 9595:

- **атрибут;**
- **атрибут набора значений.**

3.5 Определения АСН.1

В настоящем стандарте используют следующий термин, определенный в ГОСТ Р ИСО/МЭК 8824-1: **тип.**

3.6 Определения руководства по определению управляемых объектов

В настоящем стандарте используют следующий термин, определенный в ИСО/МЭК 10164-4: **шаблон.**

3.7 Определения архитектуры безопасности

В настоящем стандарте используют следующие термины, определенные в ГОСТ Р ИСО/МЭК 7498-2:

- **управление доступом;**
- **политика безопасности.**

---

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

3.8 Дополнительные определения

3.8.1 **действие:** Операция над управляемым объектов, семантика которой определена как часть определения класса управляемых объектов.

3.8.2 **фактический класс:** Класс управляемых объектов, экземпляром которого является данный управляемый объект, в отличие от алломорфного класса этого управляемого объекта.

3.8.3 **алломорфный класс** (управляемого объекта): Класс, отличный от фактического класса управляемого объекта, которым может управляться объект, используя алломорфизм.

3.8.4 **алломорфизм:** Способность объекта, который является экземпляром данного класса, быть управляемым как экземпляр одного или нескольких других классов управляемых объектов.

3.8.5 **атрибутивная группа:** Группа атрибутов, которой был присвоен один идентификатор для облегчения доступа.

3.8.6 **идентификатор атрибута:** Идентификатор, используемый для отличия атрибута класса управляемых объектов от всех других атрибутов.

3.8.7 **тип атрибутов:** Поименованное определение атрибутов конкретного вида, включающее в себя определения его синтаксиса (типа) и семантики. Атрибут является экземпляром типа атрибутов.

3.8.8 **утверждение о значении атрибута:** Утверждение, которое может быть истинным или ложным в зависимости от значения атрибута.

3.8.9 **множество значений атрибута:** Множество значений, члены которого являются допустимыми значениями атрибута.

3.8.10 **поведение:** Способ, которым определенные элементы информации административного управления связаны с моделируемыми ими ресурсами и друг с другом.

3.8.11 **характеристика:** Элемент определения класса.

3.8.12 **условный пакет:** Пакет, который присутствует в данном управляемом объекте, если удовлетворяется условие, заданное в определении его класса управляемых объектов.

3.8.13 **вмещение:** Структурированное взаимоотношение между управляемыми объектами, при котором существование управляемого объекта зависит от существования вмещающего управляемого объекта.

3.8.14 **отличающее имя:** Имя объекта, образованное последовательностью ООИ самого объекта и всех старших для него объектов.

3.8.15 **инкапсуляция:** Взаимоотношение между управляемым объектом, его атрибутами и поведением, представляющее свойство, при котором атрибуты и поведение могут наблюдаться только через операции управления над управляемым объектом или через его сообщения.

3.8.16 **наследование:** Концептуальный метод, с помощью которого характеристики заимствуются подклассом у его суперкласса.

3.8.17 **иерархия наследования:** Иерархическое упорядочение сходных классов, при котором иерархия строится на основе специализации классов.

3.8.18 **управляемый объект начальных значений:** Управляемый объект, служащий в качестве источника для получения начальных значений других управляемых объектов.

3.8.19 **реализация:** Процесс создания управляемого объекта в соответствии с определением класса управляемых объектов.

3.8.20 **инвариант:** Логический предикат, который должен оставаться истинным в заданной области действия.

3.8.21 **граница управляемого объекта:** Концептуальное место, в котором вопросы, относящиеся к нижележащим ресурсам, становятся видимыми для административного управления и которое ограничивает область определения управляемого объекта.

3.8.22 **обязательный пакет:** Пакет, который должен присутствовать во всех экземплярах данного класса управляемых объектов.

3.8.23 **кратное наследование:** Концептуальный метод, позволяющий подклассу заимствовать характеристики от нескольких похожих суперклассов.

3.8.24 **связывание имен:** Взаимоотношение между классами объектов, которое задает, что объект одного идентифицированного класса может быть старшим для объекта другого названного класса. Определение связывания имен включает в себя и другую информацию об этом отношении и может быть определено таким образом, чтобы также применять к подклассам старшего или подчиненного класса, или к тем и другим.

3.8.25 **именующая схема:** Совокупность связываний имен.

3.8.26 **именующее дерево:** Иерархическое упорядочение объектов, когда иерархия строится на основе отношения связывания имен. Объект, используемый для наименования другого управляемого

объекта, является более старшим в иерархии, чем именуемый объект. Именующий объект называется старшим именуемого объекта, который называется подчиненным.

3.8.27 **пакет:** Совокупность атрибутов, сообщений, операций и (или) поведения, которая рассматривается в спецификации класса управляемых объектов как единый модуль. При указании в определении класса управляемых объектов пакеты могут быть специфицированы как обязательные или условные.

3.8.28 **параметр:** Значение некоторого типа, который имеет соответствующую семантику и ассоциируется с идентификатором объекта и другой информацией, когда значение типа может передаваться в протоколе.

3.8.29 **множество допустимых значений:** Множество значений атрибута, включающее в себя все значения, которые разрешается принимать атрибуту заданного типа.

3.8.30 **предусловие:** Логический предикат, который должен быть истинным непосредственно перед выполнением операции или перед созданием сообщения.

3.8.31 **постусловие:** Логический предикат, который должен быть истинным непосредственно после выполнения операции или непосредственно после создания сообщения.

3.8.32 **относительное отличающее имя:** Утверждение о значении атрибута, согласно которому конкретный атрибут имеет конкретное значение, используемое для идентификации одного объекта среди всех непосредственно подчиненных данному объекту. Оно служит в качестве компонента отличающего имени объекта.

3.8.33 **множество требуемых значений:** Множество значений атрибута, включающее в себя все значения, которые должен принимать атрибут заданного типа.

3.8.34 **специализация:** Техника получения нового класса из одного или нескольких существующих похожих классов путем наследования и добавления новых характеристик.

3.8.35 **подкласс:** Класс, полученный из другого путем специализации.

3.8.36 **суперкласс:** Класс, используемый при выводе другого класса путем специализации.

3.8.37 **старший объект:** См. 3.8.26.

3.8.38 **подчиненный объект:** См. 3.8.26.

3.8.39 **нереализуемый класс управляемых объектов:** Класс, который не предназначен для реализации ни операцией административного управления системы, ни локальной операцией открытой системы.

Примечания.

1 Термины: атрибут, утверждение о значении атрибута, относительное отличающее имя, отличающее имя, используют в справочнике (ГОСТ Р ИСО/МЭК 9594-1) и настоящем стандарте в аналогичном смысле для того, чтобы подчеркнуть сходство между моделью справочника и моделью информации административного управления. Однако использование этих терминов в двух моделях не идентично в деталях.

2 Термин «класс» используют, когда не подразумевается класс какого-либо специфического вида. Термин «класс» может относиться к классу управляемых объектов или классам некоторых других видов (например классу управляемых взаимосвязей). Термин «похожие (супер)классы» означает (супер)классы одного вида.

4 Сокращения

В настоящем стандарте используют следующие сокращения:

АСН.1 — абстрактная синтаксическая нотация версии 1;

ВОС — взаимосвязь открытых систем;

ИБАУ — информационная база административного управления;

ООИ — относительное отличающее имя;

ПОИУ — протокол общей информации (административного) управления;

СИУ — структура информации (административного) управления;

УЗА — утверждение о значении атрибута;

УИУ — услуги информации (административного) управления;

УОИУ — услуги общей информации (административного) управления;

УОНЗ — управляемый объект начальных значений.

5 Информационная модель

Цель информационной модели — дать структуру управляющей информации, переносимой протоколами управления системами, и структуру модели управления соответствующими ресурсами (например протокольным автоматом Х.25). Информационная модель имеет дело с управляемыми

объектами. Управляемые объекты являются абстракциями ресурсов обработки и передачи данных (например протокольных автоматов, соединений, модемов) с целью административного управления. Ресурсы существуют независимо от необходимости ими управлять. Взаимоотношение, существующее между ресурсом и управляемым объектом как абстракцией этого ресурса, в общем случае не моделируется, т. е. мы абстрагируемся от подробностей свойств ресурсов, а конкретное влияние операций управления на ресурс должно быть установлено как часть спецификации класса управляемых объектов.

Различие между управляемым объектом, как он виден процессу управления, и ресурсом, представляемым объектом в целях управления, может быть описано так, что атрибуты, операции и сообщения видны управлению на границе управляемого объекта, а внутреннее функционирование ресурса, представленного управляемым объектом, не видно управлению каким-либо иным образом. Это понятие границы управляемого объекта не относится к реализации, но обеспечивает архитектурное различие между определениями, разрабатываемыми для классов управляемых объектов, которые относятся к тому, что находится на границе управляемого объекта и внутри ее, и определениями и международными стандартами для остальной части административного управления, относящимися к тому, что находится вне границы.

Класс управляемых объектов определяется как совокупность пакетов, каждый из которых, по определению, является совокупностью атрибутов, операций, сообщений и соответствующего поведения. Пакеты являются обязательными или условными, зависящими от явно установленных условий. Управляемый объект является экземпляром класса управляемых объектов.

Для того чтобы задокументировать спецификацию класса управляемых объектов и соответствующие характеристики, используется набор шаблонов. Шаблоны используются для системного управления, определенного в ГОСТ Р ИСО/МЭК 10165-4.

Определение класса управляемых объектов, как установлено шаблонами, состоит из:

- положения класса управляемых объектов в иерархии наследования;
- совокупности обязательных пакетов атрибутов, операций, сообщений и поведения;
- совокупности условных пакетов атрибутов, операций, сообщений и поведения вместе с условиями, при которых будет присутствовать каждый пакет;
- в структуре пакета:

  атрибуты, видимые на границе управляемого объекта;

  операции, которые могут применяться к управляемому объекту;

  поведение, демонстрируемое управляемым объектом;

  сообщения, которые может делать управляемый объект.

Другие шаблоны устанавливают возможные старшие объекты для экземпляров данного класса управляемых объектов и атрибуты, используемые для наименования (см. раздел 6) в таких случаях.

Прочие аспекты ресурсов, представленных классом управляемых объектов, не видны системному управлению.

Управляемый объект реализуется по некоторому набору правил. Эти правила задают, как спецификация класса, определенная с помощью шаблона, должна быть реализована при создании управляемого объекта. Эти правила устанавливают, что:

а) управляемый объект должен поддерживать все атрибуты, операции управления, поведение и сообщения, специфицированные во всех обязательных пакетах и во всех условных пакетах, условия которых удовлетворены;

б) управляемый объект должен поддерживать связывание имен, специфицированное соответствующим шаблоном, с которым он реализуется. Реализация будет неудачной, если запрошено неподдерживаемое связывание имен.

Каждый управляемый объект является экземпляром класса, который включает в себя все управляемые объекты, использующие одно и то же определение. Для недвусмысленного наименования каждого управляемого объекта используют отличающие имена.

С точки зрения управления, управляемый объект существует, если он имеет отличающее имя (как определено в 6.3.2) и поддерживает операции и сообщения, определенные для его класса. В противном случае, он не существует с точки зрения управления, даже если существует физический двойник.

5.1 Понятие управляемого объекта, использующее объектно-ориентированное проектирование

В формулировках международных стандартов системного управления идентифицируется потребность в добавлении новых классов управляемых объектов и функций. Следовательно, проектирование системного управления требует, чтобы был принят подход, позволяющий проводить

международную стандартизацию модульным образом и обеспечивающий расширение протоколов и процедур. Информационная модель использует принципы объектно-ориентированного проектирования потому, что они предоставляют указанные выше возможности и позволяют использовать отдельные части спецификаций.

В информационной модели объектно-ориентированное проектирование применяется для спецификации управляющей информации, как она видна в обменных протоколах открытыми системами, участвующими в деятельности по административному управлению. Она не обязательно должна применяться при реализации системы.

Объектно-ориентированное проектирование характеризуется определением объектов, в котором объект является абстракцией физического или логического предмета.

Примечание — Термин «объект» используют в настоящем стандарте при ссылке на объекты в более широком контексте, чем административное управление ВОС. Термин «управляемый объект» используют для ссылки на объекты, представляющие, в целях управления, ресурсы.

5.1.1 Инкапсуляция

Одной из сторон объектно-ориентированного проектирования является инкапсуляция. Инкапсуляция гарантирует, что сохраняется целостность объекта. Она требует, чтобы все операции, которые должны быть осуществлены, выполнялись путем отправки объекту «сообщения». Таким образом, внутренние операции управляемого объекта не видны на границе объекта, если атрибуты, операции или сообщения не определены так, чтобы показывать эту информацию. Определение класса управляемых объектов задает, какие операции могут осуществляться и какие ограничения согласованности требуются для обеспечения целостности управляемого объекта.

5.1.2 Классы управляемых объектов и их характеристики

Управляемые объекты, которые совместно используют одно и то же определение, являются экземплярами одного и того же класса управляемых объектов. Характеристики класса управляемых объектов включают в себя атрибуты, атрибутивные группы, действия, сообщения, поведение и пакеты. Различные экземпляры данного класса будут совместно использовать атрибуты, операции, сообщения и поведение, определенные в обязательных пакетах класса, а определенные в условных пакетах — в той степени, в какой экземпляры удовлетворяют условиям, связанным с этими пакетами.

5.1.2.1 *Пакеты*

Пакет является совокупностью атрибутов, атрибутивных групп, действий, сообщений и поведения, представляющей единый модуль определения класса управляемых объектов. При указании в определении управляемого объекта пакеты специфицируются как обязательные или условные. Обязательный пакет должен присутствовать во всех экземплярах данного класса управляемых объектов. Условный пакет является таким пакетом, который должен присутствовать в управляемом объекте, для которого явное условие, ассоциированное с этим пакетом в определении класса управляемых объектов, истинно. Одна и та же характеристика может присутствовать в нескольких пакетах. Условие, при котором пакет присутствует, относится либо к возможностям нижележащего ресурса, моделируемого управляемым объектом, либо к наличию или отсутствию управляющих функций, поддерживаемых управляемой системой. В случае стандартных управляемых объектов ВОС (например протокольного автомата транспортного уровня) эти пакеты будут моделировать опции, заданные как часть соответствующей спецификации.

Пакеты имеют следующие свойства:

а) в управляемом объекте может существовать только один экземпляр данного пакета;

б) так как в управляемом объекте может существовать только один экземпляр данного пакета, то пакетам не присваиваются связывания имен;

в) будучи инкапсулированными в управляемом объекте, атрибуты, операции, сообщения и поведение становятся интегральной частью управляемого объекта и доступны только как часть этого управляемого объекта;

г) пакет не может быть реализован без управляемого объекта, в котором он инкапсулирован;

д) пакет должен быть реализован в то же самое время, что и управляемый объект; реализация пакета в более позднее время не допустима;

е) пакеты должны удаляться одновременно с управляемым объектом; удаление пакета в более раннее время не допустимо;

ж) операции всегда осуществляются над управляемыми объектами, а не над пакетами.

Так как не все управляемые объекты данного класса будут включать в себя все допустимые условные пакеты, определенные для этого класса управляемых объектов, то зарегистрированные

пакеты, поддерживаемые управляемым объектом, идентифицируются в атрибуте Packages управляемого объекта (см. раздел 7).

5.1.2.2 *Атрибуты*

Управляемые объекты имеют атрибуты. Атрибут имеет ассоциированное значение, которое может иметь структуру, т. е. оно может состоять из набора или последовательности элементов. Утверждение о значении атрибута (УЗА) является истинным или ложным утверждением, в зависимости от значения атрибута.

Значение атрибута может быть наблюдаемым (на границе управляемого объекта). Значение атрибута может определять или отражать поведение управляемого объекта. Значение атрибута наблюдается или изменяется путем отправки управляемому объекту запроса прочитать (вернуть) или записать (изменить) значение. Для многозначных атрибутов определены дополнительные операции; это те атрибуты, значения которых являются наборами элементов, каждый из которых — одного и того же типа данных. Операции над атрибутами определены как осуществляющиеся над содержащим их управляемым объектом, а не непосредственно над атрибутами. Управляемый объект может наложить ограничения на значения атрибутов для обеспечения внутренней согласованности. В определении класса управляемых объектов могут быть специфицированы ограничения на значения отдельных атрибутов. Операции, которые могут осуществляться над конкретным атрибутом, специфицированы в определении класса управляемых объектов.

Определяемые атрибуты должны быть в пакетах, обязательных или условных. Следовательно, атрибуты, определяемые как часть обязательных пакетов, присутствуют во всех экземплярах класса управляемых объектов, а определяемые как часть условных пакетов — в тех экземплярах, которые удовлетворяют условиям, связанным с пакетом.

5.1.2.2.1 Множества значений атрибутов

Синтаксис атрибута есть тип АСН.1, который описывает, как экземпляры значения атрибута передаются в протоколе. Он наследуется атрибутом и остается неизменным для всех использований атрибута.

В спецификации класса управляемых объектов свойства атрибута уточняются далее в терминах множества допустимых значений и множества требуемых значений. Эти множества специфицируют ограничения на значения атрибута.

Множество требуемых значений специфицирует все значения, которые атрибут обязательно может принимать. Оно может быть пустым, если никаких конкретных значений не требуется.

Управляемый объект должен быть способен заменить значение атрибута одним из значений из множества требуемых значений в зависимости от ограничений на поведение или других ограничений, как, например, управление доступом.

Множество допустимых значений специфицирует возможные значения, которые разрешено принимать атрибуту.

Управляемый объект не сможет вернуть в ответ на операцию, требующую от управляемого объекта прочитать значение атрибута, значение, не входящее в множество допустимых значений. Управляемый объект должен отвергнуть запрос изменить значение атрибута на значение, не входящее в множество допустимых значений.

Множество допустимых значений должно быть подмножеством значений синтаксиса, а множество требуемых значений — подмножеством множества допустимых значений, причем в обоих случаях допускается идентичность.

5.1.2.2.2 Многозначные атрибуты

Многозначный атрибут — атрибут, значение которого является неупорядоченным набором членов данного типа. Размер набора — переменный; набор может быть пустым. Частью определения многозначного атрибута являются допустимые и требуемые значения кардинального числа набора. В дополнение к операциям, доступным для атрибутов всех типов, для многозначных атрибутов определяются операции, позволяющие добавлять и удалять отдельные элементы таких атрибутов.

5.1.2.3 *Атрибутивные группы*

Атрибутивная группа предоставляет способы ссылки на совокупность атрибутов в управляемом объекте. Могут быть определены два типа атрибутивных групп: фиксированные и расширяемые. Возможность или невозможность расширения указывается в определении атрибутивной группы.

Фиксированная атрибутивная группа — атрибутивная группа, набор атрибутов которой определяется как часть исходного определения атрибутивной группы и не может быть изменен никоим образом. Для фиксированных атрибутивных групп все входящие в них атрибуты должны быть определены в том же самом пакете, что и атрибутивная группа.

Расширяемая атрибутивная группа — атрибутивная группа, к которой могут быть добавлены атрибуты в результате специализации. Для расширяемых атрибутивных групп атрибуты, задаваемые для каждого расширения, должны быть определены либо в том же самом условном пакете, что и атрибутивная группа, либо в обязательном пакете.

Отдельные атрибуты, образующие атрибутивную группу, специфицируются в определении класса управляемых объектов. Атрибутивная группа сама значений не имеет. Над атрибутивными группами допустимы только такие операции, для которых не требуется задание значения.

Допустимые операции над атрибутивной группой интерпретируются как указания на соответствующие операции над каждым отдельным атрибутом, включенным в атрибутивную группу. Операция применяется к атрибутам в произвольном порядке.

Класс управляемых объектов может иметь несколько атрибутивных групп. Один атрибут может входить в несколько атрибутивных групп.

5.1.2.4 *Поведение*

Частью определения класса управляемых объектов является поведение.

Поведение может определять:

а) семантику атрибутов, операций и сообщений;

б) ответы на операции управления, осуществляемые над управляемым объектом;

в) обстоятельства, при которых могут быть переданы сообщения;

г) зависимости между значениями конкретных атрибутов, которые могут быть выражены таким образом, чтобы учитывать возможное наличие или отсутствие условных пакетов;

д) влияние взаимосвязей на участвующие в них управляемые объекты;

е) ограничения согласованности на атрибуты;

ж) предусловия, идентифицирующие условия, при которых может быть принято, что операции и сообщения должны иметь допустимый смысл;

и) постусловия, идентифицирующие результаты обработки операции управления или передачи сообщения;

к) инварианты, которые остаются справедливыми в течение всей жизни управляемого объекта и описывают условия, являющиеся истинными для операции управляемого объекта;

л) свойства синхронизации управляемого объекта.

В ГОСТ Р ИСО/МЭК 10165-4 определен набор шаблонов, которые могут быть использованы для определения всех аспектов поведения управляемого объекта.

5.1.3 Специализация и наследование

Один класс управляемых объектов специализируется из другого класса тех же объектов путем определения первого как расширения второго. Такое расширение достигается определением новых пакетов, которые могут включать в себя:

- новые операции управления;
- новые атрибуты;
- новые сообщения;
- новое поведение;
- расширения характеристик исходного класса управляемых объектов.

Способы, которыми могут быть расширены возможности данного класса управляемых объектов, подробно специфицированы в 5.2.2.

Класс управляемых объектов, который специализируется из управляемых объектов другого класса, называется подклассом этого класса (его суперкласса). Один класс управляемых объектов, называемый высшим, спроектирован как самый старший суперкласс в иерархии классов. Высший класс является нереализуемым классом управляемых объектов.

Подкласс наследует операции, атрибуты, сообщения, пакеты и поведение суперкласса. Настоящий стандарт допускает только строгое наследование характеристик, т. е. каждый экземпляр подкласса совместим с суперклассом в соответствии с правилами, определенными в 5.2.2. Специализация путем удаления каких-либо характеристик суперкласса не допускается.

Кратное наследование — возможность специализировать подкласс из нескольких суперклассов. Подкласс наследует операции, атрибуты, сообщения, пакеты и поведение от нескольких суперклассов.

Когда класс имеет кратное наследование одной и той же характеристики из нескольких суперклассов, этот класс определяется так, как если бы эта характеристика наследовалась только от одного суперкласса. Специализация не должна вносить противоречия в определение подкласса.

Пример иерархии наследования, как она может применяться для административного управления, показан на рисунке 1.

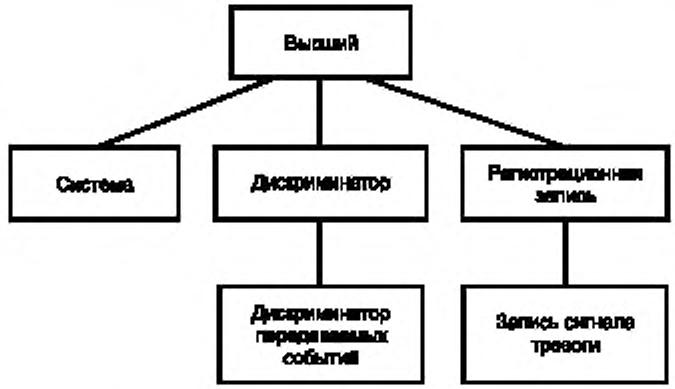

Рисунок 1 — Пример иерархии наследования

5.2 Совместимость и взаимодействие

5.2.1 Требования

Существуют требования для возможности взаимодействия между управляющей и управляемой системами. Также имеются требования по обеспечению взаимодействия в случаях, когда либо увеличивается управляемая система, либо расширяются одно или несколько определений управляемых объектов.

Конкретные требования взаимодействия системного управления для данного управляемого объекта следующие:

а) должна быть возможность управлять системой из другой системы с таким же знанием определения класса управляемых объектов данного управляемого объекта;

б) у системы должна быть возможность управлять другой системой с меньшим знанием определения класса управляемых объектов данного управляемого объекта;

в) для возможного расширения у системы должна быть возможность управлять системой с большим знанием определения класса управляемых объектов данного управляемого объекта. В частности, требуется, чтобы в случаях, когда расширенные возможности не нужны, управление должно осуществляться столь же эффективно, как если бы управляемая система их не имела.

5.2.2 Правила для совместимости

В настоящем подразделе определен набор правил, который гарантирует, что управляемый объект, являющийся экземпляром одного класса управляемых объектов (называемый расширенным управляемым объектом), совместим с определением второго класса управляемых объектов (называемых классом совместимых управляемых объектов). Эти классы управляемых объектов не обязательно связаны наследованием.

Эти правила определены в целях:

- использования в определениях строгого наследования (см. 5.1.3);
- использования в методах взаимодействия.

5.2.2.1 *Дополнительные характеристики*

Расширенный управляемый объект должен включать в себя все атрибуты, атрибутивные группы, операции управления и сообщения, которые должны были бы присутствовать в экземпляре класса совместимых управляемых объектов, реализованного при тех же самых условиях. В расширенный управляемый объект могут быть включены дополнительные атрибуты, атрибутивные группы, операции управления и сообщения.

Обязательные пакеты, реализованные в расширенном управляемом объекте, и обязательные пакеты, определенные в классе совместимых управляемых объектов, не обязательно должны быть родственными при условии, что удовлетворяются приведенные выше правила для всех характеристик в этих обязательных пакетах.

Расширенный управляемый объект должен включать в себя все условные пакеты, определенные для класса совместимых управляемых объектов, для которых удовлетворены условия их наличия для расширенного управляемого объекта.

5.2.2.2 *Условия пакетов*

Во всех случаях, когда условие наличия условного пакета в классе совместимых управляемых объектов является истинным, должно быть удовлетворено и условие для того же самого условного пакета в расширенном управляемом объекте. Это правило допускает, чтобы условный пакет в классе совместимых управляемых объектов был обязательным в расширенном управляемом объекте.

5.2.2.3 *Ограничения на значения атрибутов*

Имеются ограничения на значения, которые могут принимать атрибуты, общие для расширенного управляемого объекта и класса совместимых управляемых объектов. Общее условие для каждого такого атрибута состоит в том, что множество требуемых значений, определенное в классе совместимых управляемых объектов, должно быть подмножеством множества значений, поддерживаемого расширенным управляемым объектом, которое, в свою очередь, должно быть подмножеством множества допустимых значений, определенного в классе совместимых управляемых объектов (с возможным равенством в обоих случаях). Таким образом, расширенный управляемый объект поддерживает все значения, обязательные для класса совместимых управляемых объектов, и не поддерживает значения, не допустимые для этого класса.

К нечитаемым атрибутам относятся только условия для множеств требуемых значений. С другой стороны, к атрибутам, защищенным от записи, относятся только условия для множеств допустимых значений.

Примечание — Совместимость не гарантирует, что начальные и умалчиваемые значения, определенные в классе совместимых управляемых объектов, будут использоваться расширенным управляемым объектом.

5.2.2.4 *Ограничения на атрибутивные группы*

Расширенная атрибутивная группа в расширенном управляемом объекте должна включать в себя все атрибуты, которые должны присутствовать в соответствии с определением той же самой атрибутивной группы в классе совместимых управляемых объектов и согласно условиям для пакетов, содержащих эти атрибуты.

5.2.2.5 *Ограничения на наличие параметров действий и сообщений*

Для параметров действий и сообщений должны применяться следующие условия.

а) Параметры действия

Для данного действия, общего как для класса совместимых управляемых объектов, так и для расширенного управляемого объекта:

- все параметры действия, определенные в классе совместимых управляемых объектов, должны поддерживаться расширенным управляемым объектом;
- факультативные параметры, которые не определены для класса совместимых управляемых объектов, должны поддерживаться расширенным управляемым объектом, если только определение действия в классе совместимых управляемых объектов допускает дополнительные факультативные параметры;
- параметры, которые требуются расширенным управляемым объектом для осуществления действия и которые не определены в классе совместимых управляемых объектов, должны поддерживаться, если только в классе расширенного управляемого объекта определены умолчания на случай их отсутствия;
- все ответные параметры, определенные в классе совместимых управляемых объектов, должны поддерживаться расширенным управляемым объектом;
- ответные параметры, которые не определены для класса совместимых управляемых объектов, должны поддерживаться расширенным управляемым объектом, если только определение ответа в классе совместимых управляемых объектов допускает дополнительные факультативные параметры.

б) Параметры сообщения

Для данного сообщения, общего как для класса совместимых управляемых объектов, так и для расширенного управляемого объекта:

- все параметры сообщения, определенные в классе совместимых управляемых объектов, должны поддерживаться расширенным управляемым объектом;
- параметры, которые не определены для класса совместимых управляемых объектов, должны поддерживаться расширенным управляемым объектом, если только определение сообщения в классе совместимых управляемых объектов допускает дополнительные параметры.

5.2.2.6 *Расширения определений поведения*

Правило для расширения определения поведения состоит в том, что определение поведения для класса расширенного управляемого объекта не должно противоречить поведению класса совместимых управляемых объектов.

Для того чтобы отчасти обеспечить отсутствие такого противоречия, к поведению расширенного управляемого объекта применяются следующие правила:

- расширенный управляемый объект должен включать в себя все инварианты класса совместимых управляемых объектов;
- предусловия в расширенном управляемом объекте должны быть дизъюнктивной комбинацией предусловий, определенных в классе совместимых управляемых объектов, и любых новых предусловий, которые применяются к расширенному управляемому объекту;
- постусловия в расширенном управляемом объекте должны быть конъюнктивной комбинацией постусловий, определенных в классе совместимых управляемых объектов, и любых новых постусловий, которые применяются к расширенному управляемому объекту.

5.2.3 Методы обеспечения взаимодействия

В 5.2.3.1 и 5.2.3.2 описаны два метода обеспечения взаимодействия. Они различаются главным образом тем, предоставляет ли дополнительные возможности система-агент или управляющая система.

5.2.3.1 *Взаимодействие, обеспечиваемое системой-агентом*

Алломорфизм является способностью управляемого объекта, являющегося экземпляром данного класса, быть управляемым в качестве члена одного или нескольких других классов управляемых объектов, когда эта возможность обеспечивается системой-агентом.

5.2.3.1.1 Алломорфизм для экземпляров управляемых объектов

Алломорфизм является свойством управляемого объекта. Управляемым объектом, который поддерживает алломорфизм, можно управлять так, как если бы он был экземпляром управляемых объектов другого класса и был совместим с классом управляемых объектов, как определено в 5.2.2. Классы, которые могут управлять управляемым объектом, называются его алломорфными классами.

Алломорфный класс данного управляемого объекта может быть в иерархии наследования одним из суперклассов класса этого управляемого объекта. Однако это не обязательно для алломорфизма.

5.2.3.1.2 Определение алломорфного класса для операций

Управляемым объектом, поддерживающим алломорфизм, можно управлять так, как если бы он относился к любому из его алломорфных классов. Класс, который управляемый объект требует в качестве алломорфного, становится доступным управляемому объекту на его границе.

Управляемый объект отвечает в соответствии со следующими общими принципами за то, что:

- операция проверяется на допустимость в алломорфном классе;
- операция осуществляется в соответствии с поведением фактического класса;
- ответы генерируются в соответствии с описанием алломорфного поведения в 5.3.

5.2.3.1.3 Определение алломорфного класса для сообщений

Когда управляемый объект, поддерживающий алломорфизм, создает сообщение, он делает доступным на своей границе набор алломорфных классов, для которых сообщение определено, в сочетании с самим сообщением. Если сообщение выбрано для передачи, система-агент определяет, какой из алломорфных классов (или фактический класс) должен быть включен в отчет о событии для управляющего.

Информационное содержимое отчета о событии будет точно таким, как определено в классе управляемых объектов для управляемого объекта, создавшего сообщение, т. е. оно не изменяется последующим алломорфизмом.

5.2.3.2 *Взаимодействие, обеспечиваемое управляющей системой*

При таком подходе управляемому объекту разрешается отвечать в соответствии с определением его фактического класса. Управляющая система требуется для обработки любой дополнительной информации, которая непонятна или неожиданна (например путем ее игнорирования).

Правила совместимости 5.2.2 не обязательно должны выполняться для достижения некоторого уровня взаимодействия. Однако при выполнении правил совместимости взаимодействие может быть расширено. В частности, взаимодействие легче достигается, если управляющему известно определение класса, совместимого с управляемым объектом, которым он управляет. Если управляющий не знает о совместимом классе, то возможно ограниченное взаимодействие, когда некоторые из характеристик участвующих объектов удовлетворяют правилам совместимости.

В ГОСТ Р ИСО/МЭК 10165-4 определен специфический идентификатор объекта для использования в протоколах в качестве идентификатора класса управляемых объектов, семантика которого состоит в указании фактического класса управляемых объектов. Использование этого идентификатора позволяет управляющему без спецификации фактического класса управляемых объектов запросить, какая операция управления должна быть выполнена.

5.3 Операции системного управления

Определено два вида операций управления: те, которые могут быть переданы управляемому объекту для применения к его атрибутам, и те, которые применяются к управляемому объекту в целом. Определенные здесь операции являются примитивными, видимыми на границе управляемого объекта. Операция, осуществляемая над управляемым объектом, может быть успешной только в том случае, если вызывающая управляющая система имеет права доступа, достаточные для осуществления этой операции, а ограничения согласованности не нарушаются.

Примерами таких ограничений согласованности являются взаимоотношения, которые должны поддерживаться между значениями атрибутов. Ограничения согласованности специфицируются как часть определения поведения атрибутов или определения класса управляемых объектов. Когда выполнение операции, например изменение значения атрибута, нарушило бы заданное ограничение, операция не выполняется и возвращается указание «отказ в процессе обработки».

Указание ошибки может сопровождаться параметрами «специфическая ошибка», определенными для класса управляемых объектов.

Одни операции являются подтверждаемыми, т. е. они требуют, чтобы вызвавшему операцию был передан ответ, указывающий на успех или неудачу операции; другие операции являются неподтверждаемыми, т. е. они не требуют, чтобы вызвавшему операцию был передан ответ. Относительно использования подтверждений могут быть идентифицированы две категории операций: те операции, для которых подтверждение является внутренне присущим, и специфическая информация (если она есть) требуется как часть результата (например получение значений атрибутов и создание), и те, для которых подтверждение может быть факультативно выбрано управляющей системой в соответствии с политикой управления (например замена значения атрибута). Категория, к которой относится каждая операция, специфицируется в разделе конкретных операций.

5.3.1 Управление доступом к информации административного управления

Операции административного управления могут быть предметом управления доступом в зависимости от политики безопасности.

5.3.2 Элементарная синхронизация операций административного управления

От управляемой системы может быть запрошено осуществление операций над несколькими управляемыми объектами с элементарной синхронизацией; т. е. так, что либо все операции должны завершиться успешно, либо, если это невозможно, они не должны выполняться. Элементарная синхронизация не применяется для операции создания Create. Когда действует элементарная синхронизация, промежуточные состояния, получающиеся в ходе операции, не видны другим операциям административного управления.

Для целей элементарной синхронизации требуется определение успешного завершения для каждой операции административного управления. Определение успешного завершения для операций, направленных на атрибуты, состоит в том, что операция осуществлена успешно для всех атрибутов, заданных в списке. Для операции удаления Delete определение успешного завершения состоит в том, что управляемый объект удален. Для операции действия Action определение успешного завершения состоит в том, что действие осуществлено без индикации ошибок.

Конкретное обеспечение межобъектной синхронизации в конкретной открытой системе является локальным вопросом этой системы. Когда синхронизация для конкретных управляемых объектов не поддерживается, операция заканчивается неудачей, если запрошена элементарная синхронизация.

5.3.3 Операции, направленные на атрибуты

Управляемому объекту могут быть переданы следующие операции административного управления для применения к его атрибутам:

- получить значение атрибута;
- заменить значение атрибута;
- заменить значением по умолчанию;

- добавить член множества значений;
- удалить член множества значений.

5.3.3.1 *Поведение, общее для всех операций, направленных на атрибуты*

В данном пункте определены аспекты поведения, которые являются общими для всех операций, направленных на атрибуты.

Операции, которые применяются к атрибутам, инкапсулированным в управляемом объекте, всегда действуют на список атрибутов, т. е. считается, что все атрибуты, к которым должна быть применена запрошенная операция, доступны управляемому объекту как часть единственной операции.

Управляемому объекту доступна следующая информация для определения, должна ли и если должна, то как выполняться операция, направленная на идентификаторы атрибутов и связанные с ними операторы сравнения и значения, которые использовались при определении объектов, выбираемых для выполнения операции (см. 5.4).

На границе управляемого объекта в результате операции, направленной на атрибуты, доступна следующая информация:

- идентификаторы атрибутов и связанные с ними значения для атрибутов, над которыми могла быть выполнена операция;
- указания ошибок для атрибутов, над которыми операция не могла быть выполнена.

Различают следующие указания ошибок:

- неизвестные идентификаторы атрибутов;
- запрошенный класс управляемых объектов не является ни фактическим классом управляемого объекта, ни одним из его алломорфных классов;
- отказ при обработке запроса, с факультативным параметром «специфическая ошибка».

Примечание — Способы, при которых информация становится доступной управляемому объекту и предоставляется управляемым объектом, не являются предметом стандартизации.

Непосредственное влияние выполнения операции управления на атрибут управляемого объекта определяется этой операцией. Например непосредственное влияние операции замены на атрибут состоит в изменении значения атрибута.

Операция управления, которая осуществляется над одним или несколькими атрибутами управляемого объекта, может привести к другим наблюдаемым изменениям, называемым косвенными эффектами. Косвенные эффекты являются результатом взаимосвязей в нижележащем ресурсе. Могут происходить следующие косвенные эффекты:

- изменение атрибута в том же самом управляемом объекте;
- изменение поведения управляемого объекта;
- изменение атрибута в управляемом объекте, связанном с данным;
- изменение поведения управляемого объекта, связанного с данным, вызванное изменением одного или нескольких атрибутов в целевом управляемом объекте.

Первые два из описанных косвенных эффектов являются следствием поведения управляемого объекта, содержащего атрибут, на который направлена операция управления. Последние два косвенных эффекта являются следствием поведения управляемого объекта, связанного с данным, или следствием определения взаимосвязи.

5.3.3.2 *Получить значение атрибута*

Область действия

Операция Get attribute value применяется к атрибутам, инкапсулированным в управляемые объекты, определения классов которых допускают операцию получения значений атрибутов.

Семантика

Запрашивается чтение списка значений атрибутов или, если список не задан, чтение значений всех атрибутов; возвращаются значения атрибутов, которые могли быть прочитаны, и указания ошибок для тех значений атрибутов, которые не могли быть прочитаны.

Если в запросе задан пустой список (в отличие от отсутствия списка), то будет возвращен пустой список значений атрибутов.

Поведение

Эта операция всегда подтверждаемая.

Управляемому объекту доступна следующая дополнительная информация для определения, должна ли и если должна, то как выполняться операция Get attribute value: идентификаторы атрибутов или атрибутивных групп для атрибутов, значения которых должны быть прочитаны.

На границе управляемого объекта в результате операции Get attribute value доступна следующая дополнительная информация:

- идентификаторы атрибутов и их значения для атрибутов, значения которых могли быть прочитаны;

- указания ошибок для атрибутов, значения которых не могли быть прочитаны. Следующее указание ошибок отлично от ошибок, общих для всех операций, направленных на атрибуты: значения атрибута нечитаемы для нечитаемых атрибутов.

Примечание — Способы, которыми эти указания ошибок и значения атрибутов становятся доступными управляемому объекту и предоставляются управляемым объектом, не являются предметом стандартизации.

Алломорфное поведение

Когда управляемый объект может поддерживать алломорфные классы, используется следующее дополнительное поведение, если передана операция Get attribute value без списка атрибутов.

Управляемый объект должен:

- определить класс управляемых объектов, который применяется в этой операции, и

- ответить на запрос, предоставив идентификаторы атрибутов и их значения или указания ошибок для тех атрибутов, которые были запрошены и имеются в определении применяемого класса.

5.3.3.3 *Заменить значение атрибута*

Область действия

Операция Replace attribute value применяется к атрибутам, инкапсулированным в управляемые объекты, определения классов которых допускают операцию замены значений атрибутов.

Семантика

Значения заданных атрибутов заменяются представленными значениями. Замена осуществляется в точности, за исключением случаев, когда иное явно установлено в определении атрибута, например округление абсциссы значения с плавающей точкой.

Поведение

Каждый вызов этой операции может быть подтверждаемым или неподтверждаемым, по усмотрению вызывающего.

Управляемому объекту доступна следующая дополнительная информация для определения, должна ли и если должна, то как выполняться операция Replace attribute value: идентификаторы атрибутов и связанные с ними значения для атрибутов, значения которых должны быть заменены.

На границе управляемого объекта в результате операции Replace attribute value доступна следующая дополнительная информация:

- идентификаторы атрибутов и их значения для атрибутов, значения которых были заменены;

- указания ошибок для атрибутов, значения которых не могли быть заменены. Следующие указания ошибок отличны от ошибок, общих для всех операций, направленных на атрибуты:

- значения атрибута незаменяемы для атрибутов, защищенных от записи;

- недопустимое значение атрибута.

Алломорфное поведение

Дополнительное поведение для этой операции не используется.

5.3.3.4 *Заменить значением по умолчанию*

Область действия

Операция Replace-with-default value применяется к атрибутам, инкапсулированным в управляемые объекты, определения классов которых допускают операцию замены значением по умолчанию.

Семантика

Значения заданных атрибутов заменяются значениями по умолчанию.

Значение по умолчанию (или способ его получения) может быть определено как часть спецификации класса управляемых объектов или оставлено как локальный вопрос. Операция Replace-with-default value не обязательно восстанавливает значения атрибутов, которые управляемый объект имел при создании.

Поведение

Каждый вызов этой операции может быть подтверждаемым или неподтверждаемым, по усмотрению вызывающего.

Управляемый объект определяет значения по умолчанию и заменяет ими текущие значения атрибутов.

Управляемому объекту доступна следующая дополнительная информация для определения, должна ли и если должна, то как выполняться операция Replace-with-default value: идентификаторы

тех атрибутов и атрибутивных групп, значения которых должны быть заменены на значения по умолчанию.

На границе управляемого объекта в результате операции Replace-with-default value доступна следующая дополнительная информация:

- идентификаторы атрибутов и их значения для атрибутов, значения которых были заменены;
- указания ошибок для атрибутов, значения которых не могли быть заменены значениями по умолчанию. Следующие указания ошибок отличны от ошибок, общих для всех операций, направленных на атрибуты:
- значения атрибута незаменяемы для атрибутов, защищенных от записи;
- значение по умолчанию не определено для данного атрибута.

Алломорфное поведение

Когда управляемый объект может поддерживать алломорфные классы, используется следующее дополнительное поведение.

Управляемый объект должен заменить значения атрибутов значениями по умолчанию, заданными определением фактического класса управляемого объекта.

5.3.3.5 *Добавить член множества значений*

Область действия

Операция Add member применяется к атрибутам, инкапсулированным в управляемые объекты, определения классов которых допускают операцию добавления членов к атрибутам.

Семантика

Для каждого заданного многозначного атрибута данная операция заменяет значение атрибута (математическим) объединением множества существующих членов значения с множеством, предоставленным в операции; получающееся множество значений есть существовавшее ранее множество, к которому добавлены предоставленные в операции члены.

Попытка добавить член множества значений, который уже присутствует в атрибуте, не является ошибкой.

Поведение

Каждый вызов этой операции может быть подтверждаемым или неподтверждаемым, по усмотрению вызывающего.

Управляемому объекту доступна следующая дополнительная информация для определения, должна ли и если должна, то как выполняться операция Add member: идентификаторы атрибутов и связанные с ними значения, содержащие член множества значений, которые должны быть добавлены.

На границе управляемого объекта в результате операции Add member доступна следующая дополнительная информация:

- идентификаторы и новые значения каждого многозначного атрибута;
- указания ошибок для атрибутов, к значениям которых не могли быть добавлены новые члены. Следующие указания ошибок отличны от ошибок, общих для всех операций, направленных на атрибуты:
- член множества значения атрибута не может быть добавлен;
- недопустимое значение атрибута.

Алломорфное поведение

Дополнительное поведение для этой операции не используется.

5.3.3.6 *Удалить член множества значений*

Область действия

Операция Remove member применяется к атрибутам, инкапсулированным в управляемые объекты, определения классов которых допускают операцию удаления членов из атрибутов.

Семантика

Для каждого заданного многозначного атрибута данная операция заменяет значение атрибута (математической) разностью множества существующих членов значения и множества, предоставленного в операции; получающееся множество значений есть существовавшее ранее множество, из которого удалены предоставленные в операции члены.

Попытка удалить член, который не присутствует в атрибуте, не является ошибкой.

Поведение

Каждый вызов этой операции может быть подтверждаемым или неподтверждаемым, по усмотрению вызывающего.

Управляемому объекту доступна следующая дополнительная информация для определения, должна ли и если должна, то как, выполняться операция Remove member: идентификаторы атрибутов и

связанные с ними значения, содержащие элемент множества значений, которые должны быть удалены.

На границе управляемого объекта в результате операции Remove member доступна следующая дополнительная информация:

- идентификаторы и значения полученных многозначных атрибутов;
- указания ошибок для атрибутов, из значений которых не могли быть удалены члены. Следующие указания ошибок отличны от ошибок, общих для всех операций, направленных на атрибуты:
- член множества значений атрибута не может быть удален;
- недопустимое значение атрибута.

Алломорфное поведение

Дополнительное поведение для этой операции не используется.

5.3.4 Операции, применяемые к управляемым объектам в целом

Следующие операции управления применяются к управляемому объекту как к единому целому и, в общем случае, их действие не ограничивается изменениями значений атрибутов:

- создать;
- удалить;
- выполнить.

Ниже эти операции описаны подробнее. С помощью операции «выполнить» (Action) могут быть определены дополнительные операции. Семантика этих операций является частью определения класса управляемых объектов. В частности, должно быть специфицировано любое взаимодействие с другими связанными управляемыми объектами.

Кроме того что управляемые объекты могут быть созданы и удалены с помощью операций управления, они могут быть созданы или удалены в результате побочных эффектов обычных операций над ресурсами, например при установлении транспортного или сетевого соединения создается управляемый объект. Когда управляемые объекты создаются в результате обычных операций над ресурсами, имя управляемого объекта присваивается управляемой системой, осуществляющей операцию. После удаления управляемого объекта его имя может быть повторно использовано.

5.3.4.1 *Создать*

Область действия

Операция Create используется для создания управляемых объектов.

Семантика

Операция запрашивает создание и инициализацию управляемого объекта. Операция Create является уникальной в том смысле, что она применяется к управляемому объекту, который еще не существует. Назначение операции — создание управляемого объекта, совместимого с заданным классом управляемых объектов в иерархии наименований. Кроме создания управляемого объекта, представляющего ресурс, операция может оказывать влияние на сам ресурс. Ассоциация с представляемым ресурсом должна быть специфицирована в определении класса управляемых объектов.

Поведение

Эта операция всегда подтверждаемая.

Операция Create создает во вмещающем управляемом объекте управляемый объект заданного класса управляемых объектов или совместимый с этим классом. Вмещающий управляемый объект должен существовать до того, как может быть создан вмещаемый управляемый объект (см. 6.1). При создании управляемого объекта его атрибутам присваиваются значения, допустимые для атрибутов соответствующих типов. Эти значения извлекаются из информации в операции Create и в определении класса управляемых объектов следующим образом.

а) Обязательные начальные значения могут быть специфицированы как часть определения класса управляемых объектов. Если в запросе Create явно заданы значения, отличные от обязательных начальных значений, то этот запрос будет неудачным. Обязательные начальные значения всегда имеют старшинство над начальными значениями из всех других источников.

б) Допускается спецификация в запросе Create явных значений для отдельных атрибутов. При создании управляемого объекта явные значения могут быть присвоены атрибутам, защищенным от записи, если это явно допускается определением класса управляемых объектов. Присвоенные один раз, эти значения не могут быть изменены операциями, направленными на атрибуты. Когда значение атрибута явно задано несколько раз и две спецификации вступают в противоречие, запрос на создание будет неудачным. Дублирование спецификаций значения атрибута может быть результатом ошибки или результатом того, что некоторые параметры специфицируются в нескольких полях

запроса Create (например атрибут, используемый как ООИ, может быть задан как часть имени и в списке атрибутов).

в) Допускается в запросе Create специфицировать указываемый объект, из которого могут быть скопированы значения атрибутов. Однако значение атрибута, которое должно использоваться для наименования создаваемого управляемого объекта, не может быть скопировано из указываемого объекта.

г) Определение класса управляемых объектов может допускать использование управляемого объекта начальных значений (УОНЗ) для получения начальных значений. Могут быть заданы условия, при которых УОНЗ не в состоянии дать начальное значение.

д) Допускается в определении класса управляемых объектов специфицировать, как атрибутам присваиваются значения по умолчанию.

е) Может быть определен локальный способ присваивания начальных значений.

Значения присваиваются отдельно для каждого атрибута в соответствии с предпочтениями приведенного выше перечисления, где случай а) имеет наибольшее предпочтение.

Управляемые объекты с условными пакетами и без них являются членами одного и того же класса управляемых объектов. Для того чтобы гарантировать, что выбраны или созданы нижележащие ресурсы с требуемыми возможностями, управляющий должен быть в состоянии специфицировать возможности, которые должны иметь управляемые объекты.

Обязательные пакеты реализуются всегда. Реализация условного пакета происходит, если для реализуемого управляемого объекта удовлетворяется соответствующее условие. Управляющий может потребовать реализацию условного пакета как часть запроса создания:

а) явно, включив его в атрибут Package;

б) задав указываемый объект, который включает в себя условный пакет.

Из вышесказанного следует, что управляемая система может создать управляемый объект, содержащий пакеты в дополнение к тем, которые затребованы или которые специфицированы в указываемом объекте, если нижележащий ресурс поддерживает такие пакеты.

Запрос Create будет неудачным, если:

а) для какого-либо атрибута не может быть получено значение из информации в операции Create и в определении класса управляемых объектов описанными выше способами;

б) в запросе были нарушены явные правила создания, включающие в себя (определенные классом управляющих объектов) ограничения на значения атрибутов или на отношения между ними;

в) было задано значение атрибута из условного пакета, но сам пакет, содержащий этот атрибут, не может быть реализован;

г) создающая система не может обеспечить управляемый объект с, по крайней мере, запрошенными условными пакетами.

Имя создаваемого управляемого объекта может быть определено одним из четырех способов:

а) имя может быть полностью и явно задано управляющим как параметр операции Create;

б) управляющий может задать как параметр операции Create имя существующего управляемого объекта, который должен быть старшим нового управляемого объекта, и задать ООИ нового управляемого объекта в списке атрибутов операции Create. Это приводит к полной спецификации имени управляемого объекта, предоставленного управляющим;

в) управляющий может задать как параметр операции Create имя существующего управляемого объекта, который должен быть старшим нового управляемого объекта, и опустить ООИ нового управляемого объекта. В этом случае ООИ нового управляемого объекта присваивается управляемой системой;

г) когда управляющий не предоставляет никакой явной информации, которая может использоваться для наименования, управляемая система сама присваивает имя новому управляемому объекту.

Определение связывания имен устанавливает, какие из приведенных выше методов допустимы. Для управляемых объектов данного класса может быть разрешено несколько методов.

Если информация, связанная с запросом, некорректна или операция создания по каким-либо причинам не может быть выполнена, то управляемая система, пытавшаяся осуществить операцию, должна указать на ошибку.

Имя нового управляемого объекта и доступная информация о классе идентифицируют одно или несколько используемых связываний имен. Управляющий может потребовать конкретное связывание имен, чтобы полностью специфицировать взаимосвязь между новым управляемым объектом и его старшим. Это может быть запрошено, если, например, существует несколько применимых

связываний имен с разными определениями поведения. Если применимо несколько связываний имен и управляющий не задал конкретное связывание имен, то связывание имен может быть выбрано локально, с учетом определений связываний имен.

О том, должно или нет создаваться сообщение в результате создания управляемого объекта, следует устанавливать в определении класса управляемых объектов.

Управляемой системе, осуществляющей создание, доступна следующая информация для определения, должна ли и если должна, то как выполняться операция Create:

- идентификатор класса управляемых объектов;
- атрибут Package, посредством которого вызывается реализация соответствующих пакетов;
- идентификаторы и значения атрибутов, которые имеют явным образом заданные значения, присвоенные как часть реализации управляемого объекта;
- имя указываемого объекта, из которого должна быть получена информация для реализации управляемого объекта;
- атрибут Name Binding, посредством которого задается, какое будет использоваться связывание имен между новым управляемым объектом и его старшим.

На границе управляемого объекта в результате операции Create доступна следующая информация: полный список идентификаторов и значений всех атрибутов нового управляемого объекта.

В случае, когда управляемый объект не может быть создан, различают следующие указания ошибок:

- неизвестный идентификатор атрибута;
- недопустимое значение атрибута;
- опущено значение атрибута;
- неизвестный класс объектов;
- недопустимое имя указываемого управляемого объекта;
- недопустимая спецификация вмещения (связывания имен);
- отказ при обработке запроса Create.

Алломорфное поведение

Когда управляемый объект может поддерживать алломорфные классы, используется следующее дополнительное поведение.

Управляемая система может выполнить операцию Create для данного класса управляемых объектов, создав расширенный управляемый объект, который поддерживает класс, заданный в операции Create, в качестве алломорфного и поддерживает связывание имен, подразумеваемое именем, заданным в запросе (если было запрошено связывание имен). Фактический класс, экземпляром которого будет управляемый объект, есть класс, наиболее подходящий в контексте локальной системы. Фактический класс созданного управляемого объекта сообщается управляющему как часть ответа на запрос Create.

Созданный управляемый объект обладает всеми возможностями своего фактического класса управляемых объектов, и значения по умолчанию будут присваиваться в соответствии со спецификацией этого класса. Алломорфное поведение после создания управляемого объекта определяется поведением этого объекта и не зависит от класса, заданного в запросе Create. В последующем можно определить множество алломорфных классов, поддерживаемых управляемым объектом, прочитав атрибут Allomorphs.

5.3.4.2 *Удалить*

Область действия

Операция Delete применима ко всем управляемым объектам, которые могут быть удалены операцией управления. В частности, она может быть применена, даже если управляемый объект был создан локальными средствами.

Семантика

Операция Delete запрашивает от управляемого объекта удаление самого себя. Кроме удаления управляемого объекта, представляющего ресурс, эта операция может иметь влияние на сам ресурс. Ассоциация с представляемым ресурсом должна быть специфицирована в определении класса управляемых объектов.

Поведение

Эта операция всегда подтверждаемая.

Когда управляемый объект получает запрос удаления, он проверяет, не содержатся ли в нем другие управляемые объекты. Если в удаляемом управляемом объекте содержатся другие управляемые объекты, то поведение управляемого объекта зависит от определений класса управляемых

объектов и связывания имен. Управляемый объект может либо удалить все содержащиеся в нем управляемые объекты, чтобы обеспечить целостность имен, либо отказаться от удаления до тех пор, пока все содержащиеся в нем объекты не будут удалены. Связывание имен устанавливает для именующего дерева последовательность удаления управляемого объекта.

Осуществление успешной операции Delete для управляемого объекта, который удаляет содержащиеся в нем объекты и часть именующего дерева, подчиненную целевому управляемому объекту. Если какой-либо управляемый объект в этом поддереве был создан так, что может быть удален, только если в нем нет содержащихся объектов, но содержит управляемые объекты, то управляемый объект не удаляется и операция Delete завершается неудачно.

Аналогично, когда удаляемый управляемый объект участвует во взаимосвязях с другими управляемыми объектами, удаление этого объекта может повлиять на целостность взаимосвязи и (или) связанных управляемых объектов. Управляемые объекты и взаимосвязи должны удаляться таким образом, чтобы гарантировать целостность при каждом удалении. Если удаление управляемого объекта приведет к потере целостности, то управляемый объект может либо отвергнуть запрос удаления, либо инициировать операции, чтобы обеспечить целостность.

Должно или не должно создаваться сообщение в результате удаления управляемого объекта — зависит от его определения.

Управляемому объекту доступна следующая информация для определения, должна ли и если должна, то как выполняться операция Delete: идентификаторы атрибутов, соответствующие операторы сравнения и значения, которые были использованы при определении, какие объекты выбираются для осуществления операции (см. 5.4).

На границе управляемого объекта в результате операции Delete доступна следующая информация:

- указание, что удаление обрабатывается и не может быть отменено;
- указания ошибок в случае, когда управляемый объект не может быть удален. Различают следующее указание ошибок: отказ при обработке запроса удаления.

Если управляемый объект не может быть удален из-за ограничений, наложенных его взаимосвязями с другими управляемыми объектами, то это отмечается указанием ошибки «отказ при обработке», которое может содержать указание «специфической ошибки».

Алломорфное поведение

Дополнительное поведение для этой операции не используется.

5.3.4.3 *Выполнить*

Область действия

Операция Action может быть использована управляющими объектами всех классов.

Семантика

Операция Action запрашивает управляемый объект об осуществлении заданного действия и указании его результатов.

Действие и факультативная, связанная с ним, информация являются частью определения класса управляемых объектов.

Поведение

В отношении подтверждения операции Action могут быть определены либо тип, всегда требующий подтверждения, либо тип, который позволяет вызывающему операцию запрашивать подтверждение. Какой из типов выбран — устанавливается в определении действия.

Операции Action могут быть определены так, чтобы генерировать несколько ответов.

Конкретный эффект этой операции устанавливается классом объектов управления. Если операция Action не может быть выполнена управляемым объектом или связанная с операцией информация некорректна, то управляемый объект должен указать на ошибку.

Управляемому объекту доступна следующая информация для определения, должна ли и если должна, то как выполняться операция Action:

- идентификация конкретного действия, которое должно быть выполнено;
- аргумент, состоящий из параметров и описывающий действие. Так как можно определить действия, которые не будут иметь параметров, то этот аргумент может отсутствовать;
- идентификаторы атрибутов, соответствующие операторы сравнения и значения, которые были использованы при определении, какие объекты выбираются для осуществления операции (см. 5.4).

На границе управляемого объекта в результате операции Action доступна следующая информация.

Различают следующие указания ошибок:

- аргумент результата действия, состоящий из информации, которая требуется типом действия и классом управляющих объектов;

- указания ошибок в случае, когда управляемый объект не может выполнить действие. Различают следующие указания ошибок:

- неизвестное действие;

- неизвестный аргумент;

- недопустимое значение аргумента;

- неизвестный класс управляемых объектов;

- отказ при обработке запроса действия.

Алломорфное поведение

Дополнительное поведение, кроме установленного в 5.2.3.1.2, для этой операции не используется.

5.4 Фильтры

Фильтры, используемые в ПОИУ, позволяют специфицировать критерии, которые управляемые объекты должны согласовывать для выполнения операций управления. Вместе с определением области действия и спецификацией основного управляемого объекта, описанного в ГОСТ Р ИСО/МЭК 9595, они позволяют выбирать несколько управляемых объектов для выполнения нескольких идентичных операций. Фильтры являются факультативной возможностью.

Параметр фильтра используется для определения, должна ли выполняться операция над управляемым объектом. Он применяется для проверки, удовлетворяется ли фильтр для данного конкретного управляемого объекта. Фильтр выражается в терминах утверждений о наличии или значении конкретных атрибутов управляемого объекта и удовлетворяется только в случае, если его вычисление дает значение TRUE.

Фильтр является утверждением о наличии или значении атрибута в управляемом объекте или выражением, содержащим комбинацию более простых (в смысле вложенности) фильтров с использованием логических операторов И, ИЛИ и НЕ.

Оператор И (and) равен TRUE, если ни один из вложенных фильтров не равен FALSE.

Оператор ИЛИ (or) равен FALSE, если ни один из вложенных фильтров не равен TRUE.

Оператор НЕ (not) равен TRUE только в случае, когда вложенный фильтр равен FALSE.

Утверждение равно TRUE только в случае, когда соответствующие утверждения о значениях атрибутов равны TRUE при сравнении согласно правилам согласования, применяемым для тестируемых атрибутов.

Утверждение о значении атрибута вычисляется, только если атрибут присутствует в управляемом объекте. Правило согласования, не определенное для конкретного типа атрибута, не должно использоваться в утверждении об этом атрибуте.

Определено восемь правил согласования, которые могут быть использованы в утверждениях о значениях атрибутов. Определение типа атрибута должно специфицировать точную семантику каждого из этих правил применительно к атрибуту, но общая семантика должна сохраняться для любых таких определений. Для широко используемых типов атрибутов дальнейшая спецификация применения этих правил согласования не требуется. При отсутствии явной спецификации применяется установленная ниже семантика:

а) equality: равно TRUE только тогда, когда значение, представленное в УЗА, равно значению атрибута.

Для многозначных атрибутов УЗА равно TRUE только тогда, когда множество членов, представленных в УЗА, равно множеству членов в атрибуте;

б) greater or equal: равно TRUE только тогда, когда значение, представленное в УЗА, больше или равно значению атрибута.

Для многозначных атрибутов значение в УЗА должно содержать ровно один член. УЗА равно TRUE только тогда, когда этот член больше или равен по крайней мере одному из членов в значении атрибута;

в) less or equal: равно TRUE только тогда, когда значение, представленное в УЗА, меньше или равно значению атрибута.

Для многозначных атрибутов значение в УЗА должно содержать ровно один член. УЗА равно TRUE только тогда, когда этот член меньше или равен по крайней мере одному из членов в значении атрибута;

г) present: равно TRUE только тогда, когда такой атрибут присутствует в управляемом объекте;

д) substrings: равно TRUE только тогда, когда все подстроки, заданные в УЗА, встречаются в атрибуте в заданном порядке без перекрытия и отделены от концов значения атрибута и друг от друга нулем или несколькими элементами строки. Кроме того, чтобы УЗА было равно TRUE:

- первый элемент в начальной подстроке, если он есть, должен согласовываться с первым элементом в значении атрибута;
- любые подстроки, если они есть, должны появляться в значении атрибута в порядке, в котором эти подстроки появляются в УЗА;
- последний элемент в последней подстроке, если он есть, должен согласовываться с последним элементом в значении атрибута.

Для многозначных атрибутов каждое значение в УЗА должно содержать ровно один член. УЗА равно TRUE только тогда, когда имеется по крайней мере один член значения атрибута, в котором все подстроки, представленные в УЗА, встречаются так, как описано выше;

е) sebset of: равно TRUE только тогда, когда все указанные члены присутствуют в атрибуте.

Это правило согласования применяется только для многозначных атрибутов;

ж) superset of: равно TRUE только тогда, когда все члены атрибута присутствуют в УЗА.

Это правило согласования применяется только для многозначных атрибутов;

и) non-null set intersection: равно TRUE только тогда, когда по крайней мере один из указанных членов присутствует в атрибуте.

Это правило согласования применяется только для многозначных атрибутов.

Проверка правила present не требует, чтобы атрибут имел установленные правила согласования.

УЗА, которые встречаются в фильтрах, не должны ссылаться на атрибутивные группы.

5.5 Сообщения

Управляемые объекты могут быть определены таким образом, что будут создавать сообщения, когда происходят некоторые внешние или внутренние события. Сообщения являются специфичными для создающих их управляемых объектов. Сообщения и содержащаяся в них информация являются частью определения класса управляемых объектов, экземпляром которого является данный объект.

Будет ли передано сообщение через протокол или будет записано, зависит от конфигурации управления открытой системы. В частности, будет или не будет отправлено сообщение, зависит от того, удовлетворяются ли критерии, установленные в дискриминаторе передаваемых событий (см. ИСО/МЭК 10164-5).

Является ли сообщение результатом в подтверждении (в противоположность сообщению неподтверждаемого события) не является частью определения управляемого объекта, а определяется коммуникационными, системными или политическими требованиями, включая установки дискриминатора передаваемых событий.

6 Принципы вмещения и наименования

6.1 Вмещение

Управляемый объект одного класса может содержать другие управляемые объекты того же или других классов. Это взаимоотношение называется вмещением. Отношение вмещения является отношением между экземплярами управляемых объектов, а не классов. Управляемый объект содержится ровно в одном вмещающем объекте. Вмещающие управляемые объекты сами могут содержаться в других управляемых объектах.

Вмещение может быть изображено как направленный граф, дуги которого указывают от вмещаемого управляемого объекта на вмещающий.

Соотношение вмещения может быть использовано для моделирования иерархии частей реального мира (например совокупность, подсовокупности и компоненты) или организационной иерархии (например справочник, файлы, записи и поля).

Спецификация конкретного соотношения вмещения может (но не обязательно) определять статическое поведение вмещающего и вмещаемого управляемых объектов (например ограничивать число и классы управляемых объектов, которые могут содержаться в другом управляемом объекте).

Спецификация конкретного соотношения вмещения может (но не обязательно) определять динамическое поведение вмещающего и вмещаемого управляемых объектов (например ограничивать отношение между значениями атрибутов вмещающего и вмещаемых управляемых объектов, доступность вмещающего и вмещаемых управляемых объектов и пр.).

Любые ограничения, накладываемые на вмещаемый управляемый объект в силу его положения в иерархии вмещения, должны быть специфицированы как часть определения отношения вмещения или как часть определения классов вмещающего и вмещаемых объектов.

Примечание — Вмещение не обязательно представляет физическое размещение одного ресурса в другом.

Примеры возможного дерева вмещения показаны на рисунке 2.

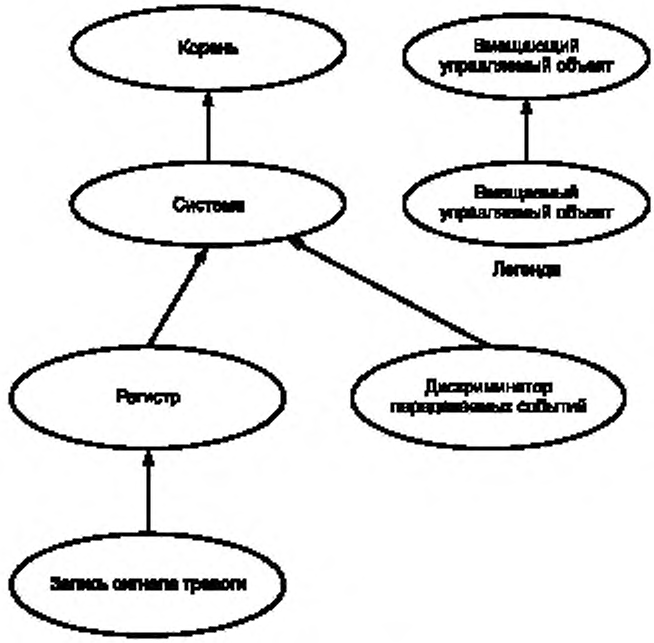

Рисунок 2 — Пример вмещения управляемых объектов

6.2 Именующее дерево

Соотношение вмещения используется для наименования управляемых объектов. Имена должны недвусмысленно различаться в заданном контексте; для административного управления этот контекст определяется вмещающим объектом.

Объекты, именуемые в терминах другого объекта, называются подчиненными этого объекта. Объект, который устанавливает контекст наименования для других объектов, называется старшим для этих подчиненных объектов.

Подчиненный объект именуется комбинацией:

- имени его старшего объекта;
- информации, однозначно идентифицирующей этот объект в пределах его старшего объекта.

Имя объекта, которое является недвусмысленным в локальном контексте наименования, может не быть таковым в более широком контексте. Однако если локальный контекст наименования является недвусмысленным в более широком контексте, то локальное имя может быть сделано недвусмысленным с помощью квалификации его контекстом наименования; имя контекста наименования используется в качестве квалификатора. Это упорядочение может быть представлено как направленный граф, дуги которого указывают от именуемого объекта к контексту наименования.

Контекст наименования сам может быть рекурсивно квалифицирован другим контекстом наименования, так что полная структура наименования может быть представлена как одновершинная иерархия. Эта иерархия называется именующим деревом. Таким образом, старшие объекты

становятся контекстами наименования и их имена — именами контекстов. Имя объекта должно быть недвусмысленным только в контексте его старшего объекта; в более широком контексте его имя всегда квалифицируется именами старших объектов.

Вмещение, наименование и существование управляемых объектов тесно связаны между собой следующим образом:

- управляемый объект может существовать, только если существует его старший объект (т. е. был создан и не был удален);
- каждый управляемый объект имеет имя, которое вытекает из соотношения вмещения, как описано выше.

Верхний уровень именующего дерева рассматривается как корень, который является вырожденным объектом null (т. е. с объектом не связаны какие-либо свойства) и который существует всегда. Для каждого определенного класса объектов должны быть идентифицированы атрибуты и классы старших объектов, экземпляры которых могут быть использованы при построении имени объекта. Взаимоотношение, идентифицирующее класс возможных старших объектов, которые могут быть использованы при наименовании, называется связыванием имен. В любое время могут быть определены дополнительные отношения связывания имен для объектов конкретного класса, т. е. не все связывания имен должны быть специфицированы при определении класса объектов. Следовательно, обеспечиваемые связывания имен являются не свойствами класса объектов в целом, и отдельные экземпляры одного и того же класса могут использовать разные связывания имен. Совокупность таких правил наименования называется именующей схемой.

Примечание — Именующее дерево не обязательно представляет физическое вхождение одного ресурса в другой.

Кроме заданных классов старших и подчиненных объектов, связывание имен может быть определено так, что будет применяться к подклассам старшего или подчиненного объекта, или к тем и другим.

6.3 Структура имени

6.3.1 Идентификация класса управляемых объектов

Класс управляемых объектов внешне идентифицируется идентификатором объекта АСН.1. Идентификатор объекта может быть представлен как последовательность целых чисел, которая управляет продвижением по дереву идентификаторов объектов к классу управляемых объектов.

Примечание — Следует учитывать, что дерево идентификаторов объектов не имеет отношения ни к вмещению, ни к именующему дереву.

Конкретный идентификатор объекта определен в ГОСТ Р ИСО/МЭК 10165-4 для использования в протоколе в качестве идентификатора класса управляемых объектов с семантикой, по которой он ссылается на фактический класс управляемых объектов рассматриваемого объекта.

6.3.2 Идентификация управляемого объекта

Каждый управляемый объект идентифицируется в пределах области действия его старшего объекта с помощью утверждения о значении атрибута (УЗА), которое специфицирует атрибут, имеющий заданное значение. При использовании такого способа наименования УЗА называется относительным отличающим именем (ООИ) и должно позволять недвусмысленно идентифицировать единственный управляемый объект в пределах области действия его старшего объекта.

Конкретный атрибут, который должен использоваться в качестве ООИ, для данного класса старшего объекта устанавливается в связывании имен. Для этой цели могут использоваться разные атрибуты даже в одном и том же старшем классе, так как можно определить несколько связываний имен. Однако требуется, чтобы ООИ могло быть недвусмысленно удовлетворено. Следовательно, каждый класс управляемых объектов, которые могут быть реализованы, должен включать в себя по крайней мере один атрибут, подходящий для наименования: такой атрибут должен быть частью обязательного пакета, должен быть проверяемым на равенство и его семантика должна допускать значения, остающиеся фиксированными на протяжении жизни каждого управляемого объекта, который использует его для наименования.

Синтаксис атрибута, используемого для ООИ, не должен быть ни одним из следующих типов АСН.1:

- вещественным;
- множество;

- множество-из;
- произвольным;
- выборочным, включающим в себя любой из перечисленных выше;
- типом, полученным из любого перечисленного выше с помощью тегирования, образования подтипов или использования селективного типа.

При удалении управляемого объекта значение, присвоенное его именующему атрибуту, становится доступным при повторном использовании для идентификации последующих управляемых объектов, созданных в том же самом старшем объекте.

6.3.2.1 *Локальная и глобальная формы имени*

В ГОСТ Р ИСО/МЭК 7498-3 описан заголовок-системы, который обеспечивает однозначную и недвусмысленную идентификацию совокупности ресурсов, соответствующих управляемой системе. Системный управляемый объект представляет управляемую систему. Каждый системный управляемый объект имеет атрибуты systemTitle и systemId. Каждый из этих атрибутов может быть использован для наименования системного управляемого объекта.

Атрибут systemId является однозначным атрибутом и его тип ACH.1 выбирается из следующих:
- CraphicString;
- INTEGER;
- NULL.

Атрибут systemTitle является однозначным атрибутом и его тип ACH.1 выбирается из следующих:
- DistinguishedName (т. е. SEQUENCE OF RelativeDistinguishedName);
- OBJECT IDENTIFIER;
- NULL.

Значение NULL для этих атрибутов используется только в случаях, когда:
- система не была первоначально сконфигурирована, или
- соответствующий атрибут не будет использоваться для наименования этого системного управляемого объекта.

Для административного управления систем ВОС могут быть использованы две формы имени:
- глобальная — задающая имя относительно глобального корня. Она не может использоваться, когда оба атрибута systemTitle и systemId равны NULL;
- локальная — задающая имя относительно предварительно определенного контекста. Системный управляемый объект представляет систему, обеспечивающую функциональные возможности агента. Для административного управления систем ВОС контекстом для локальной формы является этот системный управляемый объект. Локальной формой имени для этого системного управляемого объекта является пустая последовательность. Для административного управления системой ВОС локальная форма может всегда использоваться для управляемых объектов, именуемых в контексте системного управляемого объекта. Для управляемых объектов, не имеющих имени относительно системного управляемого объекта, требуется использовать глобальную форму. Локальная форма не обеспечивает глобально однозначную идентификацию.

Локальное имя управляемого объекта строится путем сцепления относительных отличающих имен, начиная с ООИ, которое идентифицирует управляемый объект в области действия системы, и продолжая вниз по именующему дереву до ООИ, которое идентифицирует данный управляемый объект в области действия его старшего объекта.

Примечание — Для предотвращения конфликтов при присоединении управляемых объектов к информационному дереву справочника атрибуты systemTitle и systemId зарезервированы для использования административным управлением систем.

6.3.2.2. *Примеры наименования управляемых объектов*

В приведенных ниже примерах использованы следующие обозначения:
- утверждение о значении атрибута (УЗА) обозначается как «символ атрибута» = «нотация значения», где «символ атрибута» — символ, используемый для обозначения идентификатора атрибута, а «нотация значения» — значение типа этого атрибута. (Так как примеры носят иллюстративный характер, эта нотация не идентифицирует атрибут недвусмысленно);
- последовательность заключается в фигурные скобки;
- прописные буквы используются в качестве меток объектов для ссылок, например А.

6.3.2.2.1 ООИ и отличающие имена

На рисунке 3 и в таблице 1 приведен пример, как экземплярам объектов присваиваются ООИ и отличающие имена.

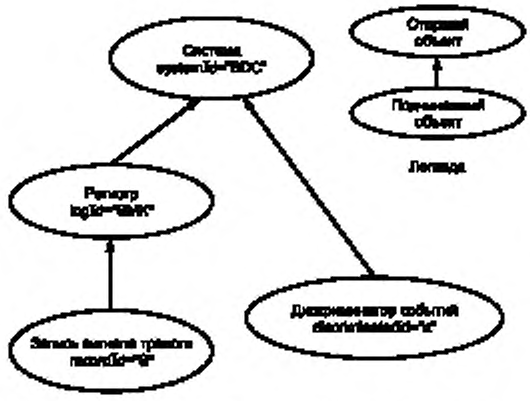

Рисунок 3 — Пример ООИ и локальных отличающих имен

Таблица 1

Относительное отличающее имя	Локальное отличающее имя
systemId = «BDC»	{}
logId = «SMK»	logId = {SMK}
recordId = «S»	{logId = SMK, recordId = «S»}

6.3.2.2.2 Локальные и глобальные имена

На рисунке 4 и в таблице 2 приведен пример, как экземплярам объектов присваиваются локальные и глобальные имена.

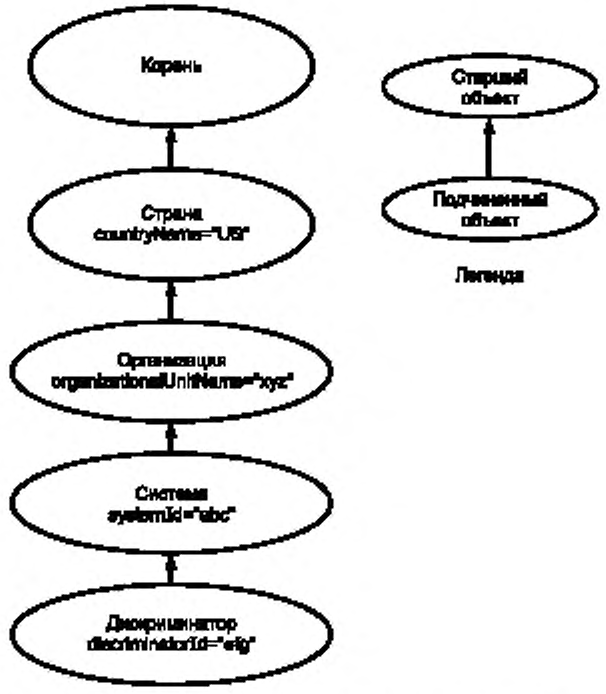

Рисунок 4 — Пример локальных и глобальных имен

Таблица 2

Объект	Глобальное имя	Локальное имя относительно системы
root	{}	Не используется
A	{countryName = «US»}	Не используется
B	{countryName = «US», organizationalUnitName = «xyz»}	Не используется
C	{countryName = «US», organizationalUnitName = «xyz», systemId = «abc»}	{}
D	{countryName = «US», organizationalUnitName = «xyz», systemId = «abc», discriminatorId = «efg»}	{discriminatorId = «efg»}

6.3.3 Идентификация атрибутов

Каждый атрибут в определении класса управляемых объектов идентифицируется идентификатором объекта ACH.1. Идентификатор объекта атрибута отличает его от всех прочих атрибутов.

7 Атрибуты высшего класса

Для того чтобы информация, содержащаяся в управляемых объектах системы, удовлетворяла требованиям расширяемости, необходимо обеспечить средства для исследования управляемых объектов, находящихся в системе. Каждый управляемый объект содержит всю информацию, необходимую для описания самого себя в целях обеспечения доступа административному управлению.

Для обеспечения этой возможности следующие атрибуты определены для всех управляемых объектов:

- ManagedObjectClass;
- Allomorphs;
- NameBinding;
- Packages.

Атрибут ManagedObjectClass идентифицирует фактический класс управляемого объекта.

Атрибут Allomorphs идентифицирует множество алломорфных классов этого управляемого объекта, за исключением его собственного. Он входит в условный пакет и присутствует только в случае, если алломорфные классы поддерживаются управляемым объектом.

Атрибут NameBinding содержит идентификатор объекта того связывания имен, которое используется между управляемым объектом и его старшим. Он входит в обязательный пакет высшего класса.

Атрибут Packages используется для установления того, какие пакеты были реализованы. Его значение является множеством идентификаторов объектов, соответствующих тем реализованным пакетам, которые зарегистрированы. Атрибут Packages входит в условный пакет, в котором он является единственным атрибутом, но идентификатор этого пакета не включается в значение атрибута. Пакет должен обязательно присутствовать в случае, если значение атрибута Packages не является пустым, т. е. если экземпляр управляемого объекта имеет какие-либо реализованные зарегистрированные пакеты, отличные от пакета, содержащего сам атрибут Packages. Если значение Packages пусто, то пакет может как присутствовать, так и отсутствовать.

Эта информация адекватна обеспечению управляющей системе доступа к управляемому объекту с целью административного управления им как членом известного алломорфного класса. Операция получения значений атрибутов в сочетании с фильтрами может быть использована для получения всей этой информации для выбранного подмножества управляемых объектов в открытой системе.

Определенная выше возможность позволяет определить, какие управляемые объекты были реализованы в данной системе. Этим методом не может быть получена информация о возможностях системы, относящихся к классам управляемых объектов, экземпляры которых не реализованы в этой системе.

УДК 681.324:006.354 ОКС 35.100.70 П85 ОКСТУ 4002

Ключевые слова: информационная технология, взаимосвязь открытых систем, обработка данных, информационный обмен, сетевое взаимодействие, административное управление, информация, модели

Редактор *В.П. Огурцов*
Технический редактор *О.Н. Власова*
Корректор *Т.И. Кононенко*
Компьютерная верстка *О.В. Арсеевой*

Изд. лиц. № 02354 от 14.07.2000. Сдано в набор 20.09.2001. Подписано в печать 09.11.2001. Усл. печ. л. 3,72. Уч.-изд.л. 3,50.
Тираж 434 экз. С 2506. Зак. 1060.

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