

Анализ и проектирование на UML

Новиков Федор Александрович

fedornovikov@rambler.ru

Курс подготовлен по заказу
ООО Сан Майкросистемс СПб

Часть 6

Курс подготовлен при поддержке Sun Microsystems
Правила использования материалов опубликованы на www.sun.ru

План лекций

- Введение в UML
- Обзор языка
- Моделирование использования
- Моделирование структуры
- Моделирование поведения
- ✓ Управление моделями
- Тенденции развития языка
- UML и процесс разработки

6. Управление моделями

- 6.1. Пакеты
- 6.2. Трассировка и гиперссылки
- 6.3. Образцы и каркасы

6.1. Пакеты

- Пакет – группирующая сущность
- Иерархия пакетов
- Анонимный корневой пакет по умолчанию
- Пакет *владеет* объявленными в нем элементами
- Владение – отношение композиции
- Пакет может владеть не только простыми элементами, но и пакетами, диаграммами и подсистемами

Свойства пакета

- Пакет задает пространство имен:
 - Имена однородных элементов в пакете уникальны
 - Составные имена через ::
- Видимость элементов пакета
 - Для каждого элемента + – #
- Зависимости со стереотипом «import» и «access»
 - Доступ по именам к открытым элементам, т.е. расширение пространства имен
- Обобщение для пакетов
 - Абстрактный пакет

Принципы структурирования

- **Форма структуры**
 - Количество сущностей на диаграмме 7 ± 3
 - Ширина ветвления дерева пакетов 7 ± 3
 - Число вхождений элемента в диаграммы 7 ± 3

- **Содержание структуры**
 - По структуре приложения
 - По фазам процесса разработки
 - По представлениям модели

UML Navigator

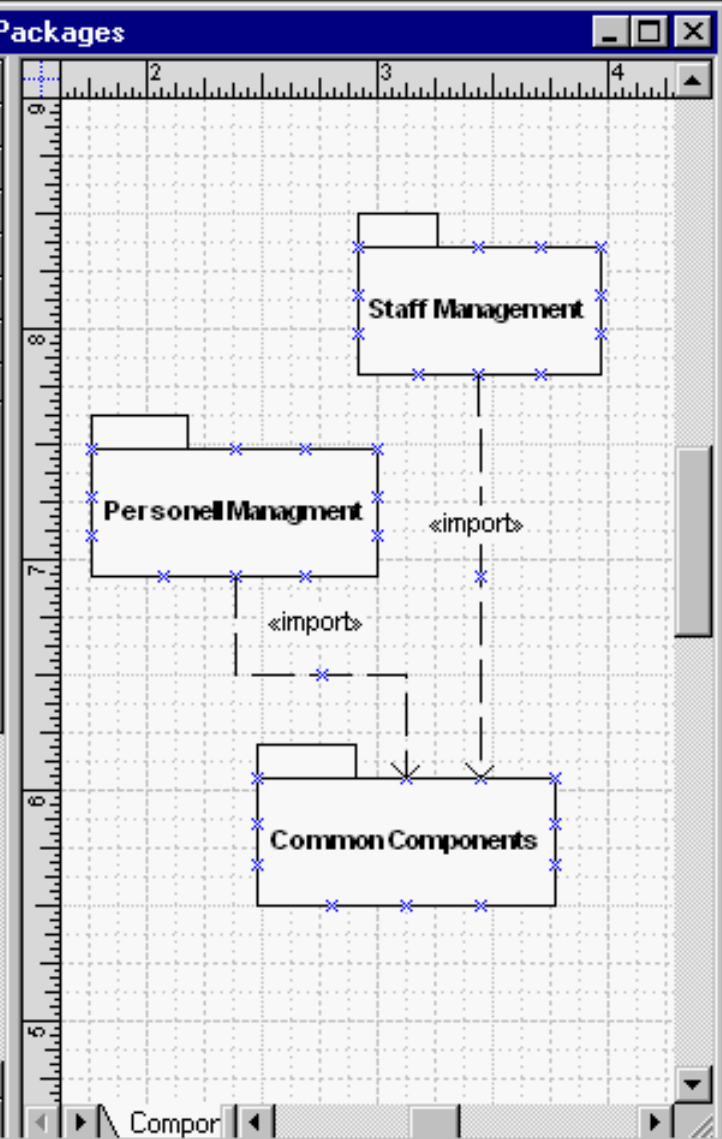
- UML Systems
 - UML System 1
 - Components
 - Top Package
 - Components Packages
 - Deployment
 - Common Components
 - Utilities
 - Personell Managment
 - Personnel Classes
 - Personnel Use Cases
 - Staff Management
 - StaffClasses
 - Staff Use Cases
 - Phases
 - Top Package
 - Phases Packages
 - Conception Analysis
 - Static Structure-4
 - Use Case-1
 - Detailed Design
 - Implementation & Deployment
 - Views
 - Top Package
 - Views Packages
 - Deployment View
 - Logical View
 - Use Case View

Packages.vsd:Components Packages

- UML Activity
- UML Collaboration
- UML Component
- UML Deployment
- UML Sequence
- UML Statechart
- UML Static Structure

Package	Class	Data Type
Interface	Interface	Generaliz...
Binary Association	Composit...	Association Class
Depende...	Utility	Subsystem
Paramete... Class	Binding	Bound Element

- UML Use Case



Files Ru...



  Model

Diagrams

Imported Elements

4_15 Metamodel x

100%

Feature

Attributes

```
private int scope
```

Instance

Classifier

Attributes

Operations

Collabora

Modeling Palette

<D> Datatype

 Artifact

 Aliased

Utility Class

 Actor

↑ Generalization

...▶ Implementation

⊕ Nested Link

▼ Robustness

 Boundary Class

 Control Class

Entity Class

▼ Dependencies

D Dependency

R → Realize

U Usage

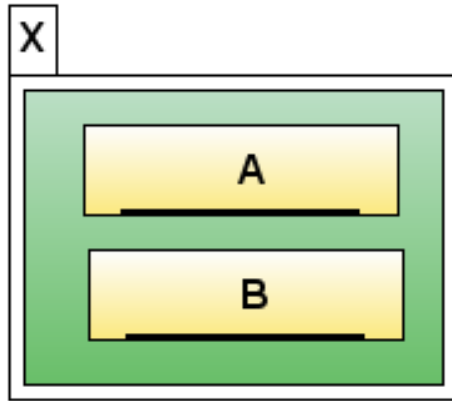
Отношения между пакетами

- **Отношения владения (вложенности)**
 - Дерево пакетов в инструменте
- **Индукцированные отношения**
 - Достаточно одной пары
- **Стереотипные зависимости**
 - «import» и «access»
- **Слияние пакетов (package merge) 👍 2.0**
- **Обобщение**
 - Абстрактные пакеты

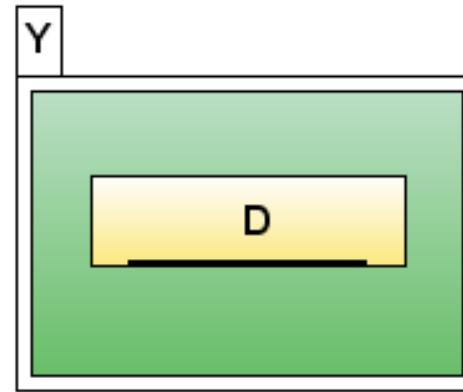
Отношение вложенности



Класс,
принадлежащий
объемлющему пакету



Пакет,
владеющий
двумя
классами

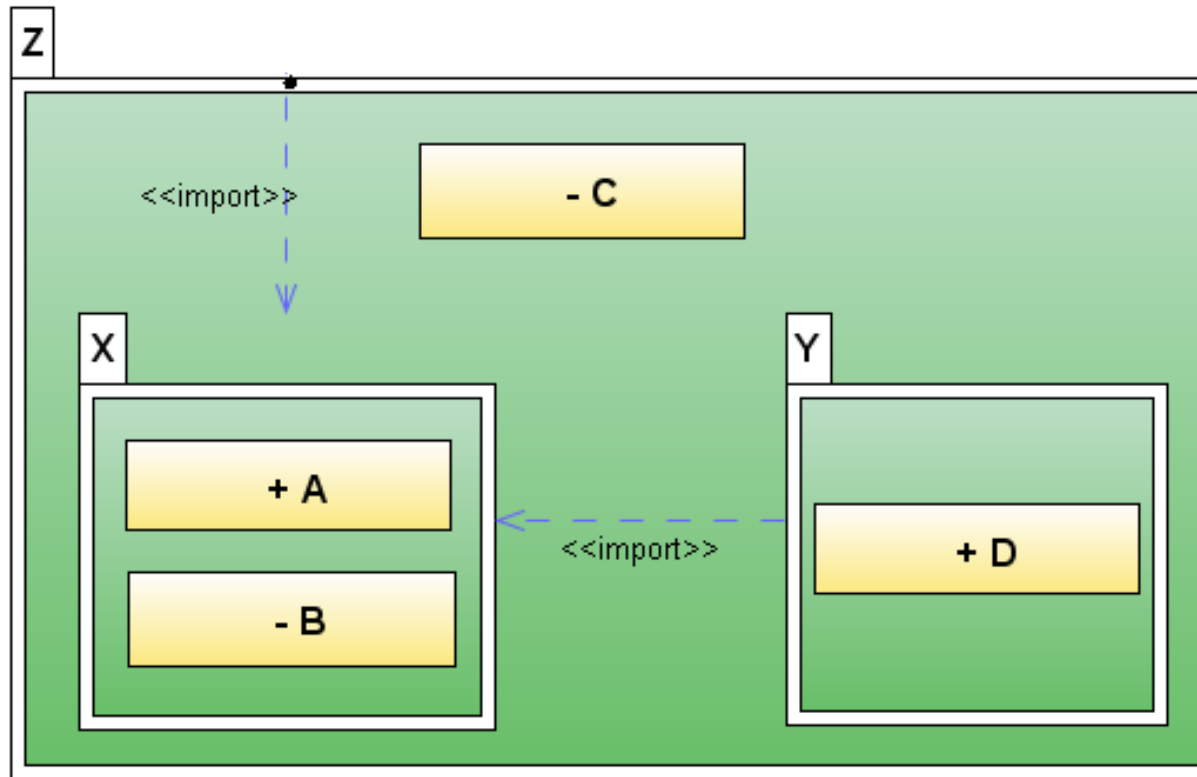



Пакет,
владеющий
одним
классом

- класс A видит классы B и C, но не видит класс D
- класс B видит классы A и C, но не видит класс D
- класс C не видит классы A, B и D
- класс D видит класс C, но не видит классы A и B

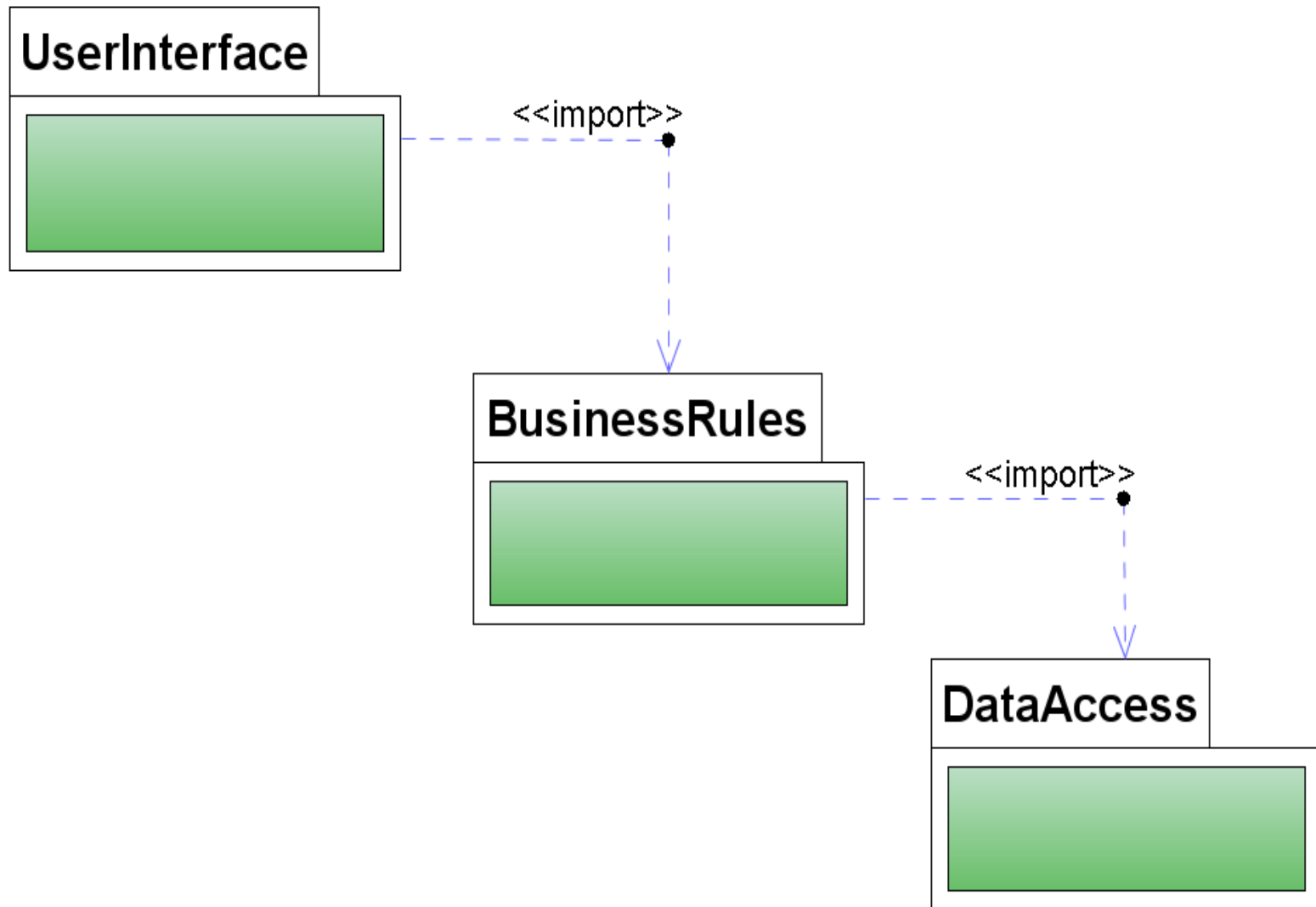


Расширение пространства имен

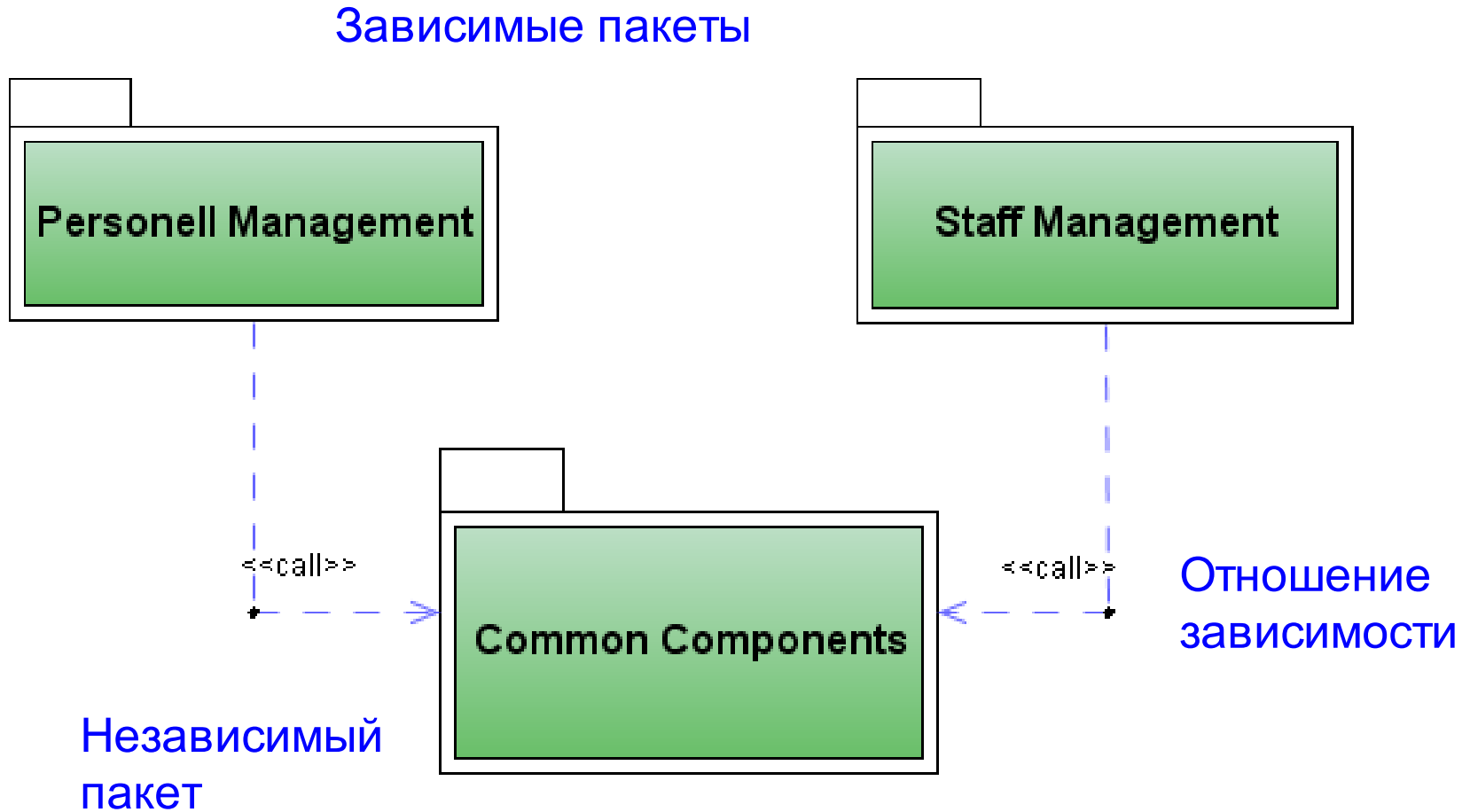


- класс A видит классы B и C, но не видит класс D
- класс B видит классы A и C, но не видит класс D
- класс C видит класс A, но не видит классы B и D
- класс D видит классы A и C, но не видит класс 

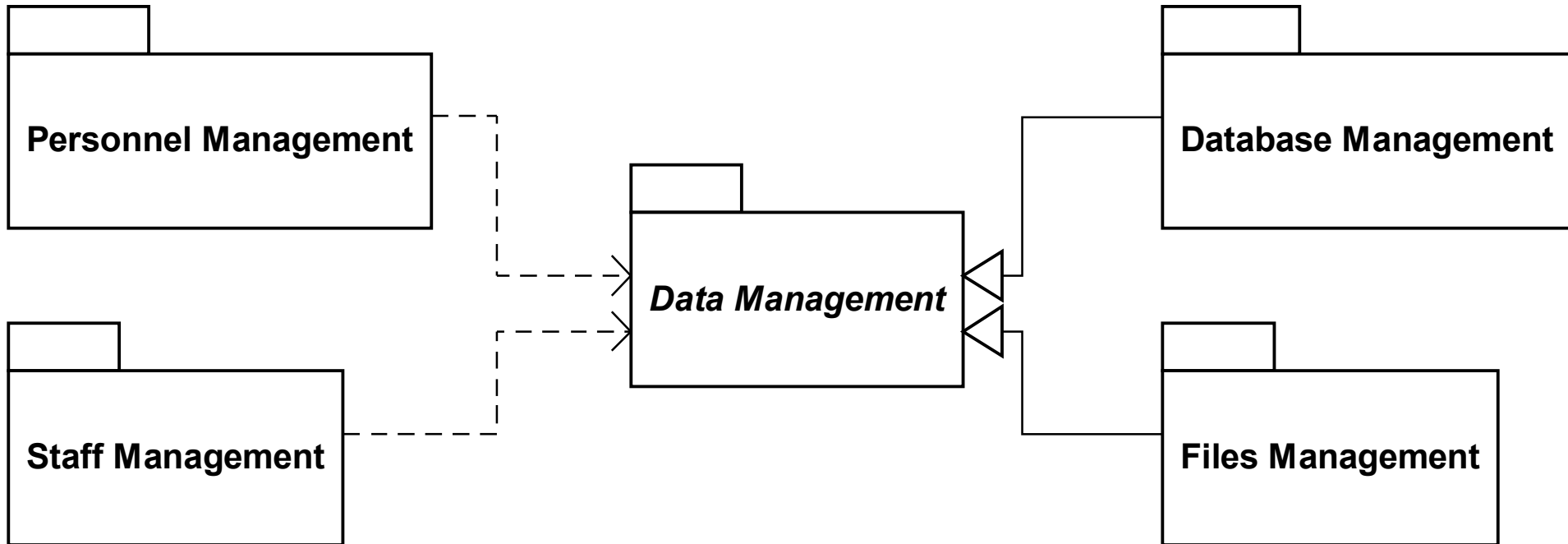
Нотация импорта



Индукцированные зависимости



Обобщение для пакетов



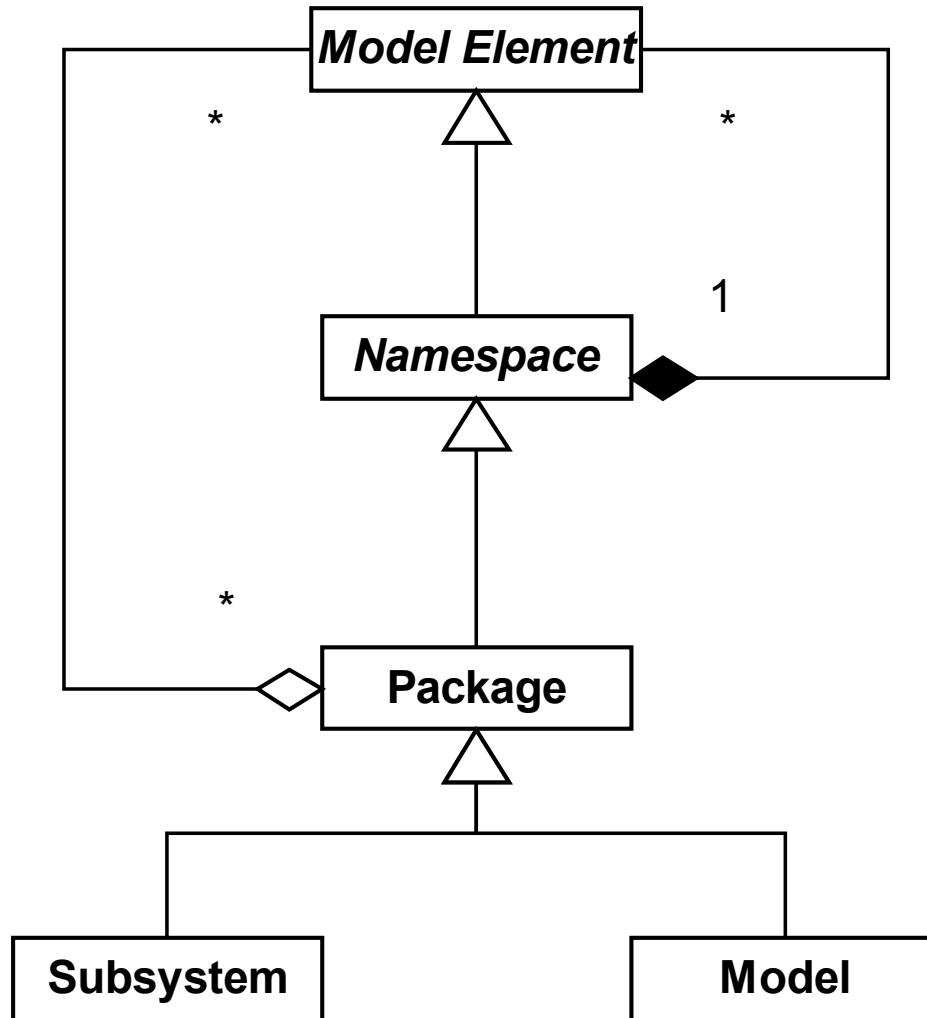
Модели, системы и подсистемы

- **Физическая система**
 - моделируемая часть реального мира
- **Модель**
 - Описание физической системы
 - Для одной физической системы м.б. несколько моделей
- **Подсистема**
 - Часть физической системы

Мета модель управления моделями

+importedElement

+ownedElement



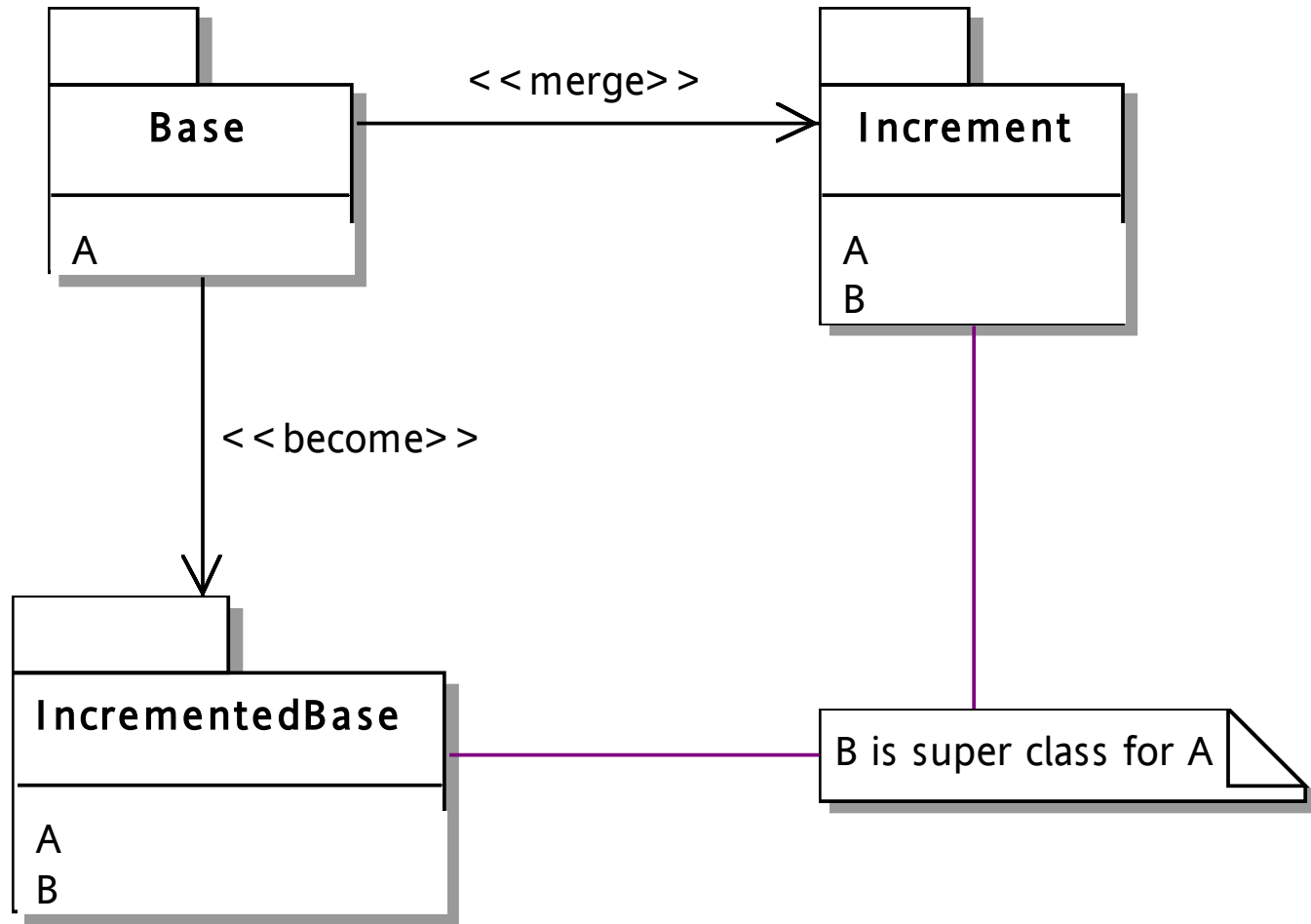
Стандартные стереотипы пакетов (1.x)

«facade»	Пакет, который содержит только <i>ссылки</i> на элементы, определенные в других пакетах
«framework»	Пакет, содержащий образцы и шаблоны
«metamodel»	Модель, которая описывает другую модель
«modelLibrary»	Пакет, содержащий определения элементов моделирования, предназначенных для использования в других пакетах
«profile»	Пакет, содержащий определения элементов моделирования, предназначенных для моделирования в определенной предметной области
«stub»	Пакет, представляющий только открытые части другого пакета
«systemModel»	Модель, содержащая несколько моделей одной физической системы
«topLevel»	Пакет, который является концом иерархии вложенности пакетов

Слияние пакетов в UML 2.0

- Очень мощный механизм повторного использования
- Нотация: зависимость со стереотипом «merge» от базы к приращению
- Элементы базы сопоставляются с элементами приращения по именам и метаклассам
- Приращение расширяет базовый пакет непротиворечивым образом
- Правила расширения свои для каждого метакласса

Пример слияния



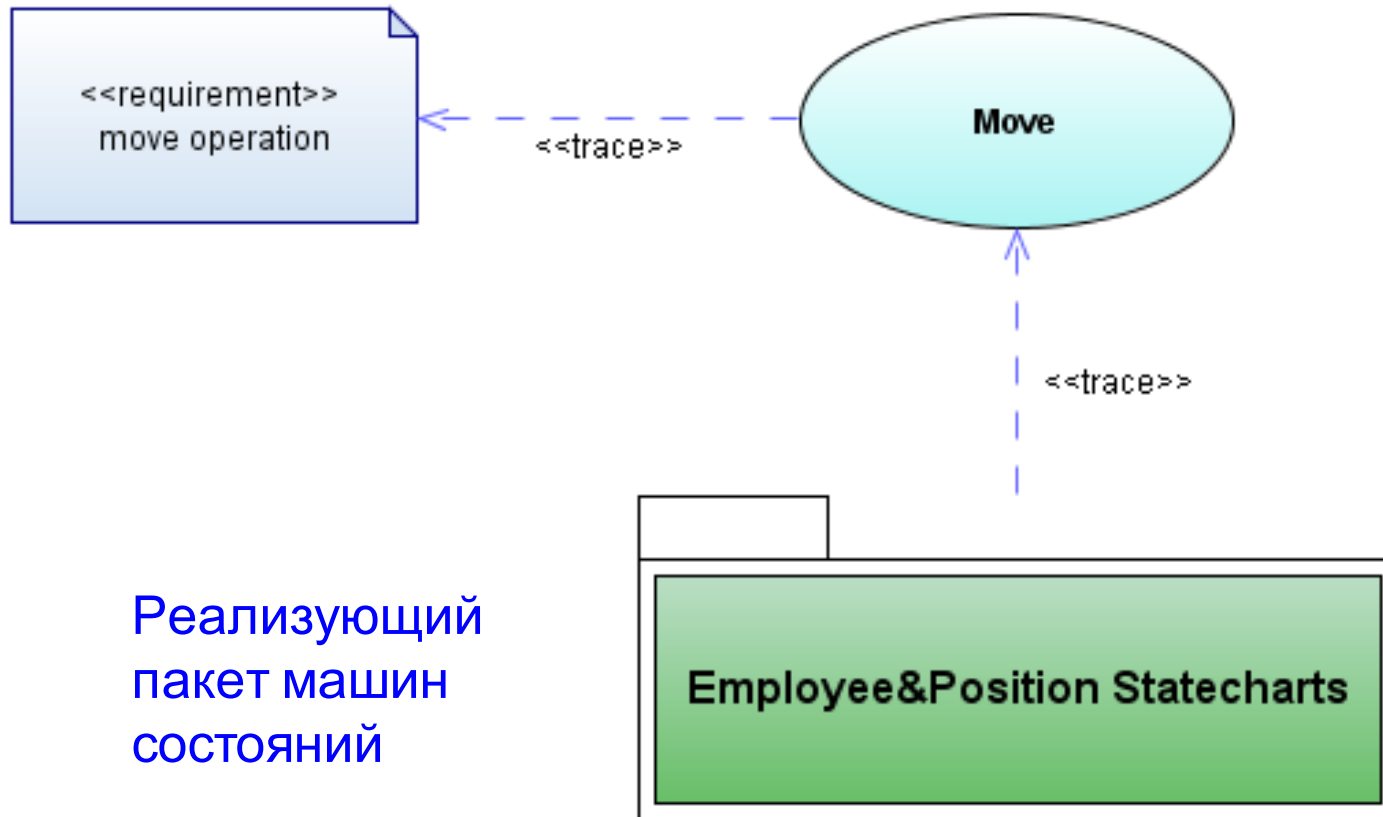
6.2. Трассировка и гиперссылки

- Сосуществование диаграмм, представлений и моделей разного уровня абстракции
- Внесистемное отношение, т.е. отношение между элементами модели, а не модель отношения между моделируемыми сущностями
- Отслеживание версий, уровней детализации и т.д.
- Зависимость со стереотипом «trace»

Пример трассировки

Исходное требование в
форме примечания

Соответствующий
вариант
использования



Реализующий
пакет машин
состояний



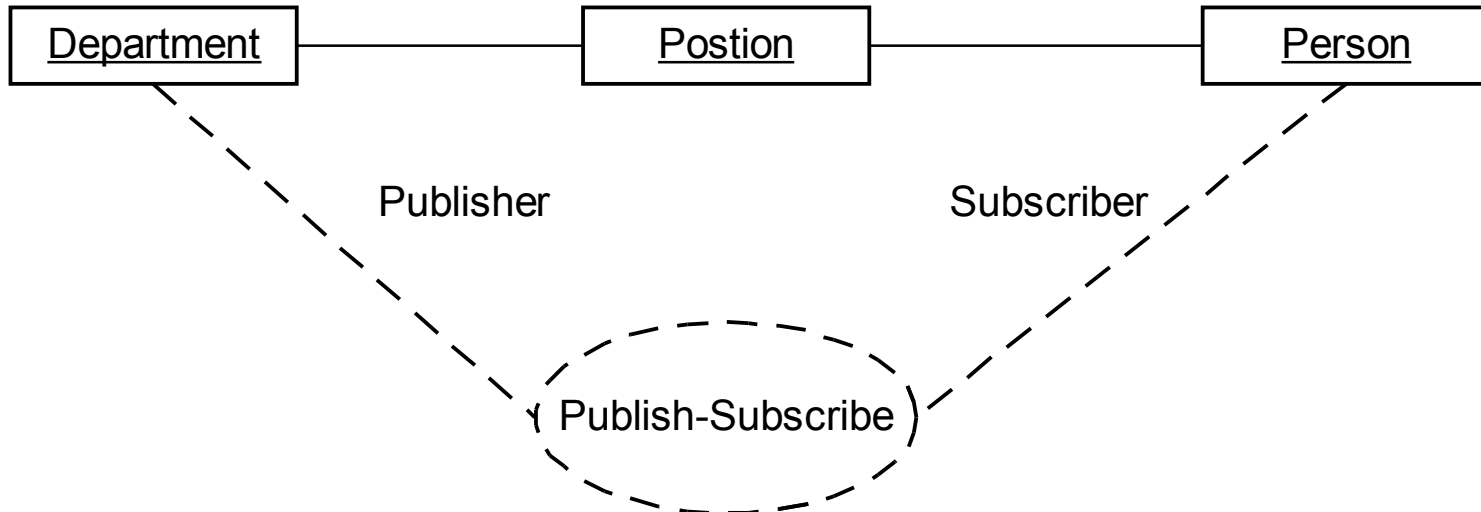
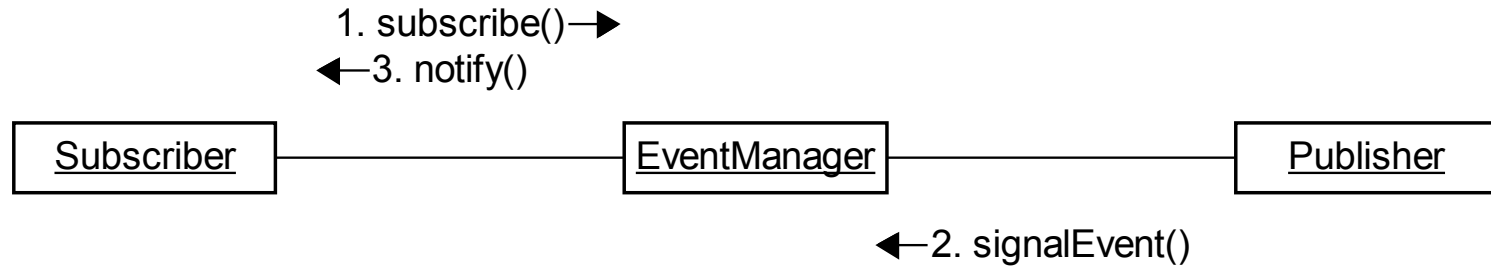
Гиперссылки

- Во многих инструментах можно установить *гиперссылку* на любой элемент или диаграмму
- Гиперссылку можно установить для любого элемента, в том числе вложенного элемента
- Гиперссылок может быть сколько угодно
- Гиперссылки – удобное средство навигации, поддержанное инструментом

6.3. Образцы и каркасы

- **Образец проектирования – типичное решение типичной проблемы в данном контексте**
 - Синтаксически образец UML (pattern) – параметрическая кооперация классов (шаблон кооперации)
- **Каркас – совокупность логически связанных образцов**
 - Синтаксически каркас UML (framework) – пакет со стереотипом «framework»

Пример применения образца Observer = Subscribe Publish



Рекомендации по использованию образцов

- **Образец = публикация обобщенного опыта профессионалов**
 - Образец полезен и разумен
- **Образец решает типичную, но узкую распространенную подзадачу**
 - Образец лаконичен и тривиален
- **Инструменты содержат библиотеки образцов (от 10 до 100)**
 - Названия нужно знать наизусть
- **Корпоративные образцы**
 - Основа реального репозитория

Выводы

- **7 ± 3 сущности на одной диаграмме**
- **Диаграмма должна охватываться «одним взглядом»**
- **Управление моделями – для того, кто моделирует, а не для компьютера**
- **В проекте сосуществуют разные модели в разных представлениях на разных уровнях абстракции**
- **Образцы проектирования полезно знать и применять**