Программная инженерия

Архитектура ПО

Лекция 4

Лекция 4 Архитектура ПО

Цели

1. Определение архитектуры ПО
2. Архитектурные проектные решения, ключевые понятия
3. Классы архитектур ПО
4. Документирование архитектуры, множественность точек зрения

Читать по Лекции 3	
Книга	**Страницы**
Д.В.*Кознов ,и др*. *Введение в программную инженерию*	***Лекция 4: Стр. 20-28***

Архитектура ПО: Структура ПО

Product line (or product family)

Продуктовая линия

Высший уровень абстракции

highest abstraction level

System or product

Система /продукт

Subsystems/Modules

Подсистемы /модули

Packages

Пакеты

Classes/Objects

Классы/объекты

Methods

Методы

Source code

Исходный код

lowest level

Наинизший уровень абстракции

Архитектура ПО

Определение 1: ***Архитектура ПО*** **-** это внутренняя структура программного продукта (компоненты и их связи)
основы пользовательского интерфейса продукта, а также
главные знания и решения, являющиеся инструментом разработки и управления проектом.

Определение 2: ***Архитектура ПО =*** набор **решений** высокого уровня, которые определяют **какими должны**
быть ключевые элементы системы и их связи друг с
другом.

Архитектура ПО

Разработка архитектуры является первым этапом борьбы со сложностью ПО, на котором реализуется принцип выделения относительно независимых компонент

Основные задачи разработки архитектуры ПС:

- выделение программных подсистем и отображение на них внешних функций (заданных во внешнем описании) ПС;
- определение способов взаимодействия между выделенными программными подсистемами.

Различают следующие основные классы архитектур программных средств:

- цельная программа;
- комплекс автономно выполняемых программ;
- слоистая программная система;
- коллектив параллельно выполняемых программ.

Архитектура ПО

Цельная программа представляет вырожденный случай архитектуры ПС: в состав ПС входит только одна программа. Такую архитектуру выбирают обычно в том случае, когда ПС должно выполнять одну какую-либо ярко выраженную функцию и ее реализация не представляется слишком сложной.

Комплекс автономно выполняемых программ состоит из набора программ, такого, что:

любая из этих программ может быть активизирована (запущена) пользователем;

при выполнении активизированной программы другие программы этого набора не могут быть активизированы до тех пор, пока не закончит выполнение активизированная программа;

все программы этого набора применятся к одной и той же информационной среде.

Таким образом, программы этого набора по управлению никак

Архитектура ПО

упорядоченной совокупности программных подсистем, называемых
слоями, такой, что:

- на каждом слое ничего не известно о свойствах (и даже существовании) более высоких слоев;
- каждый слой может взаимодействовать по управлению (обращаться к компонентам) с непосредственно соседствующим более низким слоем через заранее определенный интерфейс, ничего не зная о его внутреннем строении;
- каждый слой располагает определенными ресурсами, которые он
либо скрывает от других слоев, либо предоставляет непосредственно
более высокому слою (через указанный интерфейс) некоторые их
абстракции.

Связи между слоями ограничены передачей значений параметров
обращения каждого слоя к смежному снизу слою и выдачей

Архитектура ПО

Пример слоистой архитектуры операционной системы THE:

Коллектив параллельно действующих программ представляет собой набор программ, способных взаимодействовать между собой, находясь одновременно в стадии выполнения. Это означает, что такие программы, во-первых, вызваны в оперативную память, активизированы и могут попеременно разделять по времени один или несколько центральных процессоров, а во-вторых, выполнять между собой динамические (в процессе выполнения) обмены сообщениями, на базе которых производиться их

Архитектура ПО

В более общем случае коллектив параллельно действующих программ может быть организован в систему с *портами сообщений*.

Порт сообщений представляет собой программную подсистему,

обслуживающую некоторую очередь сообщений: она может принимать на хранение от программы какое-либо сообщение, ставя его в очередь, и может выдавать очередное сообщение другой программе по ее требованию. Сообщение, переданное какой-либо программой некоторому порту, уже не будет принадлежать этой программе (и использовать ее ресурсы), но оно не будет принадлежать и никакой другой программе, пока в порядке очереди не будет передано какой-либо программе

по ее запросу. Таким образом, программа, передающая сообщение не будет находиться в стадии ожидания пока программа, принимающая это сообщение, не будет готова его обрабатывать (если только не будет переполнен

Два способа декомпозиции сложных систем: а)создание проекций системы, б) разбиение на

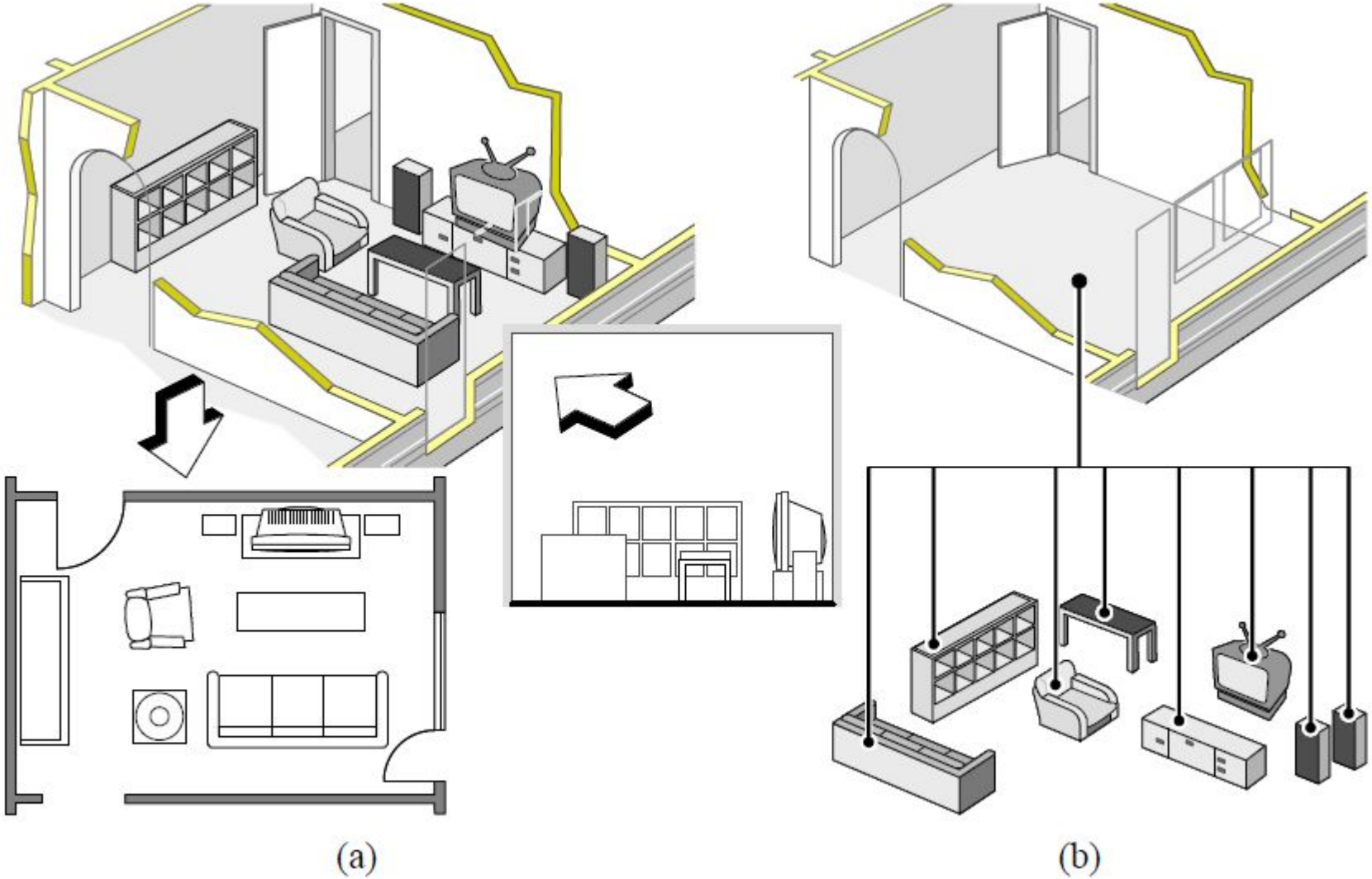
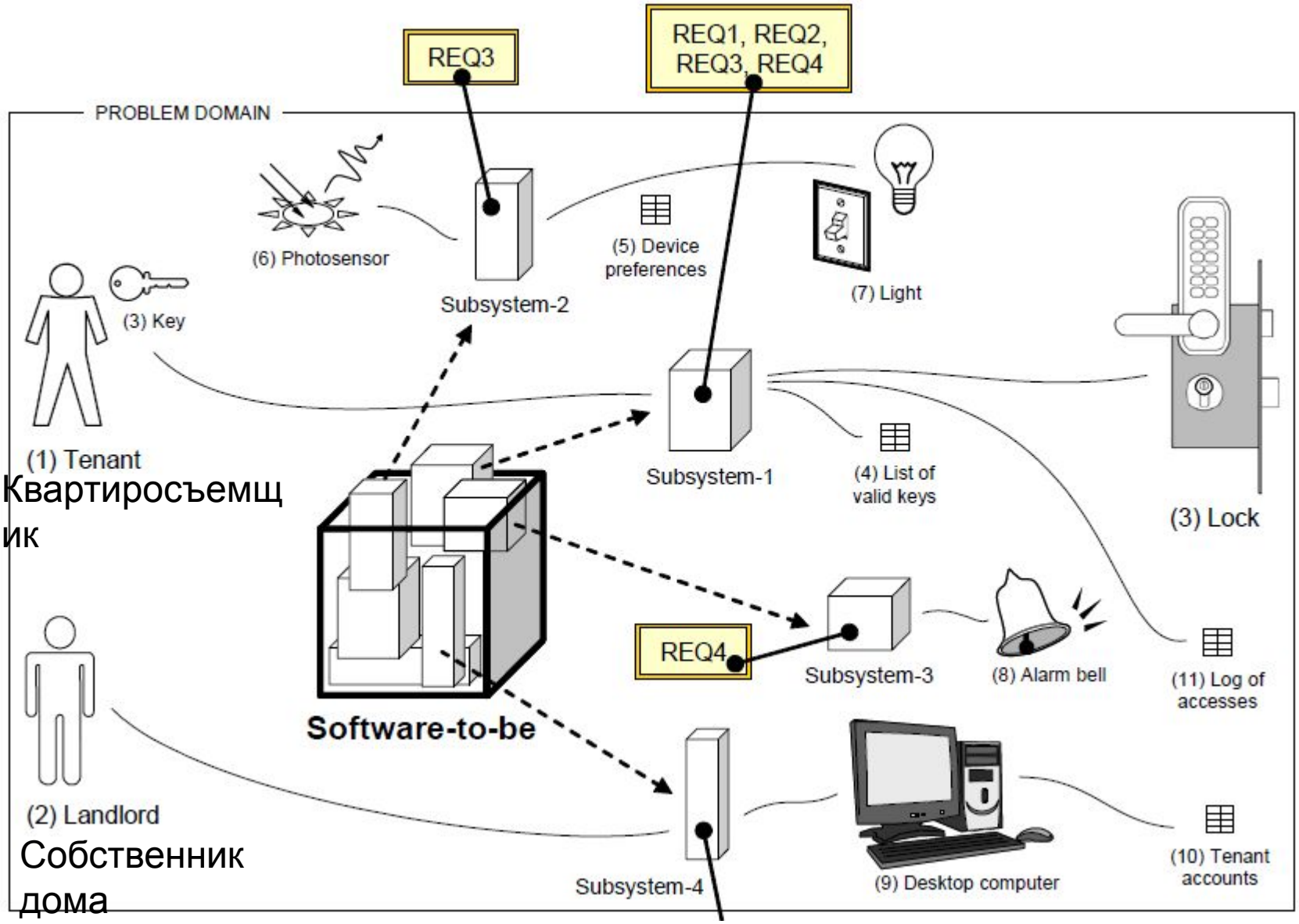


Figure 2-3: Contrasting decomposition types: (a) projection; (b) partition.

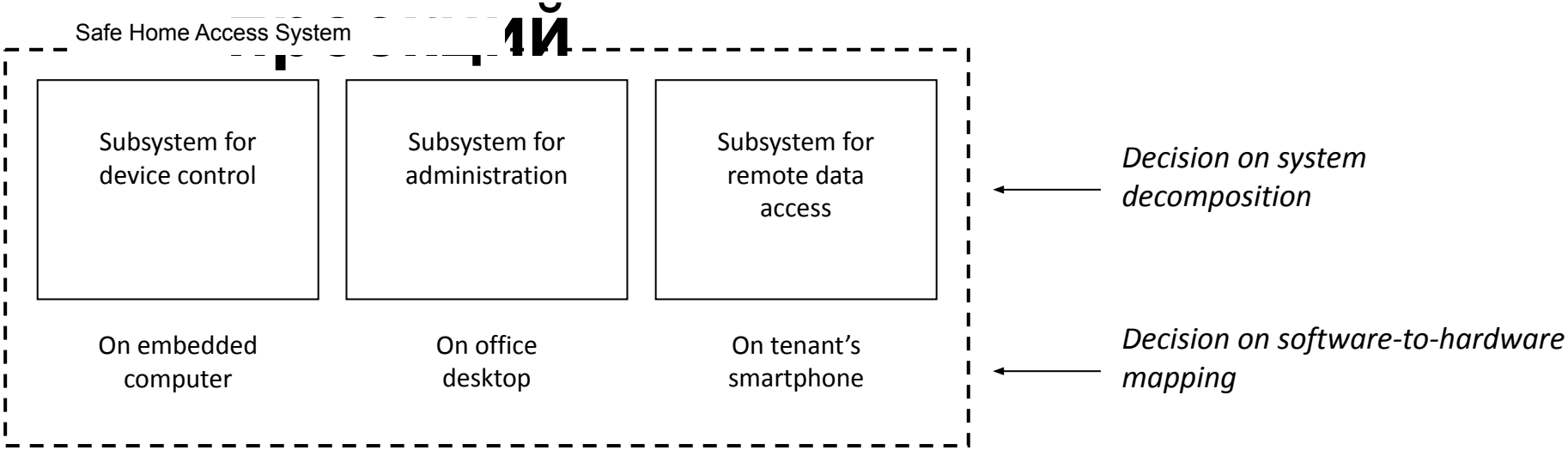
Пример системы безопасности жилого дома



Квартиросъемщик

Собственник дома

Архитектура ПО: примеры



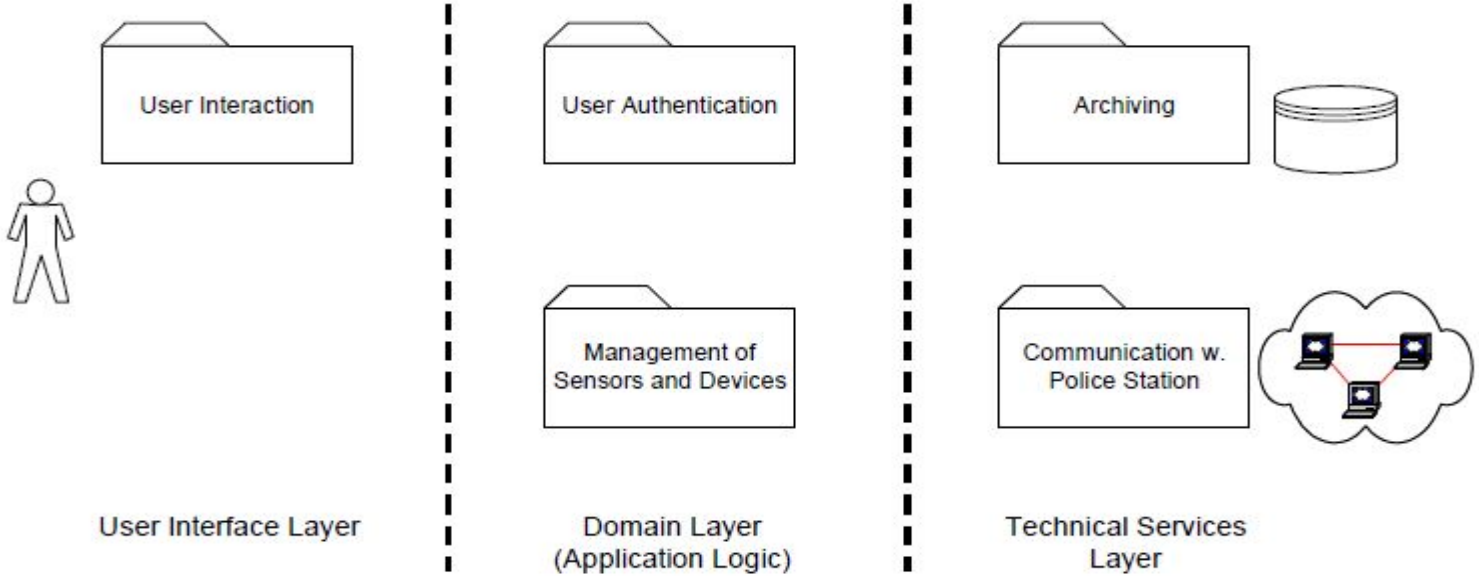


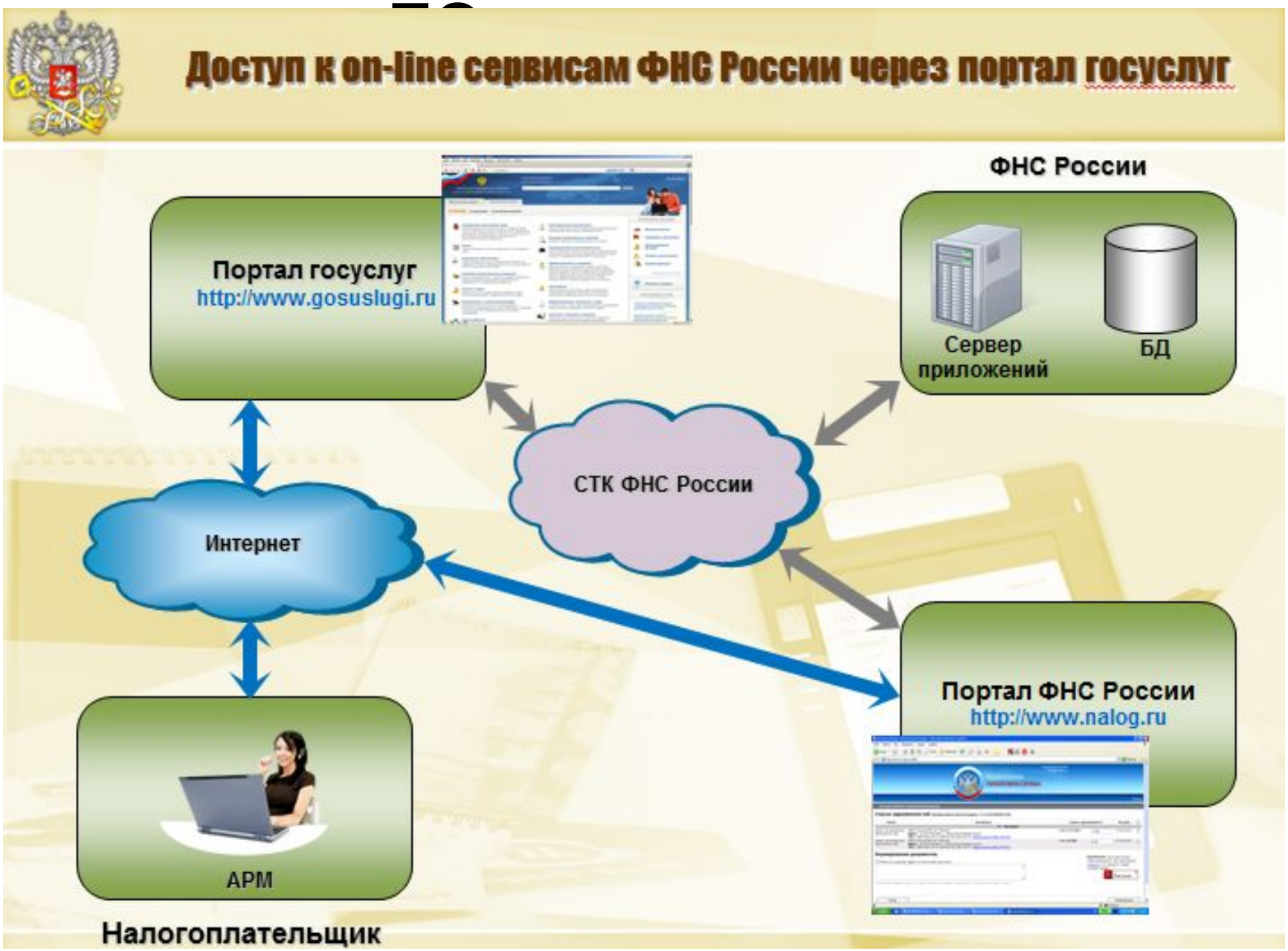
Figure 2-33: Software packages for the case study system. The system has a *layered architecture*, with the three layers as indicated.

Архитектура

Архитектура

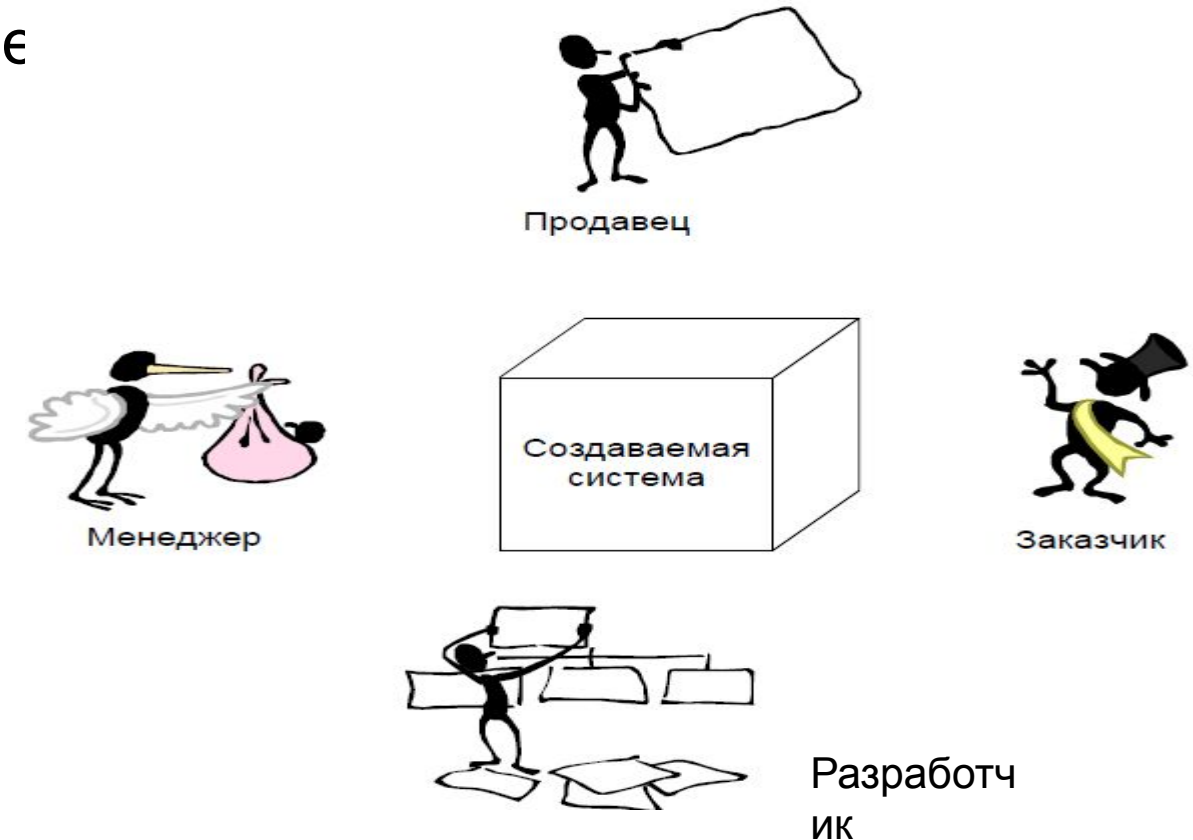
Архитектура ПО

Архитектура



Архитектура ПО: Множественность точек зрения

При разработке архитектуры ПО важным оказывается совмещение множества точек зрения. ПО оказывается настолько сложным, что его архитектуру не построить как единую модель – множество отдельных аспектов должны быть представлены в архитектуре, их связи сложны и плохо выразимы в явном виде. Полезнее оказывается создание множества моде



Архитектура ПО: Множественность точек зрения

Множественность точек зрения происходит также от того, что нет единых стандартов и норм разработки ПО. То есть разработка ПО во многом «state of art».

Точка зрения (viewpoint) — это определенный взгляд на систему, который осуществляется *для выполнения какой-то определенной задачи кем-либо из участников* проекта. Точку зрения нужно ясно осознавать при создании визуальных моделей, например, модели случаев использования. Важно понимать, что она может быть в каждом конкретном случае своя. Важнейшими характеристиками точки зрения моделирования является ***цель*** (зачем создается модель) и ***целевая аудитория*** (то есть, для кого она предназначается).

Архитектура ПО: Документирование

Часто понятие архитектуры сильно сужают, понимая под ним лишь описание основных, важных аспектов ПО, создаваемых, например, архитектором при разработке дизайна системы. Для этих целей используется язык моделирования UML (Unified Modeling Language).

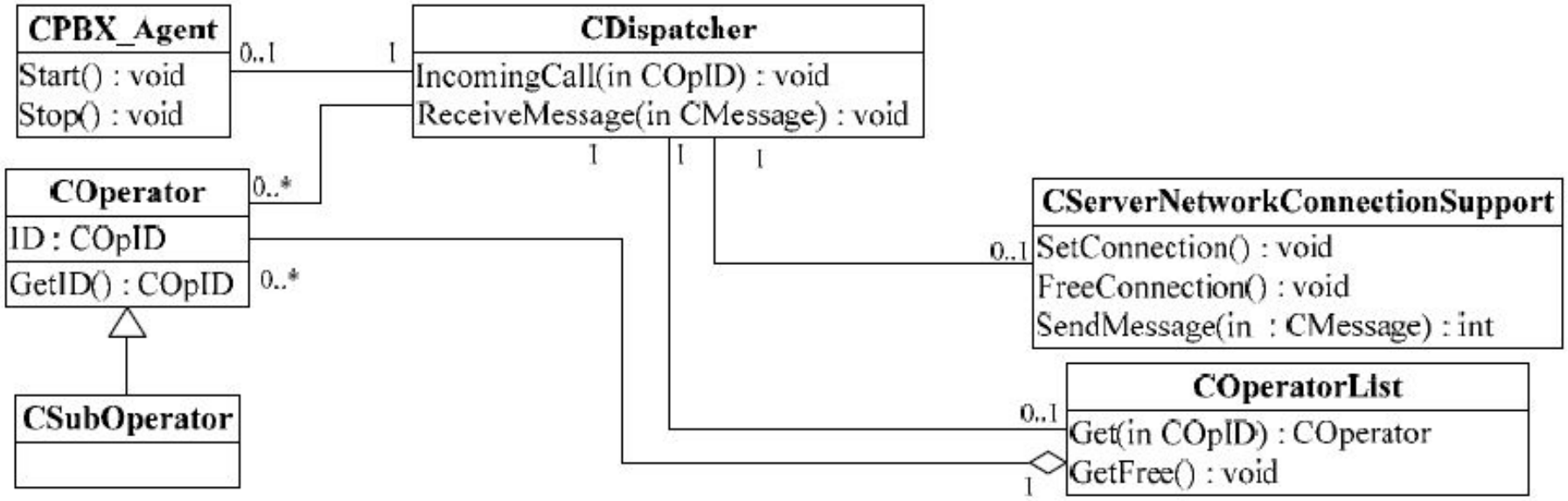


Рис. 4.3. Пример диаграмм классов.

Архитектура ПО: Документирование

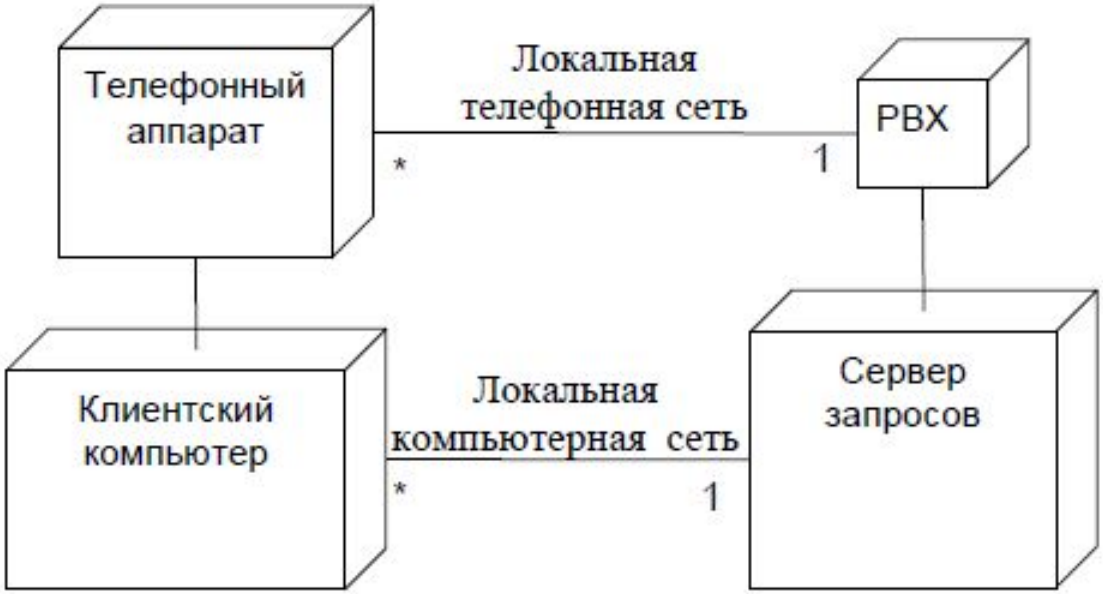
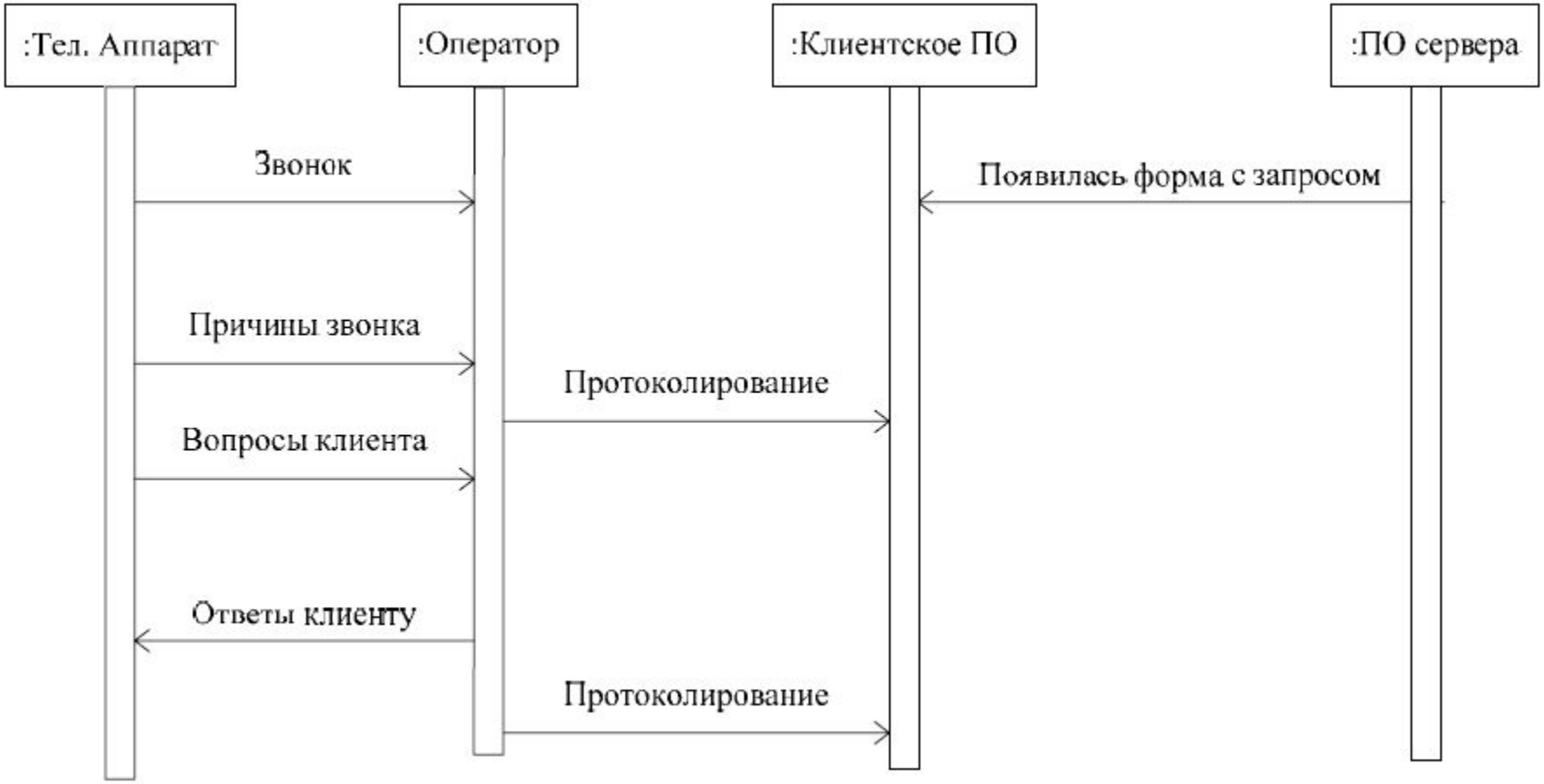


Рис. 4.4. Пример диаграмм размещений.

Архитектура ПО: Документирование



Пример диаграммы последовательностей