**Представление системной архитектуры 4+1**

Программная архитектура многомерна — она состоит из нескольких одновременно развивающихся представлений.

Модель "4+1" предлагает простой и понятный способ описания архитектуры сложных систем, который состоит в использовании пяти различных категорий или представлений (views).

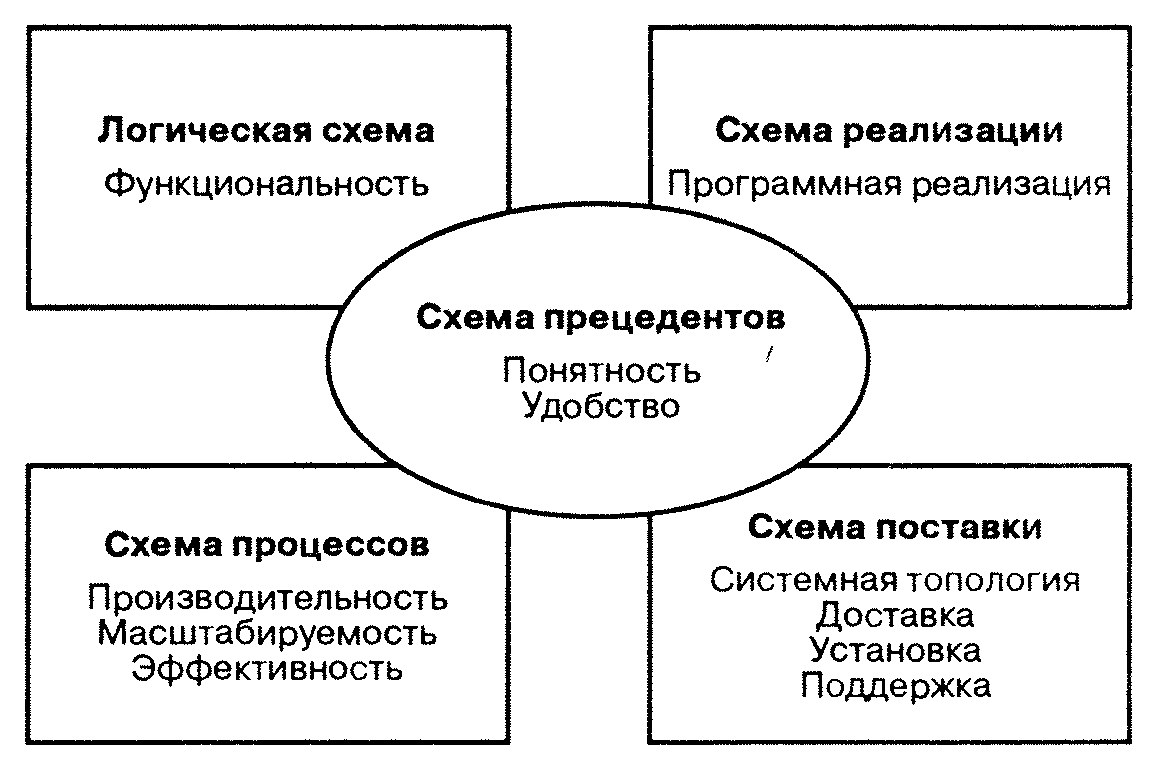


Рис. Модель 4+1

# Логическое представление

Логическое представление (logical view) архитектуры описывает функциональные требования к системе — то, что система должна обеспечивать для обслуживания пользователей. Логическая архитектура отображается на диаграмме классов, содержащей классы и отношения, которые представляют ключевые абстракции разрабатываемой системы.

Выбор ключевых механизмов системы часто называют *тактическим проектированием* (tactical design). «Плохое тактическое проектирование может уничтожить даже тщательно продуманную архитектуру, поэтому разработчики должны уменьшить этот риск, определив ключевые правила проекта». К некоторым ключевым механизмам относятся: язык разработки, хранение данных, удобный пользовательский интерфейс, обработка ошибок, механизмы взаимодействия, распределение и миграция объектов, сетевые средства.

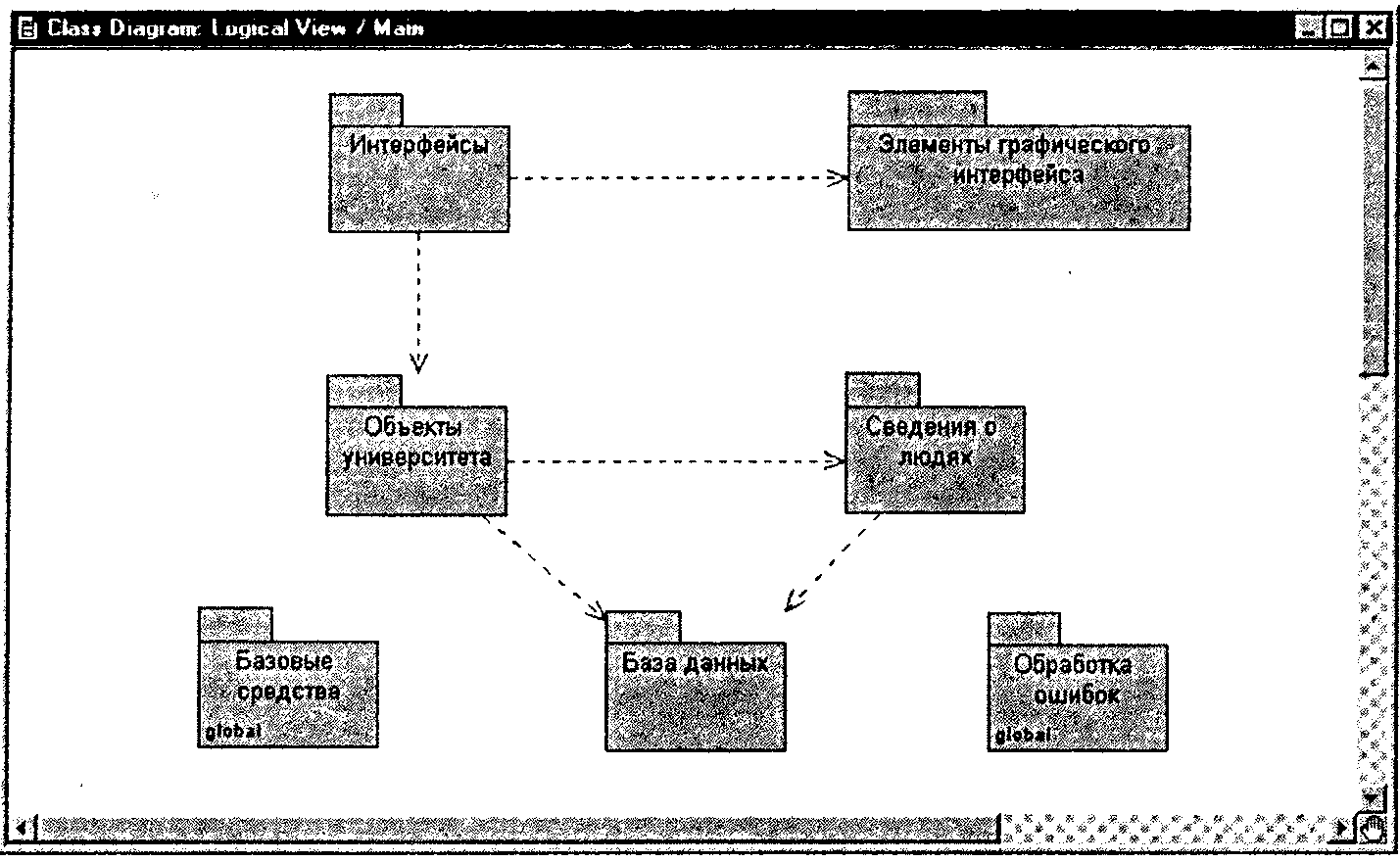


Рис. Схема логического представления

# Представление реализации

Представление реализации (implementation view) определяет реальную организацию программных модулей в среде разработки. Оно учитывает потребности в простоте разработки, управлении программными средствами, повторном использовании кода, а также языковых и инструментальных ограничениях.

Элементами моделирования в *представлении компонентов* (component view) являются пакеты, компоненты и связи между ними.

Пакет в данном представлении архитектуры — это физический раздел системы. Пакеты организованы в виде иерархии уровней или слоев, где каждый уровень имеет четко определенный интерфейс.

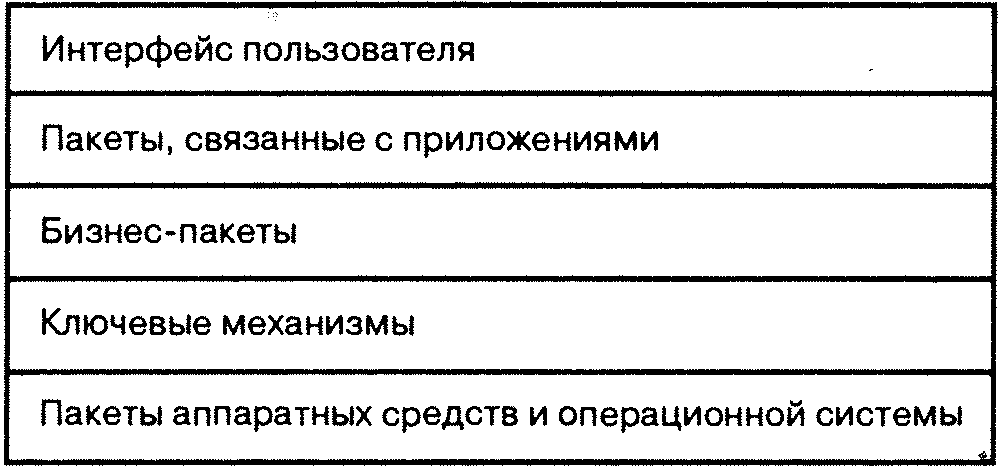


Рис. Схема уровней системы

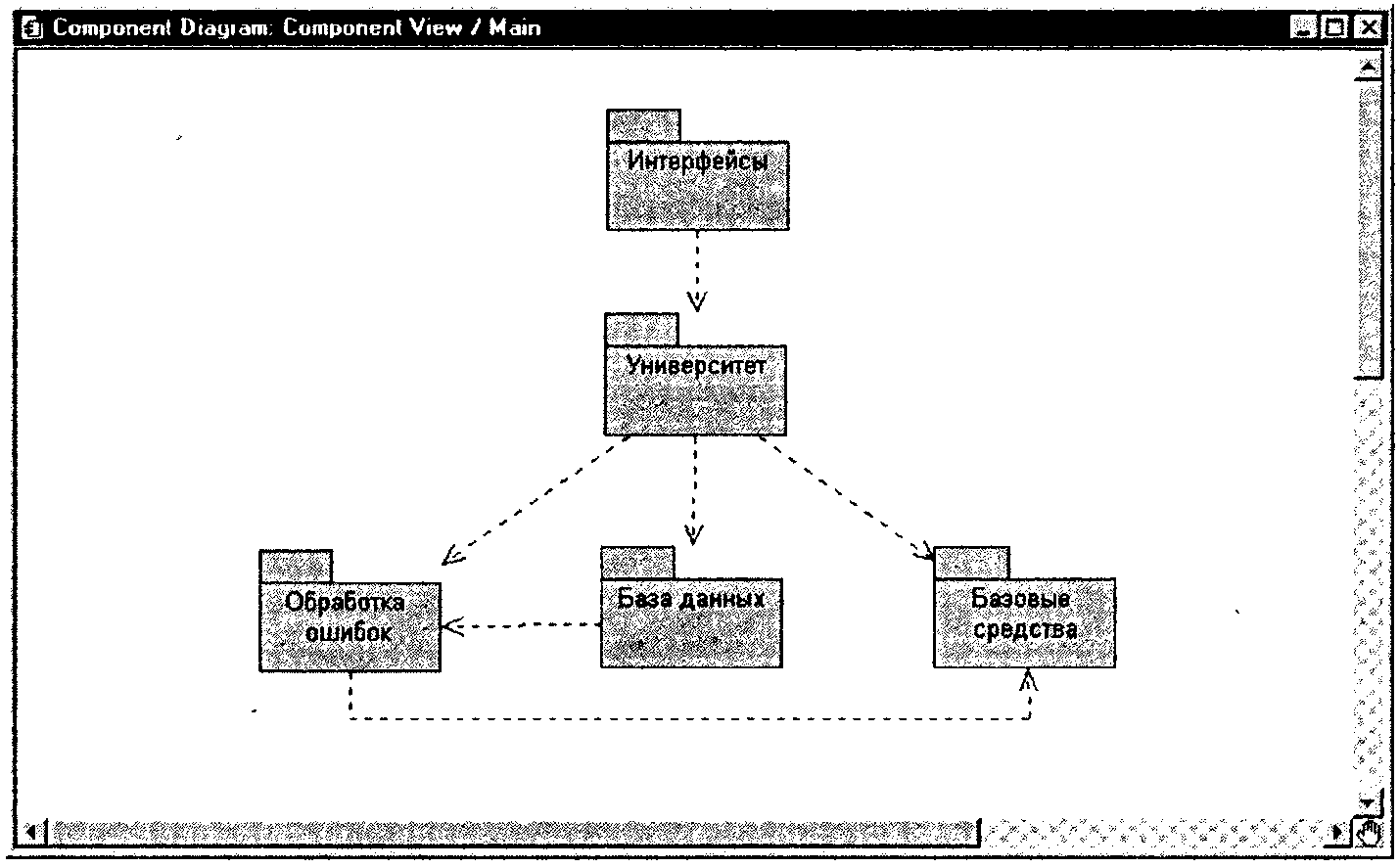


Рис. Схема представления реализации

# Представление процессов

Представление процессов (process view) отражает структуру программной реализации системы. Представление процессов учитывает такие потребности, как производительность, надежность, масштабируемость, целостность, управление системой и синхронизация. Компоненты также используются в этом представлении архитектуры.

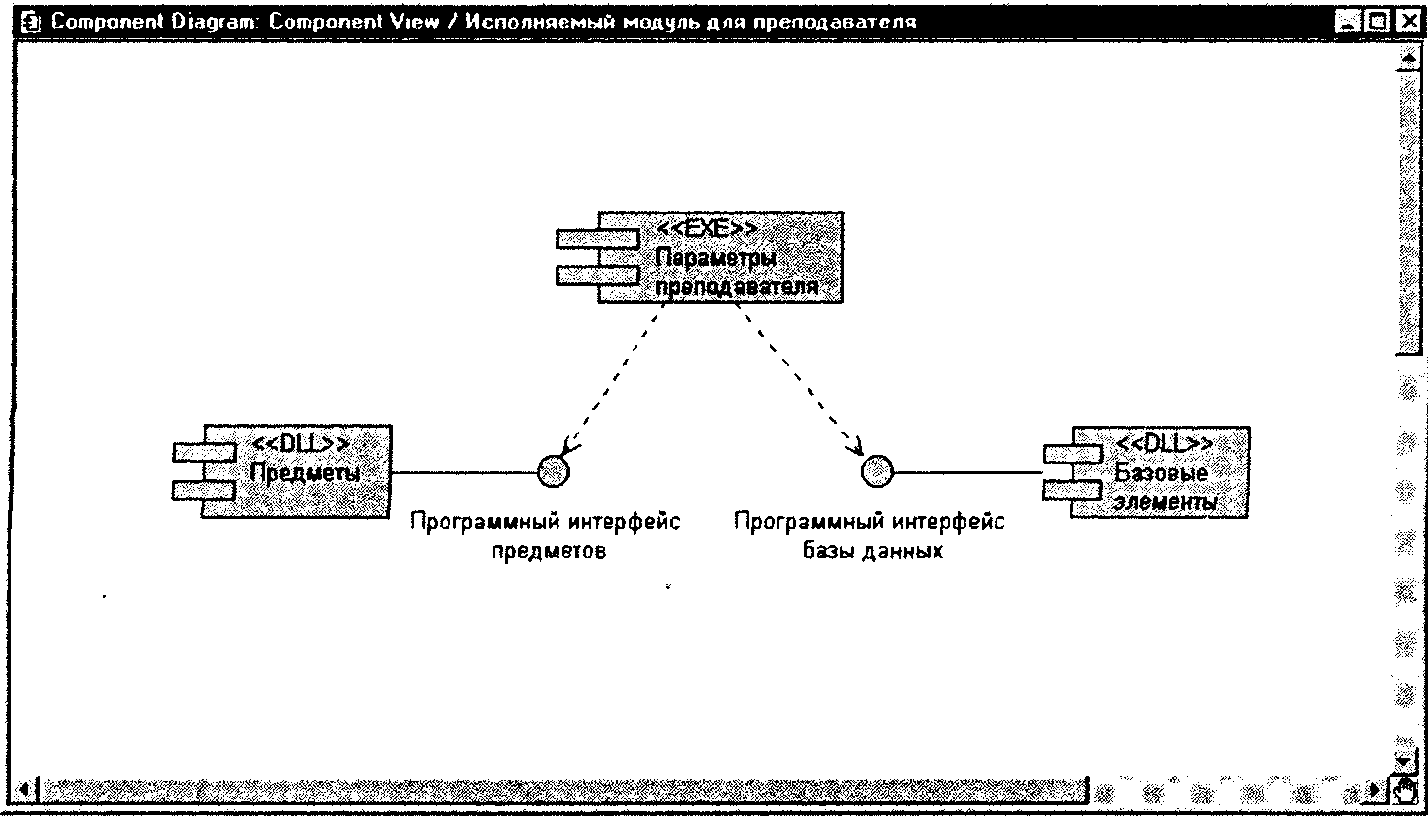


Рис. Схема представления процессов

# Представление средств внедрения

Представление средств внедрения (deployment view) отображает программные средства на узлы вычислительных систем (processing nodes). Оно показывает конфигурацию элементов обработки (processing elements) и работающих на них программных процессов. Представление средств внедрения учитывает такие потребности, как доступность системы, надежность, быстродействие и масштабируемость. Чтобы показать различные узлы вычислительных систем и связи между ними, создаются *диаграммы внедрения* (deployment diagrams). Такая диаграмма демонстрирует распределение компонентов по предприятию. Элементы обработки представлены в виде узлов вычислительных систем, которые соединены линиями, показывающими коммуникационные каналы между ними.

Программные процессы изображаются в виде текста, привязанного к узлу или группе узлов.

Такая диаграмма позволяет разработчикам архитектуры понять топологию системы и отобразить компоненты на исполняемые процессы. Здесь учитываются следующие вопросы: процессорная архитектура, скорость, емкость, пропускная способность каналов для взаимодействия процессов, физическое расположение аппаратных ресурсов, технология распределенной обработки.

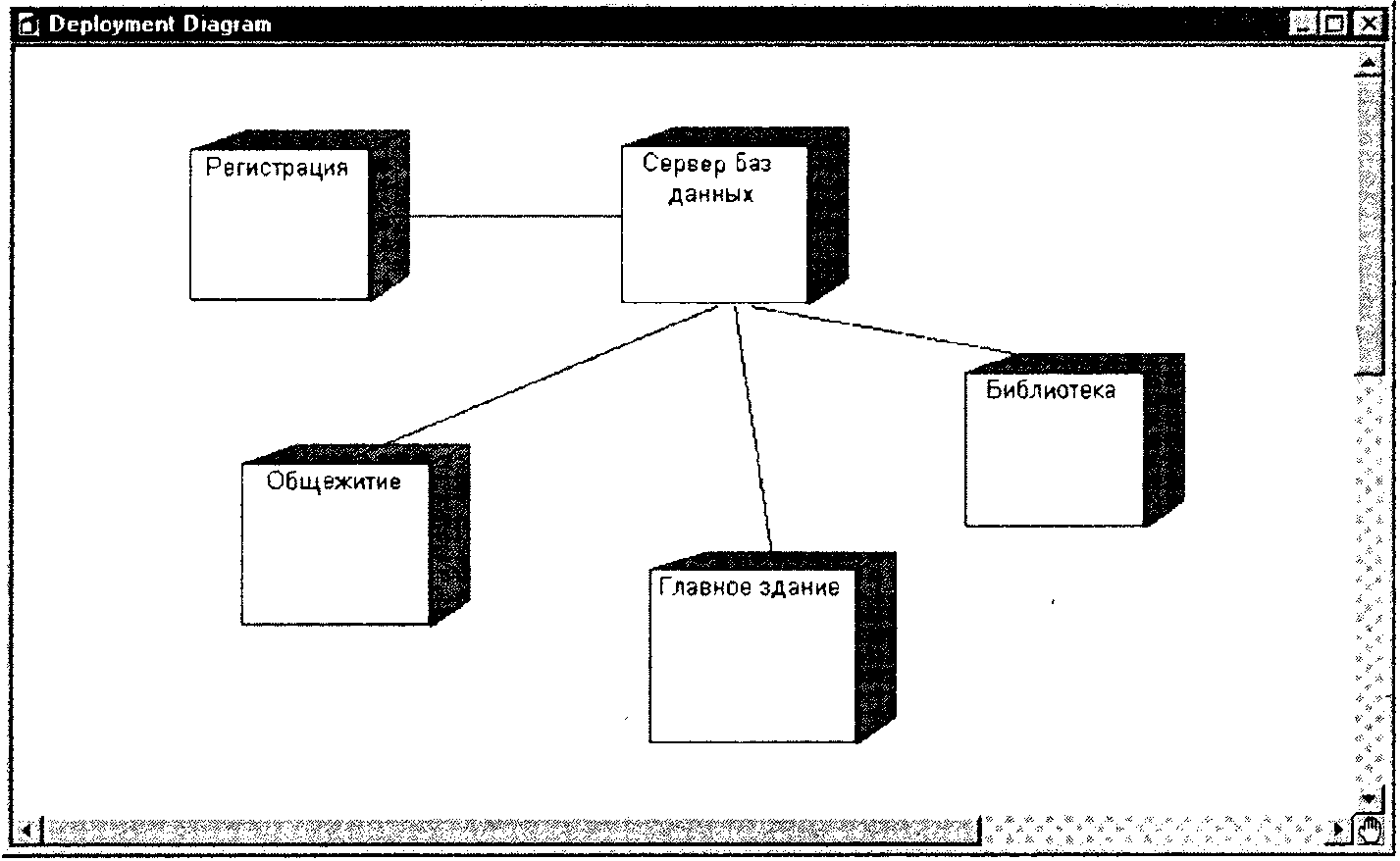


Рис. Схема средств внедрения

# Представление прецедентов

Представление прецедентов (use case view) архитектуры позволяет показать и проверить логическое представление, представления процессов, компонентов и средств внедрения. Диаграммы взаимодействий и диаграммы последовательности действий создаются для того, чтобы продемонстрировать, как различные элементы проектирования взаимодействуют для получения нужного поведения.

Схему смотри в документе 2.2.5 Прецеденты