

Практическая работа № 6

РАЗРАБОТКА ФИЗИЧЕСКОГО ПРЕДСТАВЛЕНИЯ ПРОЦЕССА ФУНКЦИОНИРОВАНИЯ ПРОГРАММНОГО СРЕДСТВА С ИСПОЛЬЗОВАНИЕМ UML

Цель занятия: научиться формировать диаграммы компонентов и диаграммы развертывания для формирования физической модели процесса в рамках которого будет функционировать проектируемое программное средство.

Задание: на основе вариантов использования разработать диаграммы компонентов и развертывания программного средства.

Общие теоретические сведения

Полный проект программной системы представляет собой совокупность моделей логического и физического представлений, которые должны быть согласованы между собой. В языке UML для физического представления моделей систем используются так называемые диаграммы реализации (implementation diagrams), которые включают в себя две отдельные канонические диаграммы: диаграмму компонентов и диаграмму развертывания.

1.1 Диаграмма компонентов

Диаграмма компонентов позволяет определить архитектуру разрабатываемой системы, установив зависимости между программными компонентами, в роли которых может выступать исходный, бинарный и исполняемый код. Во многих средах разработки модуль или компонент соответствует файлу. Пунктирные стрелки, соединяющие модули, показывают отношения взаимозависимости, аналогичные тем, которые имеют место при компиляции исходных текстов программ. Основными графическими элементами диаграммы компонентов являются компоненты, интерфейсы и зависимости между ними.

Диаграмма компонентов разрабатывается для следующих целей:

- визуализации общей структуры исходного кода программной системы;
- спецификации исполнимого варианта программной системы;
- обеспечения многократного использования отдельных фрагментов программного кода;
- представления концептуальной и физической схем баз данных.

Диаграмма компонентов обеспечивает согласованный переход от логического представления к конкретной реализации проекта в форме программного кода. Одни компоненты могут существовать только на этапе компиляции программного кода, другие — на этапе его исполнения. Диаграмма компонентов отражает общие зависимости между компонентами, рассматривая последние в качестве классификаторов.

Для представления физических сущностей в языке UML применяется специальный термин – **компонент** (component). Компонент реализует некоторый набор интерфейсов и служит для общего обозначения элементов физического представления модели. Для графического представления компонента может использоваться специальный символ – прямоугольник со вставленными слева двумя более мелкими прямоугольниками (рис.1). Внутри объемлющего прямоугольника записывается имя компонента и, возможно, некоторая дополнительная информация. Изображение этого символа может незначительно варьироваться в зависимости от характера ассоциируемой с компонентом информации.

В метамодели языка UML компонент является потомком классификатора. Он

предоставляет организацию в рамках физического пакета ассоциированным с ним элементам модели. Как классификатор, компонент может иметь также свои собственные свойства, такие как атрибуты и операции.

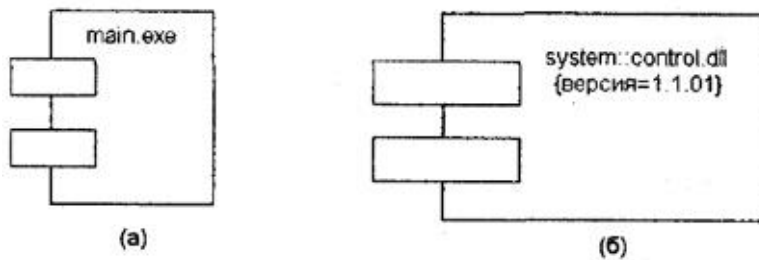


Рис. 1 Графическое изображение компонента в языке UML

Так, в первом случае (рис. 1, а) с компонентом уровня экземпляра связывается только его имя, а во втором (рис. 1, б) — дополнительно имя пакета и помеченное значение.

В языке UML выделяют три вида компонентов.

– компоненты развертывания, которые обеспечивают непосредственное выполнение системой своих функций (такими компонентами могут быть динамически подключаемые библиотеки с расширением dll (рис. 2, а), Web-страницы на языке разметки гипертекста с расширением html (рис. 2, б) и файлы справки с расширением hlp (рис. 2, в)).

– компоненты-рабочие продукты (обычно это файлы с исходными текстами программ, например, с расширениями h или cpp для языка C++ (рис. 10.2, г)).

– компоненты исполнения, представляющие исполнимые модули — файлы с расширением exe.

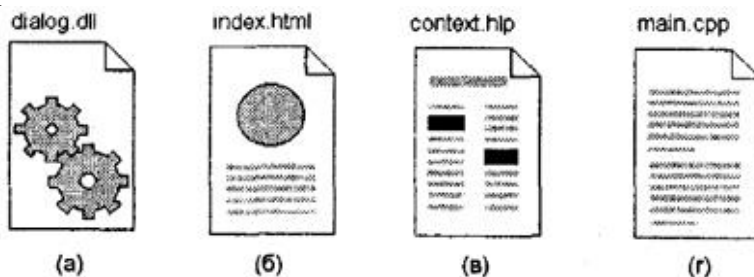


Рис. 2 Варианты изображения компонентов на диаграмме компонентов

Другой способ спецификации различных видов компонентов — явное указание стереотипа компонента перед его именем. В языке UML для компонентов определены следующие стереотипы:

– библиотека (library) – определяет первую разновидность компонента, который представляется в форме динамической или статической библиотеки;

– таблица (table) – также определяет первую разновидность компонента, который представляется в форме таблицы базы данных;

– файл (file) – определяет вторую разновидность компонента, который представляется в виде файлов с исходными текстами программ;

– документ (document) – определяет вторую разновидность компонента, который представляется в форме документа;

– исполнимый (executable) – определяет третий вид компонента, который может исполняться в узле.

Следующим элементом диаграммы компонентов являются **интерфейсы**. В общем случае интерфейс графически изображается окружностью, которая соединяется

с компонентом отрезком линии без стрелок (рис.3, а). При этом имя интерфейса, которое обязательно должно начинаться с заглавной буквы "I", записывается рядом с окружностью. Семантически линия означает реализацию интерфейса, а наличие интерфейсов у компонента означает, что данный компонент реализует соответствующий набор интерфейсов.

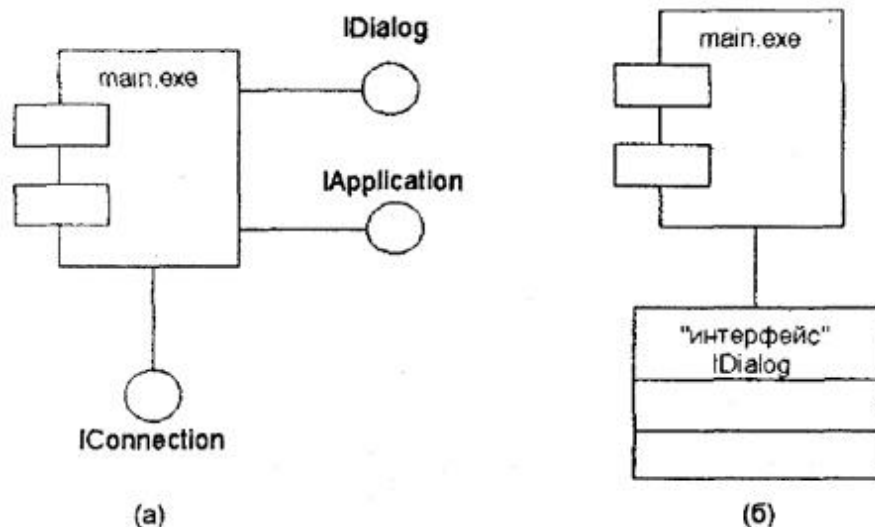


Рис. 3 Графическое изображение интерфейсов на диаграмме компонентов

Другим способом представления интерфейса на диаграмме компонентов является его изображение в виде прямоугольника класса со стереотипом «интерфейс» и возможными секциями атрибутов и операций (рис. 3, б). Как правило, этот вариант обозначения используется для представления внутренней структуры интерфейса, которая может быть важна для реализации.

При разработке программных систем интерфейсы обеспечивают не только совместимость различных версий, но и возможность вносить существенные изменения в одни части программы, не изменяя другие ее части. Таким образом, назначение интерфейсов существенно шире, чем спецификация взаимодействия с пользователями системы (актерами).

Зависимости могут отражать связи модулей программы на этапе компиляции и генерации объектного кода. В другом случае зависимость может отражать наличие в независимом компоненте описаний классов, которые используются в зависимом компоненте для создания соответствующих объектов. Применительно к диаграмме компонентов зависимости могут связывать компоненты и импортируемые этим компонентом интерфейсы, а также различные виды компонентов между собой.

В первом случае рисуют стрелку от компонента-клиента к импортируемому интерфейсу (рис. 4). Наличие такой стрелки означает, что компонент не реализует соответствующий интерфейс, а использует его в процессе своего выполнения. Причем на этой же диаграмме может присутствовать и другой компонент, который реализует этот интерфейс. Так, например, изображенный ниже фрагмент диаграммы компонентов представляет информацию о том, что компонент с именем "main.exe" зависит от импортируемого интерфейса I Dialog, который, в свою очередь, реализуется компонентом с именем "image.java". Для второго компонента этот же интерфейс является экспортируемым.

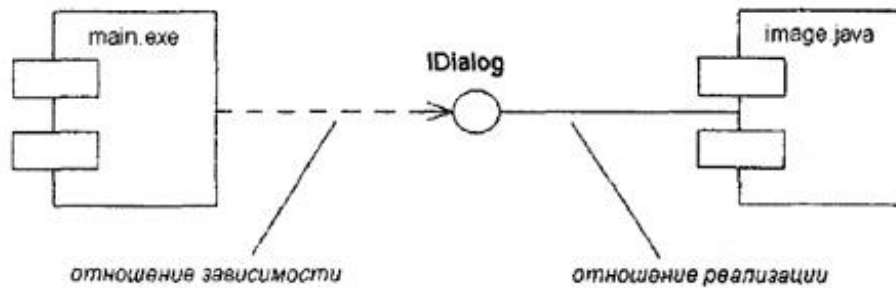


Рис. 4 Фрагмент диаграммы компонентов с отношением зависимости

Заметим, что изобразить второй компонент с именем "image.java" в форме варианта примечания нельзя именно в силу того факта, что этот компонент реализует интерфейс.

Другим случаем отношения зависимости на диаграмме компонентов является отношение между различными видами компонентов (рис. 5). Наличие подобной зависимости означает, что внесение изменений в исходные тексты программ или динамические библиотеки приводит к изменениям самого компонента. При этом характер изменений может быть отмечен дополнительно.

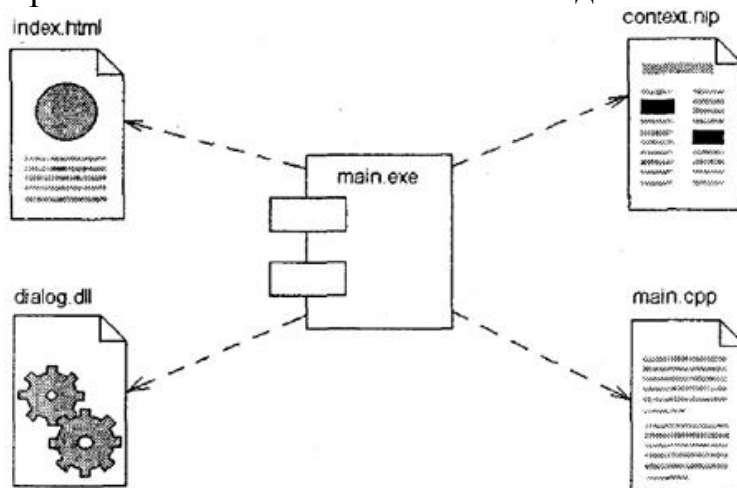


Рис. 5 Графическое изображение отношения зависимости между компонентами

На диаграмме компонентов могут быть представлены отношения зависимости между компонентами и реализованными в них классами. Ниже приводится фрагмент зависимости подобного рода, когда некоторый компонент зависит от соответствующих классов.

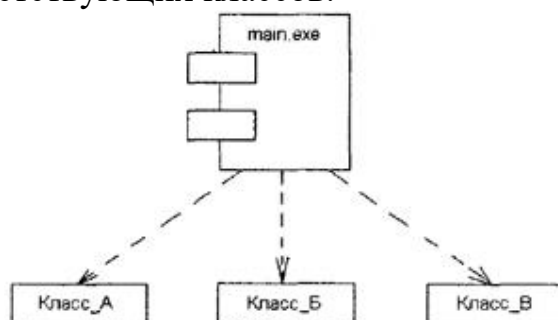


Рис. 6 Графическое изображение зависимости между компонентом и классами

Внутри символа компонента могут изображаться другие элементы графической нотации, такие как классы (компонент уровня типа) или объекты (компонент уровня экземпляра). В этом случае символ компонента изображается так, чтобы вместить эти дополнительные символы (рис. 7).

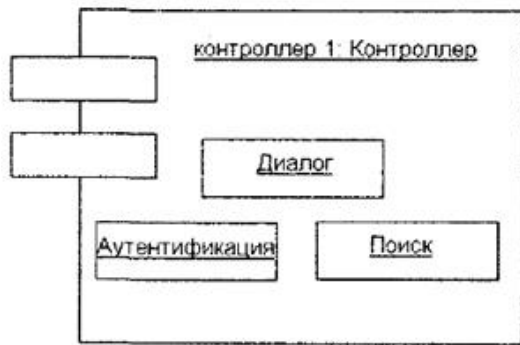


Рис. 7 Графическое изображение компонента уровня экземпляра, реализующего отдельные объекты

1.2 Диаграмма развертывания

Физическое представление программной системы не может быть полным, если отсутствует информация о том, на какой платформе и на каких вычислительных средствах она реализована. Первая из диаграмм физического представления – диаграмма компонентов – рассмотрена выше. Второй формой физического представления программной системы является диаграмма развертывания (синоним – диаграмма размещения). Она применяется для представления общей конфигурации и топологии распределенной программной системы и содержит распределение компонентов по отдельным узлам системы. Кроме того, диаграмма развертывания показывает наличие физических соединений – маршрутов передачи информации между аппаратными устройствами, задействованными в реализации системы.

Диаграмма развертывания предназначена для визуализации элементов и компонентов программы, существующих лишь на этапе ее исполнения (runtime). При этом представляются только компоненты-экземпляры программы, являющиеся исполнимыми файлами или динамическими библиотеками. Те компоненты, которые не используются на этапе исполнения, на диаграмме развертывания не показываются. Так, компоненты с исходными текстами программ могут присутствовать только на диаграмме компонентов. На диаграмме развертывания они не указываются.

Диаграмма развертывания содержит графические изображения процессоров, устройств, процессов и связей между ними. В отличие от диаграмм логического представления, диаграмма развертывания является единой для системы в целом, поскольку должна всецело отражать особенности ее реализации. Эта диаграмма завершает процесс объектно-ориентированного анализа для конкретной программной системы и её разработка, как правило, является последним этапом спецификации модели.

Цели разработки диаграммы развертывания:

- определить распределение компонентов системы по ее физическим узлам;
- показать физические связи между всеми узлами реализации системы на этапе ее исполнения;
- выявить узкие места системы и реконфигурировать ее топологию для достижения требуемой производительности.

Узел (node) представляет собой некоторый физически существующий элемент системы, обладающий некоторым вычислительным ресурсом. В качестве вычислительного ресурса узла может рассматриваться, например, некоторый объем памяти. Понятие узла может включать в себя не только вычислительные устройства (процессоры), но и другие механические или электронные устройства, такие как датчики, принтеры, модемы, цифровые камеры, сканеры и манипуляторы.

Графически на диаграмме развертывания узел изображается в форме трехмерного куба (прямоугольного параллелепипеда). Узел имеет собственное имя, указываемое внутри этого графического символа. Сами узлы могут представляться как в качестве типов (рис.8, а), так и в качестве экземпляров (рис.8, б).

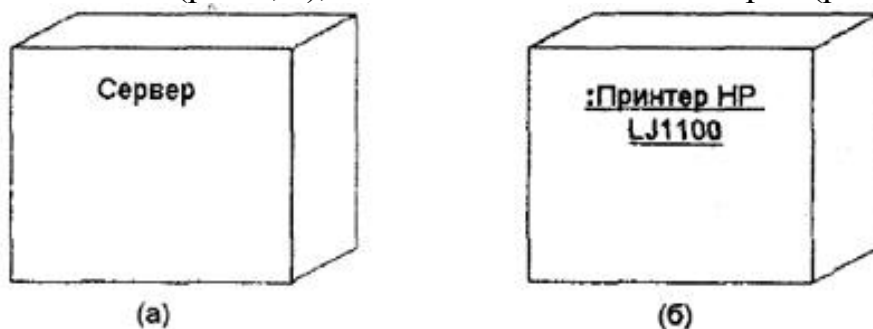


Рис. 8 Графическое изображение узла на диаграмме развертывания

В первом случае имя узла записывается без подчеркивания и начинается с заглавной буквы. Во втором имя узла-экземпляра записывается в виде <имя узла ':' имя типа узла>. Имя типа узла указывает на некоторую разновидность узлов, присутствующих в модели системы.

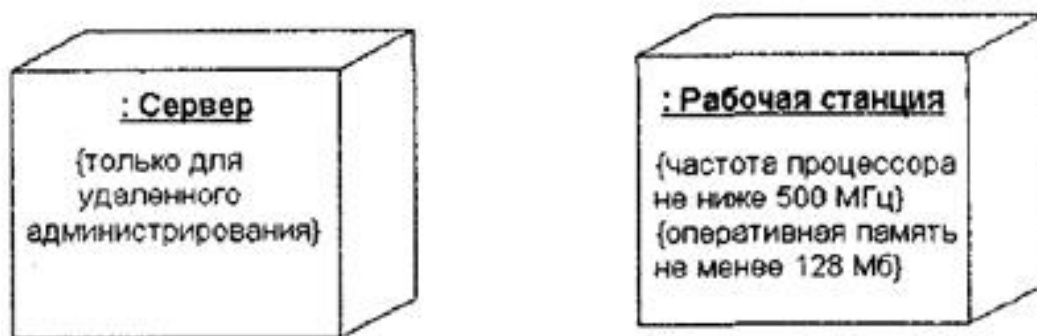


Рис. 9 Графическое изображение узла-экземпляра с дополнительной информацией в форме помеченного значения

Если необходимо явно указать компоненты, которые размещаются на отдельном узле, то это можно сделать двумя способами. Первый из них позволяет разделить графический символ узла на две секции горизонтальной линией. В верхней секции записывают имя узла, а в нижней секции — размещенные на этом узле компоненты (рис. 10, а).

Второй способ разрешает показывать на диаграмме развертывания узлы с вложенными изображениями компонентов (рис. 10, б). Важно помнить, что в качестве таких вложенных компонентов могут выступать только исполняемые компоненты.

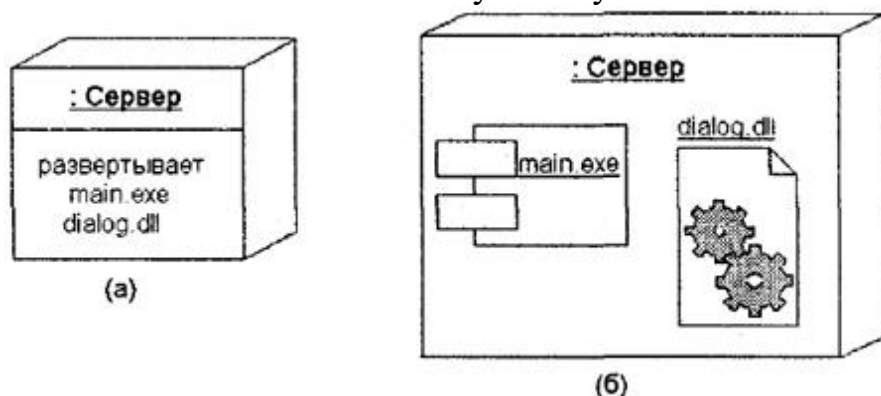


Рис. 10 Варианты графического изображения узлов-экземпляров с размещаемыми на них компонентами

Кроме собственно изображений узлов на диаграмме развертывания указываются

отношения между ними. В качестве отношений выступают физические *соединения* между узлами и зависимости между узлами и компонентами, изображения которых тоже могут присутствовать на диаграммах развертывания.

Соединения являются разновидностью ассоциации и изображаются отрезками линий без стрелок. Наличие такой линии указывает на необходимость организации физического канала для обмена информацией между соответствующими узлами. Характер соединения может быть дополнительно специфицирован примечанием, помеченным значением или ограничением. На представленном ниже фрагменте диаграммы развертывания (рис. 11) определены не только требования к скорости передачи данных по локальной сети с помощью помеченного значения, но и рекомендации по технологии физической реализации соединений в форме примечания.



Рис. 11 Фрагмент диаграммы развертывания с соединениями между узлами

Кроме соединений на диаграмме развертывания могут присутствовать отношения зависимости между узлом и развернутыми на нем компонентами. Подобный способ является альтернативой вложенному изображению компонентов внутри символа узла, что не всегда удобно, поскольку делает этот символ излишне объемным. Поэтому при большом количестве развернутых на узле компонентов соответствующую информацию можно представить в форме отношения зависимости (рис. 12).

Диаграммы развертывания могут иметь более сложную структуру, включающую вложенные компоненты, интерфейсы и другие аппаратные устройства. На изображенной ниже диаграмме развертывания (рис. 13) представлен фрагмент физического представления системы удаленного обслуживания клиентов банка. Узлами этой системы являются удаленный терминал (узел-тип) и сервер банка (узел-экземпляр).

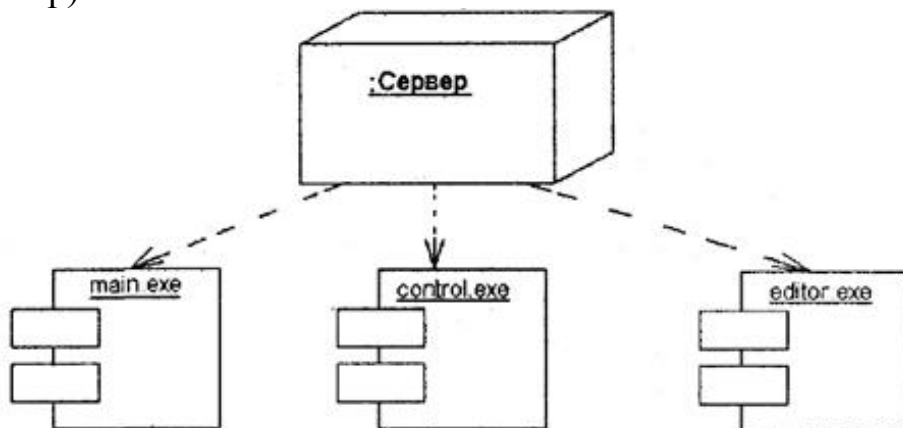


Рис. 12 Диаграмма развертывания с отношением зависимости между узлом и развернутыми на нем компонентами

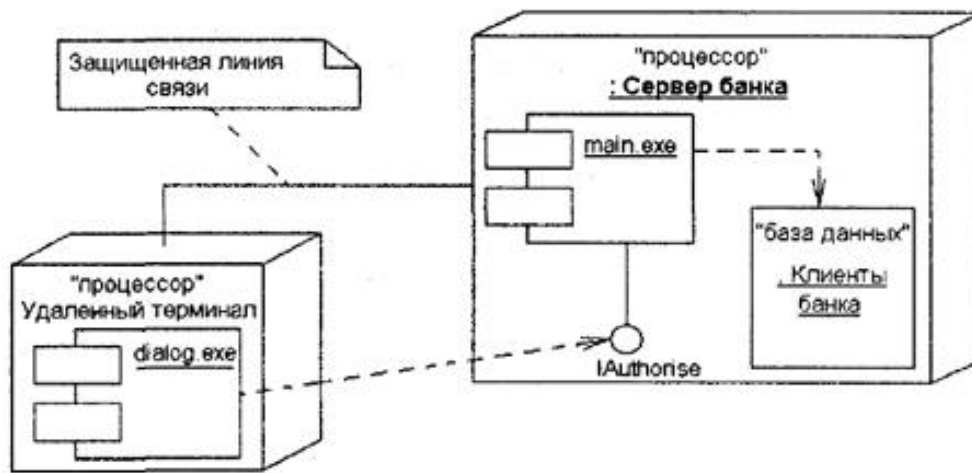


Рис. 13 Диаграмма развертывания для системы удаленного обслуживания клиентов банка

Вариант физического представления этой транспортной системы показан на следующей диаграмме развертывания (рис. 14).

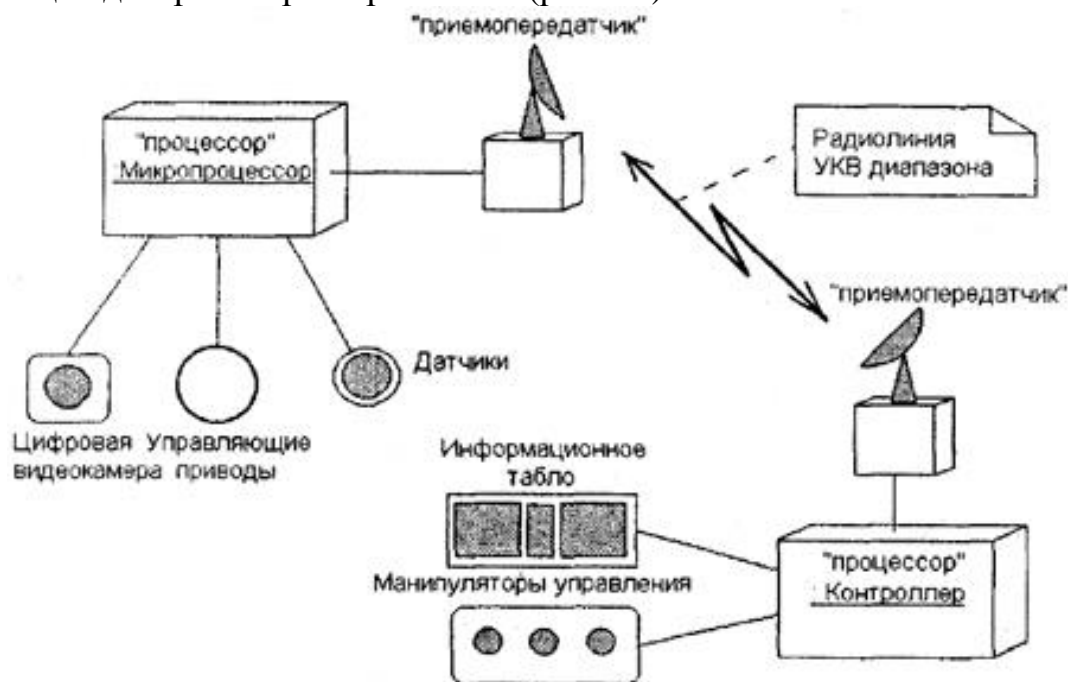


Рис. 14 Диаграмма развертывания для модели системы управления транспортной платформой

Данная диаграмма содержит самую общую информацию о развертывании рассматриваемой системы и в последующем может быть детализирована при разработке собственно программных компонентов управления.