

Глава 9

Системная архитектура: Решения

Решение 4.1 Организация модели системы

Это задание состоит из двух частей, относящихся к одной из двух рассматриваемых в этой книге систем. Первое решение показывает организацию модели для системы управления дорожным перекрестком Рoadраннер. Она показана как в представлении обозревателя модели, так и на диаграмме пакетов.

Сначала, на Рис. 9-1 мы видим диаграмму пакетов, отображающая набор пакетов, используемых для организации модели всей системы. Назначение каждого пакета указано в коротком комментарии. На следующем, Рис. 9-2, показано содержимое системной области модели. Обратите внимание на то, что пакет для системных элементов называется "_Система". Это сделано для того, чтобы он отображался первым в списке, отсортированном в алфавитном порядке в обозревателе модели ¹.

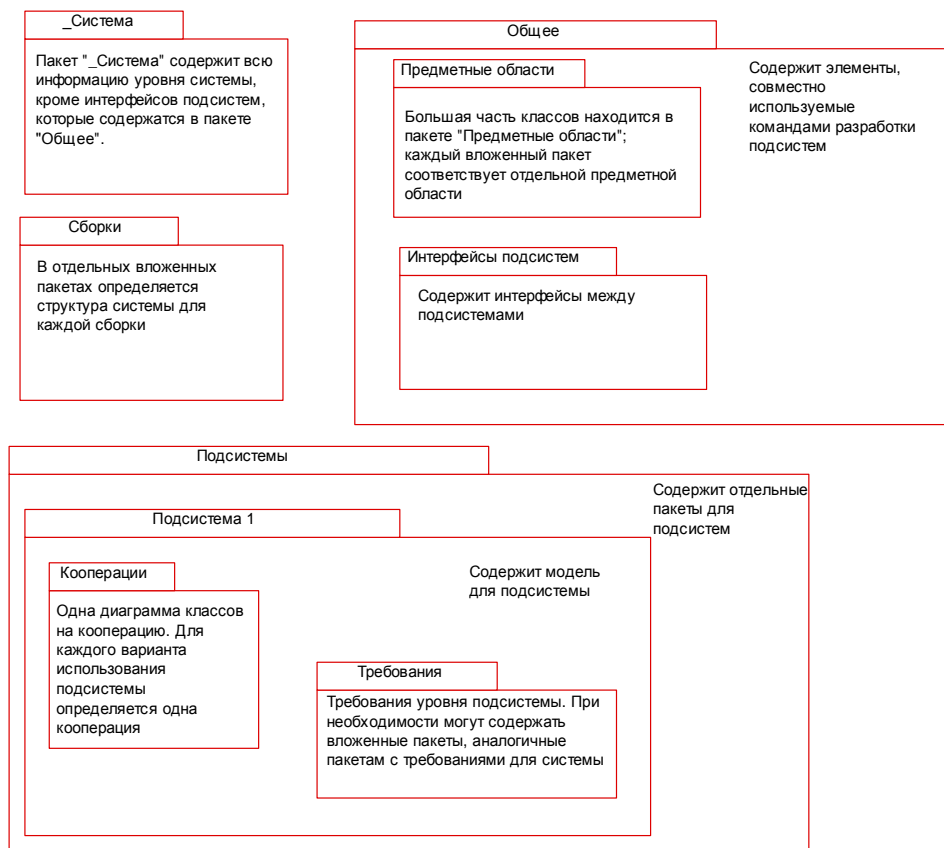


Рис. 9-1 Общая организация модели для Рoadраннер

¹ Это не обязательно делать в Rhapsody, поскольку в Rhapsody вы можете контролировать порядок отображения пакетов в обозревателе модели; это просто моя привычка, которой я всегда следую.

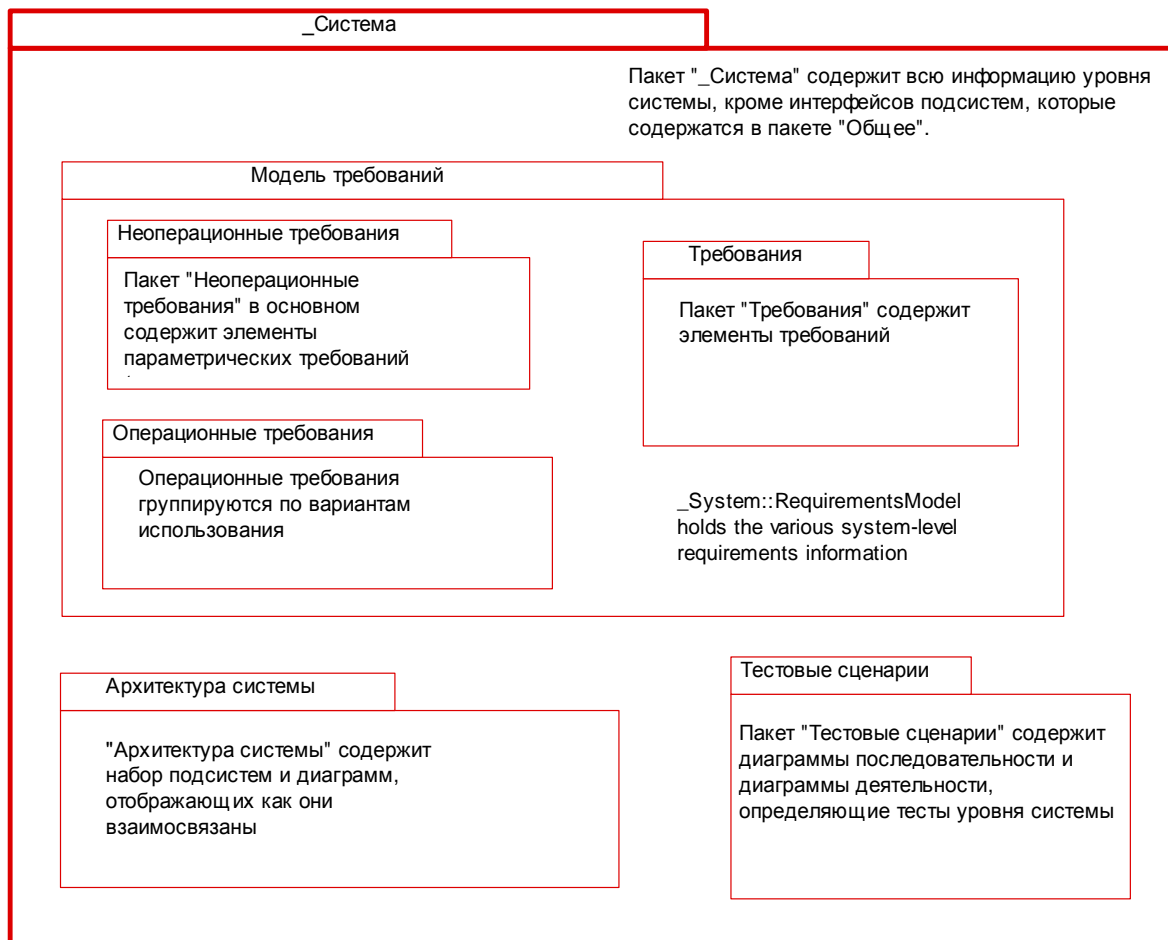


Рис. 9-2 Организация системной области модели для Роадраннер

Пакет “Требования”, находящийся внутри пакета “Модель требований” на Рис. 9-2, содержит формулировки требований, которые относятся к вариантам использования или более детальным операционным элементам; формулировки требований представлены как элементы требований языка SysML. Пакет “Операционные требования” содержит варианты использования и их детальные спецификации с помощью конечных автоматов, диаграмм деятельности и диаграмм последовательности. Пакет “Неоперационные требования” содержит требования, которые относятся к объекту Система и другим неоперационным конструкциям.

На следующих двух рисунках та же самая информация показана в иерархическом виде в окне обозревателя модели. Данный вид представления модели помогает быстро перемещаться между элементами модели; однако, оба этих представления бывают полезны.

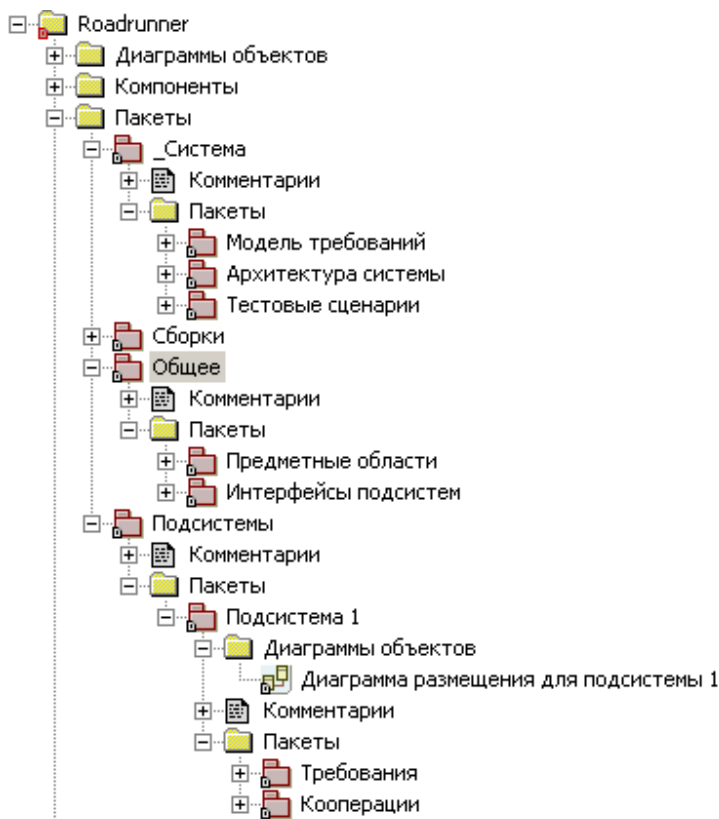


Рис. 9-3 Организация модели для Роадраннер в обозревателе модели

Вторая часть задания состоит в разбиении модели БЛА Койот на несколько взаимодействующих моделей в соответствии с инструкциями в Главе 4. Размеры системы БЛА Койот на 2 или 3 порядка больше размеров системы управления дорожным перекрестком, и поэтому, чтобы обеспечить эффективность работы, она должна описываться с использованием нескольких моделей:

- Системная модель
- Общая модель
- (n) Модели подсистем

Системная модель будет использоваться для определения системных требований и системной архитектуры. Данная модель содержит пакеты для подсистем, как раз те пакеты, которые будут переданы командам разработки подсистем при переходе проекта к фазе разработки подсистем. Эти пакеты содержат модели со спецификациями подсистем, на основе которых будут разрабатываться детальные модели подсистем. Они будут содержать только требования и элементы, относящиеся к требованиям. Внутренняя структура подсистем будет определяться командами разработки подсистем непосредственно в моделях подсистем.

Общая модель содержит элементы, совместно используемые командами разработки подсистем; к таким элементам относятся интерфейсы подсистем и спецификации классов, относящимся к нескольким предметным областям (более подробно о предметных областях будет рассказываться далее в этой главе). Наконец, для каждой команды разработки подсистем существует отдельная модель подсистемы, которую можно рассматривать как модель системы меньшего масштаба. Другими словами, модели подсистемы организуется точно также, как модель системы. Она содержит пакет для требований к подсистеме и пакет для определения структуры подсистемы, которая состоит из взаимодействующих объектов, входящих в подсистему.

Центральным вопросом декомпозиции системы на несколько взаимодействующих друг с другом моделей, является вопрос как совместно использовать эти более или менее независимые, но связанные друг с другом модели. Внешние модели могут добавляться в текущую модель в виде копий и в виде ссылок. Когда вы добавляете внешнюю модель как копию, пользователь импортирует в модель копию внешней модели. При этом не существует никаких ограничений для манипуляции пользователем элементами скопированной модели; можно свободно добавлять или удалять элементы к созданной копии, не изменяя исходную модель. Если вы импортируете внешнюю модель в виде ссылке, то все изменения, которые выполняются в импортированной модели, сразу же отображаются в исходной модели. В Rhapsody, импорт внешних моделей реализуется при помощи команды “Добавить в Модель”. Эта команда позволяет загрузить внешнюю модель целиком, часть ее пакетов или отдельный элемент, как в виде копии, так и в виде ссылки. Для моделей, которые вы возьмете и будете уточнять— например, модели подсистем — импортирование модели в виде копии является наиболее предпочтительным вариантом. Модели, из которых вы хотите совместно использовать имеющиеся элементы, например, модели, содержащей общие классы, используемые в различных подсистемах, лучше импортировать в виде ссылки.

Итак, к данному моменту мы определили несколько моделей:

- Системная модель
- Модель сборки
- Общая модель
- (n) Модель подсистем

Содержимое системной модели практически совпадает с содержимым пакета “_Система” в решении для Роадраннер. Оно отличается в основном тем, что, хоть и содержит отдельные пакеты для каждой подсистемы, эти пакеты содержат только требования и связанные с ними элементы (варианты использования и т.п.). Они не содержат определения внутренней структуры подсистем, то есть результатов работы команд разработки подсистем, которые определяют их в отдельных моделях.

Модель сборки можно рассматривать как «мастер» модель, которая содержит очень мало собственных элементов, но импортирует все модели подсистем, чтобы создать согласованный исполняемый экземпляр системы (т.е. ее «прототип»). Ранние сборки могут быть неполными: в них могут отсутствовать некоторые подсистемы, не использоваться электронные или механические компоненты, либо представлена не вся

функциональность отдельных подсистем. Тем не менее, даже эти первые сборки системы могут быть исполнены и оценены на соответствие требованиям. В дальнейшем сборки системы будут включать в себя все больше функций и использовать все больше целевой аппаратуры.

Общая модель служит для тех же целей, что и пакет “Общее” в решении с единой моделью: в нее помещаются совместно используемые элементы, благодаря чему они могут быть доступны в различных моделях, в которых они нужны. .

И, наконец, разработка моделей подсистем начинается с импорта спецификаций для этих подсистем из системной модели. Эти модели затем независимым образом уточняются и детализируются в процессе анализа и проектирования подсистем.

Системная модель БЛАК

Системная модель показана на Рис. 9-5 (на диаграмме пакетов) и на Рис. 9-6 (в обозревателе модели).

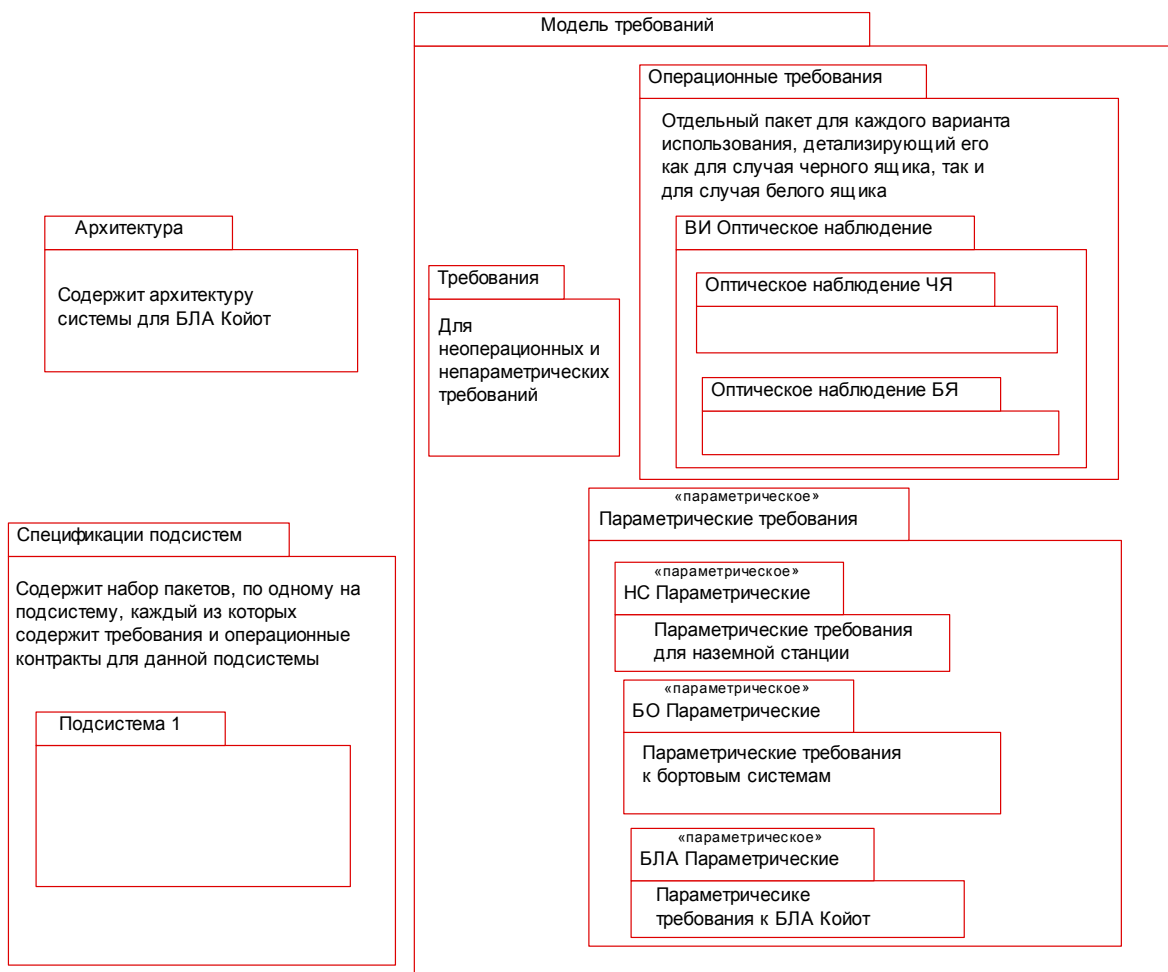


Рис. 9-5 Организация системной модели БЛАК на диаграмме пакетов

Организация общей модели показана на Рис. 9-7.

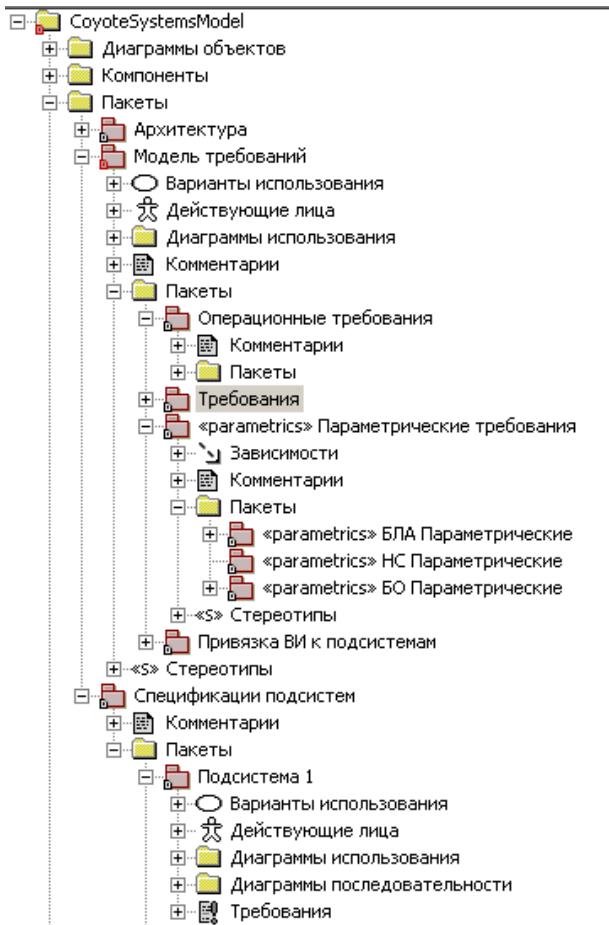


Рис. 9-6 Организация системной модели БЛАК в обозревателе модели

Структура общей модели Койот

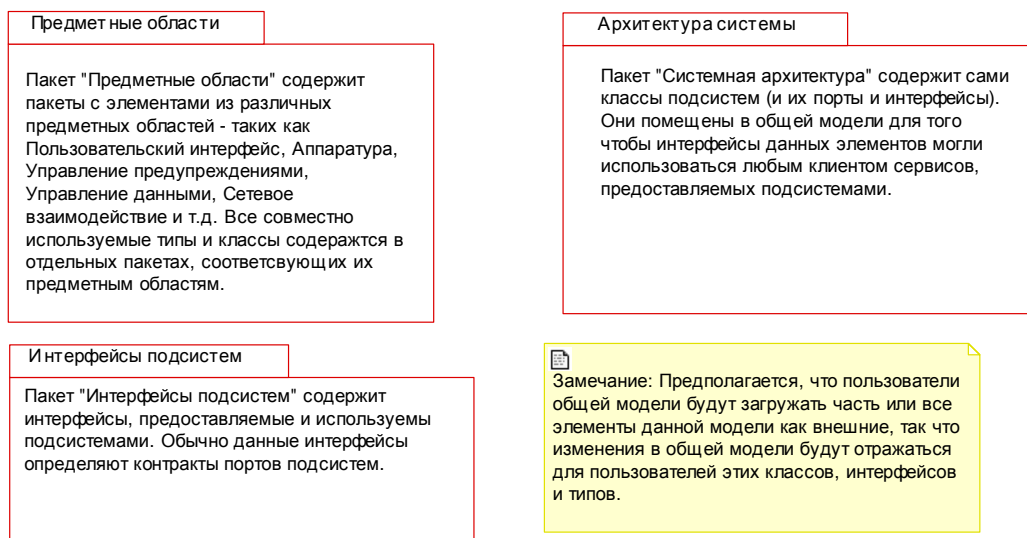


Рис. 9-7 Организация общей модели БЛАК

И, наконец, пример модели подсистемы (только одной, поскольку мы пока еще не определили подсистемы нашей системы) показана на Рис. 9-8. По мере того как архитектура будет уточняться, для каждой из определенных подсистем будет создана отдельная модель подсистемы.

Организация модели подсистемы

Назначение: Отображение логической архитектуры (т.е. структуры модели) для модели подсистемы

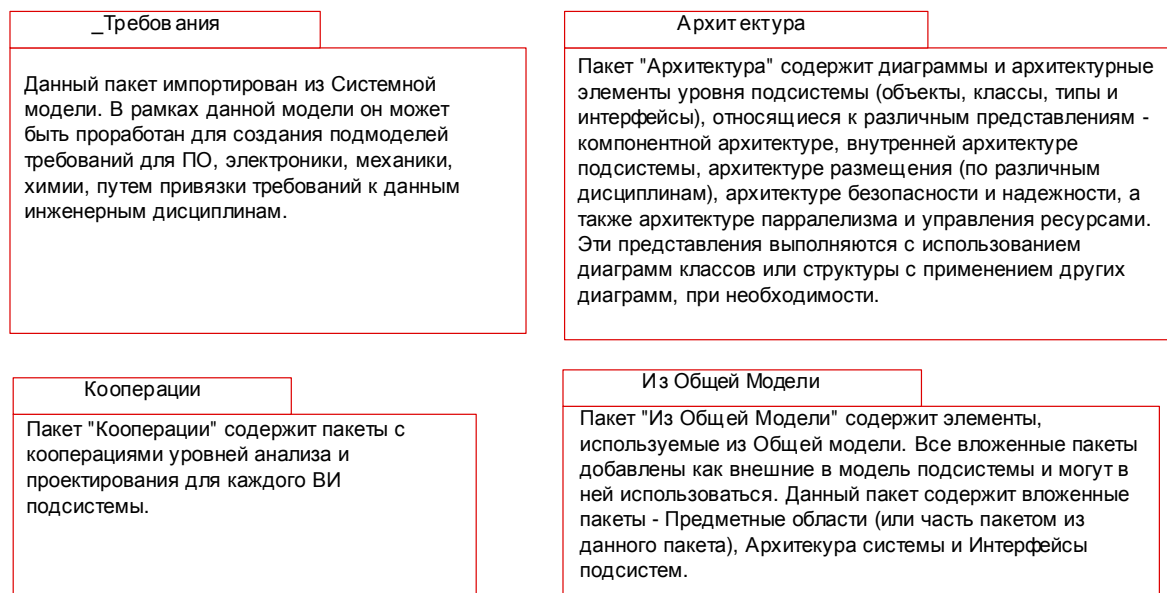


Рис. 9-8 Организация модели подсистемы БЛАК

Решение 4.2 Определение подсистем

Система управления дорожным перекрестком Родраннер – небольшая система, поэтому диаграмма с определением системной архитектуры для нее будет достаточно простой. Для отображения системной архитектуры я воспользовался структурной диаграммой UML. Обратите внимание на стереотипы «Система» и «Подсистема», используемые для элементов архитектуры. Их использование необязательно, но я считаю их полезными для пояснения назначения отдельных элементов модели. Следует заметить, что ни один из элементов на Рис. 9-9 не является «примитивным» — каждый из них содержит большое количество внутренних частей.

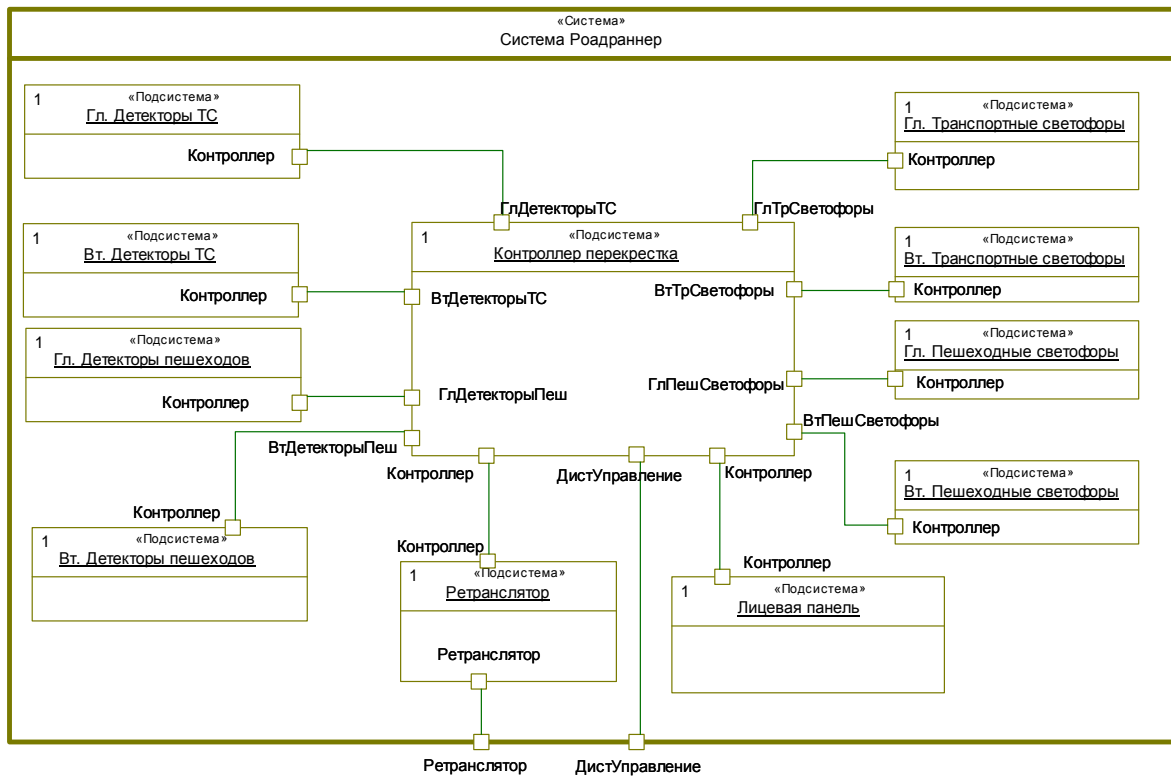


Рис. 9-9 Диаграмма с системной архитектурой для Родраннер

Порты между подсистемами системы Родраннер еще (пока) не типизированы соответствующими интерфейсами, однако связи между элементами уже понятны. Вы легко можете представить себе какие связи будут необходимы для работы системы, а также, немного подумав, определить сервисы, предоставляемые и запрашиваемые через эти порты. Эти интерфейсы будут определены позднее при выполнении заданий в этой главе.

Обратите внимание на подсистемы, определенные для системы управления дорожным перекрестком (все подсистемы показаны на Рис. 9-9). Определены отдельные подсистемы для транспортного светофора на основной дороге (Гл. Транспортный светофор) и транспортного светофора на второстепенной дороге (Вт. Транспортный светофор). Вполне вероятно, что структура и поведение этих двух подсистем будут идентичными, однако их роли в системе различны. В представленной на рисунке модели для каждой из подсистем используется множественность 1; другой альтернативой является использование одного блока подсистемы с названием “Транспортный светофор” и множественностью 2. Я выбрал первый вариант, поскольку роли двух этих подсистем для их клиента (контроллера управления перекрестком) различны, и использование отдельных связей делает это различие более наглядным. Но при моделировании может использоваться любой из этих подходов ².

² Существует общее правило: когда несколько объектов играют одинаковую роль по отношению к другому объекту (воспринимаются им примерно одинаково), тогда я использую ассоциацию с множественностью *. Если же несколько объектов играют разные роли, я использую различные ассоциации (для одного и того же класса) для каждой роли.

Система Беспилотного Летательного Аппарата Койот (СБЛАК) примерно на два порядка превосходит по своим размерам систему управления дорожным перекрестком. Поэтому, ее архитектура представлена не на одной, а на нескольких диаграммах. На первой из этих диаграмм, показанной на Рис. 9-10, представлены связанные системы, составляющие весь комплекс БЛАК. Мы видим три типа систем – наземные системы (для планирования и управления операциями), летательный аппарат Койот и бортовые системы. Определены четыре бортовых системы: ракета, тепловизионная система переднего обзора, система видеонаблюдения, и радар с синтезированной апертурой. Для каждой из них применен стереотип «Система» и для каждой из этих систем будет выполнена декомпозиция на подсистемы на отдельных диаграммах. Хотя всю эту информацию можно разместить на одной диаграмме, такая диаграмма будет слишком сложной.

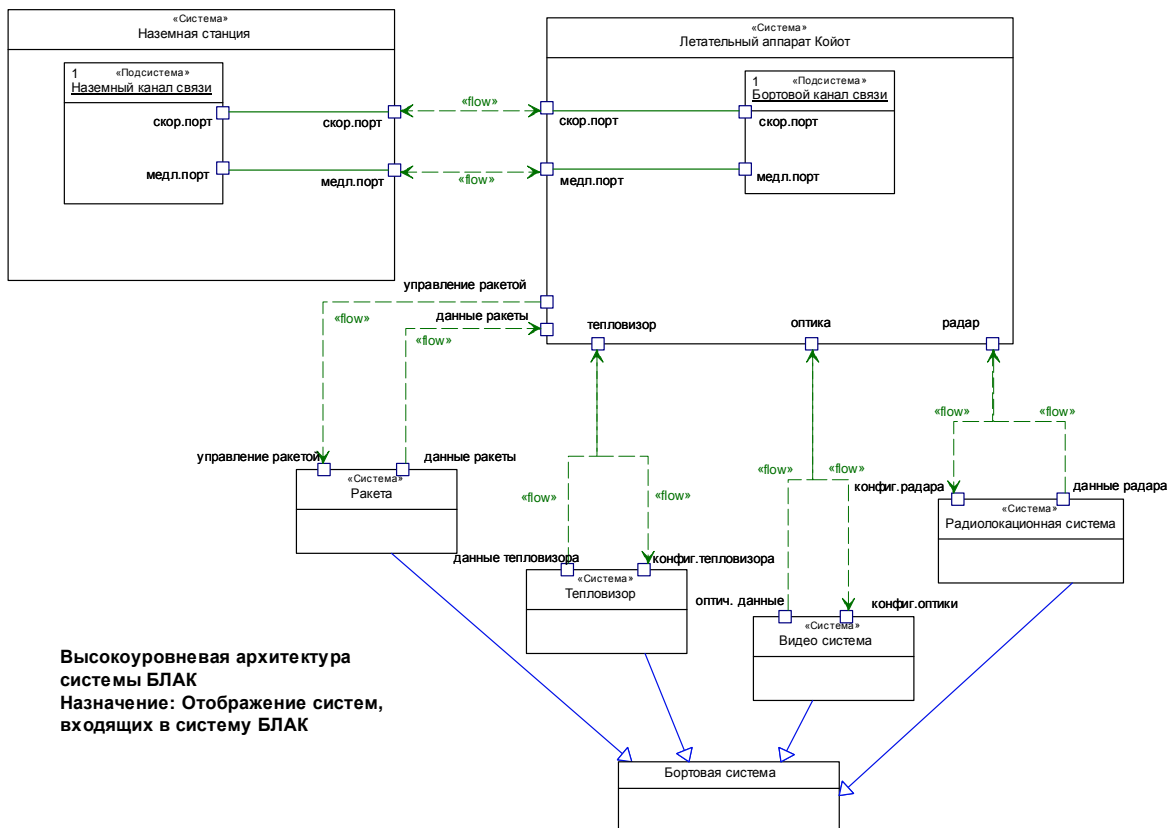


Рис. 9-10 Диаграмма с системной архитектурой для БЛАК

На Рис. 9-10 представлена диаграмма классов вместо несколько ограничивающей структурной диаграммы, исключительно для примера еще одного способа отображения архитектуры. Поскольку на этой диаграмме представлены классы, а не объекты, то связи между портами на этой диаграмме не могут быть определены³. По этой причине на

³ Связи могут быть определены между объектов и между портами объектов; отношения ассоциации в UML 2.0 могут быть определены между классами, но не между портами. Конечно же, связь является экземпляром

диаграмме изображены информационные потоки между соответствующими портами. Та же самая информация, но с использованием структурной диаграммы показана на Рис. 9-11, только в этом случае основные элементы архитектуры являются составными частями всего проекта, и, поэтому, они могут быть связаны друг с другом при помощи связей.

Для блока "Проект БЛАК" применен стереотип «Система систем» для того чтобы подчеркнуть, что он «больше», чем входящие в него системы. Поскольку эта структурная диаграмма и ее составные части являются экземплярами, то они могут быть соединены при помощи связей, а не с помощью информационных потоков. Мы также можем указать множественность для каждой составной части. Части "Ракета", "Тепловизор", "Видео система" и "Радар" могут устанавливаться на летательный аппарат только в одном экземпляре, однако для них используется множественность *, поскольку в одной системе могут иметься несколько активных летательных аппаратов.

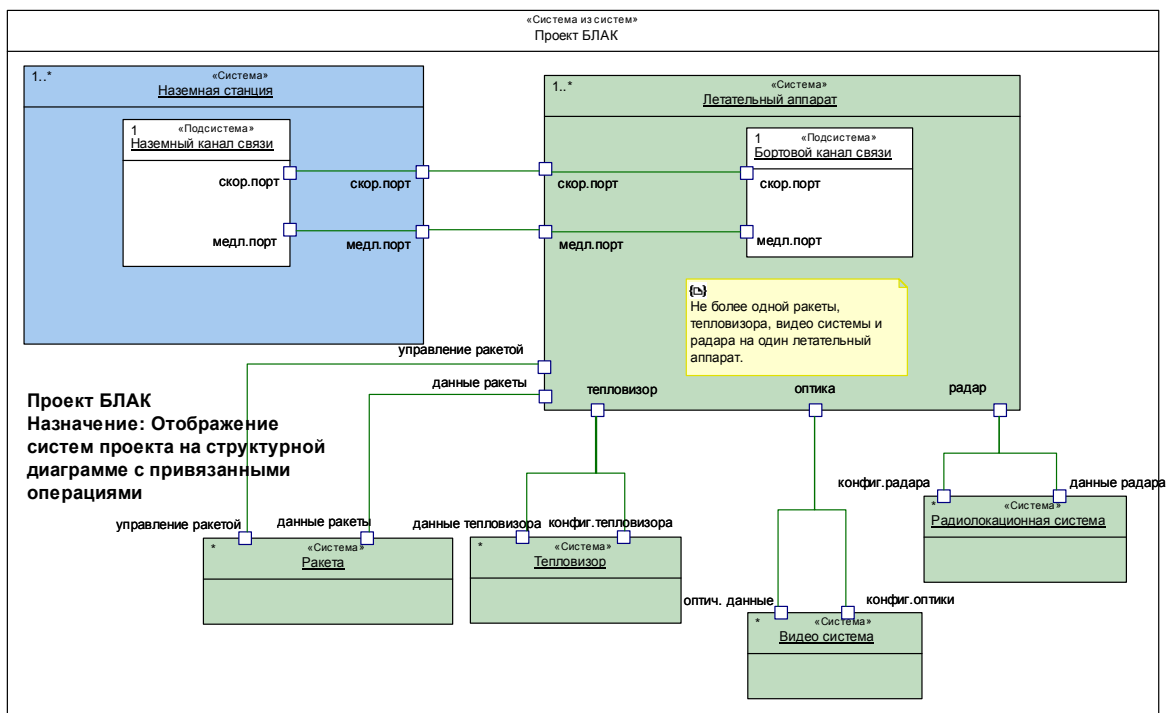


Рис. 9-11 Системная архитектура БЛАК на структурной диаграмме

На предыдущих рисунках для бортовых систем был применен стереотип «Система», и эти системы моделировались на том же уровне абстракции, что и летательный аппарат, и наземная станция. Можно рассмотреть вариант в котором они являются подсистемами летательного аппарата - в конце концов, они устанавливаются и сбрасываются с летательного аппарата. Хотя такой вариант моделирования бортовых системы не является не правильным, в нашем случае я выбрал другой вариант. Я руководствовался тем соображением, что бортовые системы снимаются с борта, могут

ассоциации. Связи на диаграммах могут быть определены только тогда, когда на диаграмме отображены экземпляры классов - объекты.

быть заменены, а также установлены на различные летательные аппараты, которые могут даже разрабатываться не в данном проекте. Чтобы облегчить задачу повторного использования бортовых подсистем, я выделил их в отдельные системы, а не сделал подсистемами летательного аппарата.

Для всех систем, определенных в проекте БЛАК, должна быть выполнена декомпозиция на входящие в них подсистемы. Эта декомпозиция выполнена на следующей серии диаграмм. Диаграмма Рис. 9-12 иллюстрирует подсистемы для наземной станции. Обратите внимание на то, что подсистема обмена данными с наземной станцией (Наземный канал связи) и подсистема хранения данных (Хранилище данных) существуют в единственном экземпляре в составе наземной станции. В наземной станции может быть до четырех станций управления, каждая из которых может включать в себя два пульта – один для управления БЛА и один для приема и обработки данных наблюдения. Ассоциация между ручными пультами позволяет управлять с одного пульта всеми остальными пультами при работе в режиме низкой боевой готовности.

Правила, используемые для определения связей между этими подсистемами, весьма просты. Если одна из подсистем использует какие-либо сервисы или данные другой подсистемы, то между ними определяется связь. Все связи (на уровне подсистем) связывают порты, чтобы гарантировать высокую степень инкапсуляции в системе.

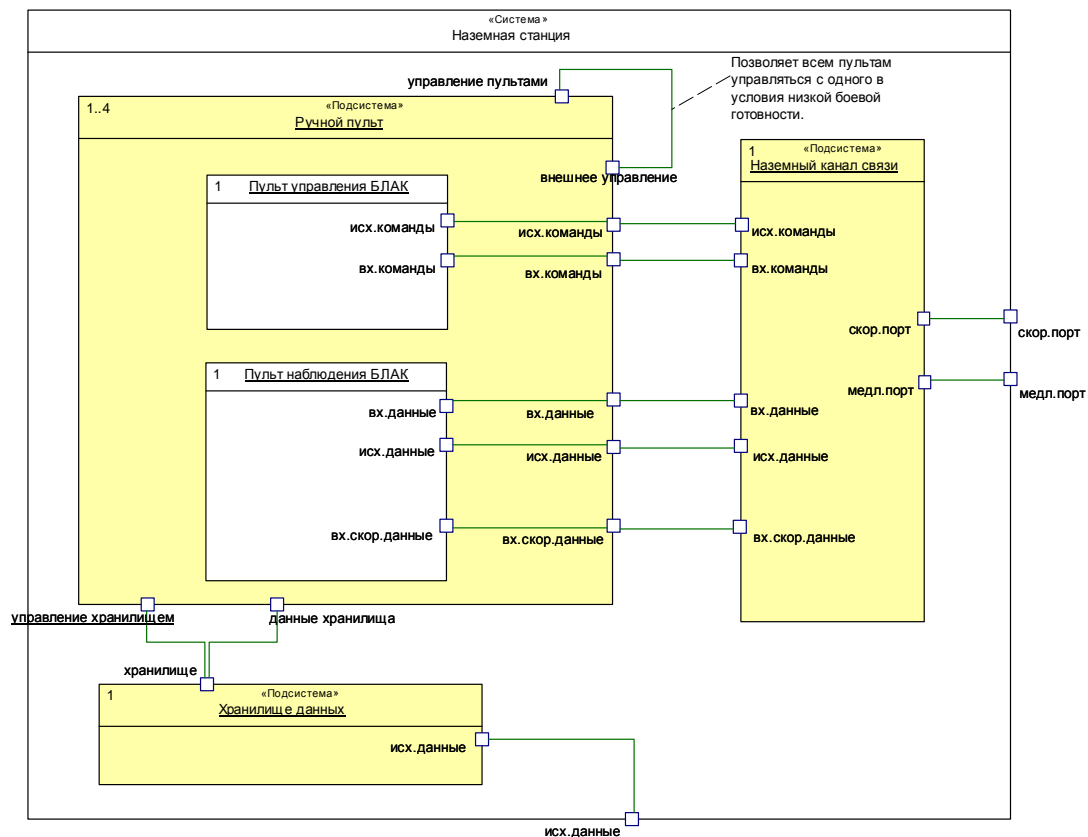


Рис.

9-12 Подсистемы наземной станции системы БЛА Койот

На Рис. 9-13 показан сам летательный аппарат. В его состав входят несколько подсистем. Все подсистемы перечислены в Таб. 9-1.

Таб. 9-1 Назначение подсистем БЛА Койот

Подсистема летательного аппарата	Связана с	Назначение
Бортовой канал связи	Внешний высокоскоростной порт Внешний низкоскоростной порт Управление полетом Управление стрельбой Управление положением Управление двигателем Навигация Наблюдение	Обеспечивает связь летательного аппарата с наземной станцией как для передачи низкоскоростных (команд) данных и статуса, так и для передачи высокоскоростных данных. Также выполняет разбор пакетов данных, полученных через каналы связи, и передает их в соответствующие подсистемы.
Контроль положения	Бортовой канал связи Управление гидравликой Управление полетом	Контролирует положение летательного аппарата (углы тангажа, наклона и рыскания), отдавая команды на смену положения для рулевого управления летательного аппарата.
Управление двигателем	Бортовой канал связи Управление полетом	Управляет двигателем и контролирует его состояние.
Управление стрельбой	Бортовой канал связи Управление целями Порт управления ракетой Порт данных ракеты	Управляет системами вооружения, размещенными на борту БЛА, осуществляя управление и проверку состояния, в том числе пуском ракет Хелфайер.
Управление полетом	Бортовой канал связи Управление положением Управление двигателем Управление гидравликой Навигация Управление топливом	Осуществляет общее управление полетом, которое может быть полностью или частично автоматизировано или полностью быть управляемо с земли.
Управление топливом	Управление полетом	Следит за состоянием топлива и при необходимости перераспределяет топливо для обеспечения баланса веса.
Управление гидравликой	Управление положением Управление полетом	Обеспечивает непосредственное управление рулевыми плоскостями летательного аппарата – крыльями, рулевыми плоскостями, рулями высоты

		и крена. Также управляет шасси летательного аппарата
Навигация	Бортовой канал связи Управление полетом	Используется для управления и контроля за местонахождением в воздушном пространстве. Также содержит навигационные данные и инструменты, такие как электронные карты и полетные задания.
Управление наблюдением	Бортовой канал связи Управление целями Порт тепловизора Порт видео системы Порт радара	Собирает и обрабатывает данные наблюдения как для сопровождения цели на борту, так и для удаленного наблюдения (путем передачи данных на наземную станцию).
Управление целями	Управления стрельбой Управления наблюдением	Обнаруживает и сопровождает цели.

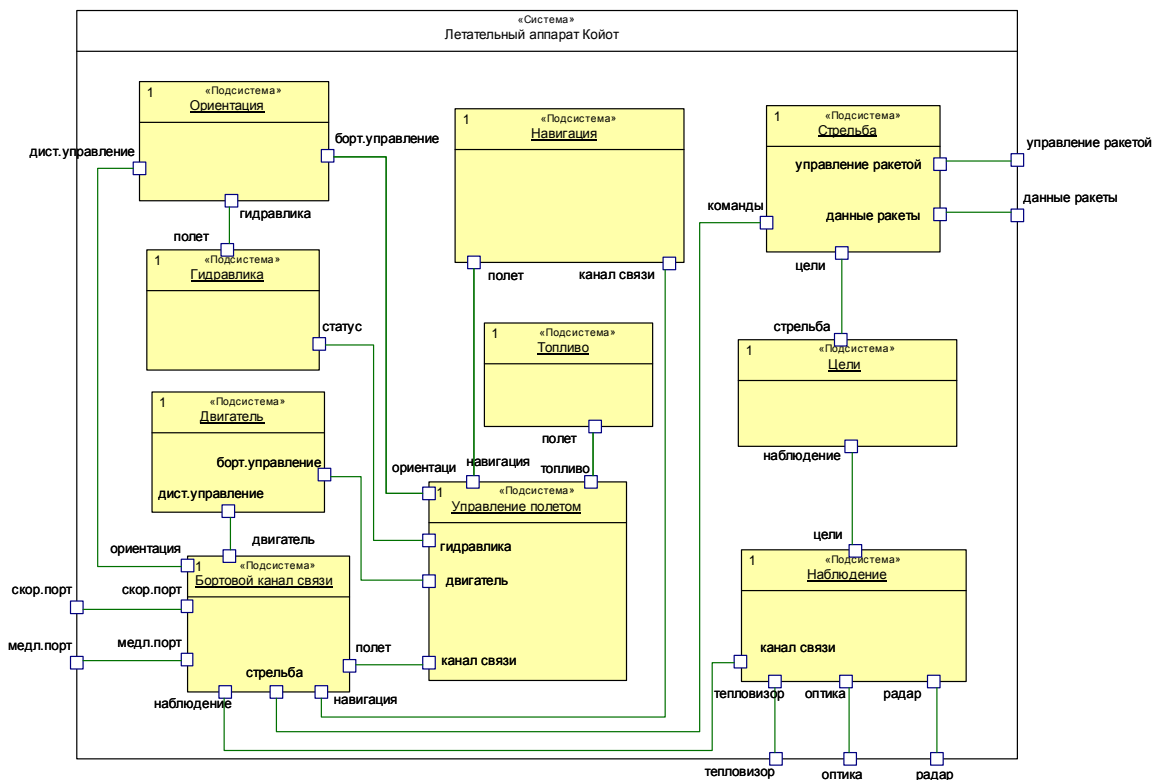


Рис. 9-13 Подсистемы летательного аппарата Койот

Мы не будем здесь декомпозировать бортовое оборудование на подсистемы. Одна из причин состоит в том, что каждая бортовая система будет разрабатываться отдельной командой разработчиков, поэтому не существует острой необходимости выполнять эту декомпозицию уже сейчас. Во-вторых, бортовые системы гораздо менее сложны, чем наземная станция или летательный аппарат.

Еще одно замечание по поводу декомпозиции систем на подсистемы, выполненной нами: все подсистемы являются междисциплинарными. Другими словами, каждая подсистема включает программное обеспечение, электронику, механику и, возможно, химические компоненты. Данная декомпозиция рассматривается при выполнении следующих заданий в этой главе.

Решение 4.3 Привязка операционных контрактов к подсистемам

В процессе разработки систем традиционно сначала определяются требования к системе. При использовании MDA/UML, требования группируются по вариантам использования. Далее мы определяем архитектуру системы. Этот вопрос мы рассмотрели для обеих систем в предыдущем задании этой главы. Следующим шагом нам необходимо привязать требования к различным подсистемам. Для этого нам необходимо выполнить следующие шаги:

1. Привязать операционные контракты (сервисы) к элементам архитектуре
2. Сгруппировать эти сервисы на уровне подсистем в варианты использования

В этом задании мы выполним первый из этих шагов. Концептуально, это достаточно простой процесс ⁴. Каждый вариант использования реализуется несколькими взаимодействующими между собой элементами системы. На архитектурном уровне такими элементами являются системы, подсистемы и компоненты. Каждый сервис уровня системы (уровень «система» для системы управления дорожным перекрестком Родраннер и «система из систем» для системы БЛА Койот) декомпозируются на сервисы, предоставляемые различными подсистемами. На данном шаге процесса мы декомпозируем сервисы уровня системы на сервисы уровня подсистем. Задействование этих сервисов мы главным образом отображаем как сообщения на диаграммах последовательностей.

Технически, декомпозиция сервисов системы может выполняться путем «добавления» или посредством декомпозиции линии жизни варианта использования. Под «добавлением» мы имеем ввиду простое копирование исходных диаграмм последовательностей и добавление к ним линий жизни подсистем. Данный способ применим только для небольших систем. Для подсистем большего масштаба, или систем с большим количеством частей на следующем уровне абстракции, более подходит способ, когда линия жизни диаграммы последовательностей верхнего уровня ссылается на более детальную диаграмму последовательностей. И уже на ней определяются взаимодействия между внутренними частями линии жизни, происходящие в рамках того же самого сценария. Это новая возможность UML 2.0 и одно из наиболее значимых дополнений в UML. В Rhapsody создание диаграммы последовательностей для декомпозиции линии жизни очень просто – нужно дважды щелкнуть по линии жизни на диаграмме для открытия диалога “Характеристики”, затем в списке “Декомпозированные” выберите <новая> (или выберите имя уже существующей диаграммы последовательностей). При этом будет создана (или

⁴ Однако, не забывайте закон Дугласа, который гласит: «Разница между теорией и практикой гораздо ощутимее на практике, чем в теории!»

выбрана) вложенная диаграмма последовательностей, используемая для декомпозиции линии жизни. Перейти к вложенной диаграмме последовательностей очень просто - просто выберите интересующую вас линию жизни, щелкните по ней правой кнопкой мыши и выберите команду “Открыть вложенную диаграмму последовательностей”.

При выполнении декомпозиции линии жизни диаграммы последовательностей возникает вопрос: «Каким образом отображать входящие и исходящие сообщения на вложенной диаграмме последовательности?» Наиболее простым и часто применяемым способом является использование “Внешнего окружения” – специальной линии жизни, представляющей все сущности, не изображенные на диаграмме в явном виде. Все входящие и исходящие сообщения, относящиеся к вложенной диаграмме последовательностей, будут начинаться или заканчиваться на линии жизни “Внешнее окружение”. Если вы определяете вложенную диаграмму последовательностей для линии жизни исходной диаграммы последовательностей, то все входящие сообщения приходят от «границы линии жизни исходной диаграммы», а исходящие сообщения отправляются к той же границе.

Альтернативой использованию “Внешнего окружения” состоит в копировании действующих лиц из исходной диаграммы последовательностей. Если все элементы, отправляющие или принимающие сообщения, являются действующими лицами, то несомненно хорошим подходом будет скопировать их во вложенную диаграмму последовательностей. Оба этих способа эквивалентны друг другу.

В Rhapsody имеется «режим анализа» для создания диаграмм последовательностей – этот режим удобно использовать для набрасывания сценариев, поскольку никаких изменений в репозитории в этом режиме не происходит. В других инструментах подобные действия могут выполняться несколько по-другому. Как только сценарий становится стабильным, диаграмма последовательностей может быть переведена в «режим проектирования», в котором все изменения диаграммы последовательностей отражаются в репозитории. В режиме проектирования все сообщения, входящие в линию жизни (для классификатора), становятся операциями (в случае синхронных сообщений) или реакциями на события (в случае асинхронных сообщений) для данного классификатора.

Итак, давайте сначала рассмотрим систему управления дорожным перекрестком. Интеллектуальность в этой системе сосредоточена в основном в контроллере перекрестка (см. Рис. 9-9). Сначала возьмем диаграмму последовательностей для сценария 2 и выполним декомпозицию линии жизни для варианта использования “Режим реагирования на сигналы датчиков”. При этом будет создана пустая диаграмма последовательностей, на которую будет ссылаться линия жизни исходной диаграммы. Откройте новую диаграмму последовательностей и перетащите на нее элементы (части системы управления дорожным перекрестком), чтобы создать линии жизни на диаграмме. Добавьте линию жизни “Внешнее окружение”, чтобы иметь возможность добавлять исходящие и входящие сообщения для данной диаграммы последовательностей. Теперь для каждого сообщения на исходной диаграмме последовательностей от действующего лица к линии жизни варианта использования, создайте на вложенной диаграмме сообщение от линии жизни “Внешнее окружение” к

соответствующей линии жизни подсистемы. Аналогично, для каждого сообщения от линии жизни варианта использования к действующему лицу, создайте на вложенной диаграмме соответствующее сообщение от какой-либо линии жизни к линии жизни “Внешнее окружение”. Добавьте необходимые внутренние сообщения между элементами, чтобы наладить взаимодействие этих элементов друг с другом. Вложенная диаграмма последовательностей для сценария 2 (Рис. 4-3) показана на Рис. 9-14. Аналогично, более детальное представление для сценария 3 (Рис. 4-4) показано на Рис. 9.15.

Сценарий 2: Операционные контракты, привязанные к подсистемам

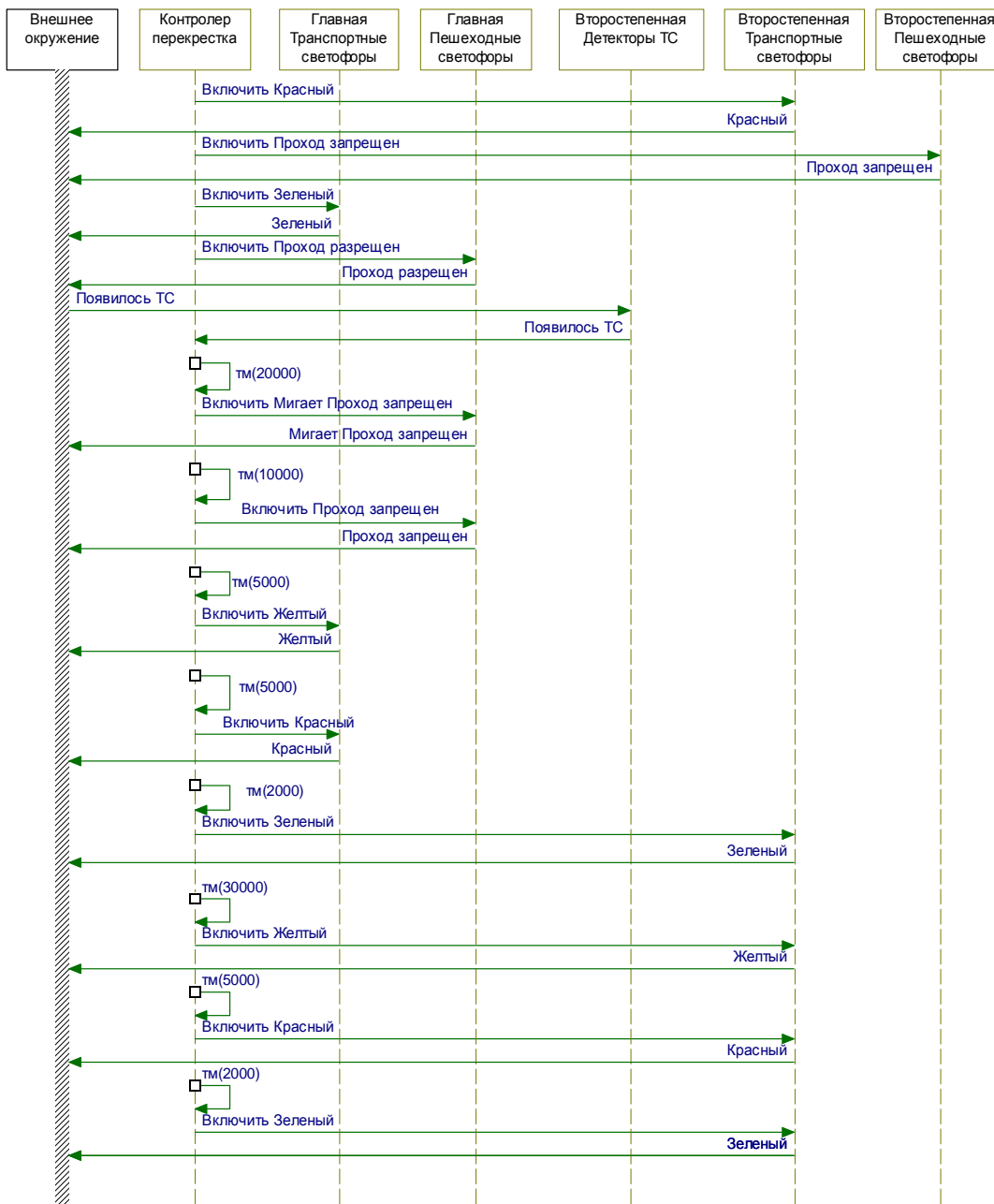


Рис. 9-14 Сценарий 2 для Родраннер с детализацией для подсистем

Сценарий 3: Операционные контракты, привязанные к подсистемам

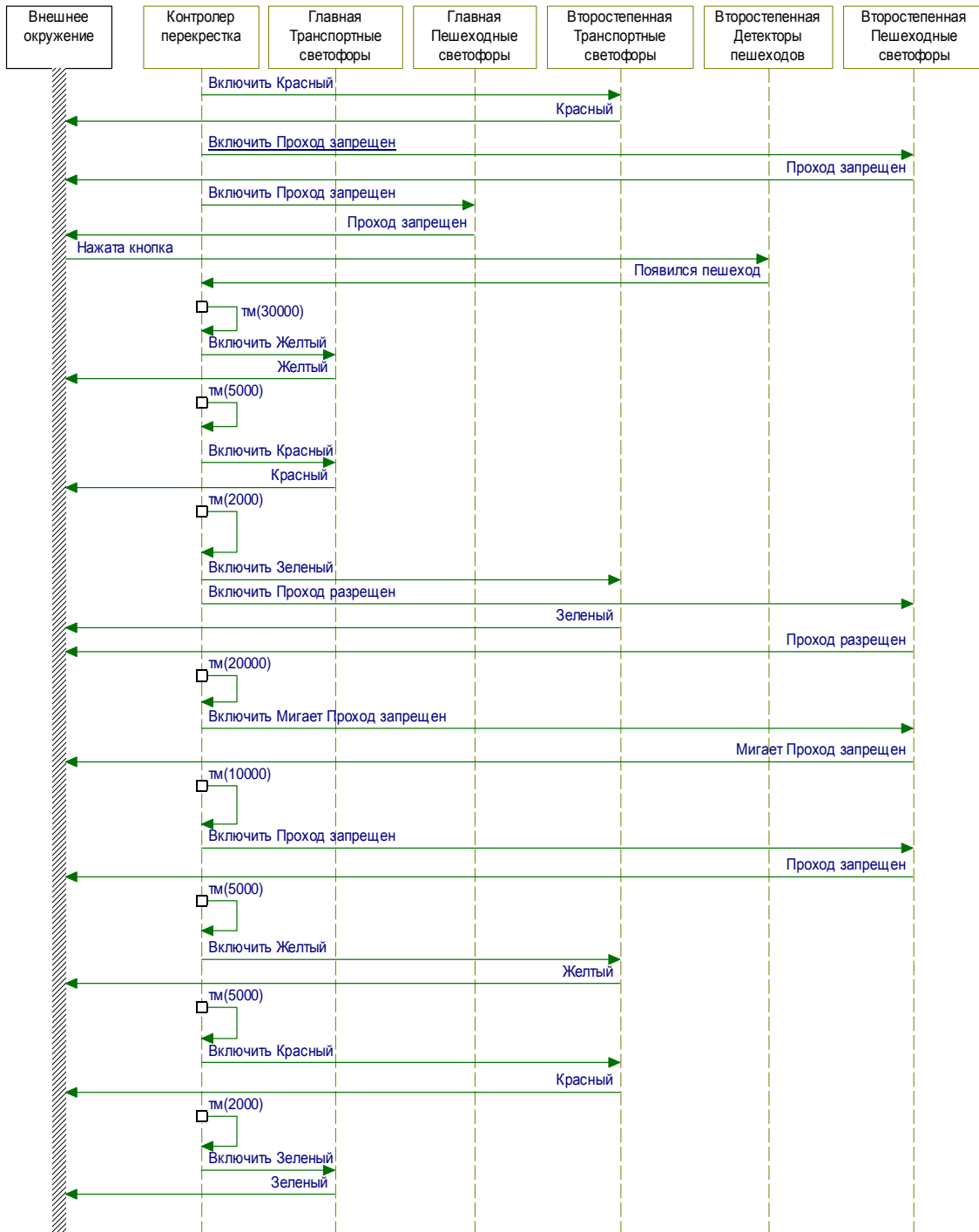


Рис. 9- 15 Сценарий 3 для Родраннер с детализацией для подсистем

Итак, что же это означает для подсистем? Поскольку сообщения реализуются подсистемами, то к подсистемам добавляются соответствующие сервисы (операции и реакции на события). В нашем случае мы сделали подсистемы экземплярами классов, и эти реализованные сообщения стали операциями, определенными для классов. После

этого их можно добавить к соответствующим портам, добавив эти сервисы в интерфейсы, которые специфицируют контракты портов. На Рис. 9-16 интерфейсы портов не показаны, чтобы не перегружать диаграмму.

Системная архитектура Роадраннер:
Назначение: Отображение входящих подсистем

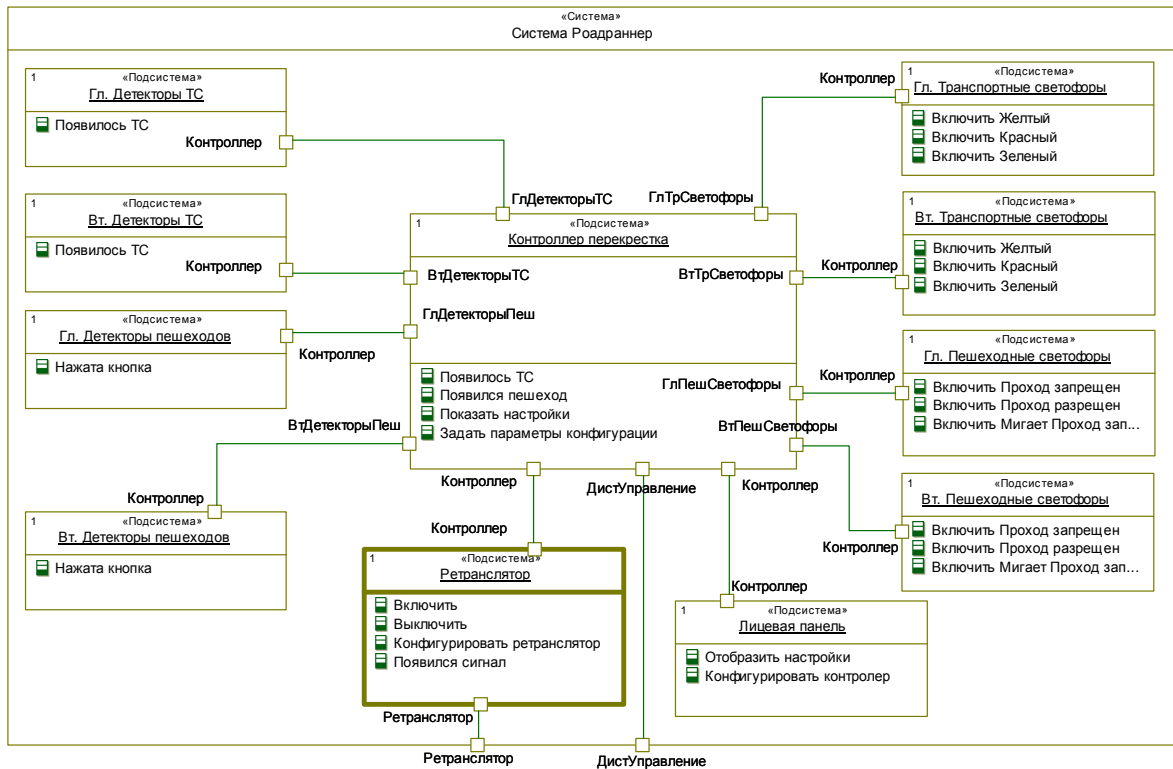


Рис. 9-16 Подсистемы Роадраннер с операциями

Система управления дорожным перекрестком – относительно простая система, но этот пример хорошо иллюстрирует процесс определения сервисов для подсистем. В данном примере интерфейсы между подсистемами просты, однако это наглядно иллюстрирует распределение ролей между архитектурными частями системы. Для того чтобы получить полный набор интерфейсов между подсистемами, на Рис. 9-16 я добавил несколько дополнительных операций, появляющихся в некоторых других сценариях (например, использование лицевой панели для конфигурирования системы и использование ретрансляторов для обнаружения транспортных средств специальных служб), не показанных в данной главе.

Теперь выполним те же действия для системы БЛАК. Конечно же, это гораздо большая система, поэтому нам необходимо будет использовать оба способа декомпозиции диаграмм последовательностей для управления информацией, отображаемой на диаграммах.

Мы начнем с взаимодействия между наземной станцией (и ее наземным каналом связи), летательным аппаратом (и его бортовым каналом связи) и одним из видов бортового

оборудования – в данном случае, видеосистемой. Как сказано в задании, мы рассмотрим вариант использования "Поиск по территории", еще точнее Сценарий 1, показанный на Рис. 4-7. Поскольку диаграмма, изображенная на этом рисунке, содержит только ссылки на более подробные диаграммы последовательностей, мы начнем с Рис. 4-8, на котором показано как происходит взлет летательного аппарата. На более детальном представлении, изображенном на Рис. 9-17, мы выполним декомпозицию для линии жизни варианта использования "Поиск по территории", определив вложенную диаграмму последовательностей.

Детализация Сценария 1 "Взлет ЛА" для ВИ "Поиск по территории"



Рис. 9-17 Детализация сценария 1 для ВИ "Поиск по территории" – Основная диаграмма

Аналогично предыдущему примеру для системы управления дорожным перекрестком, все сообщения от действующих лиц исходят от линии жизни "Внешнее окружение" (на диаграмме обозначенной как ENV), которая связывает сообщения с действующими лицами на высокоуровневой диаграмме. Также, как мы уже отмечали, описание детального взаимодействия будет слишком длинным, чтобы поместить его на одну диаграмму последовательностей; поэтому, мы добавили ссылки на внешние сценарии, в которых будет представлено большинство деталей: первый сценарий содержит подробное описание взаимодействия, относящегося к подготовке летательного аппарата, а второй – описание взаимодействия, относящегося к взлету летательного аппарата с взлетной полосы.

Первый фрагмент взаимодействия разбит на Рис.9-18 и Рис.9-19.

Детализация сценария 1 "Взлет летательного аппарата" для ВИ "Поиск по территории"

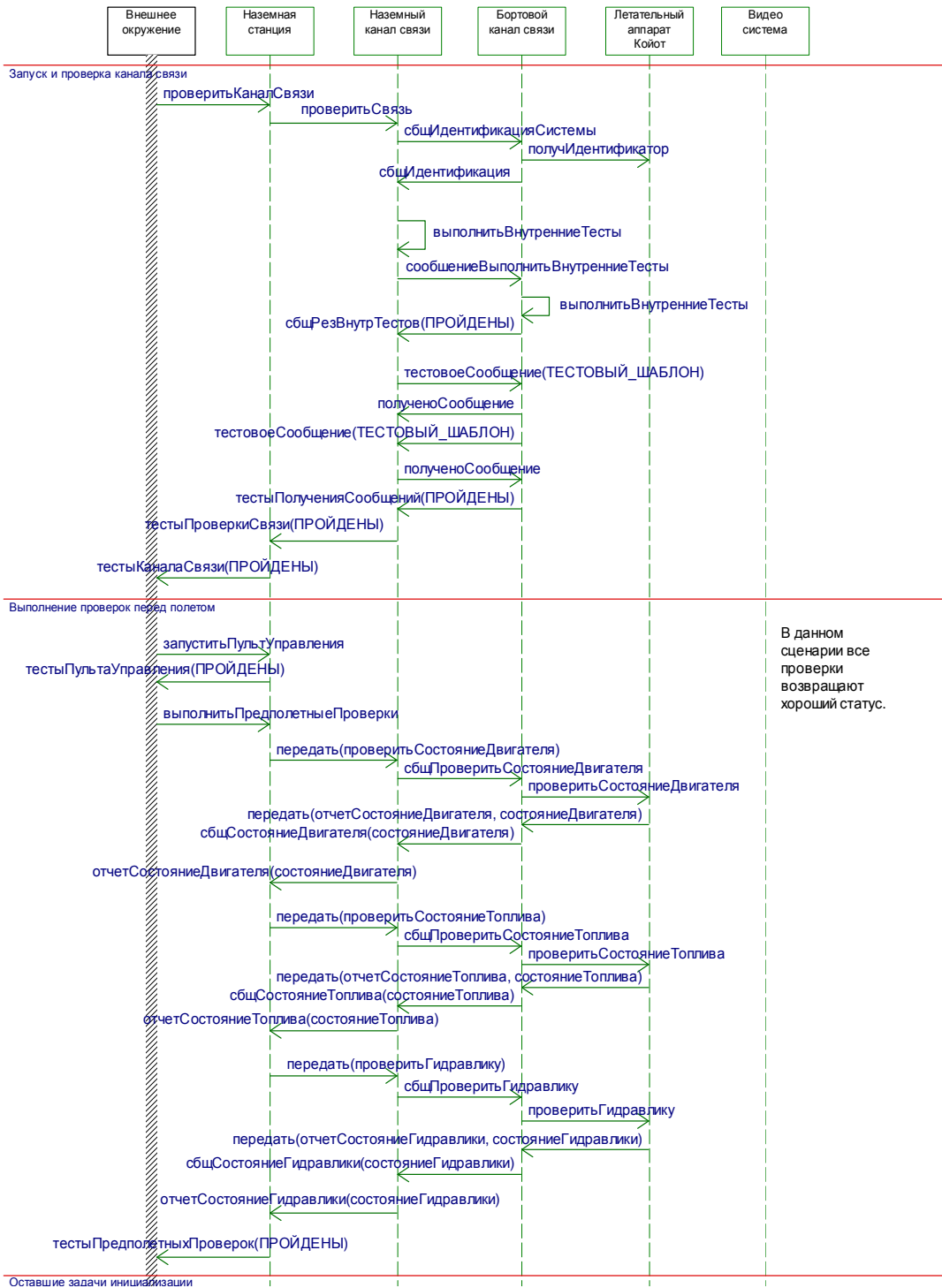


Рис. 9-18 Подготовка летательного аппарата (часть А)

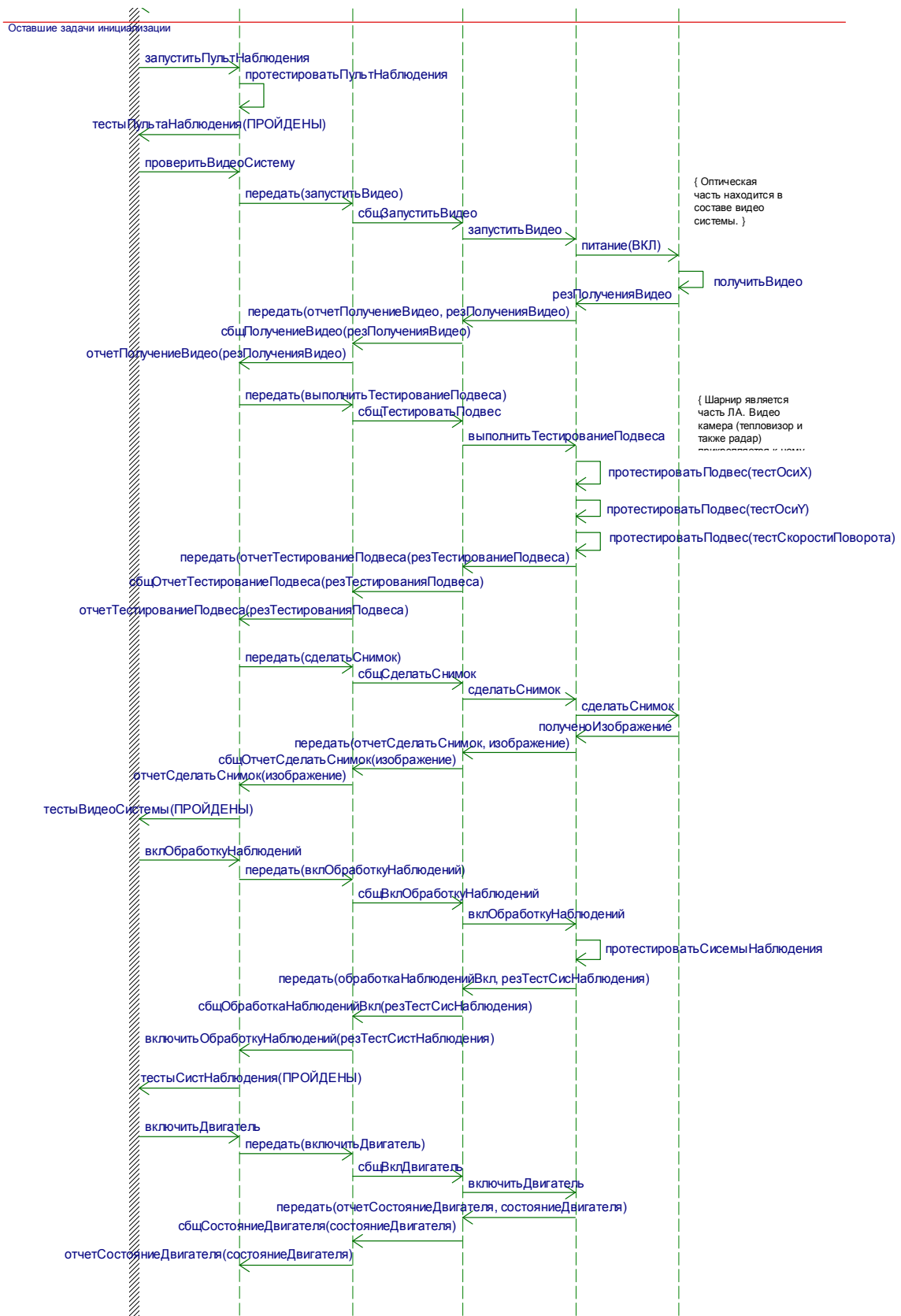


Рис. 9-19 Подготовка летательного аппарата (часть Б)

Как видно на примере предыдущей диаграммы, отображение более детального уровня приводит к чрезвычайному увеличению количества информации, поэтому, управление диаграммами и информацией может сильно усложниться. Возможности декомпозиции диаграмм последовательностей, реализованные в UML 2.0, позволяющие переходить по ссылкам к вложенным уровням очень сильно облегчают эту задачу. При желании, можно добавить ссылки, обеспечивающие возможность обратного перехода; но их добавление приходится делать вручную. Это обусловлено тем, что на один и тот же фрагмент взаимодействий может быть несколько ссылок из разных диаграмм последовательностей, поэтому и вам может понадобиться добавить несколько таких обратных ссылок. На следующем рисунке, Рис. 9-21, показана такая ссылка (в элементе комментария). Если щелкнуть эту ссылку в Rhapsody, будет открыта высокоуровневая диаграмма. Я очень советую добавлять такие обратные ссылки при создании больших моделей. Это значительно облегчает процесс навигации между диаграммами; более того, их добавление лучше делать непосредственно при создании диаграмм, а не впоследствии, когда будет создано много различных диаграмм.

Также как с предыдущей диаграммой, которая является единой диаграммой в инструменте, данная диаграмма разбита на два рисунка из соображений форматирования. В данном случае, линия разбиения диаграммы проходит между двумя параллельными фрагментами на диаграмме. Она отображена пунктирной линией внутри фрагмента взаимодействия "ПАР" на рисунке.

Сценарий 1 "Взлет ЛА со взлетной полосы" для ВИ "Поиск по территории"

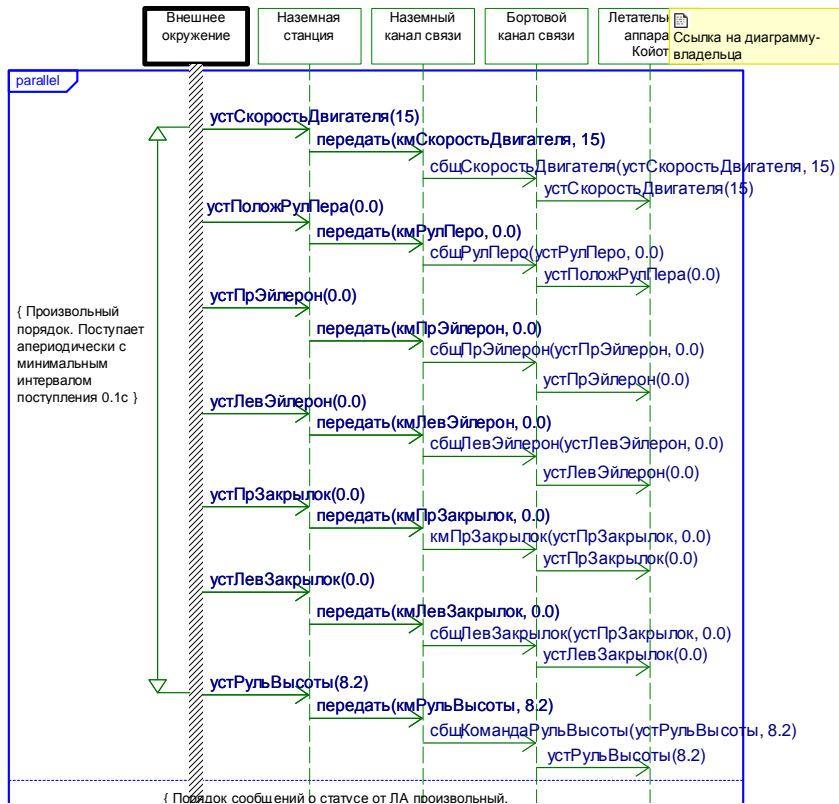


Рис. 9-20 Детализация для взлета летательного аппарата (Часть А)



Рис. 9-21 Детализация для взлета летательного аппарата (Часть Б)

После того как все операции привязаны, (как это сделано на предыдущих диаграммах последовательности), привязанные операции могут быть показаны на структурной диаграмме или диаграмме классов (см. Рис. 9-22).

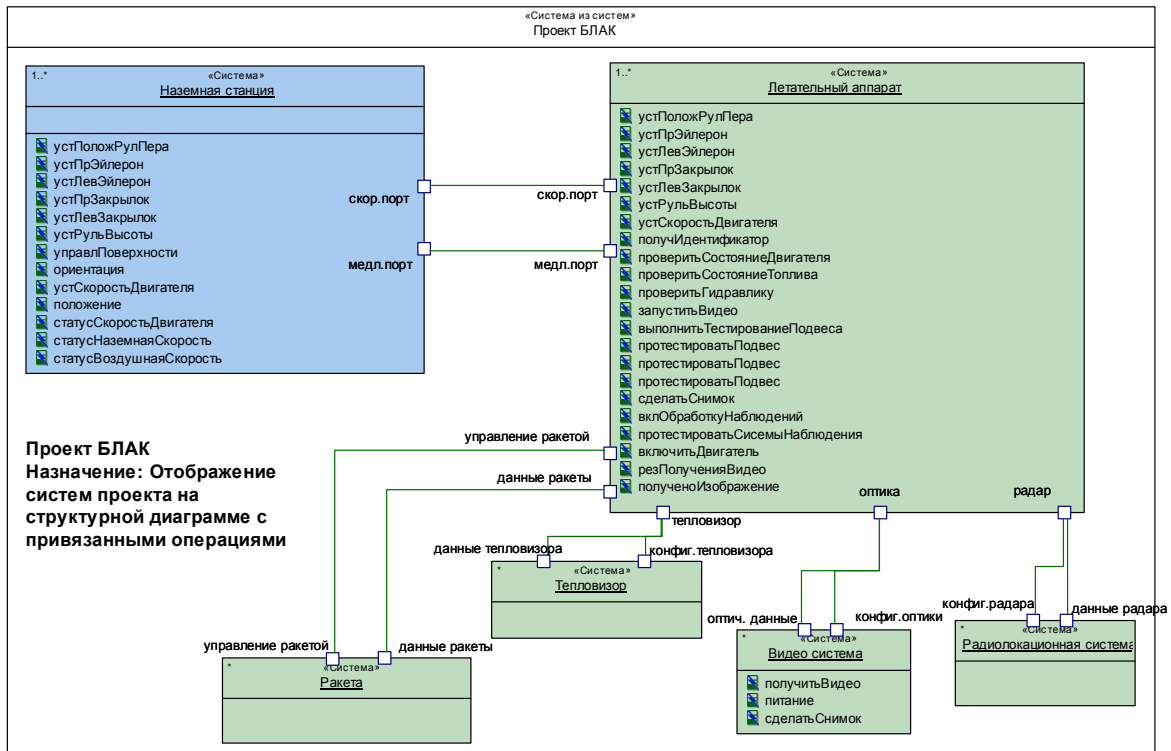


Рис. 9-22 Операции, привязанные к подсистемам БЛАК

Рис. 9-22 получается из диаграммы Рис. 9-11 путем переключения из структурного представления в представление спецификации для ее частей, что приводит к отображению всех операций⁵.

Решение 4.4 Определение вариантов использования для подсистем

В первой части этого задания нам предстоит определить варианты использования для системы управления дорожным перекрестком Рoadраннер, при этом мы ограничимся двумя вариантами использования уровня системы – "Детектирование ТС" и "Конфигурирование системы". Разумеется, в реальном проекте мы бы декомпозировали все варианты использования уровня системы и привязали бы их ко всем подсистемам. Но в данном задании мы ограничимся только двумя вариантами использования уровня системы.

⁵ Возможно, вам также потребуется настроить свойства отображения (из контекстного меню) для частей, выбрав те операции, которые вы хотите показать.

Для первого из них, варианта использования "Детектирование ТС", определены два специализированных варианта использования "Детектирование ТС подземными датчиками" и "Детектирование ТС надземными датчиками". При выбранной архитектуре системы управления дорожным перекрестком, большинство действий в рамках данных вариантов использования будут выполняться в двух подсистемах – "Обнаружения ТС" и "Контроллер перекрестка". Часть требований относятся к обоим специализированным вариантам использования. Так, например, независимо от способа детектирования ТС, система должна выполнять подсчет автомобилей, а также вычислять и управлять статистическими данными о движении ТС. Также, все детекторы должны поддерживать обмен данными по проводному каналу передачи данных. Поэтому на Рис. 9-23 вариант использования системы "Детектирование ТС" декомпозирован на два варианта использования уровня подсистем "Управление статистикой движения ТС" (привязанный к подсистеме "Контроллер перекрестка") и "Передача по проводу" (привязанный к подсистемам "Контроллер перекрестка" и "Детектирование ТС"). Декомпозиция вариантов использования выполнена с использованием отношения зависимости со стереотипом «включает». Привязка к подсистемам отображена на рисунке при помощи элементов ограничений; кроме того, варианты использования помещены в пакеты, относящиеся к соответствующим подсистемам. Варианты использования, относящиеся к нескольким подсистемам, могут помещаться в пакете "Общее" системной области модели или дублироваться в соответствующих подсистемах, на ваше усмотрение.

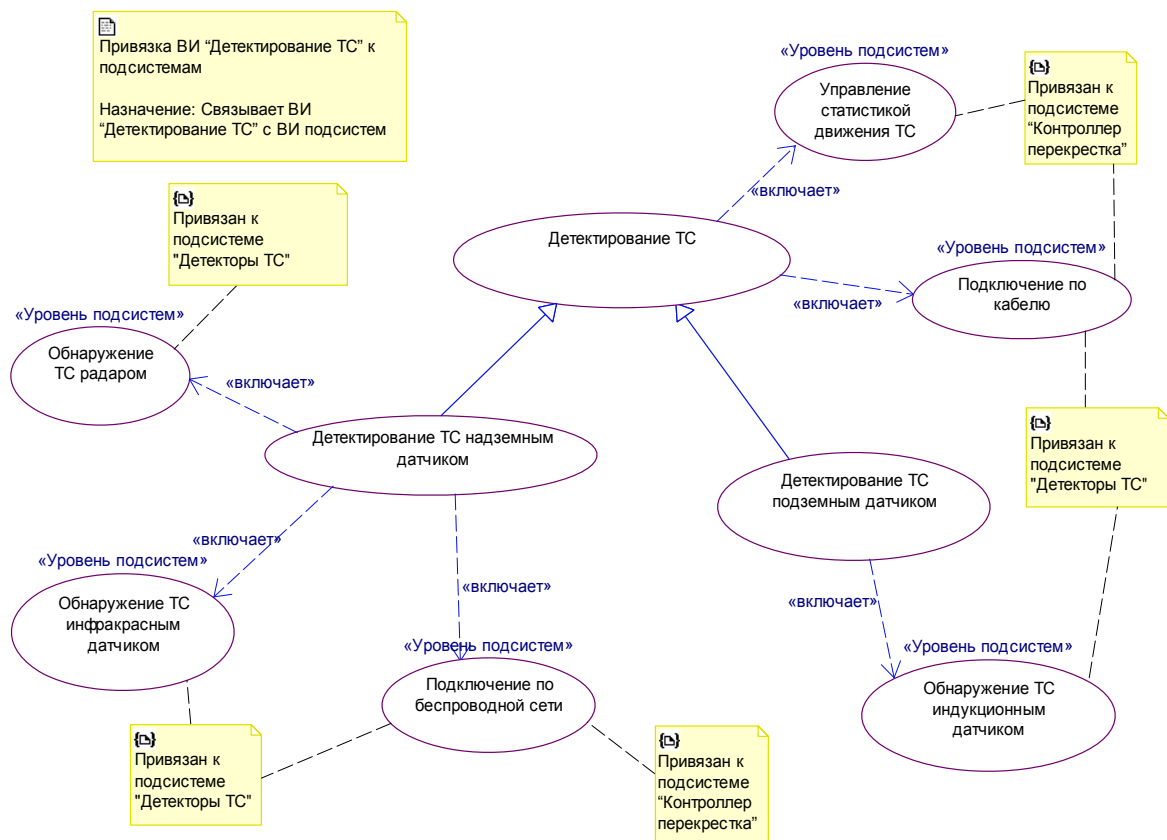


Рис. 9-23 Привязка вариантов использования Радраннер к подсистемам

Специализированные варианты использования аналогичным образом декомпозируются на варианты использования уровня подсистем, что отображается с помощью отношений зависимости со стереотипом «включает».

Вариант использования "Конфигурирование системы" аналогичным образом декомпозируется на несколько более мелких вариантов использования, каждый из которых привязывается к определенной подсистеме. Исключение составляет вариант использования "Конфигурирование беспроводной связи", который является вариантом использования уровня системы, для которого уже выполняется декомпозиция на три варианта использования, привязываемых к подсистемам.

Несколько дополнительных комментариев к Рис. 9-24. Во-первых, вариант использования "Конфигурирование режимов работы" привязан к подсистеме "Контроллер перекрестка". Из этого следует, что все более специализированные варианты использования, такие как "Конфигурирование режима реагирования на ТС специальных служб" также привязаны к этой подсистеме. Во-вторых, специализированные варианты использования для конфигурирования беспроводных сетей привязаны к подсистемам "Ретранслятор", "Детектор ТС" и "Контроллер перекрестка". Напомним, что подсистема "Ретранслятор" принимает сигналы от ТС с преимуществом движения и ТС специальных служб, а подсистема "Детектор ТС" может быть подключена к "Контроллеру перекрестка" либо при помощи кабеля, либо с использованием беспроводного канала связи. Поэтому для каждой из этих систем должна быть предоставлена возможность настроить беспроводное соединение.

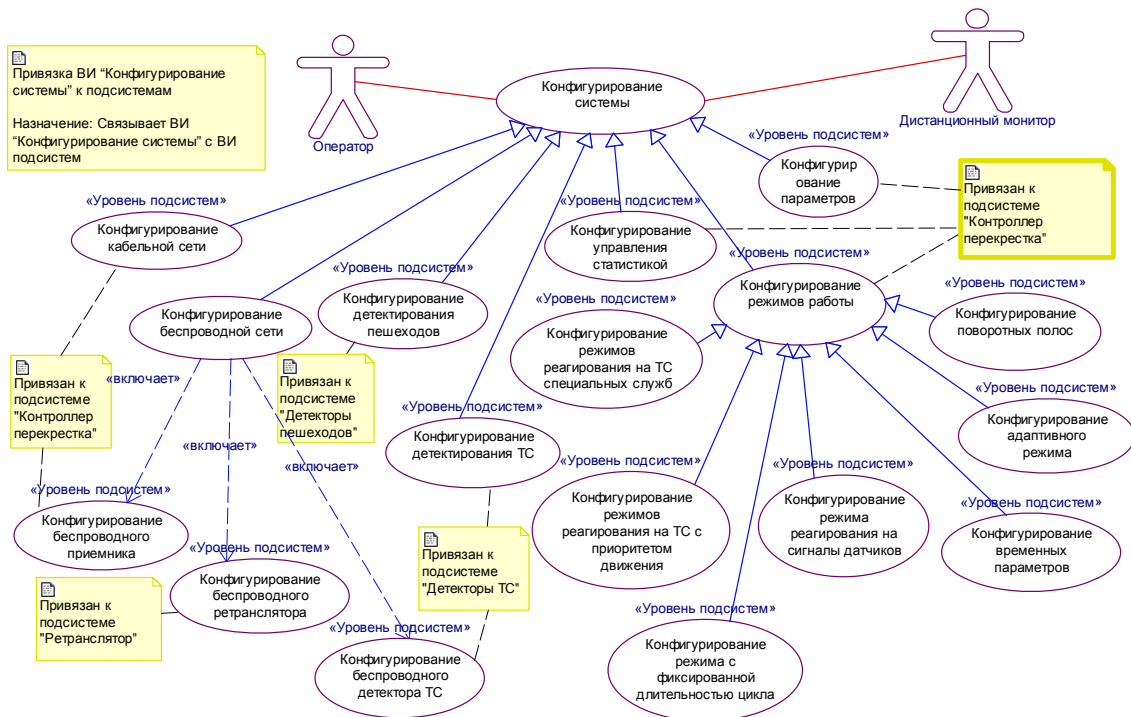


Рис. 9-24 Привязка вариантов использования для конфигурирования к подсистемам

Следующая часть задания состоит в привязке вариантов использования "Ручная подстройка положения" и "Поиск по территории" к подсистемам БЛАК. Привязка к подсистемам для варианта использования "Ручная корректировка положения" показана на Рис. 9-25. На этом рисунке используются другой способ для отображения привязки вариантов использования к подсистемам. В предыдущем примере мы использовали элементы ограничений для отображения привязки. На Рис. 9-25 мы отображаем привязку с помощью элементов "Граница системы", по одному на подсистему.

Я создал отдельную диаграмму для каждого варианта использования уровня системы, чтобы показать их декомпозицию на варианты использования уровня подсистем и провести привязку к различным подсистемам. Я называю такую диаграмму «диаграммой привязки вариантов использования», но на самом деле это просто диаграмма использования, назначение которой – показать как варианты использования уровня системы декомпозируются по подсистемам.

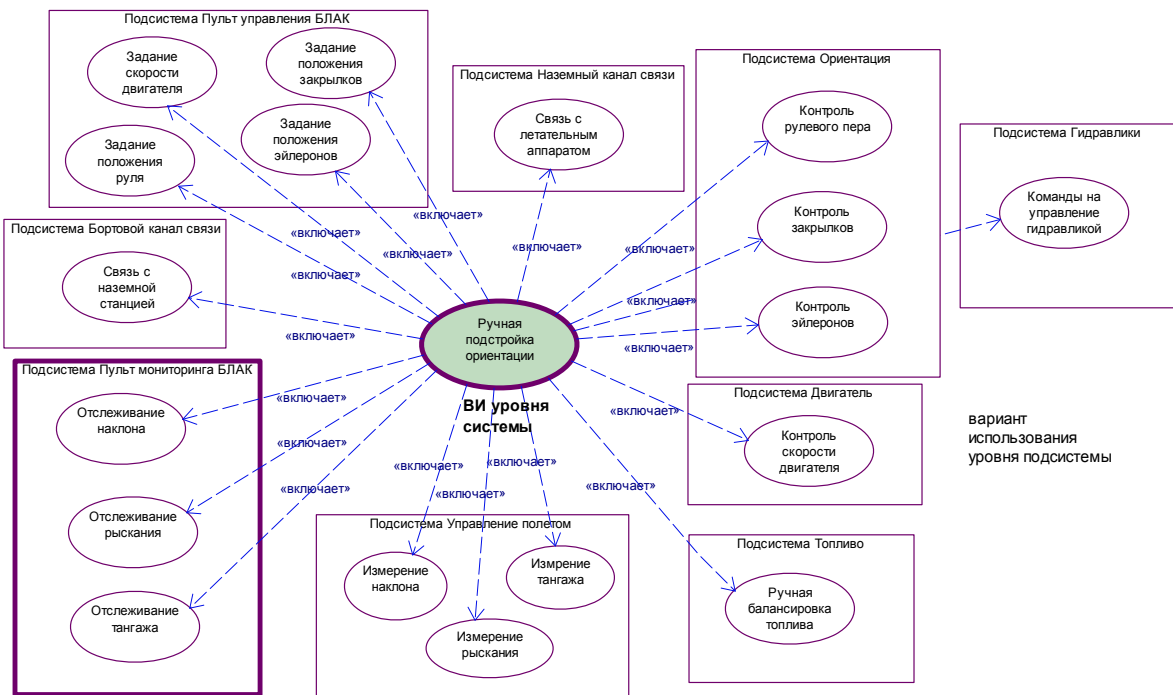


Рис. 9-25: Привязка варианта использования "Ручная корректировка положения" к подсистемам БЛАК

На Рис. 9-26 показана аналогичная информация, что и на предыдущем рисунке, но для варианта использования "Поиск по территории". Если вы посмотрите на диаграммы использования для системы, то обнаружите, что вариант использования "Поиск по территории" является специализацией варианта использования "Выполнение наблюдения", который, в свою очередь, также является специализацией варианта использования "Выполнение миссии". Последний включает в себя вариант использования "Управление полетом БЛА". Следовательно, вместо того чтобы явным образом отображать привязку всех включаемых вариантов использования, Рис. 9-26

отображает включение варианта использования "Управление полетом БЛА" (который в свою очередь включает ВИ "Ручная подстройка положения").

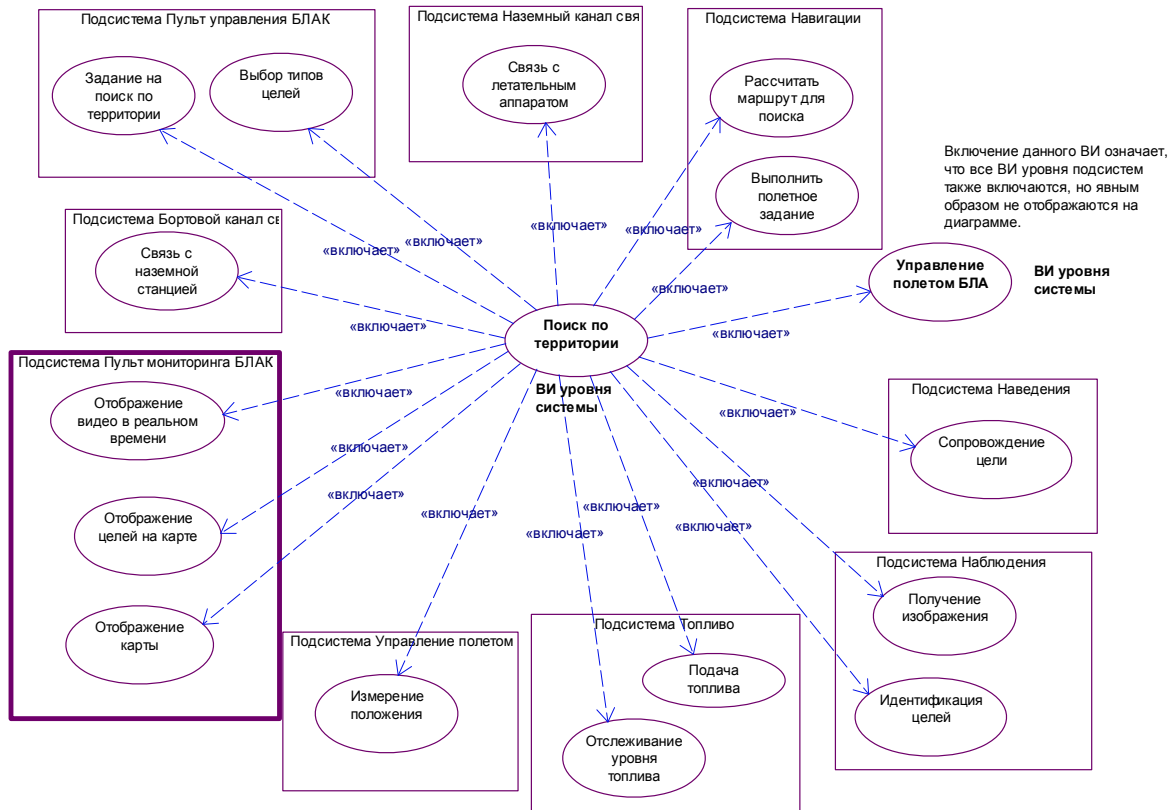


Рис. 9-26 Привязка к подсистемам ВИ "Поиск по территории"

В результате данного процесса привязки мы получим для каждой из подсистем набор вариантов использования, привязанных к ней. Например, диаграмма использования для подсистемы "Навигация" могла бы выглядеть, как показано на Рис. 9-27. На этом рисунке для всех действующих лиц используется стереотип «внутренний». Это означает, что с точки зрения подсистемы данные элементы являются действующими лицами, в тоже время эти элементы являются внутренними для разрабатываемой системы. Это означает, что другие подсистемы становятся действующими лицами для подсистемы "Навигация". Строго говоря, в языке UML действующие лица не являются представлениями объектов, а являются ролями, которые играют объекты. Поэтому в Rhapsody необходимо создавать действующих лиц в виде отдельных элементов, независимых от подсистем, которые они представляют. Поэтому для имен действующих лиц я всегда использую префикс "ДЛ" (означающий «Действующее лицо»), чтобы отличать их от подсистем с аналогичными именами, хотя является или нет данный объект действующим лицом зависит от контекста .

Обратите внимание на то, что для вариантов использования используется стереотип «Уровень подсистем». Использование этого стереотипа не является обязательным; это

сделано, чтобы при просмотре диаграмм можно было сразу определить уровень абстракции вариантов использования.

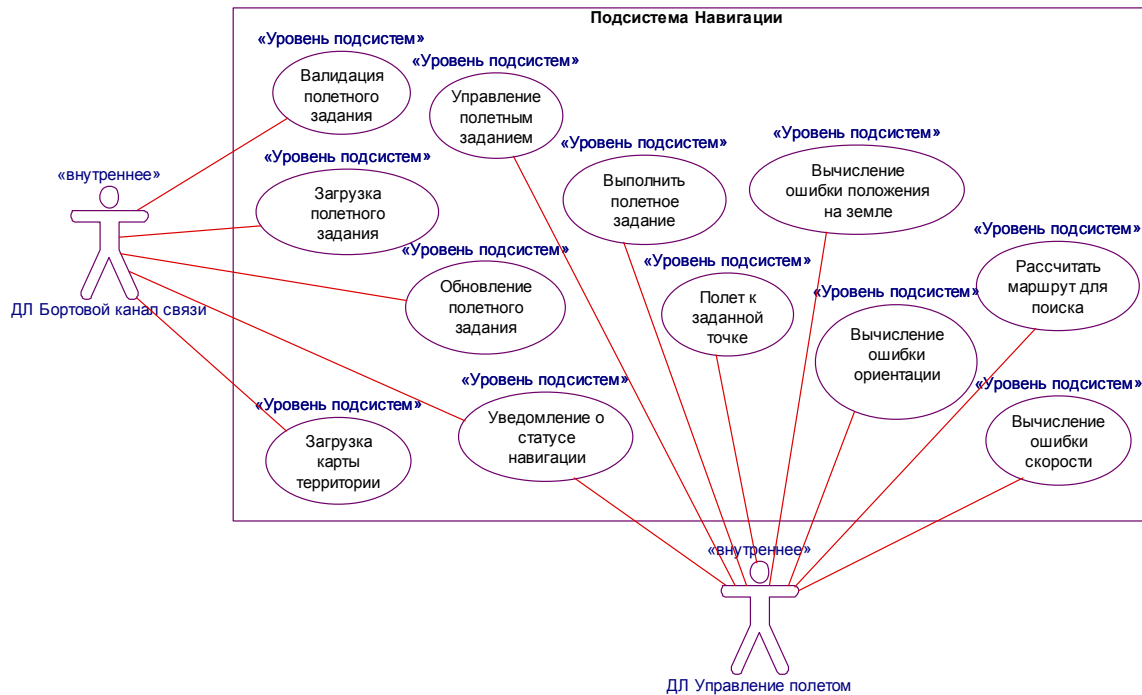


Рис. 9-27 Варианты использования подсистемы “Навигация”

В следующем примере, для подсистемы “Управление полетом” определено большее количество вариантов использования, и, соответственно, большее количество действующих лиц. Как и в предыдущем случае, они определены на основании вариантов использования уровня системы, путем привязки частей варианта использования уровня системы к рассматриваемым подсистемам. Поскольку некоторые варианты использования (например, вариант использования “Управление ВТ”⁶) связаны сразу с несколькими действующими лицами, некоторые действующие лица отображены по несколько раз на диаграмме. Это лишь означает, что они отображены несколько раз на диаграмме. Это позволяет уменьшить число пересечений линий на диаграмме и упростить внешний вид диаграммы.

И наконец, после выполнения декомпозиции на варианты использования для подсистем, результирующие варианты использования могут быть перемещены в пакеты модели, содержащие спецификаций подсистем. Для вариантов использования подсистемы “Управление полетом”, пакета подсистемы в представлении обозревателя модели будет выглядеть, как показано на Рис. 9-29.

⁶ Аббревиатура ВТ означает “Встроенный Тест”, стандартное сокращение для этого термина.

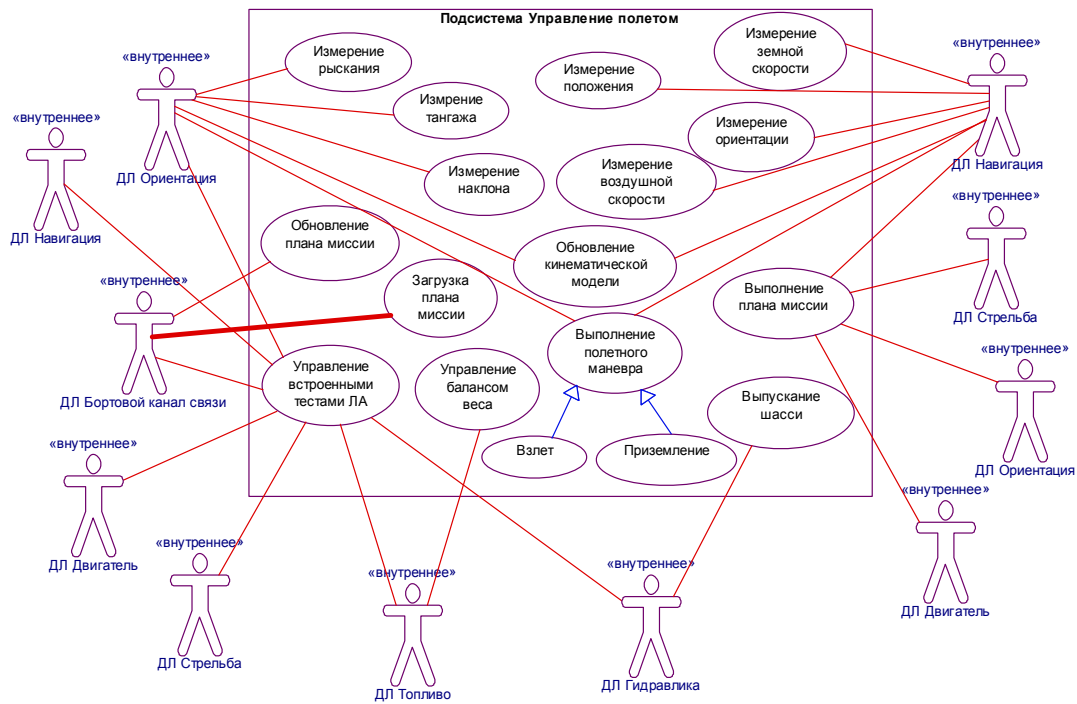


Рис. 9-28 Варианты использования подсистемы “Управление полетом”

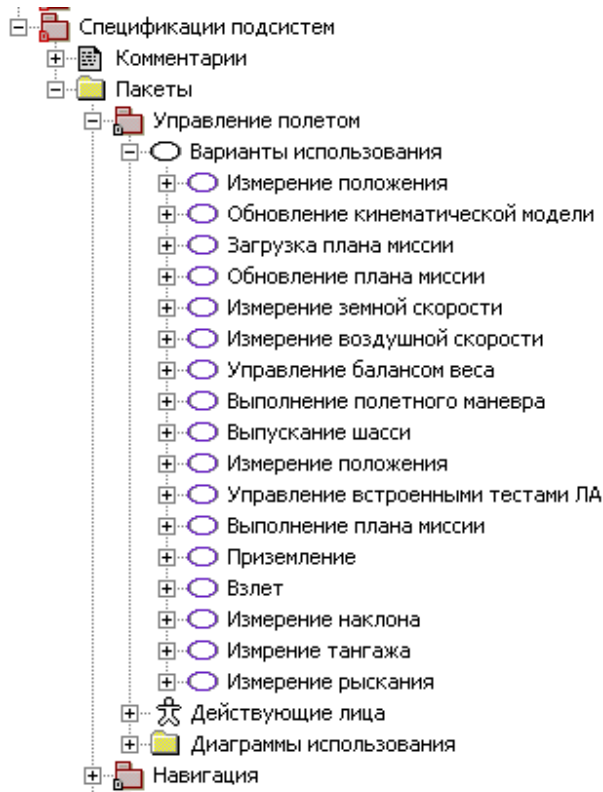


Рис. 9-29: Пакет со спецификацией подсистемы "Управление полетом"