**ФГБОУ ВО**

**Уфимский университет науки и технологий**

**Кафедра Автоматизированных Систем Управления**

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 100 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
| 90 |  |  |  |  |  |  |  |  |  |  |  |  |
| 80 |  |  |  |  |  |  |  |  |  |  |  |  |
| 70 |  |  |  |  |  |  |  |  |  |  |  |  |
| 60 |  |  |  |  |  |  |  |  |  |  |  |  |
| 50 |  |  |  |  |  |  |  |  |  |  |  |  |
| 40 |  |  |  |  |  |  |  |  |  |  |  |  |
| 30 |  |  |  |  |  |  |  |  |  |  |  |  |
| 20 |  |  |  |  |  |  |  |  |  |  |  |  |
| 10 |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |

UML проектирование систем реального времени

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

**к курсовой работе**

**по**   «Проектирование систем реального времени»

(наименование дисциплины)

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Группа |  |  | Фамилия, И., О. | Подпись | Дата | Оценка |
| ЭАС-412С |  |
|  |  |
| Студент | | | Мустафина М. Р. |  | 04.12.2023 |  |
| Консультант | | | Сулейманова А. М. |  |  |  |
| Принял | | |  |  |  |  |

**Уфа 2023 г.**

Оглавление

[ПОЯСНИТЕЛЬНАЯ ЗАПИСКА 1](#_Toc152297858)

[1. Описание задачи 3](#_Toc152297859)

[2. Модель прецедентов 4](#_Toc152297860)

[2.1. Прецедент «Выбор Типа Бензина» 5](#_Toc152297861)

[2.2. Прецедент «Количество бензина» 6](#_Toc152297862)

[2.3. Абстрактные прецеденты 6](#_Toc152297863)

[2.4. Абстрактный прецедент «Заливки бензина» 7](#_Toc152297864)

[2.5. Абстрактный прецедент «Планирование Заливки Типа Бензина» 7](#_Toc152297865)

[2.6. Конкретный прецедент «Выбор Типа Бензина» 8](#_Toc152297866)

[2.7. Конкретный прецедент «Количество бензина» 8](#_Toc152297867)

[3. Статическая модель предметной области 9](#_Toc152297868)

[4. Разбиение на объекты 10](#_Toc152297869)

[5. Динамическая модель 11](#_Toc152297870)

[5.1. Диаграмма кооперации для прецедента «Выбор Типа Бензина» 11](#_Toc152297871)

[5.2. Диаграмма кооперации для прецедента «Количество Бензина» 12](#_Toc152297872)

[5.3. Диаграмма кооперации для прецедента «Заливка бензина» 13](#_Toc152297873)

[5.4. Абстрактный прецедент «Осуществить Заливку Типа Бензина» 16](#_Toc152297874)

[6. Модель состояний 18](#_Toc152297875)

[7. Консолидация диаграмм кооперации 21](#_Toc152297876)

[8. Разбиение на подсистемы 22](#_Toc152297877)

[9. Разбиение системы на задачи 24](#_Toc152297878)

[9.1. Выделение задач в подсистеме бензоколонки 25](#_Toc152297879)

[9.2. Выделение задач в подсистеме этажа 27](#_Toc152297880)

[9.3. Выделение задач в подсистеме планировщика 28](#_Toc152297881)

[9.4. Определение интерфейсов задач 28](#_Toc152297882)

[9.5. Проектирование класса абстрагирования данных 29](#_Toc152297883)

[9.6. Обсуждение альтернативных архитектур 31](#_Toc152297884)

[10. Проект распределенной системы управления бензоколонки 31](#_Toc152297885)

[10.1. Структура подсистемы бензоколонки 32](#_Toc152297886)

[10.2. Структура подсистемы типа бензина 33](#_Toc152297887)

[10.3. Структура подсистемы планировщика 35](#_Toc152297888)

[10.4. Интерфейсы подсистем 36](#_Toc152297889)

[11. Проектирование скрывающих информацию классов 36](#_Toc152297890)

[11.1. Проектирование классов интерфейса устройств 36](#_Toc152297891)

[11.2. Проектирование класса, зависящего от состояния 38](#_Toc152297892)

[12. Разработка детального проекта программы 38](#_Toc152297893)

[12.1. Проектирование объектов-разъемов для бензоколонки 38](#_Toc152297894)

[12.2. Проектирование составных задач 40](#_Toc152297895)

[13. Конфигурирование целевой системы 41](#_Toc152297896)

[14. Анализ производительности нераспределенной системы управления бензоколонками 42](#_Toc152297897)

[14.1. Последовательности событий 42](#_Toc152297898)

[14.3. Назначение приоритетов 44](#_Toc152297899)

[14.4. Планирование в реальном времени для нераспределенной архитектуры 47](#_Toc152297900)

[14.5. Последовательность событий «Выбор Типа Бензина» 48](#_Toc152297901)

[14.6. Последовательность событий «Количество Бензина» 49](#_Toc152297902)

[14.7. Последовательность событий «Заливка Бензина» 50](#_Toc152297903)

[15. Анализ производительности распределенной системы управления бензоколонками 51](#_Toc152297904)

[15.1. Сценарий для анализа производительности 51](#_Toc152297905)

[15.2. Планирование в реальном времени для распределенной архитектуры 53](#_Toc152297906)

[15.3. Последовательность событий «Выбор Типа Бензина» 54](#_Toc152297907)

[15.4. Последовательность событий «Количество Бензина» 55](#_Toc152297908)

[15.5. Последовательность событий «Заливка Бензина» 56](#_Toc152297909)

[Вывод 58](#_Toc152297910)

[Список Литературы 59](#_Toc152297911)

1. Описание задачи

В каждой бензоколонке есть:

* Топливораздаточные колонки: устройства для подачи топлива в автомобиль. Они могут быть ручными или автоматическими, а также иметь различные типы пистолетов для разных видов топлива (например, бензин, дизельное топливо, сжатый природный газ и т.д.).
* Насосное оборудование: используется для перекачки топлива из резервуара в топливораздаточную колонку.
* Резервуары для хранения топлива: емкости для хранения различных видов топлива, обычно подземные или надземные.
* Система управления: компьютерная система, которая контролирует работу всей станции, включая управление раздаточными колонками, отчетность о продажах и учет топлива.
* Отчеты и системы учета: программное обеспечение для обработки и анализа данных о продажах топлива, мониторинге уровня топлива в резервуарах, а также для контроля качества топлива.
* Устройства безопасности: такие как датчики утечки топлива, противопожарное оборудование и системы блокировки и сигнализации для обеспечения безопасности на станции.
* Кассовый аппарат или система оплаты: для приема платежей от клиентов и ведения отчетности о продажах.
* Зона ожидания и обслуживания клиентов: может включать в себя скамейки, мусорные баки и другие удобства для клиентов.
* Освещение и знаки: для видимости и безопасности на станции в ночное время и в условиях плохой видимости.
* Автоматизированные системы самообслуживания: позволяют клиентам самостоятельно заправлять свои автомобили без помощи персонала станции.

2. Модель прецедентов

В системе заправкой автомобилей есть два актера: один представляет Клиент заправки, а второй – Сотрудник Заправки. Клиент заправки взаимодей­ствует с системой с помощью табло, которое находится на бензоколонке.

Клиент заправки инициирует два прецедента (рис.1), указанные в описании задачи:

– выбор типа бензина. Клиент, находящийся на заправке, нажимает кнопки «92», «95», «ДТ» для выбора типа, который ему необходим для автомобиля;

– количества бензина. Клиент, находящийся на заправке, вбивает количество бензина от 1 до 100 литров, для количества, который ему необходим для автомобиля;

Эти прецеденты рассматриваются ниже:

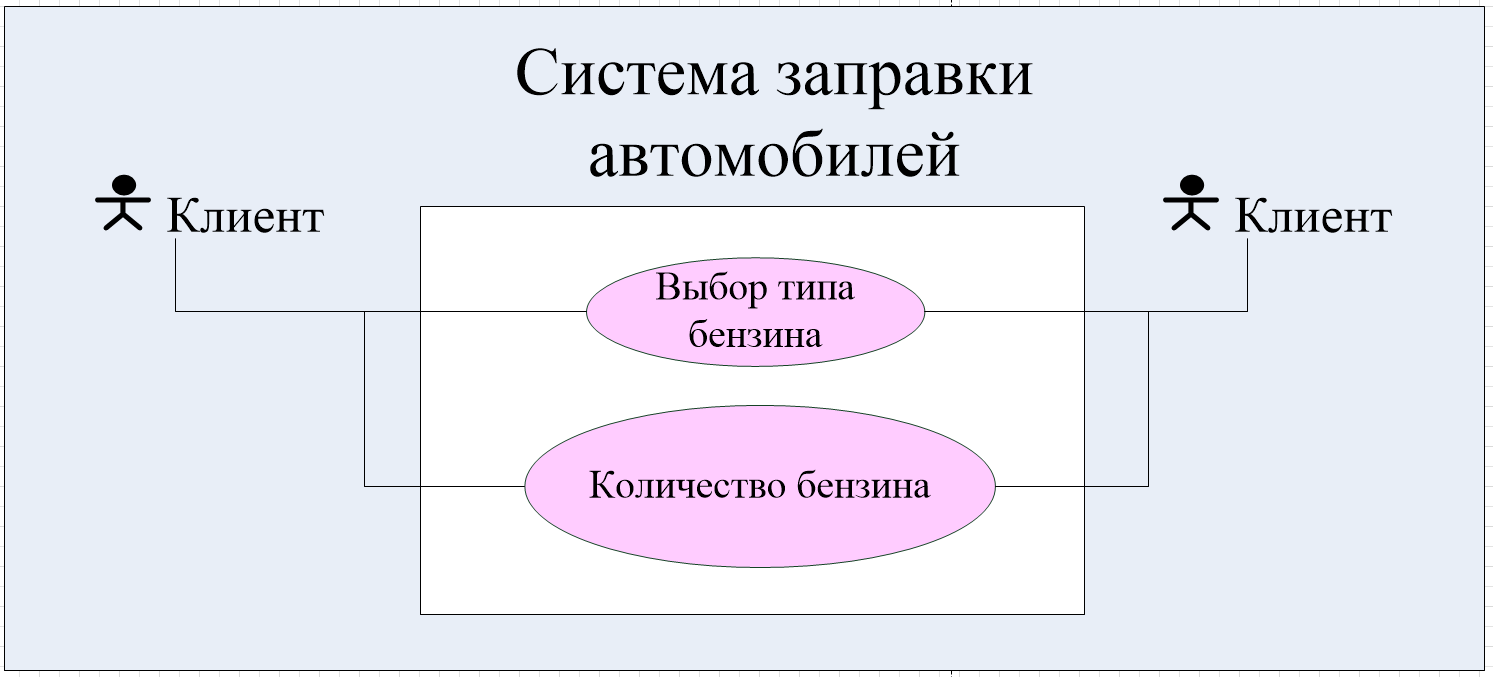


Рис.1. Актеры и прецеденты в системы заправки автомобилей

2.1. Прецедент «Выбор Типа Бензина»

Актеры. Клиент АЗС (главный), датчик залива бензина. Предусловие. Пользователь находится на АЗС у табло. Описание:

Клиент нажимает на табло кнопку «выбор типа бензина». Датчик кнопки табло посылает системе запрос, в котором указан выбранный клиентом тип бензина.

Новый запрос добавляется в список количества бензина, которое должна использовать бензоколонка. Если бензоколонка не заливает бензин, то система определяет, какой тип бензина выбран, чтобы обслужить очередной запрос. Система отдает бензоколонке команду выбора типа бензина. Как только тип бензина выбран, система выводит запрос на количество бензина.

По мере заливания бензина датчики топлива распознают количество залитого бензина и извещают об этом систему. Система проверяет, залито ли нужное количество бензина. Если да, то система отдает бензоколонке команду остановиться.

**Альтернативы:**

– клиент выбирает тип бензина 92, 95 или ДТ. Реакция системы такая же, как в глав­ной последовательности;

– если бензоколонка не заливает бензин и новых вызовов нет, то она остает­ся в покое.

**Постусловие.** Бензоколонка залила определенный тип бензина, выбранный клиентом.

2.2. Прецедент «Количество бензина»

**Актеры.** Клиент АЗС (главный), датчик залива бензина.

Пользователь находится на АЗС у табло.

**Предусловие.** Пользователь находится у табло и выбирает количество бензина.

**Описание:**

Клиент, находящийся на заправке, вбивает количество бензина от 1

до 100 литров, для количества, который ему необходим для автомобиля. Датчик количества бензина посылает системе запрос, содержащий указание на количества, которое необходимо клиенту.

Система выбирает, какую бензоколонку необходимо использвать.

Новый запрос добавляется в список количества бензина, которое должна использовать бензоколонка. Если бензоколонка не заливает, система определяет, в какой бензоколонке выбрано количество бензина для удовлетворения запроса. Система отдает бензоколонке команду выбора количества бензина. Как только количество бензина выбрано, система приказывает бензоколонке заливать бензин – 92, 95 или ДТ.

По мере заливания бензина датчики топлива распознают количество

залитого бензина и извещают об этом систему. Система проверяет, залито ли нужное количество бензина. Если да, то система отдает бензоколонке команду остановиться.

**Альтернативы:**

– клиент выбирает тип бензина 92, 95 или ДТ. Реакция системы такая

же, как в глав­ной последовательности;

– если бензоколонка не заливает бензин и новых вызовов нет, то она

остает­ся в покое.

**Постусловие.** Бензоколонка залила определенный тип бензина,

выбранный клиентом.

2.3. Абстрактные прецеденты

Анализ данных прецедентов показывает, что есть две общие последовательно­сти, которые допустимо вынести в абстрактные прецеденты, а затем включить в новые, более простые варианты исходных прецедентов. Первый абстрактный прецедент — это планирование заливки типа бензина в ответ на запрос клиента. Его можно так и назвать - Планирование заливки типа бензина. Другую общую последователь­ность, связанную с заливкой бензина, удобно вынести в абстрактный прецедент Заливка бензина. Отношения между прецедентами показаны на Рис.2. Оба прецедента – Выбор типа бензина и Количество бензина - включа­ют абстрактные прецеденты, которые описываются ниже.

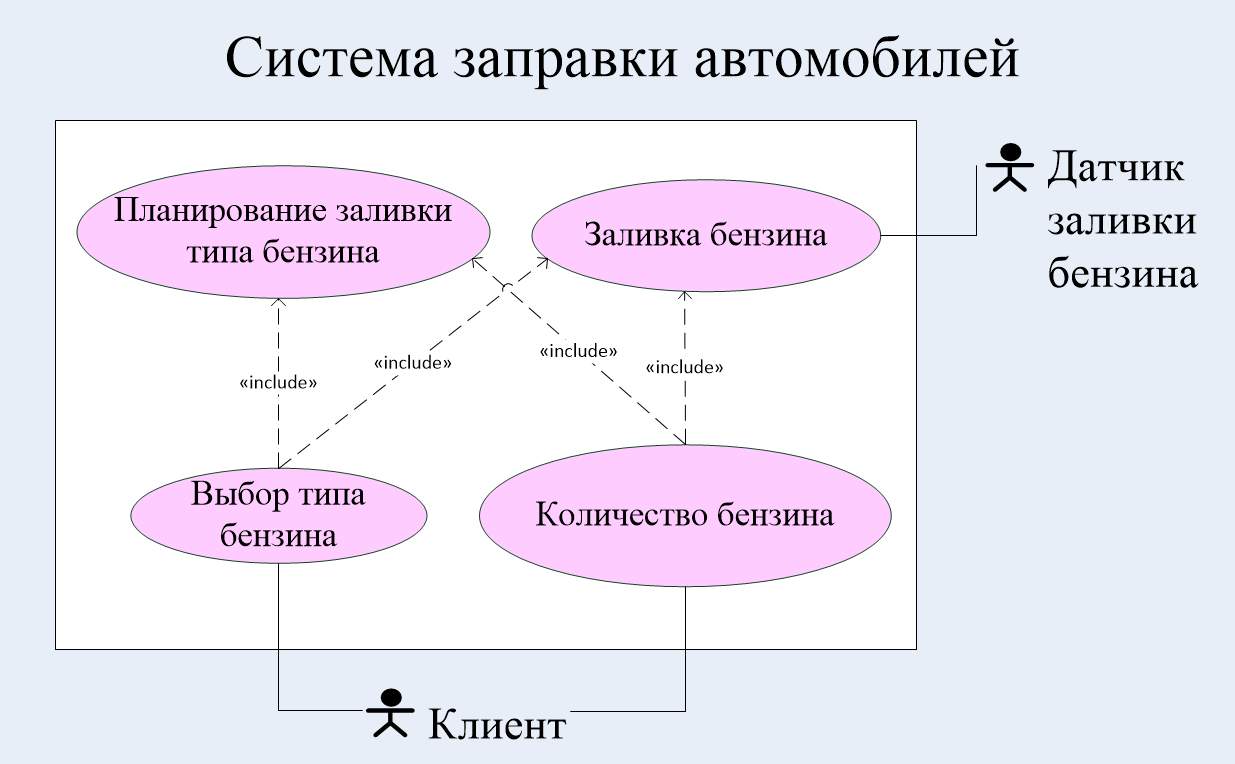


Рис.2. Модель прецедентов с абстрактными прецедентами

2.4. Абстрактный прецедент «Заливки бензина»

**Актер.** Датчик заливки бензина.

**Предусловие.** Бензоколонка заливает.

**Описание.**

Датчик заливки бензина распознает, что тип бензина выбран и можно начинать заливку и извещает об этом систему. Система проверяет, должна ли бензоколонка начать подачу бензина. Если да, то система отдает бензоколонке команду начать.

**Альтернатива.** Бензоколонке не нужно останавливаться на данном этапе, и она продолжает заливку бензина.

**Постусловие.** Бензоколонка начала заливку.

2.5. Абстрактный прецедент «Планирование Заливки Типа Бензина»

**Предусловие.** Бензоколонка должна залить хотя бы один тип бензина.

**Описание.**

Система выбирает, тип бензина выбрать, чтобы удовлетворить очередной запрос. Система отдает бензоколонке команду заливать бензин. Как только бензина зальется нужное количество, бензоколонка остановится.

**Альтернатива.** Если вызовов нет, то бензоколонка остается в покое.

**Постусловие.** Бензоколонка заливает выбранный тип бензина.

2.6. Конкретный прецедент «Выбор Типа Бензина»

Перепишем прецедент Выбор Типа Бензина с использованием абстракт­ных прецедентов.

**Актеры**. Клиент АЗС (главный), датчик залива бензина.

**Предусловие.** Пользователь находится на АЗС у табло.

**Описание:**

Клиент нажимает на табло кнопку «выбор типа бензина». Датчик кнопки

табло посылает системе запрос, в котором указан выбранный клиентом тип бензина.

Новый запрос добавляется в список добавляется в список количества

бензина, которое должна использовать бензоколонка. Если бензоколонка не заливает бензин, то система определяет, какой тип бензина выбран, чтобы обслужить очередной запрос. Система отдает бензоколонке команду выбора типа бензина. Как только тип бензина выбран, система выводит запрос на количество бензина.

По мере заливания бензина датчики топлива распознают количество

залитого бензина и извещают об этом систему. Система проверяет, залито ли нужное количество бензина. Если да, то система отдает бензоколонке команду остановиться.

**Альтернативы:**

– клиент выбирает тип бензина 92, 95 или ДТ. Реакция системы такая же, как в глав­ной последовательности;

– если бензоколонка не заливает бензин и новых вызовов нет, то она остает­ся в покое.

**Постусловие.** Бензоколонка залила определенный тип бензина, выбранный клиентом.

2.7. Конкретный прецедент «Количество бензина»

Перепишем прецедент Количество бензина с использованием абстрактных преце­дентов.

**Актеры.** Клиент АЗС (главный), датчик залива бензина.

Пользователь находится на АЗС у табло.

**Предусловие.** Пользователь находится у табло и выбирает количество бензина.

**Описание:**

Клиент, находящийся на заправке, вбивает количество бензина от 1

до 100 литров, для количества, который ему необходим для автомобиля.

Датчик количества бензина посылает системе запрос, содержащий указание на количества, которое необходимо клиенту.

Система выбирает, какую бензоколонку необходимо использовать.

Новый запрос добавляется в список количества бензина, которое должна использовать бензоколонка. Если бензоколонка не заливает, система определяет, в какой бензоколонке выбрано количество бензина для удовлетворения запроса. Система отдает бензоколонке команду выбора количества бензина. Как только количество бензина выбрано, система приказывает бензоколонке заливать бензин – 92, 95 или ДТ.

По мере заливания бензина датчики топлива распознают количество

залитого бензина и извещают об этом систему. Система проверяет, залито ли нужное количество бензина. Если да, то система отдает бензоколонке команду остановиться.

**Альтернативы:**

– клиент выбирает тип бензина 92, 95 или ДТ. Реакция системы такая

же, как в глав­ной последовательности;

– если бензоколонка не заливает бензин и новых вызовов нет, то она

остает­ся в покое.

**Постусловие.** Бензоколонка залила определенный тип бензина,

выбранный клиентом.

3. Статическая модель предметной области

В статической модели отражаются статические отношения, существующие в системе заправкой автомобилями. Для начала рассмотрим физические классы предметной области (рис.3). Бензоколонка — это составной класс, содержащий Кнопку Типа Бензина, Кнопку Количества Бензина.

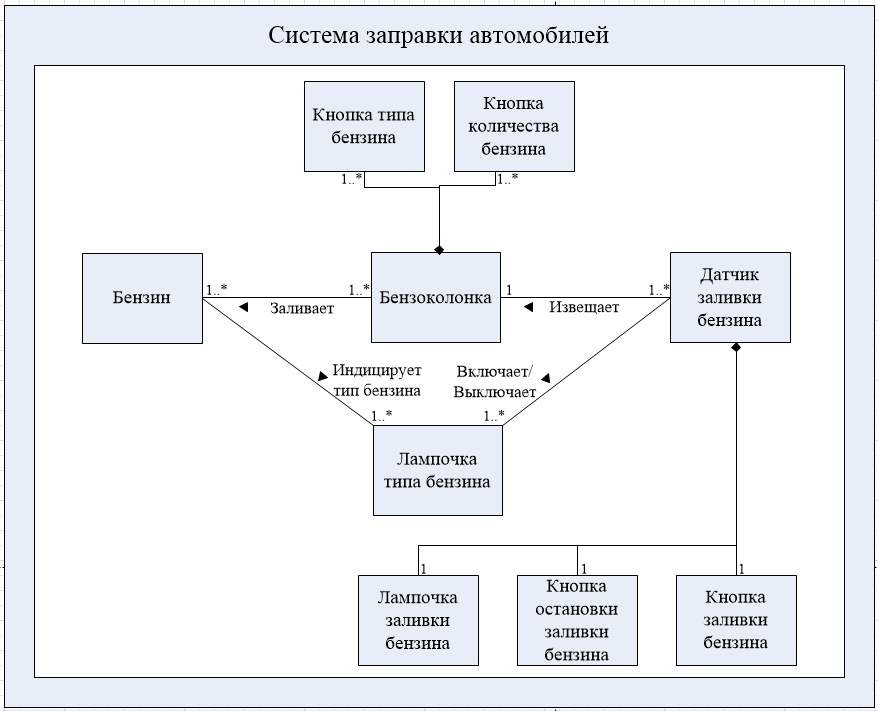


Рис.3. Концептуальная статическая модель системы управления

Класс Бензоколонка ассоциирован также с классом Датчик Заливки Бензина, который изве­щает о заливке бензина в бак, и с классом Лампочка Типа Бензина, кото­рую Бензоколонка включает и выключает. Класс Лампочка Типа Бензина ассоциирован с классом Бензин.

Из диаграммы контекста системы видно, что, за исключением перечисленных составных классов, остальные классы на рис.3, представляющие объекты ре­ального мира, — это внешние устройства, то есть датчики или приводы. Поэтому они изображены в виде классов внешних устройств ввода или вывода, которые осуществляют интерфейс с системой заправки автомобилей (рис. 4).

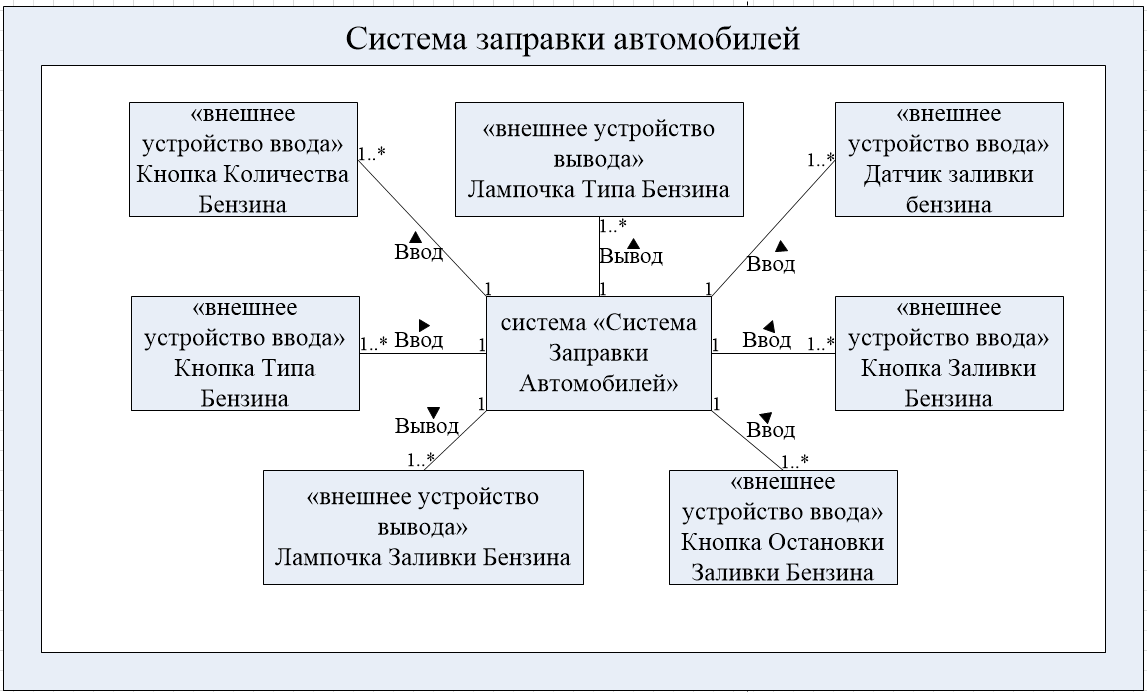


Рис.4. Диаграмма контекста классов для системы заправки автомобилей

4. Разбиение на объекты

Чтобы подготовиться к динамическому моделированию, рассмотрим программ­ные объекты в системе заправкой автомобилей. Объект Заправка Автомобилей составлена из несколь­ких объектов. Поскольку она получает информацию от внешних объектов и управ­ляет внешними объектами, то некоторые его компоненты — это объекты интерфейса устройств ввода/вывода, а именно датчиков и приводов бензоколонок.

Для каждого объекта внешнего устройства существует соответствующий объект программного интерфейса. Так, Бензоколонка получает вызовы от объектов Интерфейс Кнопки Заливки Бензина. В Бензоколонке есть также объект Интерфейс Кнопки остановки Заливки Бензина, взаимодействующие соответственно с фи­зической кнопкой Остановки бензина. Помимо этого, Бензоколонка посылает команды объекту Ин­терфейс Лампочки Заливки Бензина. На каждой бензоколонке находится по 3 экземпляра класса Интерфейс Кнопки Типа Бензина. Существует также объект Интерфейс Датчика Заливки Бензина, который посылает сообщения объекту Управление Бензоколонкой, когда бензин практически залит, и объект Интерфейс Лампочки Типа Бензина для взаи­модействия с физической лампочкой.

Помимо объектов интерфейса устройств, выявленных с помощью диаграммы контекста классов, нужны еще сущностные и управляющие объекты. Для каждо­й бензоколонки необходим сущностный объект Состояние и План Работы Бензоколонки. Под состоянием понимается информация о том, работает ли бензоколонка вообще, сколько заливает бензина и какой тип бензина. Сюда же относят план Работы Бензоколонки, включающий список запросов, которые бензоколонка должна выполнить.

С каждой бензоколонкой связан зависящий от состояния управляющий объект Управление Бензоколонкой, в ведении которого находятся заливка и остановка заливки бензина. Он управляет заливкой и остановкой бензина. Состояния этого объекта изображены на диаграмме состояний бензоколонки. Поскольку запросы на использование бензоколонки могут поступать в любое время, то необходимо создать отдельный объект-координатор Диспетчер Бензоколонки, который будет при­нимать запросы и обновлять план работы Бензоколонки. Еще одна проблема - выбор Бензоколонки, которая обслужит поступивший запрос. Избежать сложностей можно, включив еще один координирующий объект Планировщик, который бу­дет заниматься поиском подходящей или свободной бензоколонки.

5. Динамическая модель

Далее следует определить взаимодействия между объектами, соответствую­щие каждому прецеденту. В этом примере мы будем пользоваться диаграммами кооперации. Для каждого прецедента разрабатывается своя диаграмма кооперации, на которой изображены участвующие в нем объекты и последовательность их взаи­модействий. Кроме того, если в кооперацию вовлечен зависящий от состояния объект Управление Бензоколонкой, то последовательность событий также изображается на диаграмме состояний. Ниже приводятся описания сообщений для каждого прецедента.

5.1. Диаграмма кооперации для прецедента «Выбор Типа Бензина»

Диаграмма кооперации для прецедента Выбор Типа Бензина изображена на рис.5. Поскольку Запрос Кнопки Заливки Бензина может поступить в момент, ког­да Бензоколонка (точнее, объект Управление Бензоколонкой) занят обслуживанием предыдуще­го запроса, то ответственность за его обработку возлагается на объект Диспетчер Бензоколонкой. Последовательность сообщений описывается следующим образом:

Е1: Поступил Запрос Кнопки Заливки Бензина объекту Интерфейс Кнопки Заливки Бензина.

Е2: Интерфейс Кнопки Заливки Бензина посылает Запрос Бензоколонки объекту Диспетчер Бензоколонки.

Е3: Диспетчер Бензоколонки просит объект Состояние и План Работы Бензоколонки добавить запрос в список работы, которые необходимо сделать

Е4: Обновляется Работы Бензоколонки. Объекту Диспетчер Бензолонки возвра­щается подтверждение.

Е5: Объект Диспетчер Бензоколонки отправляет Планировщику сообщение Обя­зательство Бензоколонки, информирующее о том, что Бензоколонке необходимо выполнить запрос.

Е5а: Если Бензоколонки не заливает, то Диспетчер Бензоколонки посылает сообщение 92 (95 или ДТ) объекту Управление Бензоколонки, извещающее, какой тип бензина заливать. Этот случай обрабатывается в абстрактном прецеденте Осуществить Заливку Типа Бензина.



Рис.5. Диаграмма кооперации для прецедента *Выбор Типа Бензина*

5.2. Диаграмма кооперации для прецедента «Количество Бензина»

Диаграмма кооперации для прецедента Количество Бензина изображена на рис.6. Поскольку вызов поступает от кнопки количества бензина, нужно решить, какая бензоколонка обслу­жит запрос. Решение принимается Планировщиком, у которого есть информация о состоянии и план. Вот описание последовательности сообщений:

F1: Поступил Запрос Кнопки Заливки Бензина объекту Интерфейс Кнопки Количества Бензина.

F2: Объект Интерфейс Кнопки Количества Бензина посылает Запрос на Обслуживание объекту Планировщик.

F3: Планировщик выбирает Бензоколонку и передает Запрос Планировщика объекту Диспетчер Бензоколонки, входящему в состав выбранной Бензоколонки.

F4: Объект Диспетчер Бензоколонки отправляет сообщение Обновить объекту Со­стояние и План Движения Бензоколонки, чтобы тот добавил новый запрос в список работы, который необходимо выполнить.

F5: Объекту Диспетчер Бензоколонки возвращается подтверждение, где, в частно­сти, содержится информация о том, заливает бензоколонка или нет.

F6: Объект Диспетчер Бензоколонка посылает Планировщику сообщение Обяза­тельство Бензоколонки.

F6a: Если бензоколонка не заливает, то Диспетчер Бензоколонка посылает сообщение 92 (95 или ДТ) объекту Управление Бензоколонкой, извещающее, какой тип бензина заливать. Этот случай обрабатывается в абстрактном прецеденте Осуществить Заливку Типа Бензина.



Рис.6. Диаграмма кооперации для прецедента Количество Бензина

5.3. Диаграмма кооперации для прецедента «Заливка бензина»

Поскольку прецедент Осуществить Заливку Типа Бензина на Бензоколонке зависит от состояния, он изображается одновременно на диаграмме кооперации (рис.7) и на диаграм­ме состояний (рис.8). Состояния, в которые переходит объект, перечислены в описании последовательности сообщений ниже; предусловие состоит в том, что бензоколонка находится в состоянии Бензоколонка Работает:

А1: Объект Интерфейс Датчика Заливки Бензина получает информацию от внешнего устройства.

А2: Интерфейс Датчика Заливки Бензина посылает тип бензина в сообщении Обслуживание Запроса объекту Управление Бензоколонкой.

A3: Объект Управление Бензоколонкой отправляет сообщение Проверить Тот Ли Тип Бензина объекту Состояние и План Работы Бензоколонки, который проверяет, какой тип бензина выбран.

А4: объект Состояние и План Работы Бензоколонки передает сообщение Выполняю Запрос объекту Управление Бензоколонкой. В сообщении содержится тип бензина.

А5: Объект Управление Бензоколонкой посылает сообщение Запуск Заправочного Пистолета объекту Интерфейс Заправочного Пистолета.

А6: Объект Интерфейс Заправочного Пистолета посы­лает объекту Заправочный Пистолет сообщение Запустить Заправочный Пистолет.

А7: Получает ответ от Заправочного Пистолета.

А8: Бензоколонка заливает.

А9: Объект Управление Бензоколонкой посылает сообщение Начинаю Заливку объекту Планировщик.

А10: Объект Управление Бензоколонкой посы­лает объекту Интерфейс Лампочки Типа Бензина сообщение Включить Лампочку Типа Бензина (с параметром «92», «95» или «ДТ»), которое зажи­гает лампочку (А10а.1).

А11: в результате перехода объект Управление Бензоколонкой передает сообщение Проверить Следующий Запрос объекту Состояние и План Работы Бензоколонки. Его цель - выяснить, нужно ли будет еще бензоколонке заливать бензин или можно завершить работу. Если у Бензоколонки нет невыполненных запросов, то она переходит в состояние Бензоколонка не заливает (событие А12), в противном случае выполняется пре­цедент Осуществить Заливку Типа Бензина.

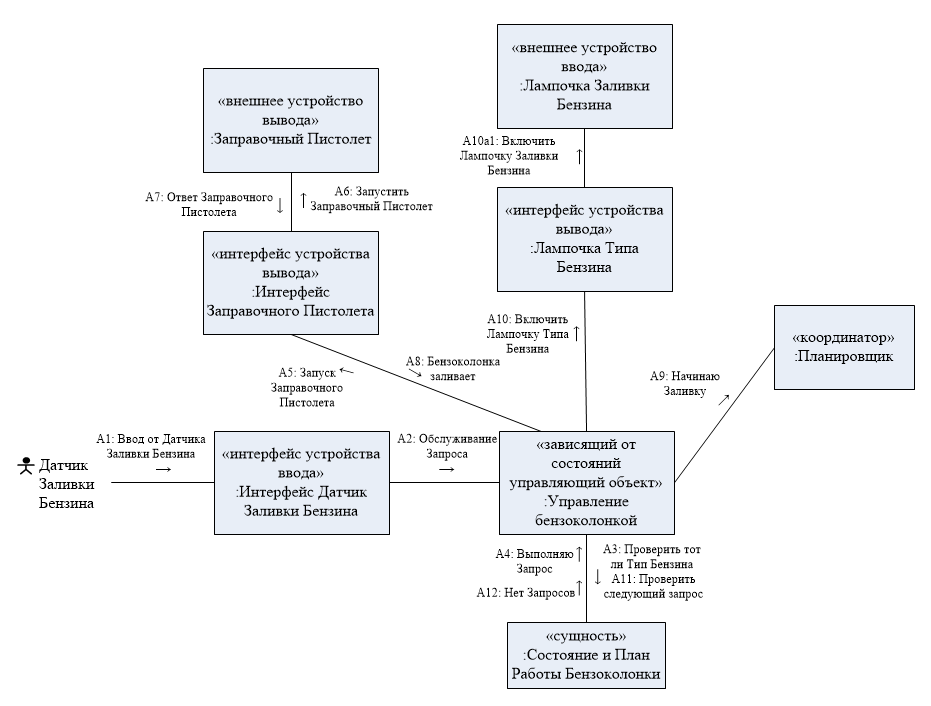


Рис.7. Диаграмма кооперации для прецедента Заливка Бензина

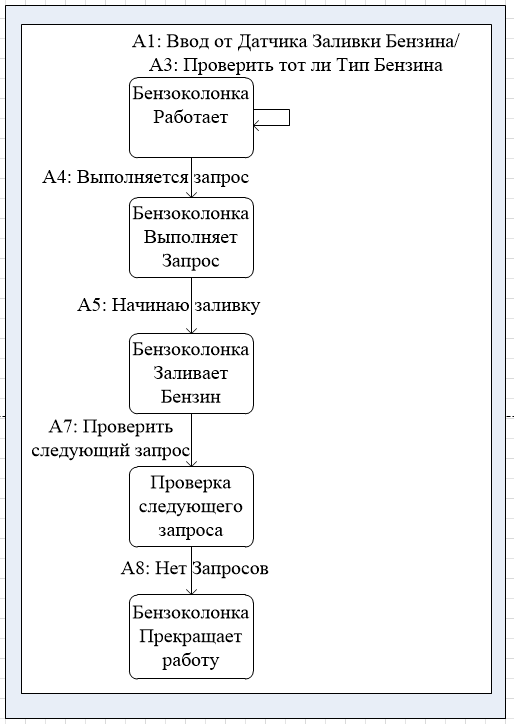


Рис.8. Прецедент Заливка Бензина: диаграмма состояния объекта Управление Бензоколонкой

5.4. Абстрактный прецедент «Осуществить Заливку Типа Бензина»

Диаграмма кооперации для абстрактного прецедента Осуществить Заливку Типа Бензина.

D1: {Объект-источник} отправляет сообщение Запрос 92 объекту Управление Бензоколонкой.

D2: Объект Управление Бензоколонкой дает команду Включить Лампочку Заливки Бензина объекту Интерфейс Лампоч­ки Заливки Бензина, который включает физическую лампочку (D2. l).

D3: Объект Управление Бензоколонкой дает команду Включить Лампочку Типа Бензина объекту Интерфейс Лампоч­ки Типа Бензина «92», который включает физическую лампочку (D3. l).

D4: Объект Управление Бензоколонкой отправляет сообщение Начинаю Заливку объекту Планировщик.

D5: Объект Управление Бензоколонкой отправляет сообщение Заливка Происходит объекту Состояние и План Работы Бензоколонки.

D6: Объект Управление Бензоколонкой отправляет сообщение Бензоколонка Работает объекту Состояние и План Работы Бензоколонки.

Последовательность шагов будет такой же и при получении сообщения Запрос 95, только слово 92 надо всюду заменить словом 95.

Последовательность шагов будет такой же и при получении сообщения Запрос ДТ, только слово 92 надо всюду заменить словом ДТ.

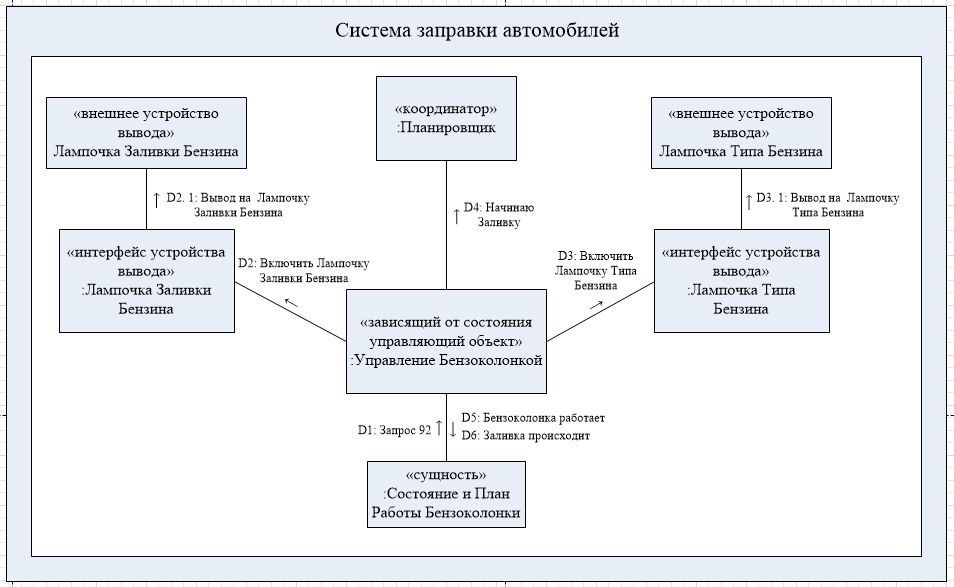


Рис.9. Диаграмма кооперации для прецедента Заливка Бензина

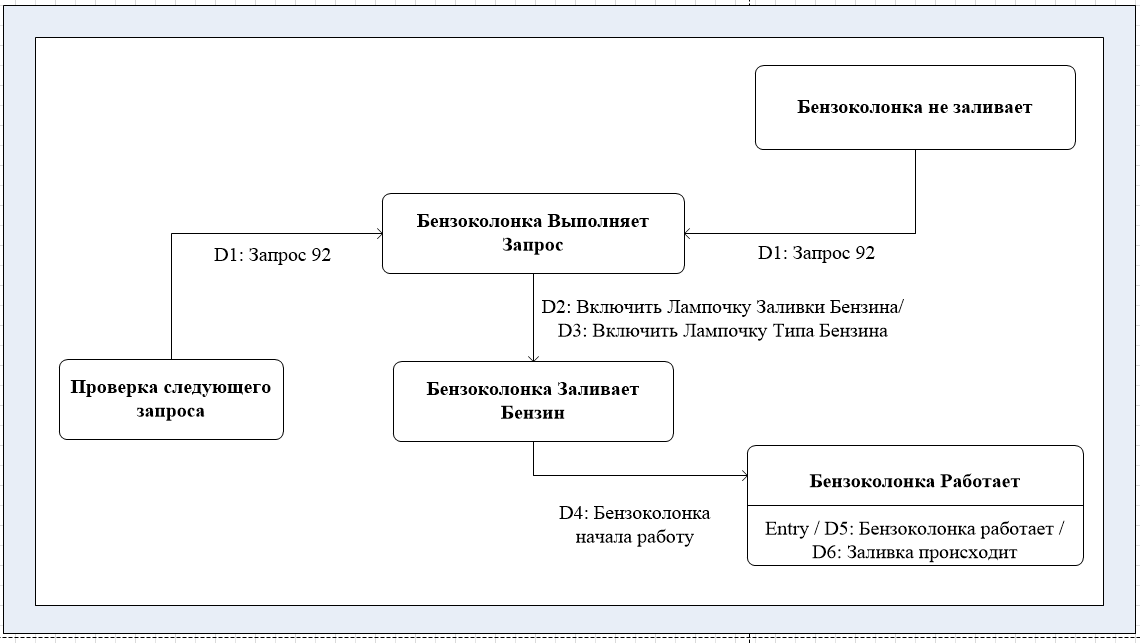


Рис.10. Прецедент Заливки Бензина: диаграмма состояния объекта Управление Бензоколонкой

6. Модель состояний

Поскольку мы получили три зависящих от состояния прецедента, то для создания полной диаграммы состояний необходимо консолидировать три частичные диаграммы и рассмотреть альтернативные ветви. Полная диаграмма, содер­жащая последовательности событий Заливка Бензина и Остановки Заливки Бензина, изображена на рис.11.

Это плоская диаграмма состояний, но ее можно превратить в иерархическую, определив надсостояния, соответствующие основным группам состояний бензоколонки. Надсостояния и подсостояния представлены на иерархической диаграмме и опи­саны ниже. Диаграмма верхнего уровня (включающая события, но не действия), показана на рис.12, а полная диаграмма состояний - на рис.13.

Бензоколонка не заливает. Бензоколонка находится в спокойном состоянии, необслуженных вызовов нет.

Подготовка к Заливке Бензина. Бензоколонка Выполняет Запрос «92». Это надсостояние включает следующие подсостояния:

– Включить Лампочку Заливки Бензина.

– Включить Лампочку Типа Бензина «92».

– Бензоколонка начинает работу. В такое состояние Бензоколонка переходит, когда Включилась Лампочка Заливки Бензина и Включилась Лампочку Типа Бензина.

Подготовка к Заливке Бензина Бензоколонка. Выполняет Запрос «95». Это надсостояние включает следующие подсостояния:

– Включить Лампочку Заливки Бензина.

– Включить Лампочку Типа Бензина «95».

– Бензоколонка начинает работу. В такое состояние Бензоколонка переходит, когда Включилась Лампочка Заливки Бензина и Включилась Лампочку Типа Бензина.

Подготовка к Заливке Бензина. Бензоколонка Выполняет Запрос «ДТ». Это надсостояние включает следующие подсостояния:

– Включить Лампочку Заливки Бензина.

– Включить Лампочку Типа Бензина «ДТ».

– Бензоколонка начинает работу. В такое состояние Бензоколонка переходит, когда Включилась Лампочка Заливки Бензина и Включилась Лампочку Типа Бензина.

5. Заливка Бензина. Надсостояние включает следующие подсостояния:

Бензоколонка работает. Бензоколонка переходит в такое состояние, когда начинает заливку бензина;

Заливка Происходит. Переход в указанное состояние осуществляется, когда Бензоколонка начала заливку;

5. Проверка Следующего Запроса. В таком состоянии Бензоколонка про­веряет, следует ли ей выполнить другой запрос (какой именно Тип Бензина) или нужно перейти в состояние Нет Запросов.

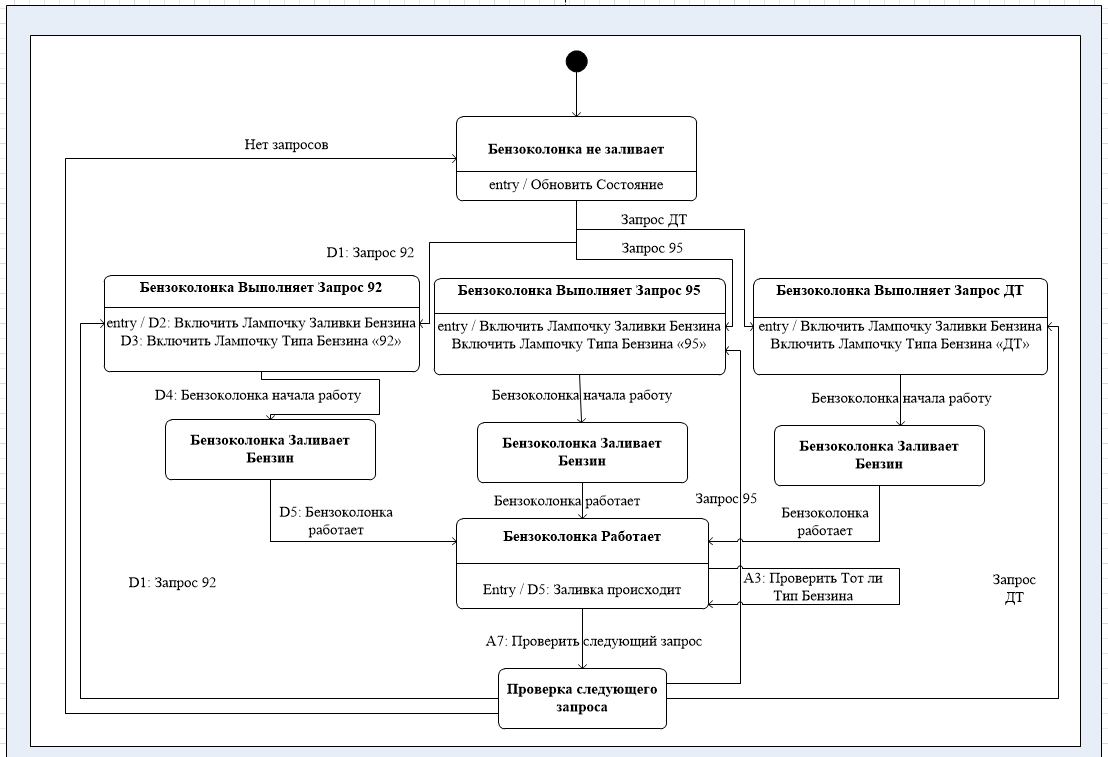


Рис.11. Диаграмма состояния объекта Управление Бензоколонкой

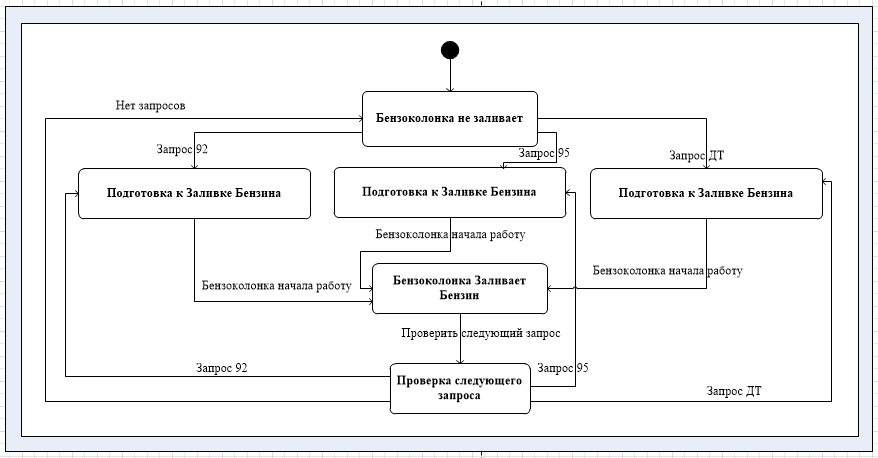


Рис.12. Диаграмма состояний верхнего уровня объекта Управления Бензоколонкой

7. Консолидация диаграмм кооперации

Результат консолидации диаграмм кооперации, описывающих все четыре преце­дента, показан на рис.14. Некоторые объекты принимают участие в нескольких прецедентах. Другие объекты задействованы только в одном прецеденте.

Для объектов, которые принимают участие лишь в одном прецеденте, все взаи­модействия изображены на диаграмме кооперации, описывающей этот прецедент. Если же объект занят в нескольких прецедентах, его взаимодействия представлены на разных диаграммах и объединены на консолидированной диаграмме кооперации.

На консолидированный диаграмме кооперации должны присутствовать все взаимодействия объектов, в том числе и альтернативы, которые обычно на отдель­ных диаграммах кооперации не изображаются. Поэтому на рис.14 показаны все сообщения «92», все сообщения «95», все сообщения «ДТ». Кроме того, имена сообщений можно агрегировать. Так, Команда Лампочке Направления — это результат аг­регирования четырех сообщений: Включить Лампочку Типа Бензина «92», Выключить Лампочку Типа Бензина, Включить Лампочку Заливки Бензина «92», Выключить Лампочку Заливки Бензина.

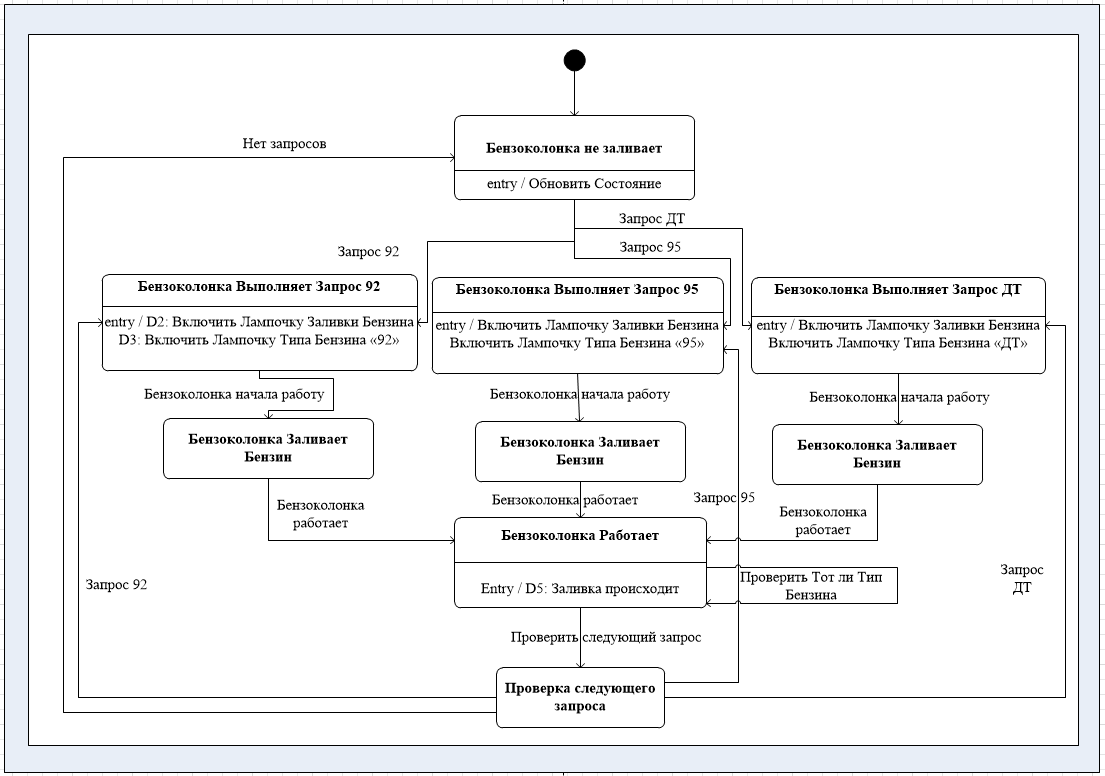


Рис.13. Иерархическая диаграмма состояний объекта Управление Бензоколонкой

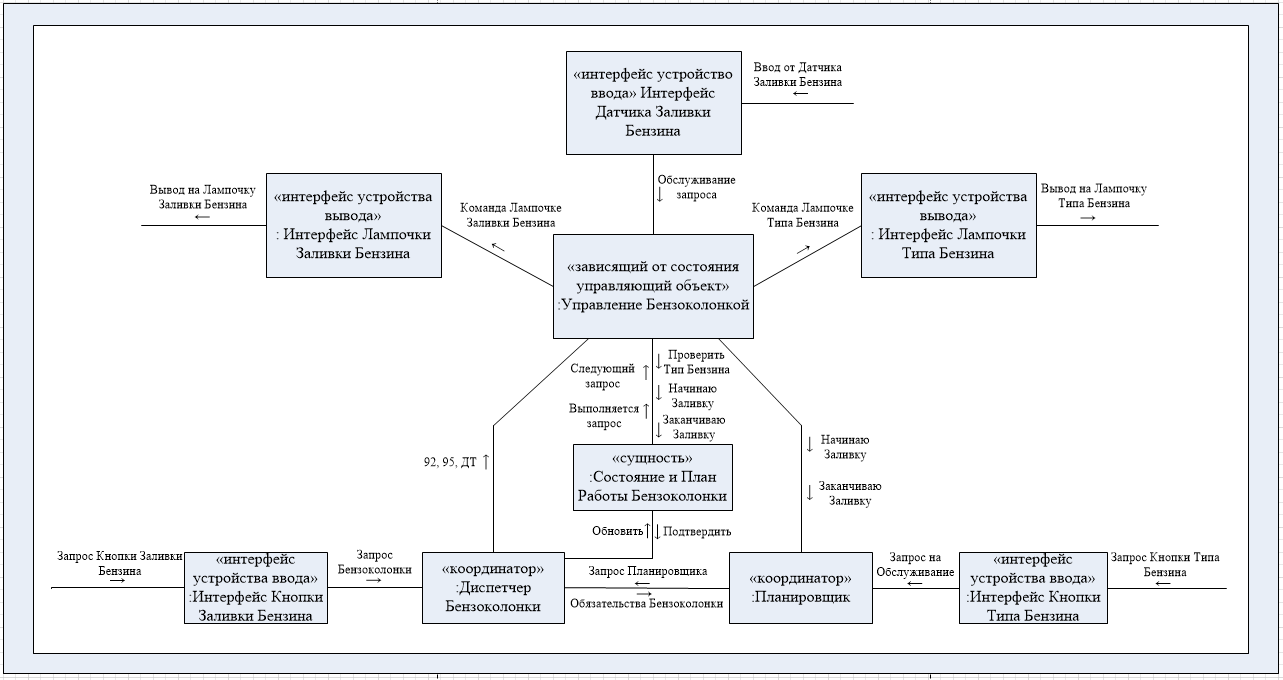


Рис.14. Система управления бензоколонками: консолидированная диаграмма кооперации

8. Разбиение на подсистемы

На следующем шаге система разбивается на подсистемы. Поскольку потенци­ально это приложение является распределенным, то прежде всего применяются рекомендации, касающиеся географического местоположения и агрегирования/композиции. В частности, все объекты-бензоколонки (числом n) являются частями со­ставного объекта Подсистема Бензоколонки. Каждой бензоколонке нужны объекты Управле­ние Бензоколонкой, Диспетчер Бензоколонки и Состояние и План Движения Бензоколонки. Составной объект Подсистема Типа Бензина включает в себя Интерфейс Лампочки Типа Бензина и Интерфейс Кнопки Типа Бензина. Другие объекты мо­гут относиться к Подсистеме Бензоколонки, Подсистеме Типа Бензина или к какой-то другой подсистеме. В качестве примера назовем Интерфейс Датчика Заливки Бензина. Этот объект отнесен к Подсистеме Бензоколонки, поскольку связан с ней наиболее тесно, о чем свидетельствует его участие в прецеденте Заливка Бензина. С другой стороны, объект Интерфейс Лампочки Типа Бензина принадлежит Подсистеме Типа Бензина. Наконец, объект-координатор Планировщик помещается в сво­ей собственной подсистеме, поскольку не зависит ни от числа типов бензина, ни от числа бензоколонок.

На рис.15 показана общая структура разбиения на подсистемы бензоколонки, типа бензина и планировщика. Подсистема Бензоколонки– это управляющая подсистема, Подсистема Типа Бензина– подсистема сбора данных, а подсистема Планировщик–координирующая. Структура Подсистемы Бензоколонки представлена на рис.16, а Подсистемы Типа Бензина– на рис.17.

Дополнительно разрабатывается уточненная статическая модель, изображаемая на диаграмме классов. Эта диаграмма выводится из общей архитектуры подсистем и структуры каждой подсистемы. На диаграмме классов отражены все классы, из которых создаются объекты, представленные на диаграммах кооперации, а также отношения между этими классами, то есть сами кооперации. На рис.18 приведе­на уточненная статическая модель, в которой каждая подсистема предстает в виде составного класса. Становится очевидным, как программные понятия соответству­ют статической модели предметной области, созданной ранее (см. рис.3).

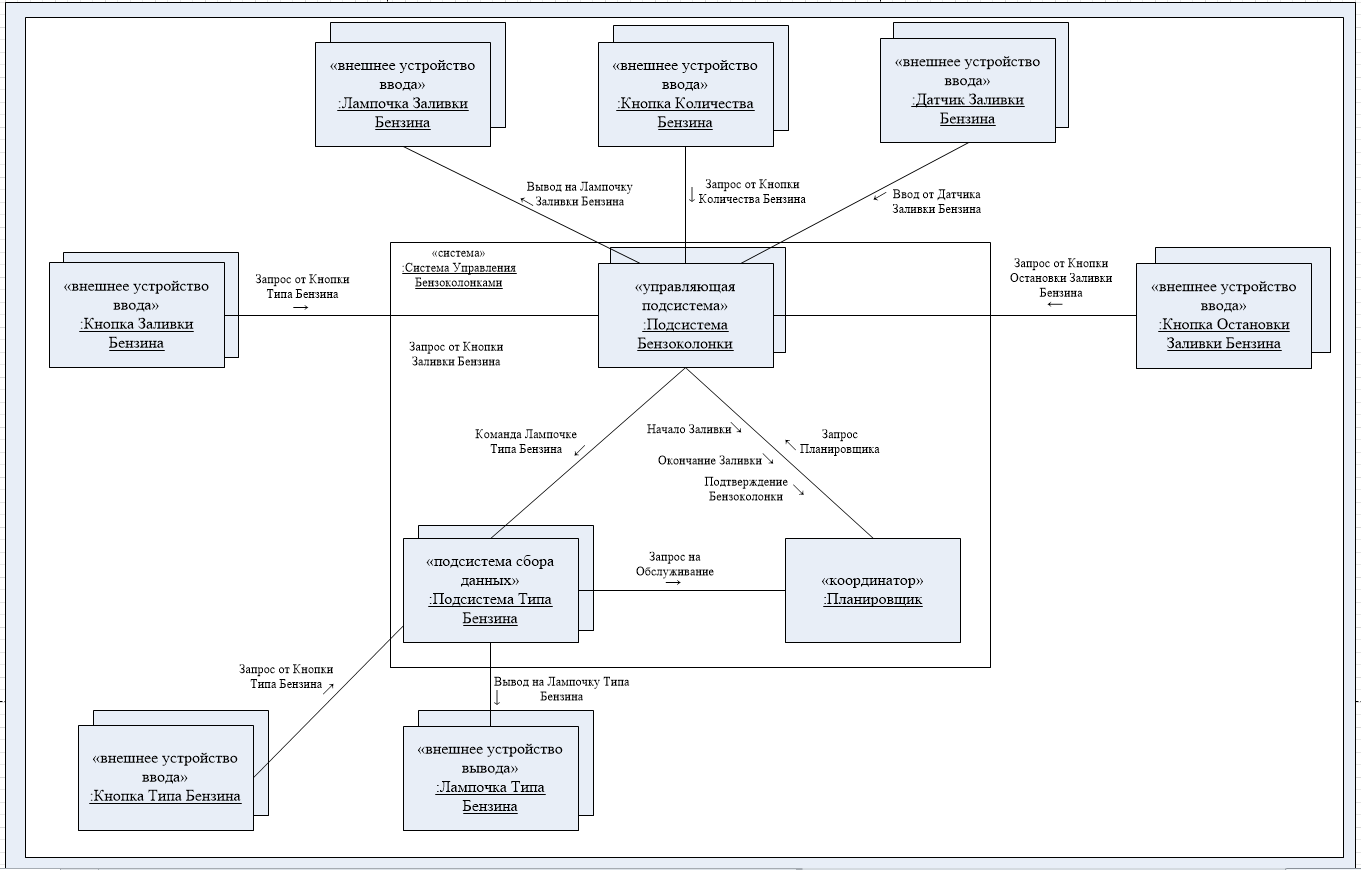


Рис.15. Разбиение на подсистемы

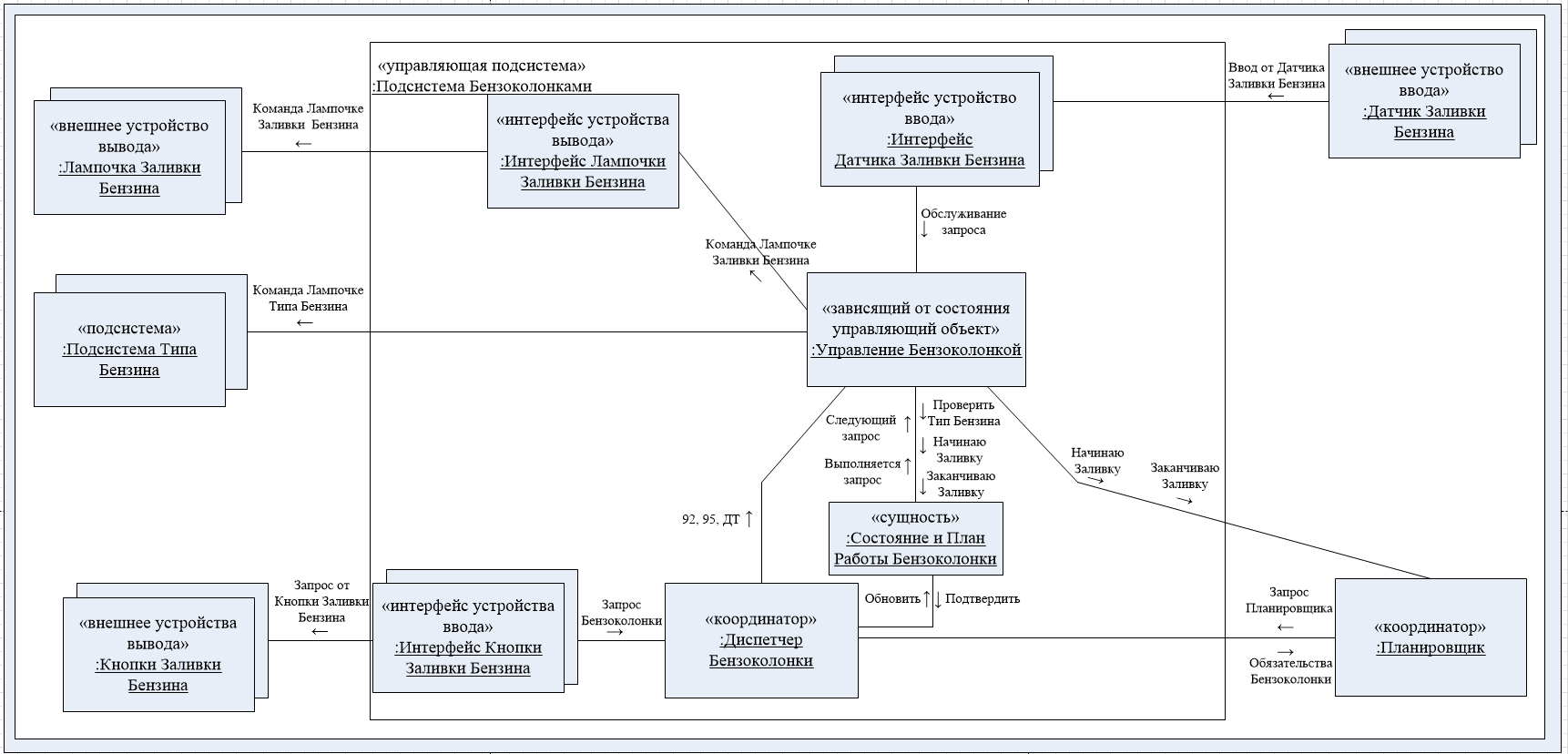


Рис.16. Структура подсистемы бензоколонки

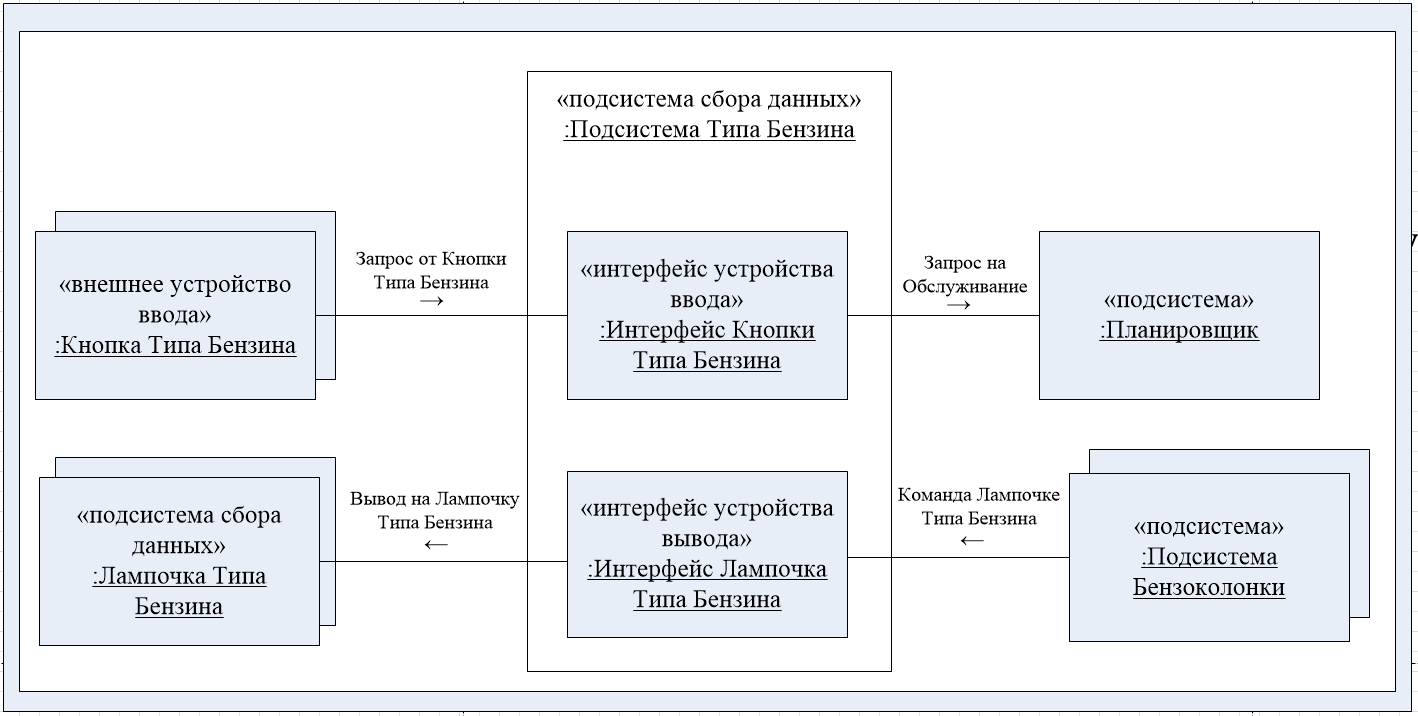


Рис.17. Структура подсистемы типа бензина

9. Разбиение системы на задачи

Далее рассматривается разбиение на задачи. Для этого необходимо проанали­зировать все объекты на диаграммах кооперации и применить критерии выделе­ния задач. Мы сделаем это для каждой диаграммы по очереди.

В распределенной системе управления бензоколонками имеется по одному экземпляру Подсистемы Бензоколонки для каждой бензоколонки и по одному экземпляру Подсис­темы Типа Бензина для каждого Типа Бензина (см. рис.15). Но, если трактовать систему как нераспределенную, можно кое-что упростить. В данном случае система управле­ния бензоколонками отображается на один процессор или сильно связанную многопро­цессорную конфигурацию (с общей памятью).

Важным аспектом нераспределенного решения является то, что сущностный объект Состояние и План Работы Бензоколонки доступен всем бензоколонкам, равно как и Планировщику, то есть его можно расположить в централизованном хранили­ще данных. Для слабо связанной распределенной системы, в которой нет разделя­емой памяти, этот подход не годится. Распределенное решение опишем в раз­деле 10.

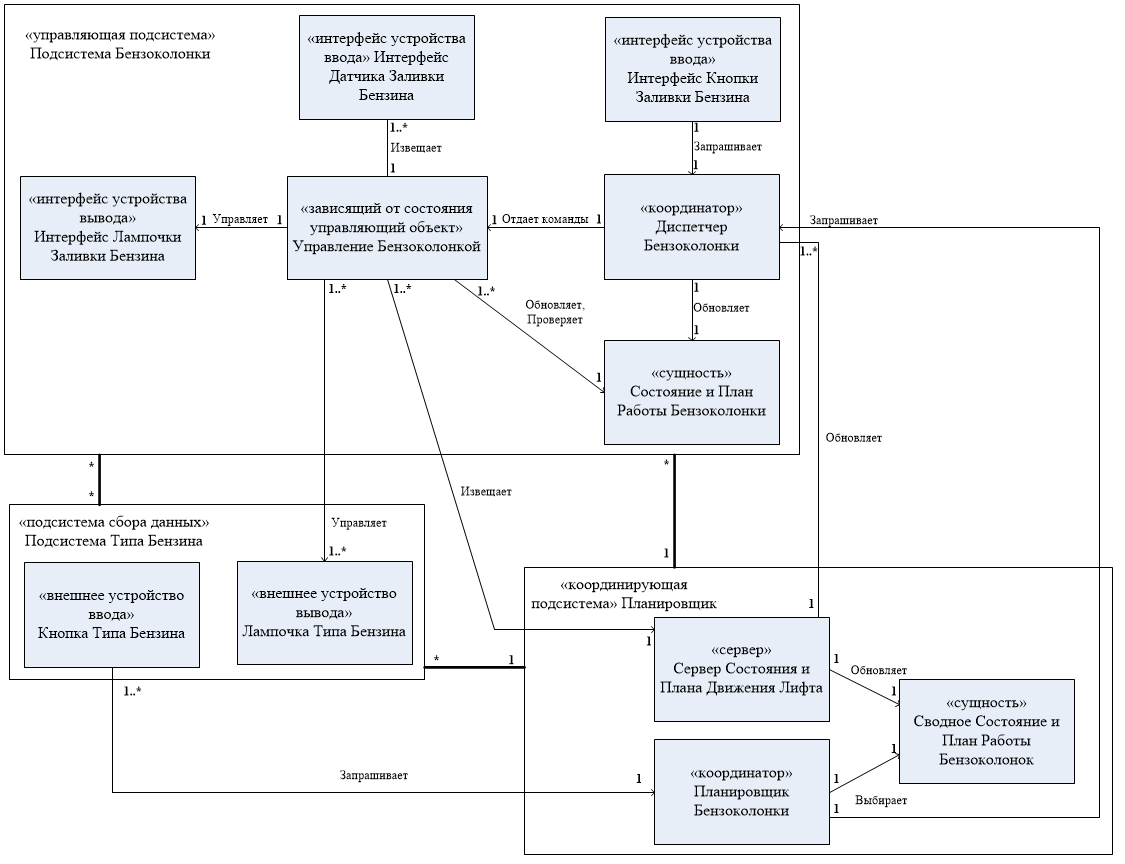


Рис.18. Уточненная статическая модель системы управления бензоколонками

9.1. Выделение задач в подсистеме бензоколонки

Рассмотрим архитектуру задач Подсистемы Бензоколонки для нераспределенного случая.

На диаграмме кооперации Выбор Количества Бензина (см. рис.5) нужно обратить внимание на объект интерфейса устройства, который получает входную информацию от актера, и затем проследить цепочку взаимодействий. Объект Интерфейс Кнопки Заливки Бензина выделяется в самостоятельную задачу Интерфейс Кно­пок Заливки Бензина с помощью критерия асинхронного интерфейса устройства ввода. Применив инверсию задач, мы проектируем одну задачу, которая будет отвечать за все кнопки бензоколонки, вместо того чтобы иметь по одной задаче на каждую кнопку. Задача Интерфейс Кнопок Бензоколонки активизируется прерыванием, возникающим при нажатии любой кнопки. Затем она считывает входную информацию от кноп­ки и посылает запрос Диспетчеру Бензоколонки, а сама тем временем готовится к при­ходу следующего прерывания. Диспетчер Бензоколонки, который является объектом-координатором, получает сообщения от Интерфейса Кнопок Бензоколонки в данном прецеденте и от Планировщика в прецеденте Количество Бензина (см. рис.6). Он выделяется в координирующую задачу, активизируемую приходом сообщения Запрос Планировщика или Запрос Бензоколонки. Состояние и План Работы Бензоколонки — это пассивный объект абстрагирования данных, у которого нет своего потока управления.

Рассмотрим далее прецедент Заливки Бензина на рис.7. Объект Интерфейс Датчика Заливки Бензина становится асинхронной задачей интерфейса устройства ввода - Интерфейс Датчиков Заливки Бензина - по тем же причинам, что и в случае с задачей Интерфейс Кнопок Бензоколонки. Он посылает Тип Бензина объек­ту Управление Бензоколонкой, о котором речь пойдет ниже.

Разберем диаграмму состояний Управление Бензоколонкой на рис.13. Здесь есть зависящий от состояния управляющий объект, который исполняет диаграмму состояний. На этапе анализа все аспекты физического объекта «бензоколонка», связанные с управлением, были отображены на управляющий объект Управление Бензоколонкой (см. рис.7 и 9). Если имеется несколько бензоколонок, то управляются они неза­висимо, так что в модели присутствует по одному такому объекту для каждой бензоколонки. В ходе разбиения на задачи каждый объект Управление Бензоколонкой отобра­жается на отдельную задачу Контроллер Бензоколонки. Любая задача исполняет диа­грамму состояний для своей бензоколонки, как показано на рис.13.

Задача Контроллер Бензоколонки взаимодействует с несколькими объектами интер­фейса устройств вывода, которые работают непосредственно с внешней средой, в частности и лампочками бензоколонки. Все эти устройства пассив­ны (то есть не генерируют прерываний), следовательно, асинхронные задачи не нужны. Каждый запрос на вывод информации выполняется по требованию, по­этому не нужны и периодические задачи. Кроме того, вызывающая задача обяза­на дожидаться завершения вывода, поэтому нет необходимости и в пассивной задаче вывода. Таким образом, объект вывода на устройство не стоит выделять в самостоятельную задачу, он объединяется с задачей Контроллер Бензоколонки в соот­ветствии с критерием группировки задач.

Рассмотрим выполнение задачи Контроллер Бензоколонки подробнее. Она полу­чает сообщение Выполняется Запрос, находясь в состоянии Бензоколонка работает (см. рис.8). Затем Контроллер Бензоколонки посылает сообщение объекту Состоя­ние и План Работы Бензоколонки, требуя Проверить Этот Запрос. Состояние изменя­ется только в том случае, если в соответствии с планом бензоколонка должна выполнять этот запрос.

Диспетчер Бензоколонки (всего один экземпляр в нераспределенном решении) ре­ализуется асинхронно по отношению к задаче Контроллер Бензоколонки, так как запрос к бензоколонки может прийти в любой момент. Поэтому Диспетчер Бензоколонки вычленяет­ся в отдельную задачу-координатор, которая активизируется асинхронно по со­бытию Запрос Бензоколонки или Запрос Планировщика и обычно исполняется неза­висимо от задачи Контроллер Бензоколонки. Единственное взаимодействие между двумя задачами возникает, когда бензоколонка находится в состоянии Не Заливает и поступа­ет новый запрос. В этом случае Диспетчер Бензоколонки активизирует задачу Кон­троллер Бензоколонки.

Итак, в нераспределенной системе управления бензоколонками Подсистема Бензоколонки разбивается на четыре задачи: Интерфейс Кнопок Бензоколонки, Интерфейс Датчи­ков Заливки Бензина, Диспетчер Бензоколонки и Контроллер Бензоколонки. Имеется по одному экземпляру первых трех задач и столько экземпляров четвертой задачи, сколько есть бензоколонок. Все экземпляры задачи Контроллер Бензоколонки идентичны и исполня­ют собственную копию диаграммы состояний Управление Бензоколонки. Предвари­тельная архитектура задач изображена на начальной диаграмме параллельной кооперации (рис.19).

9.2. Выделение задач в подсистеме этажа

Перейдем к архитектуре задач в Подсистеме Типа Бензина, изображенной на рис.17. Объект Интерфейс Кнопки Типа Бензина выделяется в самостоятельную задачу Интерфейс Кнопок Типа Бензина на основе критерия асинхронного интерфейса устройства ввода и критерия группировки задач. В случае нераспределенного ре­шения задача Интерфейс Кнопок Типа Бензина отвечает за обработку ввода от всех кно­пок этажа. Она активизируется прерыванием, обрабатывает его и посылает За­прос на Обслуживание объекту Планировщик, после чего готова к приходу следующего прерывания.

Лампочки типа бензина — это пассивное устройство вывода. С ним взаимодействует объект Интерфейс Лампочки Типа Бензина. Бывает так, что различные экземпляры задачи Контроллер Бензоколонки одновременно посылает запросы лампочке. В таком случае необходимо иметь для каждого устройства задачу-монитор ресурса, которая сможет сериализовать обработку параллельных запросов. Итак, выявилась потребность в задачах Монитор Лампочки Типа Бензина. Все объекты Интерфейс Лампочки Типа Бензина отображается на задачу Монитор Лампочки Типа Бензина. Задачи, из которых состоит Подсистема Типа Бензина в нераспределенной системе управления бензокнолонкой, показаны на рис.19. Вмес­то этого допустимо завести единственный монитор ресурсов для лампочки типа бензина, однако первоначальное решение обладает большей гибкостью.

9.3. Выделение задач в подсистеме планировщика

В случае нераспределенного решения Планировщик — это подсистема, кото­рая состоит из одного объекта-координатора, выделенного в задачу. Задача Пла­нировщик активизируется по запросу, в частности после получения Запроса на Обслуживание, и выбирает бензоколонку, наиболее подходящую для его удовлетворения.

Поскольку речь идет о нераспределенном решении, очевидно, что Планиров­щик может напрямую читать Состояние и План Работа Бензоколонки. Следователь­но, различные экземпляры задачи Контроллер Бензоколонки не обязаны посылать Пла­нировщику сообщения Начал Заливку и Закончил Заливку. Архитектура задач для нераспределенного случая представлена на рис.19.

9.4. Определение интерфейсов задач

Рассмотрим теперь, как определяются интерфейсы задач. В случае интерфей­сов обмена сообщениями между параллельными задачами возможен либо слабо связанный, либо сильно связанный обмен. Необходимо исследовать только ин­терфейсы между объектами, выделенными в самостоятельные задачи. Кроме того, следует точно описать сообщения, включая их имена и параметры.

Взаимодействие между задачами Интерфейс Кнопок Бензоколонки и Диспетчер Бензоколонки, показанными на рис.19, отображается на слабо связанный обмен сооб­щениями (рис.20). Тем самым гарантируется, что исполнение задачи Интерфейс Кнопок Бензоколонки не будет приостановлено после отправки сообщения задаче Дис­петчер Бензоколонки. Планировщик посылает сообщения запрос Планировщика в ту же очередь. Поскольку Планировщик в момент передачи ему сообщения часто бы­вает занят, то интерфейс между задачами Планировщик и Интерфейс Кнопок Бензоколонки также отображается на слабо связанный обмен сообщениями (см. рис.20).

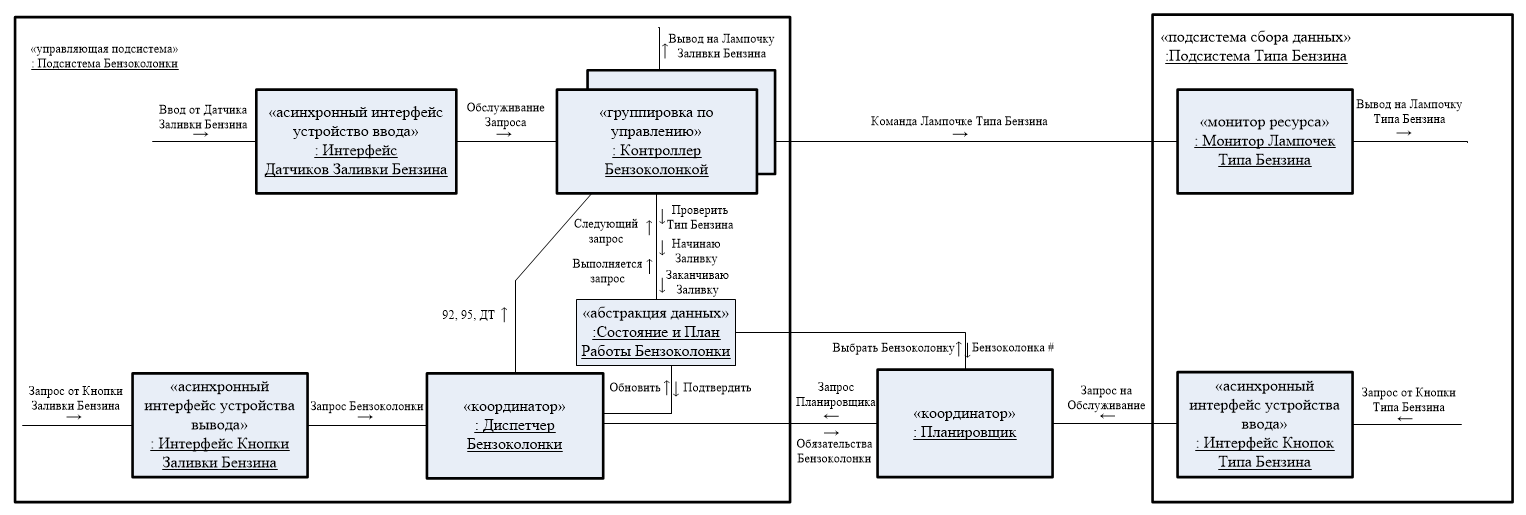


Рис.19. Нераспределенная система управления бензоколонками: архитектура задач

Интерфейс между задачами Диспетчер Бензоколонки и Контроллер Бензоколонки (см. рис.19) реализуется как сильно связанный обмен сообщениями (см. рис.20). Диспетчер Бензоколонки может посылать сообщения «92», «95» или «ДТ». Интерфейс является сильно связанным, поскольку Диспетчер Бензоколонки способен отправлять сообщения Контроллеру Бензоколонки только тогда, когда последний находится в состоянии Бензоколонка Не Заливает и, следовательно, должен быть активизирован.

Рассмотрим теперь интерфейс между Контроллером Бензоколонки и зада­чей-монитора ресурса: Монитор Лампочки Типа Бензина. Контроллер Бензоколонки дает команды управления лампочке типа бензина Монитору Лампочек типа бензина и команды управления лампочке типа бензина-Монитору Лампочке Типа Бензина (см. рис.19).

Проанализируем пассивные сущностные объекты, к которым обращается сра­зу несколько задач. Состояние и План Работы Бензоколонки - объект абстрагирова­ния данных, который инкапсулирует состояние и план движения бензоколонки. В нерас­пределенном варианте есть только один экземпляр этого объекта, так что можно использовать централизованное хранилище. К объекту осуществляют доступ не­сколько экземпляров задач Контроллер Бензоколонки, Диспетчер Бензоколонки, а также Пла­нировщик (см. рис.20). Доступ к пассивному объекту должен быть синхрони­зирован, чтобы его операции исполнялись взаимно исключающим образом.

9.5. Проектирование класса абстрагирования данных

При централизованном подходе есть только один класс абстрагирования дан­ных - Состояние и План Работы Бензоколонки. Состояние бензоколонки — это его текущее положение и тип бензина (92, 95, ДТ). Планработы представляет собой список запросов, которые бензоколонка должна выполнить. По­скольку есть только один экземпляр указанного класса, мы вправе прибегнуть к централизованному хранилищу, как показано на рис.21а.

Чтобы определить операции класса абстрагирования данных, необходимо по­нять, как к нему обращаются. На рис.19 показаны три разные задачи, обращающиеся к такому объекту: Планировщик, Диспетчер Бензоколонки и Контроллер Бензоколонки (последняя существует в нескольких экземплярах - по одному для каждой бензоколонки). Планировщик читает план и состояние каждого бензоколонки с целью выбрать бензоколонку, которой будет поручено обслуживание нового запроса. Эта функция вы­полняется внутри операции выбрать Бензоколонку. Диспетчер Бензоколонки обновляет план движения бензоколонки и проверяет, не находится ли бензоколонка в состоянии покоя. Эта функ­ция реализуется операцией обновить План. Задача Контроллер Бензоколонки обраща­ется к объекту Состояние и План Работы Бензоколонки тремя способами (путем отправки трех разных сообщений): чтобы обновить состояние, когда Бензоколонка начала заливку или закончила, чтобы проверить Тип Бензина, и чтобы проверить Следующий Тип Бензина. Каждое из названных сообщений отображается на опера­цию объекта абстрагирования данных (см. рис.21).

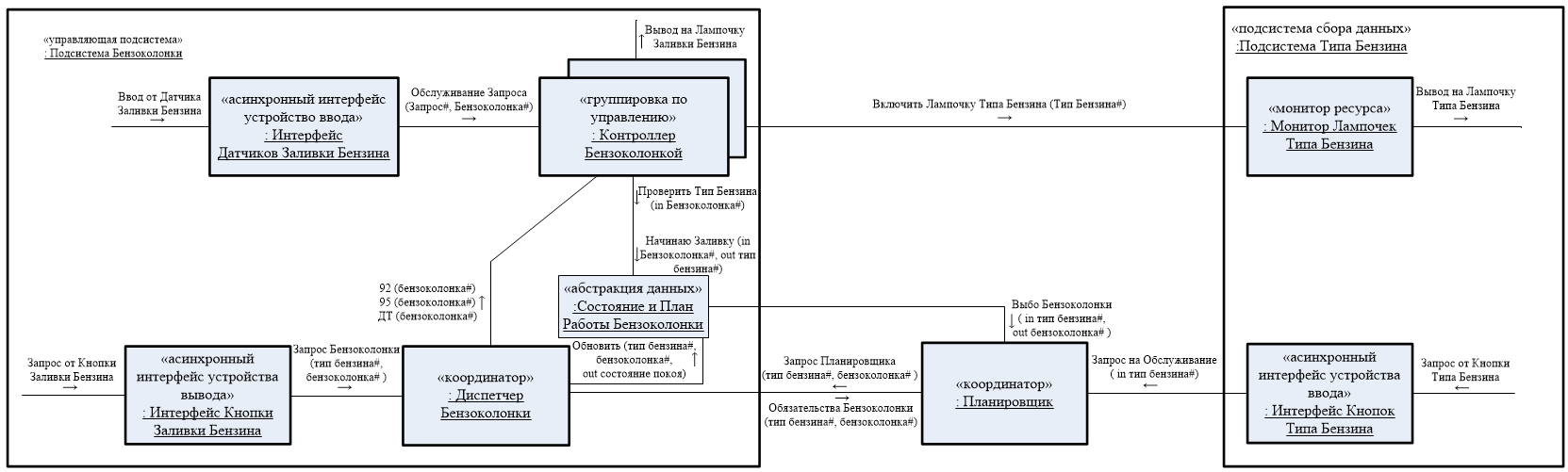


Рис.20. Нераспределенная система управления бензоколонками: интерфейсы задач

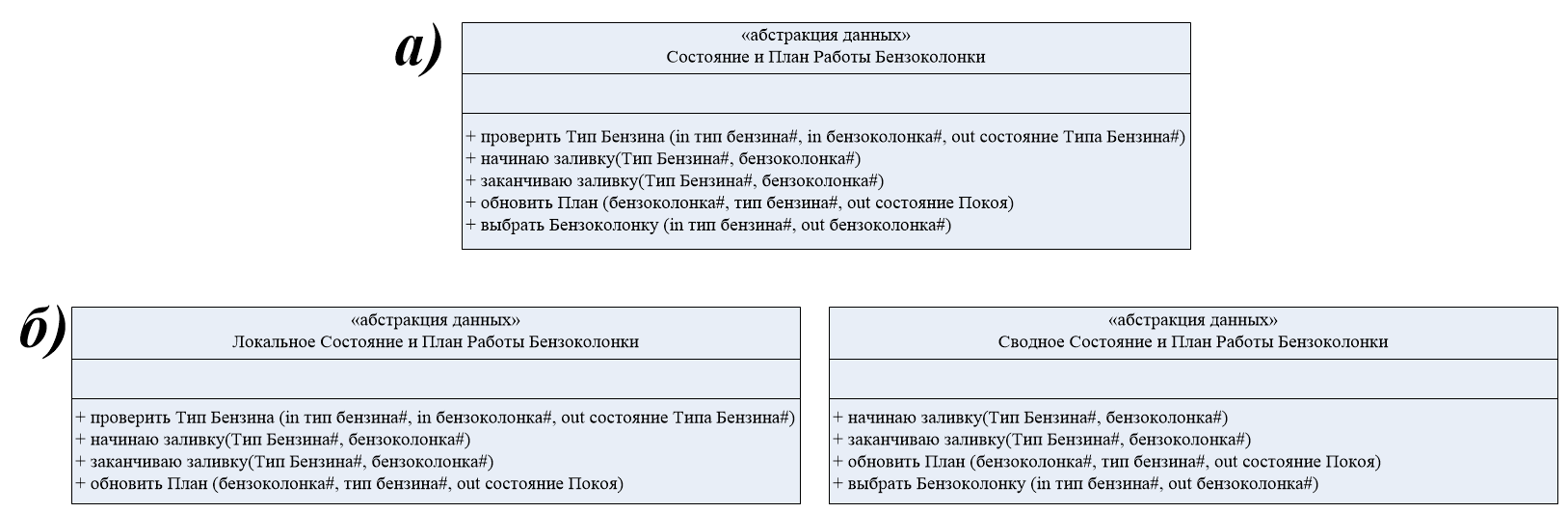


Рис.21. Классы абстрагирования данных: а -для централизованного решения; б - для распределенного решения

Операция проверить Тип Бензина вызывается с параметрами Тип Бензина# и Бензоколонка#, проверяет, должна ли Бензоколонка выбрать этот Тип Бензина, и обновляет состояние и план работы бензоколонки.

Операция проверить Следующий Тип Бензина (вызываемая после этого) проверяет, какой тип бензина будет дальше. Она уста­навливает состояние бензоколонки равным «92», «95», «ДТ» или «покой» (если невыпол­ненных запросов нет) и возвращает направление, которое будет равно соответ­ственно запрос 92, запрос 95, запрос ДТ или нет Запросов.

9.6. Обсуждение альтернативных архитектур

Вместо только что описанной архитектуры мы могли бы ввести для каждой Бензоколонки по одному экземпляру задач Интерфейс Кнопок Бензоколонки, Интерфейс Дат­чика Заливки Бензина и Диспетчер Бензоколонки в дополнение к задаче Контроллер Бензоколонки. При наличии одного процессора такое решение нежелательно из-за слишком больших накладных расходов.

Но в многопроцессорной среде для каждой Бензоколонки допустимо иметь свой про­цессор, на котором будут выполняться экземпляры задач Интерфейс Кнопок Бензоколонки, Интерфейс Датчика Заливки Бензина, Диспетчер Бензоколонки и Контроллер Бензоколонки. Задачи Планировщик, Интерфейс Кнопок Типа Бензина, Монитор Лампочки Типа удобно разместить на отдельном процес­соре. В случае нескольких ЦП с общей памятью объект абстрагирования данных Состояние и План Работы Бензоколонки по-прежнему хранился бы в разделяемой памяти.

Если же процессоры не располагают общей памятью, то несколько задач не смогут напрямую обращаться к объекту Состояние и План Работы Бензоколонки. Далее мы опишем проект распределенной системы управления бензоколонкой.

10. Проект распределенной системы управления бензоколонки

Физическая конфигурация распределенной системы управления бензоколонками состоит из нескольких узлов, соединенных локальной сетью. В такой среде необходимо установить дисциплину, согласно которой все коммуникации между рас­пределенными подсистемами происходят путем обмена сообщениями. Общая ар­хитектура распределенного ПО показана на рис. 22, где представлено несколько экземпляров Подсистемы Бензоколонки (по одной на каждую бензоколонку), несколько экземпляров Подсистемы Типа Бензина (по одной на тип бензина) и один экземпляр подсистемы Планировщик. Все коммуникации между подсистемами осуществляются посред­ством слабо связанного обмена сообщениями.

В распределенной конфигурации нет разделяемой памяти, а значит, Планировщик и экземпляры Подсистемы Бензоколонки не могут напрямую обратиться к объ­екту абстрагирования данных Состояние и План Работы Бензоколонки, как было в предыдущем случае. Один из способов решить проблему - погрузить такой объект в серверную задачу. Вместо того чтобы вызывать операцию объекта абстрагирова­ния данных, клиентская задача отправит синхронное сообщение с ответом задаче Сервер Состояния и Плана Работы Бензоколонки. Но при этом сервер может стать узким местом, поскольку у него есть довольно много клиентов: Планировщик, а также несколько экземпляров Диспетчера Бензоколонки и Контроллера Бензоколонки.

Альтернативное решение - применение репликации данных. Каждый экземпляр Подсистемы Бензоколонки хранит собственный локальный объект Локальное Состояние и План Работы Бензоколонки. Есть такой объект и у Планировщика, толь­ко в нем содержатся состояния и планы работы всех бензоколонок и называется он Сводное Состояние и План Работы Бензоколонки.

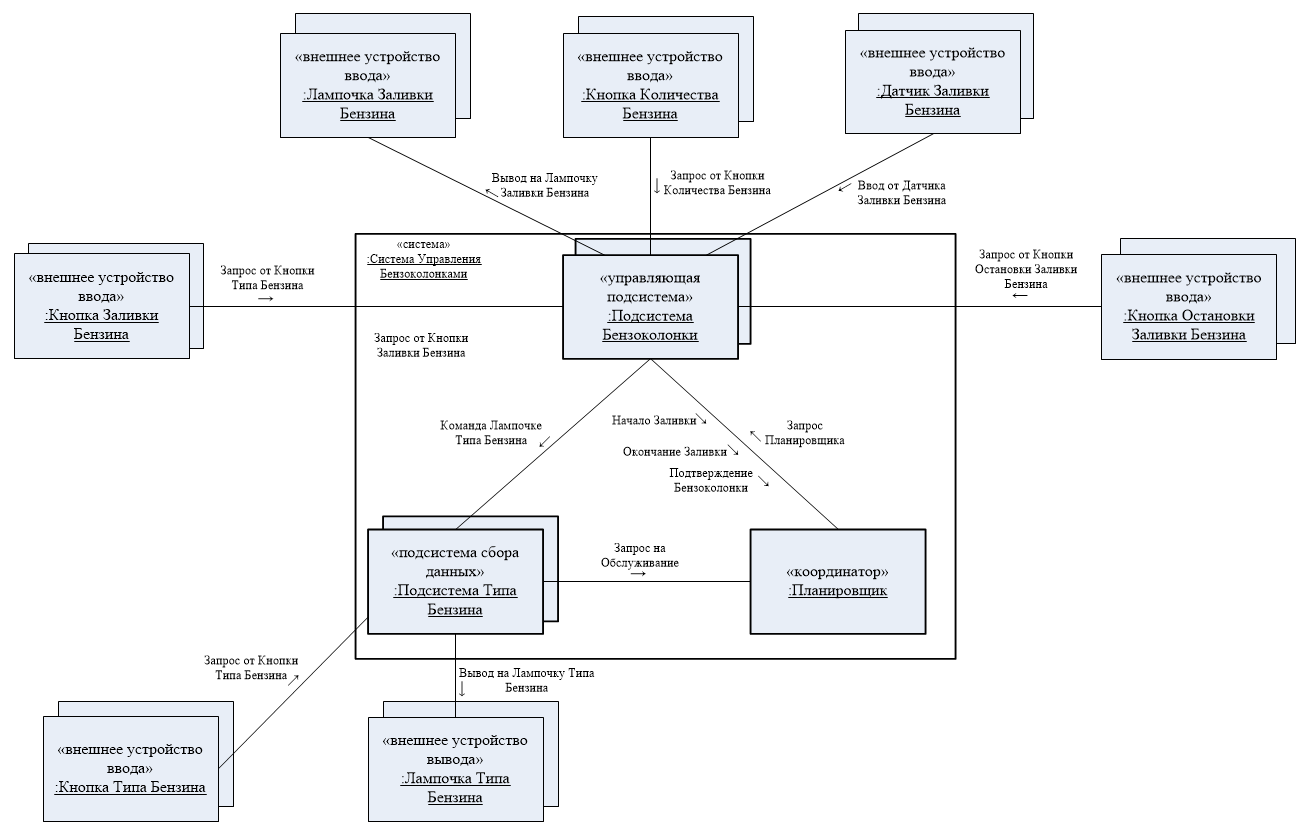


Рис.22. Архитектура распределенного ПО

10.1. Структура подсистемы бензоколонки

В распределенном проекте для каждой бензоколонки имеется по одному экземпляру Подсистемы Бензоколонки, включающему по одному экземпляру задач Контроллер Бензоколонки, Интерфейс Кнопок Бензоколонки, Интерфейс Датчиков Заливки Бензина и Диспет­чер Бензоколонки. Кроме того, каждый экземпляр Подсистемы Бензоколонки хранит собствен­ный локальный экземпляр объекта абстрагирования данных Состояние и План Работы Бензоколонки.

Архитектура задач для Подсистемы Бензоколонки показана на рис.23. Задача Диспетчер Бензоколонки получает сообщения Запрос Бензоколонки от задачи Интерфейс Кнопок Бензоколонки, сообщения Запрос Планировщика от Планировщика и посыла­ет сообщения Обязательство Бензоколонки задаче Планировщик. Контроллер Бензоколонки отправляет сообщения о состоянии и Подсистеме Бензоколонки, и Планировщику.

Пересмотренная архитектура задач показана на рис.24. Здесь изображены интерфейсы задач, а также способ доступа к объектам, скрывающим информацию. Проект объекта абстрагирования данных Локальное Состояние и План Работы Бензоколонки проще, чем для централизованного решения (см. рис.21а), по­скольку он поддерживает только одну бензоколонку и, следовательно, операция выбрать Бензоколонку не нужна.

На этапе конфигурирования целевой системы (см. раздел 13) каждый эк­земпляр Подсистемы Бензоколонки обычно отображается на узел физической бензоколонки, поэтому программы в каждом узле исполняются независимо от всех остальных.

10.2. Структура подсистемы типа бензина

В описанном выше распределенном решении существует по одному экзем­пляру Подсистемы Типа Бензина для каждого типа бензина. В каждом таком экземпляре есть по одному экземпляру задач Интерфейс Кнопок Типа Бензина, Монитор Лампочки Типа Бензина, то есть для каждого устройства вво­да/вывода имеется отдельная задача.

Архитектура задач для Подсистемы Типа Бензина показана на рис.25. Она ана­логична архитектуре для нераспределенного решения (см. рис.19) с тем отли­чием, что экземпляров подсистем существует несколько.

Задача Интерфейс Кнопок Типа Бензина посылает Запросы на Обслуживание, содер­жащие информацию о вызове на данный тип бензина, Планировщику. И Монитор Лампо­чки Типа Бензина, получает сообщения о состоянии в форме команд лампочки от нескольких экземпляров задачи Контроллер Бензоколонки. Интерфейсы задач для пересмотренной архитектуры изображены на рис.26.

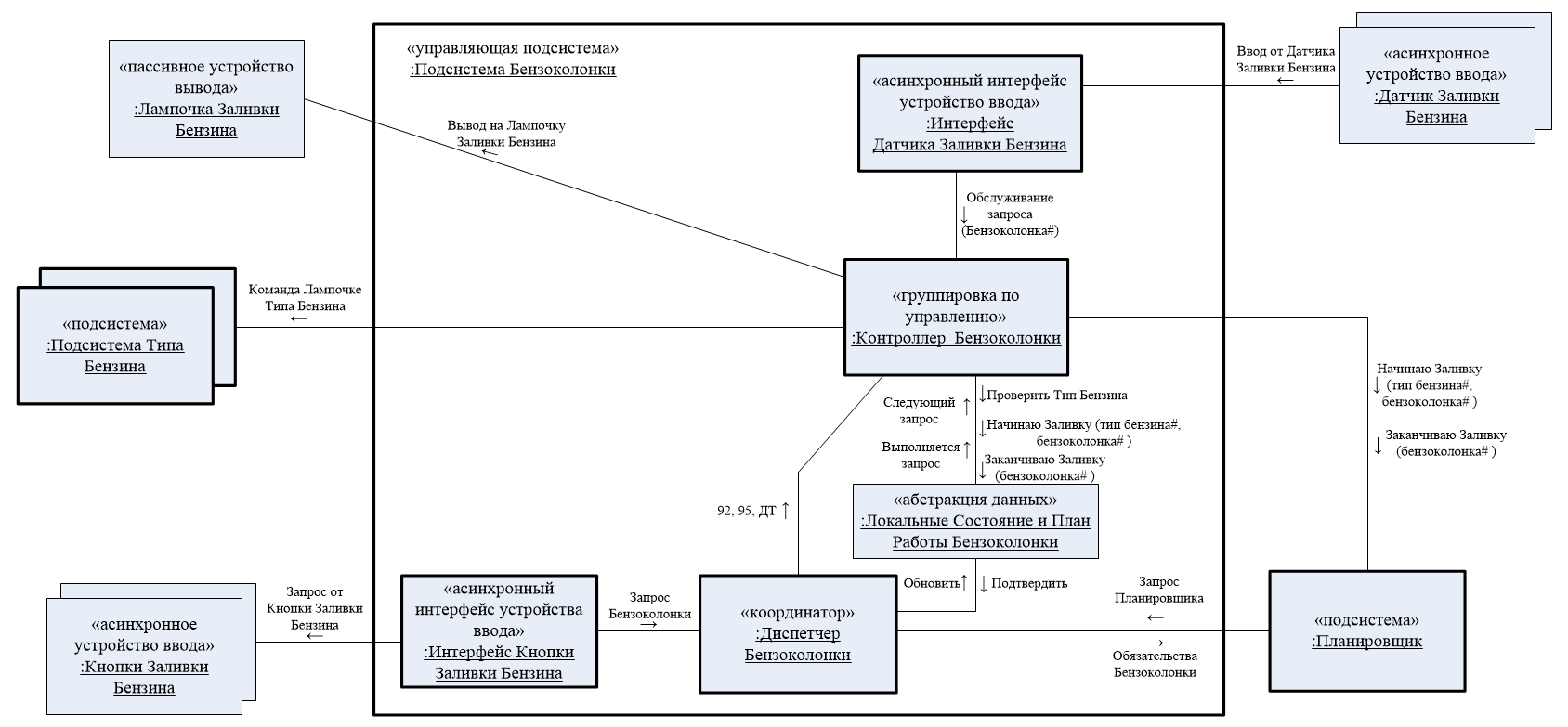


Рис.23. Архитектура задач для Подсистемы Бензоколонки

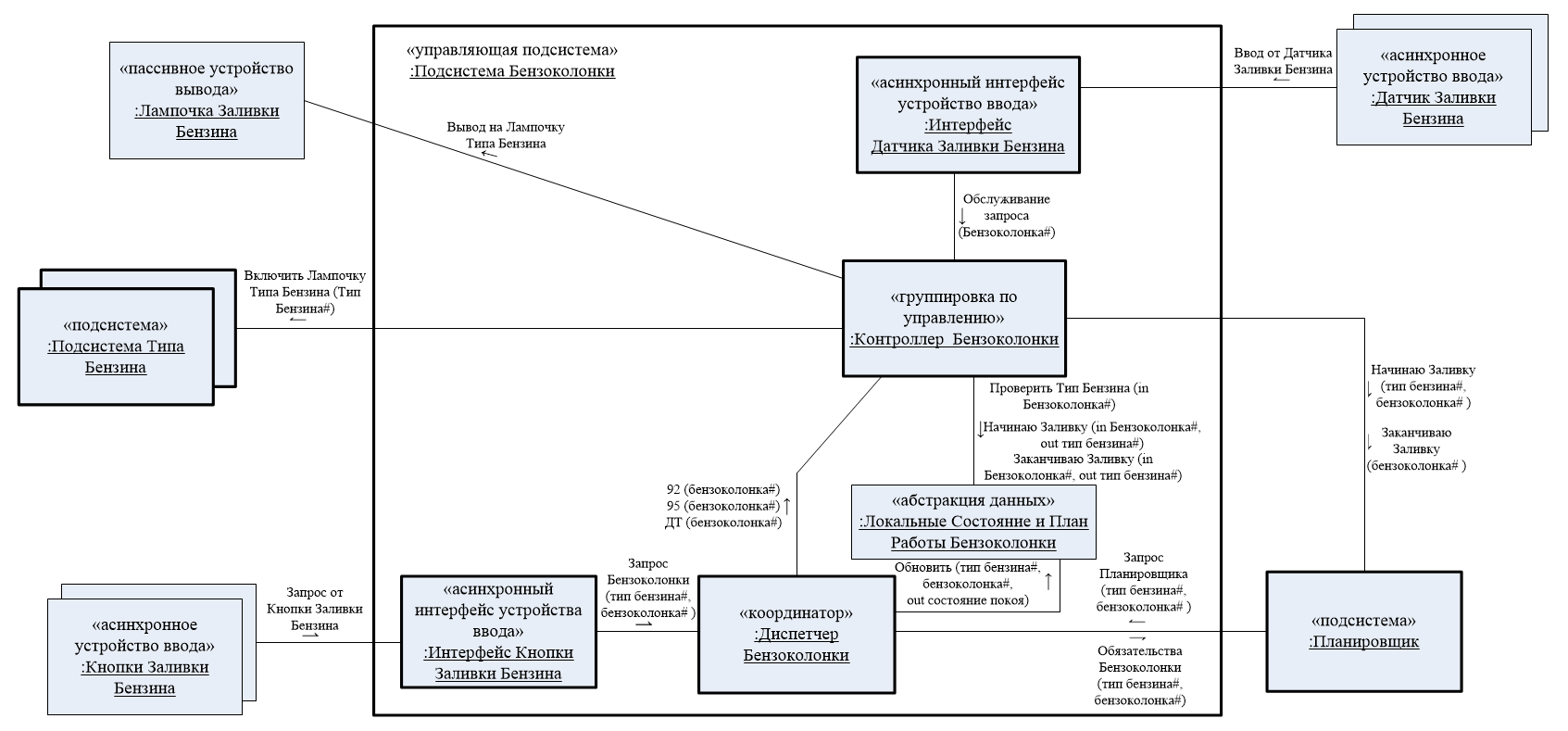


Рис.24. Архитектура задач для подсистемы Бензоколонки: интерфейсы задач

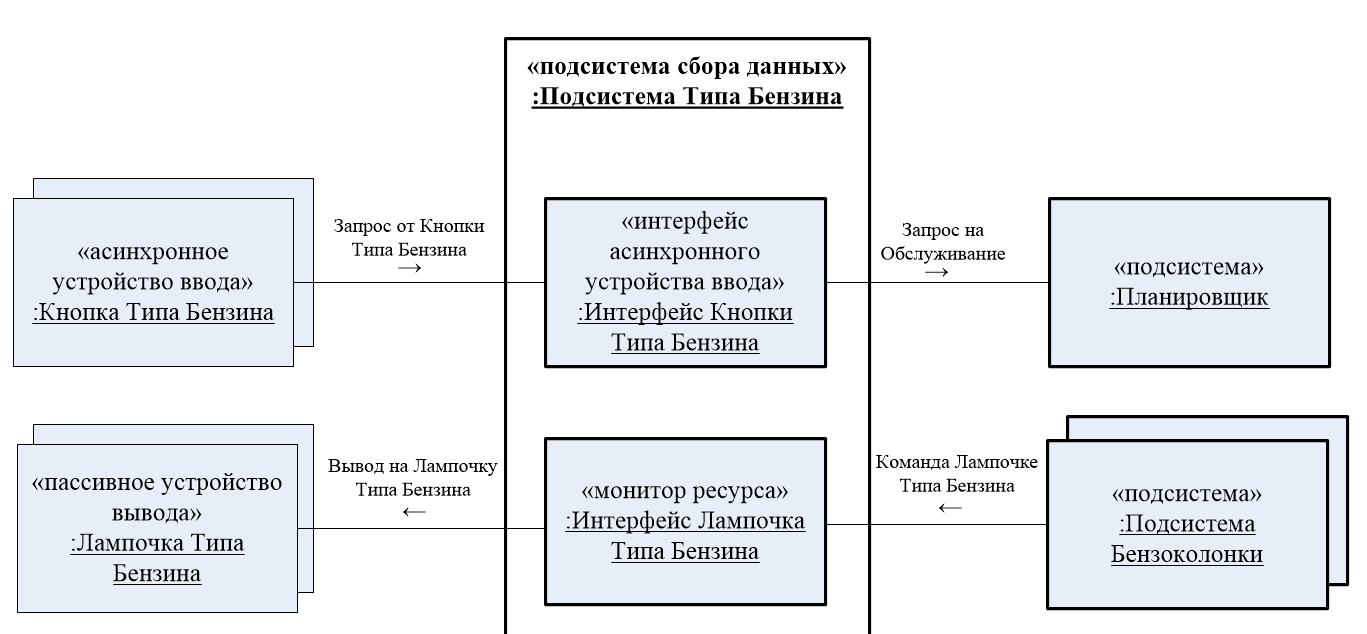


Рис.25. Архитектура задач для Подсистемы Бензоколонки

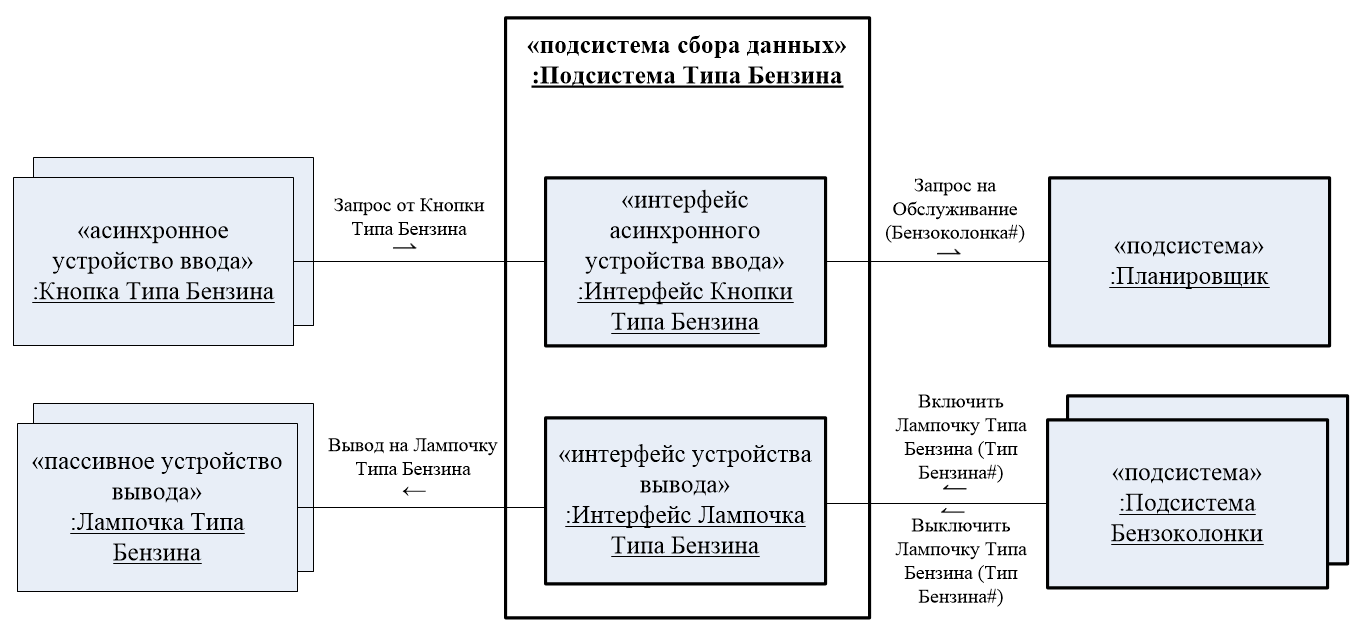


Рис.26. Архитектура задач для Подсистемы Типа Бензина: интерфейсы задач

10.3. Структура подсистемы планировщика

Существует единственный экземпляр подсистемы Планировщик, которая со­стоит из двух задач и одного скрывающего информацию объекта. Это объект аб­страгирования данных Сводное Состояние и План Работы Бензоколонки, который содержит текущее состояние каждой бензоколонки и план ее работы, описывающий, какие типы бензинов бензоколонка должна залить (см. рис.216).

В любом узле доступ к объекту Локальное Состояние и План Работы Бензоколонки осуществляется со стороны задач Контроллер Бензоколонки и Диспетчер Бензоколонки. Но, чтобы Планировщик знал состояния и планы движения всех бензоколонок, каждый Контроллер Бензоколонки посылает ему сообщения о состоянии, извещающие о начале заливки бензина бензоколонки и окончание заливки. Кроме того, Диспетчер Бензоколонки пере­дает Планировщику два вида сообщений об обязательствах бензоколонки с уведомлени­ем о том типе, который бензоколонка планирует залить:

– уведомления о том, что клиент нажал кнопку типа бензина; Планировщик та­ким образом информируется о типе бензина;

– подтверждения, то есть ответы Подсистемы Бензоколонки на Запросы Планиров­щика, требующие, чтобы бензоколонки начала работу с целью заливки выбранного типа бензина.

Подсистема Планировщика разбита на две задачи: Сервер Состояния и Плана Работы Бензоколонки (серверная задача) и Планировщик Бензоколонки (коорди­нирующая задача). Первая принимает сообщения о состоянии и обязательствах бензоколонки и обновляет объект Сводное Состояние и План Работы Бензоколонки, а вто­рая принимает Запросы на Обслуживание от нескольких экземпляров задачи Интерфейс Кнопок Типа Бензина. Каждый раз при получении сообщения Запрос на Обслуживание задача Планировщик Бензоколонки проверяет, собирается ли какая-ни­будь бензоколонка залить указанный тип бензина. Если нет, она выбирает бензоколонку и посылает задаче Диспетчер Бензоколонки, соответствующей этой бензоколонке, сообщение Запрос Планировщика. Архитектура задач для Подсистемы Планировщика показана на рис.27. Интерфейсы задач для пересмотренной архитектуры изображены на рис.28.

Объект абстрагирования данных Сводное Состояние и План Работы Бензоколонки предоставляет операции заливка началась, заливка закончилась, обновить План и выбрать Бензоколонку (см. 28). Задача Сервер Состояния и План Работы Бензоколонки вызывает операции заливка началась или заливка закончилась, когда получает сообщение о состоянии. При получении сообщения об обязательстве бензоколонки она вызывает операцию об­новить План.

10.4. Интерфейсы подсистем

После определения интерфейсов задач во всех трех подсистемах можно обно­вить архитектуру распределенной программы с целью показа интерфейсов под­систем (см. рис.29). На самом деле все интерфейсы принадлежат той или иной из ранее рассмотренных задач.

11. Проектирование скрывающих информацию классов

Сами классы были определены на этапе разбиения на объекты, теперь речь пойдет о проектировании их операций. Классы абстрагирования данных описа­ны выше; в этом разделе мы разработаем остальные классы, скрывающие инфор­мацию.

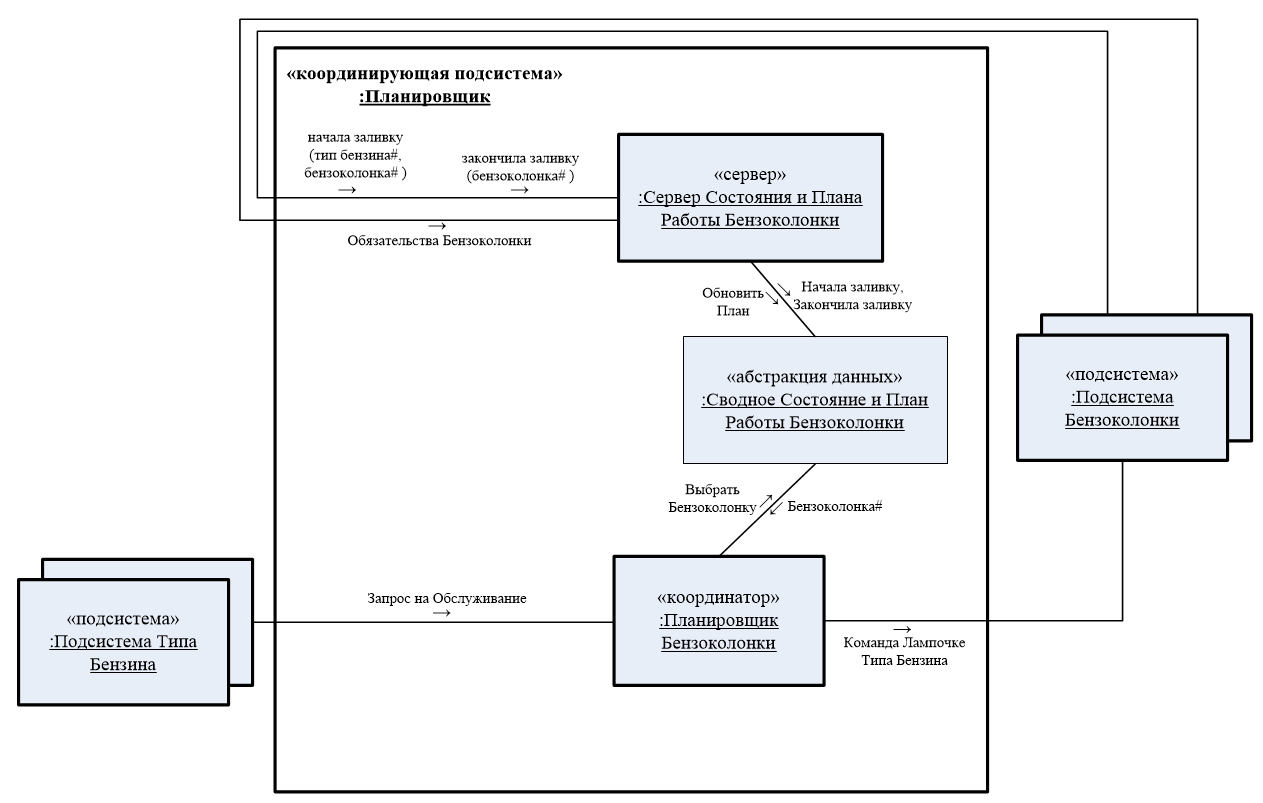


Рис.27. Архитектура задач для Подсистемы Планировщика

11.1. Проектирование классов интерфейса устройств

Класс интерфейса устройства скрывает истинный интерфейс с физическим устройством, предлагая вместо него виртуальный интерфейс. Для каждого типа устройств ввода/вывода имеется отдельный интерфейсный класс. Состав опера­ций, поддерживаемых таким классом, зависит от функций, которые должен под­держивать объект интерфейса. Классы интерфейса устройств изображены на рис.30 и описаны ниже:

–Интерфейс Кнопки Заливки Бензина. Предоставляет две операции: читать (считы­вает значение датчика кнопки типа бензина) и инициализировать;

–Интерфейс Датчика Заливки Бензина. Предоставляет две операции: читать (считывает значение датчика заливки бензина) и инициализировать;

–Интерфейс Лампочки Типа Бензина. Предоставляет операцию выключить для вы­ключения лампочки типа бензина. Предполагается, что лампочка включается аппаратно, так что программная реализация операции включить не нужна. Опе­рация выключить соответствует сообщению Выключить Лампочку Типа Бензина, которое получает объект Интерфейс Лампочки Типа Бензина (см. рис.14);

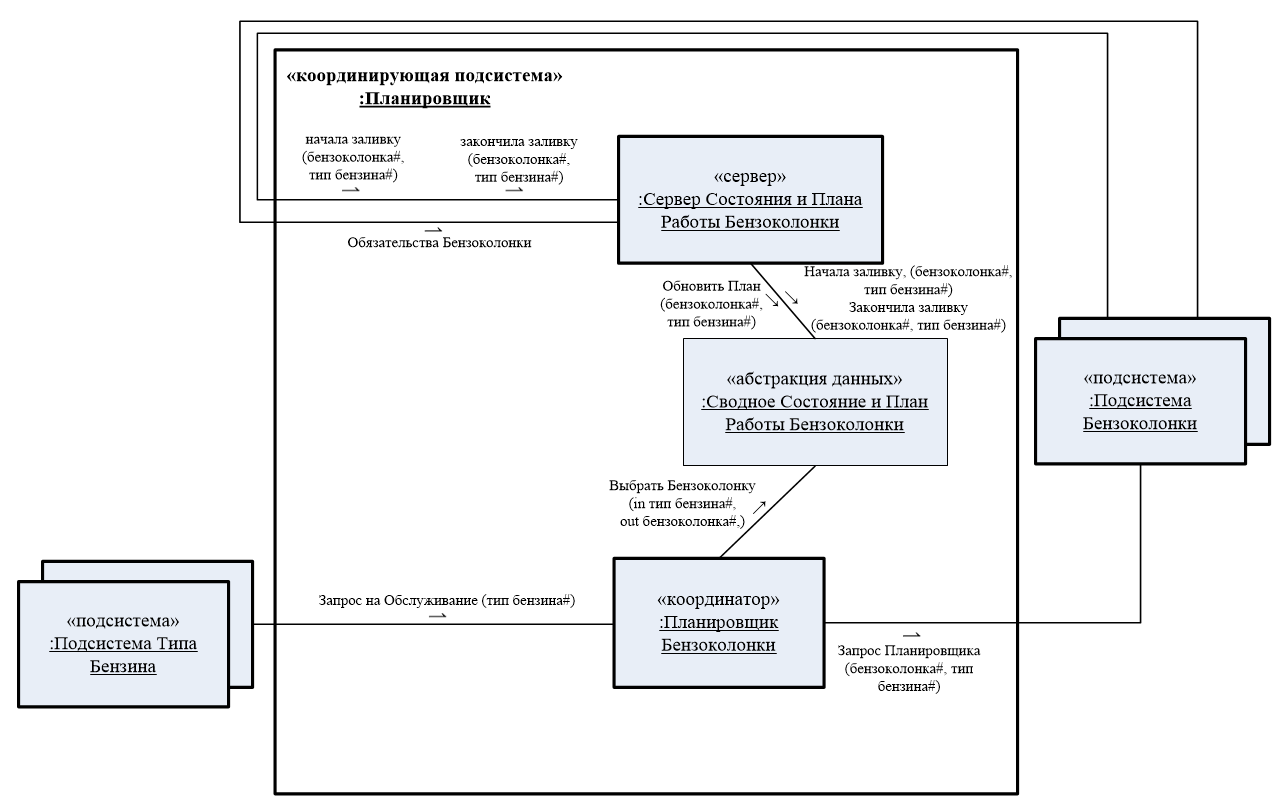


Рис.28. Архитектура задач для Подсистемы Планировщика: интерфейсы задач

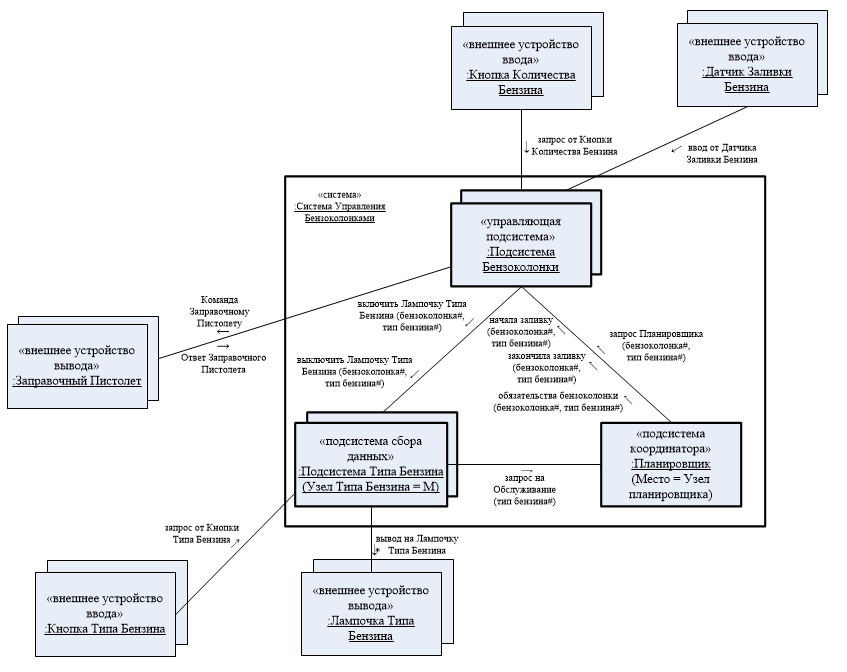


Рис.29. Архитектура распределенной программы: интерфейсы подсистем

11.2. Проектирование класса, зависящего от состояния

В системе есть только один зависящий от состояния класс Управление Бензоколонкой, который инкапсулирует диаграмму состояний, реализованную в виде табли­цы переходов, – см. рис.13. Он поддерживает две операции: обработать Событие и текущее Состояние. Этот объект вложен в задачу Контроллер Бензоколонки. Поскольку задача существует в нескольких экземплярах, то и экземпляров класса будет несколько – по одному для каждой бензоколонки. Спецификация класса Управ­ление Бензоколонкой представлена на рис.31.

12. Разработка детального проекта программы

Итак, задачи и скрывающие информацию классы определены. Далее нужно выполнить детальное проектирование. Сюда входит проектирование объектов-разъемов и составных задач, в которые вкладываются скрывающие информацию объекты.

12.1. Проектирование объектов-разъемов для бензоколонки

Изображенные на рис.24 интерфейсы задач в распределенной Подсисте­ме Бензоколонки имеют концептуальный характер: они показывают тип обмена сообщениями - слабо или сильно связанный. На этапе детального проектирования ин­терфейсы описываются более точно с помощью объектов-разъемов, как показано на рис.32. Так, объекту Контроллер Бензоколонки поступают сильно связанные со­общения, не требующие ответа. Данный интерфейс отображается на объект-разъем буфер Сообщений Контроллера Бензоколонки. Обычно такого рода разъем су­ществует между одним производителем и одним потребителем, однако в нашем случае есть две задачи-производителя –Интерфейс Датчиков Заливки Бензина и Дис­петчер Бензоколонки, которые отправляют сообщения Контроллеру Бензоколонки (при этом последний находится в различных состояниях). Задача Интерфейс Датчиков Заливки Бензина посылает сообщение началась заливка, когда Контроллер Бензоколонки находится в состоянии Бензоколонка заливает, а задача Диспетчер Бензоколонки передает сооб­щение следующее Тип Бензина, когда контроллер находится в состоянии Бензоколонки не заливает. Сообщения Контроллеру Бензоколонки никогда не перекрываются, так что мож­но использовать один буфер вместо двух.

Контроллер Бензоколонки является производителем в одном интерфейсе со слабо связанным обменом. В этом случае потребители находятся в других распределенных подсистемах. Следовательно, мы будем использовать три очереди-разъема (см. рис.32), которые скрывают детали асинхронного обмена сообщениями с потенциально удаленными задачами, очередь Сообщений Планировщика инкапсулирует детали обмена с Планировщиком, а очередь Сообщений Лам­почки Типа Бензина – деталь обмена с Подсистемой Типа Бензина.



Рис.30. Классы интерфейса устройств

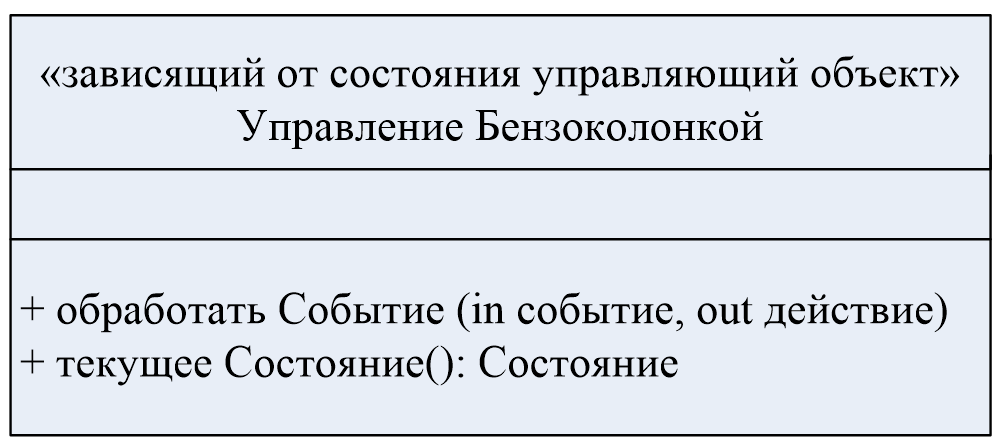


Рис.31. Зависящий от состояния управляющий класс

В распределенном приложении разъемы сле­дует проектировать так, чтобы они могли во время выполнения определить, находится задача-получа­тель в том же или в удаленном узле. Отправители не должны знать о расположении получателей. Такая независимость от места позво­лит реализовать гибкую стратегию конфигурирова­ния, при которой распределенные подсистемы проектируются в виде распределенных компонентов. Экземпляры таких компонентов отображаются на физические узлы на этапе конфигурирования.

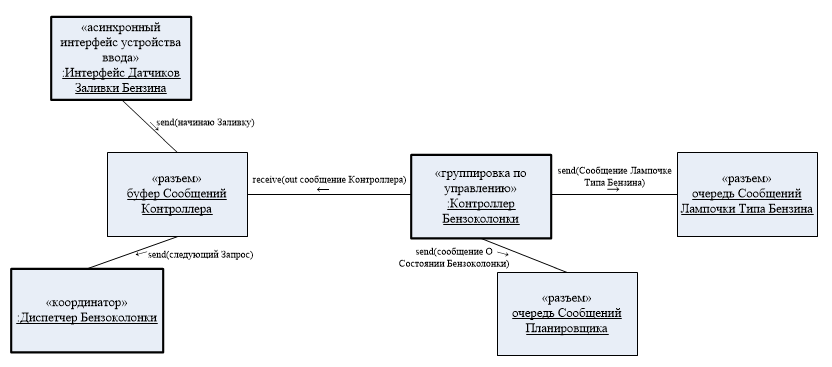


Рис.32. Проектирование разъемов Контроллера Бензоколонки

12.2. Проектирование составных задач

Рассмотрим теперь проектирование составной сгруппированной задачи Кон­троллер Бензоколонки, чтобы показать вложенные в нее объекты, скрывающие информа­цию. Объекты, описывающие каждую бензоколонку, размещаются внутри соответствующегоКонтроллера Бензоколонки. Это зависящие от состояния объекты Управление Бензоколонки и объект интерфейса пассивного устройств: Интерфейса Лампочки Заливки Бензина. Имеется также объект Координатор Бензоколонки, который выполняет координирующие функции для задачи в целом. На рис.33 приведен деталь­ный проект задачи Контроллер Бензоколонки.

Задачи мониторинга ресурсов также проектируются в виде составных. Объек­ты интерфейсов пассивных устройств ввода/вывода, обслуживающих несколько бензоколонок (Интерфейс Лампочки Типа Бензина), помещаются в задачу-монитор для соответствующего устройства, то есть в Мони­тор Лампочек Типа Бензина. Эти задачи получа­ют сообщения от нескольких бензоколонок и обеспечивают последовательный доступ к интерфейсному объекту Интерфейс Лампочки Типа Бензина (см. рис.26). Так, Монитор Лампочек Типа Бензина принимает сообщения от Контроллеров Бензоколонок с требовани­ем включить или выключить лампочку. При этом он вызывает операцию включить или выключить соответствующего объекта Интерфейс Лампочки Типа Бензина, передавая ей номер бензоколонки и типа бензина, взятые из сообщения.

Любой объект интерфейса асинхронного устройства ввода/вывода помещает­ся внутрь асинхронной задачи, поддерживающей это устройство. Например, все экземпляры объекта Интерфейс Кнопки Заливки Бензина принадлежат задаче Интерфейс Кнопок Заливки Бензина.

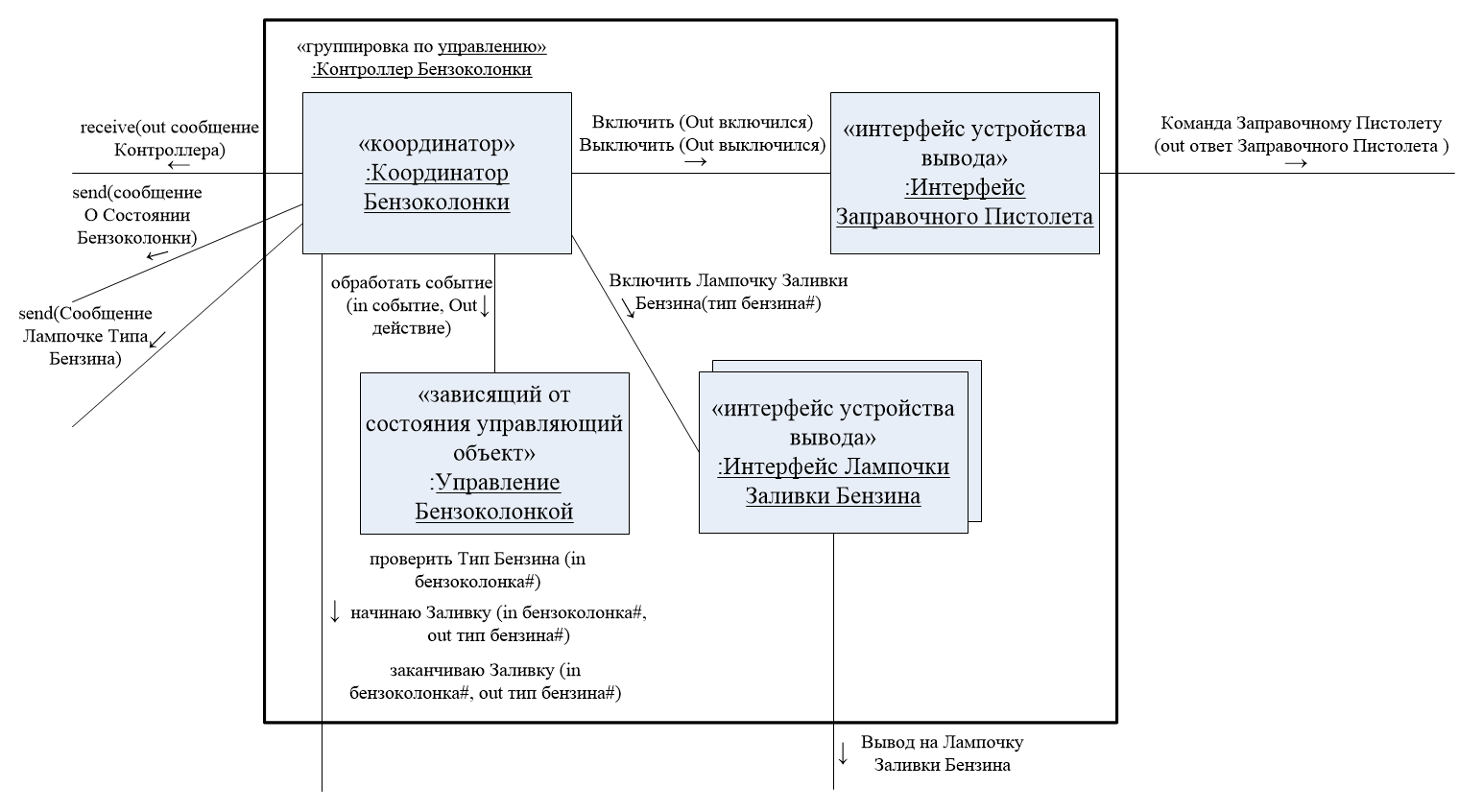


Рис.33. Детальный проект Контроллера Бензоколонки

13. Конфигурирование целевой системы

На этапе конфигурирования целевой системы производится отображение под­систем на физические узлы. Одна из возможных конфигураций такова: по одно­му узлу для каждого экземпляра Подсистемы Бензоколонки (один узел – одна бензоколонка), по одному узлу для каждого экземпляра Подсистемы Типа Бензина (один узел - одна бензоколонка) и еще один узел для Планировщика. Следовательно, если есть n бензоколонок и m типов бензинов, то физическая конфигурация будет состоять из n + m + 1 узлов. Такой вари­ант изображен на диаграмме развертывания (рис.34).



Рис.34. Диаграмма развертывания распределенной системы управления бензоколонками

Допустима и другая конфигурация, когда все экземпляры Подсистемы Типа Бензина отображаются на один узел. При этом каждая из задач, входящих в состав Под­системы Типа Бензина, будет отвечать за устройства ввода/вывода на всех типах бензина, а не на каком-то одном. Так, задача Интерфейс Кнопок Типа Бензина станет отслеживать состояние всех кнопок, задача Монитор Лампочек Типа Бензина– всех лампочек типа бензина. При этом никаких изменений в архитектуре задач для Подсистемы Типа Бензина не потре­буется, она останется такой же, как на рис.26.

Подсистему Планировщик можно оставить в отдельном узле или разместить в том же узле, что и Подсистему Типа Бензина. В последнем случае физическая конфи­гурация будет состоять из n + 1 узлов.

14. Анализ производительности нераспределенной системы управления бензоколонками

В этом разделе мы применим теорию планирования в реальном времени к ана­лизу производительности нераспределенного варианта системы управления бензоколонками, а потом проделаем то же самое для распределенного варианта.

14.1. Последовательности событий

Рассмотрим сначала соответствующие прецедентам последовательности событий в нераспределенной системе (рис.35).

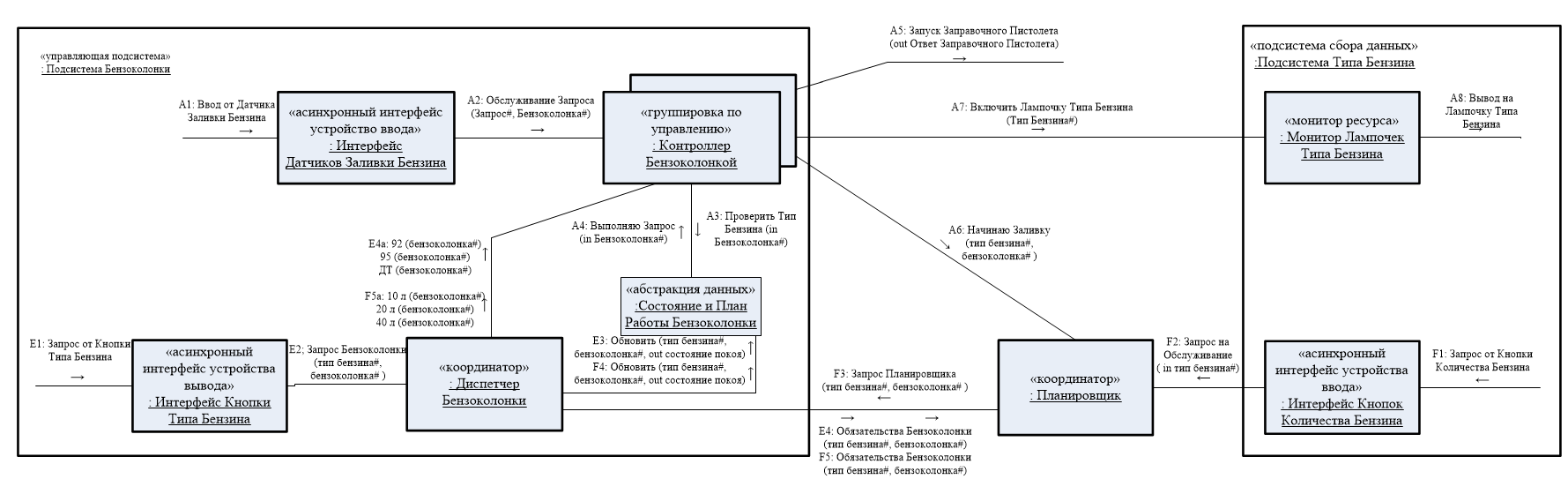


Рис.35. Последовательность событий в нераспределенной системе управления бензоколонками

Последовательность событий «Выбор Типа Бензина» (период = Тb):

Е1: Поступил Запрос Кнопки Типа Бензина Интерфейсу Кнопки Типа Бензина.

Е2: Интерфейс Кнопки Типа Бензина посылает Запрос Бензоколонки Диспетчеру Бензоколонки.

Е3: Диспетчер Бензоколонки просит Состояние и План Работы Бензоколонки добавить запрос в список работы, которые необходимо сделать

Е4: Диспетчер Бензоколонки отправляет Планировщику сообщение Обя­зательство Бензоколонки, информирующее о том, что Бензоколонке необходимо выполнить запрос.

Е4а: Диспетчер Бензоколонки посылает сообщение 92 (95 или ДТ) Контроллеру Бензоколонки, извещающее, какой тип бензина заливать.

Последовательность событий «Количество Бензина» (период = Тс):

F1: Поступил Запрос Кнопки Количества Бензина Интерфейсу Кнопки Количества Бензина.

F2: Интерфейс Кнопки Количества Бензина посылает Запрос на Обслуживание Планировщик.

F3: Планировщик выбирает Бензоколонку и передает Запрос Планировщика Диспетчеру Бензоколонки, входящему в состав выбранной Бензоколонки.

F4: Диспетчер Бензоколонки отправляет сообщение Обновить Со­стоянию и План Движения Бензоколонки, чтобы тот добавил новый запрос в список работы, который необходимо выполнить.

F5: Диспетчер Бензоколонки посылает Планировщику сообщение Обяза­тельство Бензоколонки.

F5a: Если бензоколонка не заливает, то Диспетчер Бензоколонка посылает сообщение 10 (20 или 40 литров) Контроллер Бензоколонкой, извещающее, какое количество бензина заливать.

Последовательность событий «Заливка Бензина» (период = Та):

А1: Интерфейс Датчика Заливки Бензина получает информацию от внешнего устройства.

А2: Интерфейс Датчика Заливки Бензина посылает тип бензина в сообщении Обслуживание Запроса Контроллеру Бензоколонкой.

A3: Контроллер Бензоколонкой отправляет сообщение Проверить Тот Ли Тип Бензина Состоянию и План Работы Бензоколонки, который проверяет, какой тип бензина выбран.

А4: Состояние и План Работы Бензоколонки передает сообщение Выполняю Запрос Контроллеру Бензоколонкой. В сообщении содержится тип бензина.

А5: Контроллер Бензоколонкой вызывает операцию Запуск Заправочного Пистолета.

А6: Контроллер Бензоколонкой посы­лает Планировщику сообщение Начинаю Заливку.

А7: Контроллер Бензоколонкой посы­лает Монитору Лампочки Типа Бензина сообщение Включить Лампочку Типа Бензина (с параметром «92», «95» или «ДТ»)

А8: Монитор Лампочек Типа Бензина включает Лампочку Типа Бензина.

14.3. Назначение приоритетов

Параметры задач в нераспределенной системе управления бензоколонками приведе­ны в табл.1. Время ЦП для каждой задачи включает затраты на контекстное переключение (не более двух переключений на задачу). Затраты на обработку со­общений поровну разделены между задачей-отправителем и задачей-получателем. Периоды всех задач, участвующих в данной последовательности обработки собы­тий, одинаковы: они определяются поступлением внешнего события, послужив­шего началом последовательности.

Заметим также, что периоды трех асинхронных задач интерфейса устройств (все они управляются прерываниями) кратны друг другу, то есть эти задачи могут быть готовы к активизации практически одновременно. Поскольку прерывания нужно обрабатывать максимально быстро, данным задачам следует назначить наивысшие приоритеты.

Задачи, управляемые прерываниями, исполняются с более высоким приоритетом, чем остальные задачи в данной последовательности. Из-за этого всю сово­купность задач нельзя привести к одной эквивалентной задаче с тем же периодом, но потребляющей большее время ЦП. Придется проектировать их как разные задачи с оди­наковым периодом.

Таблица 1. Параметры задач для планирования в реальном времени на примере системы управления бензоколонками.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Задача | Время ЦП Сi | Период Тi | Коэффициент использования Ui | Назначенный приоритет |
| Последовательность событий «Выбор Типа Бензина» | | | | |
| Интерфейс Кнопок Типа Бензина | 5 | 85 | 0,01 | 1 |
| Диспетчер Бензоколонок (случай «б») | 7 | 85 | 0,03 | 2 |
| Состояние и План Работы Бензоколонки | 3 | 85 | 0,05 | 3 |
| Диспетчер Бензоколонок (случай «б») | 7 | 85 | 0,04 | 4 |
| Планировщик | 7 | 85 | 0,05 | 5 |
| Диспетчер Бензоколонок (случай «б») | 7 | 85 | 0,05 | 6 |
| Контроллер Бензоколонки | 4 | 85 | 0,05 | 7 |
| Полное затраченное время = 40 мс |  |  | Полный коэффициент использования = 0,28 |  |
| Последовательность событий «Количество бензина» | | | | |
| Интерфейс Кнопок Количества Бензина | 5 | 110 | 0,01 | 8 |
| Планировщик | 7 | 110 | 0,05 | 9 |
| Диспетчер Бензоколонки (случай «в») | 7 | 110 | 0,04 | 10 |
| Состояние и План Работы Бензоколонки | 3 | 110 | 0,05 | 11 |
| Диспетчер Бензоколонки (случай «в») | 7 | 110 | 0,04 | 12 |
| Планировщик | 7 | 110 | 0,05 | 13 |
| Диспетчер Бензоколонки (случай «в») | 7 | 110 | 0,04 | 14 |
| Контроллер Бензоколонки | 4 | 110 | 0,05 | 15 |
| Полное затраченное время = 47 мс |  |  | Полный коэффициент использования = 0,33 |  |
| Последовательность событий «Заливка бензина» | | | | |
| Интерфейс Датчиков Заливки Бензина | 3 | 220 | 0,03 | 16 |
| Контроллер Бензоколонки | 4 | 220 | 0,05 | 17 |
| Состояние и План Работы Бензоколонки | 3 | 220 | 0,05 | 18 |
| Контроллер Бензоколонки | 4 | 220 | 0,05 | 19 |
| Контроллер Бензоколонки | 4 | 220 | 0,05 | 20 |
| Контроллер Бензоколонки | 4 | 220 | 0,05 | 21 |
| Планировщик | 7 | 220 | 0,05 | 22 |
| Контроллер Бензоколонки | 4 | 220 | 0,05 | 23 |
| Монитор Лампочек Типа Бензина | 4 | 220 | 0,001 | 24 |
| Полное затраченное время = 36 мс |  |  | Полный коэффициент использования = 0,39 |  |

Изображение выглядит как текст, диаграмма, число, График

Автоматически созданное описание

Примечание. На этом рисунке используется расширенная нотация UML.для временных диаграмм.

Рис.36. Система управления бензоколонками: диаграмма последовательности с временными метками

14.4. Планирование в реальном времени для нераспределенной архитектуры

Сложение коэффициентов использования ЦП для всех задач в табл.1 дает полный коэффициент 0,901, то есть намного меньше значения 0,999, полученного из теоремы о верхней границе. Но, поскольку присвоение приоритетов произведено с нарушением принципа частотной монотонности, необходим более детальный разбор.

Анализ выполняется для каждой последовательности, так как критично время, требующееся для завершения именно последовательности, а не каждой зада­чи в отдельности. Нужно принять во внимание вытеснение задачами с более вы­соким приоритетом, а также блокировку низкоприоритетными задачами.

Вытеснение иногда возникает из-за задач в других последовательностях событий, имеющих более короткий период и более высокий приоритет. Они способ­ны вытеснять любую задачу несколько раз. Причиной вытеснения бывают также более приоритетные задачи с большими периодами, например высокоприоритетные задачи, управляемые прерываниями, которые принадлежат последовательностям событий с длинными периодами. Но такого рода вытеснения не могут происходить более одного раза. Затраты времени на блокировку связаны с низкоприоритетны­ми задачами, захватившими ресурс, который нужен высокоприоритетной задаче, -в данном случае речь идет об Состояние и План Работы Бензоколонки.

Ниже приводится анализ планирования в реальном времени для каждой последовательности событий. Его результаты представлены на временной диаграм­ме (рис.36): предполагается худший случай, когда все внешние прерывания возникают одновременно. На рисунке изображены только задачи и обмен сооб­щениями между ними. Доступ к пассивным не показан, так как он про­исходит в контексте потока управления задачи.

14.5. Последовательность событий «Выбор Типа Бензина»

Задачи, входящие в последовательность событий: Интерфейс Кнопки Типа Бензина и Диспетчер Бензоколонки, Состояние и План Работы Бензоколонки. Из табл.1 видно, что период этой последовательности Тb равен 85 мс.

Рассмотрим те же четыре фактора, что и выше:

– время выполнения задач в последовательности событий: 5 мс для задачи Интерфейс Кнопок Типа Бензина, затем 7 мс для задачи Диспетчер Бензоколонки, Состояние и План Работы Бензоколонки 3 мс, и т.д. по таблице 1, что в сумме дает Сb = 40 мс. Коэффициент использования ЦП за счет выполнения Ub = Сb / Тb = 40/85 = 0,47;

– время вытеснения более приоритетными задачами с меньшими периодами. В течение 85 мс по два раза могут выполниться задачи Диспетчер Бензоколонок и Контроллер Бензоколонки, Состояние и План Работы Бензоколонки, что приведет к вытеснению на 28 мс;

– вытеснение более приоритетными задачами с большими периодами: 5 мс способна занять задача Интерфейс Кнопок Типа Бензина на обработку прерыва­ния (она вытесняет Диспетчера Бензоколонки). Следовательно, общее время вы­теснения более приоритетными задачами Сp =28 + 5 = 33 мс, а соответству­ющий коэффициент использования Up = Сp / Тp = 33 / 85 = 0,388;

– время блокировки задачами с более низким приоритетом. Задача Планировщик может занимать критическую секцию на 7 мс, блокируя задачу Диспетчер Бензоколонки. Время блокировки Вb в худшем случае равно 7 мс, а коэффициент использования Ub = Вb / Тb = 7 / 85 = 0,0824.

Таким образом, получаем:

Суммарное затраченное время = время выполнения + время вытеснения + + время блокировки в худшем случае = 40 + 33 + 7 = 80 мс, что меньше периода, равного 85 мс.

Полный коэффициент использования = коэффициент использования за счет выполнения + коэффициент использования за счет вытеснения + коэффициент использования за счет блокировки = Up + Ue + Ub = 0,0824 + 0,47= 0,5524, что меньше верхней границы 0,999.

Согласно обобщенной теореме о верхней границе коэффициента использова­ния и обобщенной теореме о времени завершения, задачи в последовательности событий Выбор Типа Бензина удовлетворяют временным ограничениям.

14.6. Последовательность событий «Количество Бензина»

**Задачи, входящие в последовательность событий:** Интерфейс Кнопок Количества Бензина, Планировщик и Диспетчер Бензоколонки (участвует сразу в двух последовательностях событий). Из табл.1 видно, что период этой последовательности Тс равен 110 мс.

Рассмотрим все те же четыре фактора:

– время выполнения задач в последовательности событий. 5 мс для задачи Интерфейс Кнопок Количества Бензина, затем 7 мс для задачи Диспетчер Бензоколонки. После этого Состояние и План Работы Бензоколонки 3 мс, т.д. по таблице 1, что в сумме дает Сс = 40 мс. Коэффициент использования ЦП за счет выполнения Uc = Сс / Тс= = 0,364;

– время вытеснения более приоритетными задачами с меньшими периодами. По четыре раза могут выполниться задачи Интерфейс Кнопок Количества Бензина и Контроллер Бензоколонки (вытесняют Диспетчера Бензоколонки и Планиров­щика, Состояние и План Работы Бензоколонки), что в сумме дает 36 мс.

Задачи Интерфейс Кнопок Типа Бензина и Диспетчер Бензоколонки (вытесняют Пла­нировщика) в состоянии выполниться по два раза, на что уйдет еще 24 мс. Следовательно, общее время вытеснения более приоритетными задачами Сp = 36 + 24 = 60 мс, а соответствующий коэффициент использования Up = Cp/Tp =0,545;

– вытеснение более приоритетными задачами с большими периодами. Поскольку эта последовательность имеет наибольший период, таких задач нет;

– время блокировки. Блокировка из-за доступа к разделяемому объекту другими задачами уже принята во внимание, дополнительных затрат на блокировку нет.

Таким образом, получаем:

Суммарное затраченное время = время выполнения + время вытеснения + время блокировки в худшем случае = 40 + 60 = 100 мс, что меньше периода, равного 110 мс.

Полный коэффициент использования = коэффициент использования за счет выполнения + коэффициент использования за счет вытеснения + коэффициент использования за счет блокировки = Up + Ue = 0,545 + 0,364 = 0,909, что меньше верхней границы 0,999.

Согласно обобщенной теореме о верхней границе коэффициента использова­ния и обобщенной теореме о времени завершения, задачи в последовательности событий Количество Бензина удовлетворяют временным ограничениям.

1***4.7. Последовательность событий «Заливка Бензина»***

**Задачи, входящие в последовательность событий:** Интерфейс Датчиков Заливки Бензина и Контроллер Бензоколонки. Из табл.1 видно, что период этой последова­тельности Тa составляет 220 мс.

Рассмотрим четыре фактора, вытекающие из обобщенной теории планирова­ния в реальном времени:

– время выполнения задач в последовательности событий: 3 мс для задачи Интерфейс Датчиков Заливки Бензина, затем 4 мс для задачи Контроллер Бензоколонки, 3 мс Состояние и План Работы Бензоколонки и т.д. по таблице 1, что в сумме дает Са = 36 мс. Коэффициент использования ЦП за счет выполнения Ua = Са / Та = 36 / 220 = 0,164;

– время вытеснения более приоритетными задачами с меньшими периодами. Так как у этой последовательности наименьший период, то таких задач нет;

– вытеснение более приоритетными задачами с большими периодами. Таких задач нет, т.к. у данной задачи все периоды приоритеты больше.

– время блокировки задачами с более низким приоритетом.

Задача Планировщик способна занимать критическую секцию на 7 мс, пока работает с разделяемым объектом, и все это время задача Контроллер Бензоколонки будет блокирована. Время блокировки Ва в худшем случае равно 7 мс, а коэффи­циент использования Ub = Ва / Та = 7 / 220 = 0,032.

Теперь можно определить суммарное затраченное время и полный коэффици­ент использования ЦП.

Суммарное затраченное время = время выполнения + время вытеснения + время блокировки в худшем случае = Са *+* Ва = 36 + 7 = 43 мс, что меньше периода, равного 220 мс.

Полный коэффициент использования = коэффициент использования за счет выполнения + коэффициент использования за счет вытеснения + коэффициент использования за счет блокировки = Ue + Ub = 0,032 + 0,164= 0,196, что меньше верхней границы 0,999.

Согласно обобщенной теореме о верхней границе коэффициента использова­ния и обобщенной теореме о времени завершения, задачи в последовательности событий Заливка Бензина всегда укладываются в срок.

15. Анализ производительности распределенной системы управления бензоколонками

При увеличении чис­ла бензоколонок и типов бензина нагрузка на процессор возрастет, и в конце концов система справляться перестанет.

15.1. Сценарий для анализа производительности

Теперь рассмотрим показанный на рис.29 распределенный проект, в котором имеется по одному узлу на каждую бензоколонку, по одному узлу на каждый тип бензина и еще один узел для Планировщика. Предположим, что используются те же самые процессоры (то есть время выполнения не меняется) и локальная сеть со ско­ростью 100 Мбит/с, характеризуемая фиксированными задержками.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Задача | Время ЦП Сi | Период Тi | Коэффициент использования Ui | Назначенный приоритет |
| Подсистема Бензоколонки | | | | |
| Интерфейс Датчиков Заливки Бензина | 3 | 80 | 0,05 | 1 |
| Контроллер Бензоколонки | 4 | 80 | 0,11 | 2 |
| Диспетчер Бензоколонки | 7 | 110 | 0,08 | 3 |
| Подсистема Типа Бензина | | | | |
| Интерфейс Кнопок Типа Бензина | 5 | 110 | 0,08 | 1 |
| Монитор Лапочек Типа Бензина | 4 | 80 | 0,01 | 2 |
| Подсистема планировщик | | | | |
| Сервер Состояния и Плана Работы Бензоколонки | 3 | 220 | 0,01 | 1 |
| Планировщик Бензоколонки | 7 | 220 | 0,12 | 2 |

Таблица 2. Параметры задач для планирования в реальном времени на примере распределенной системы управления бензоколонками

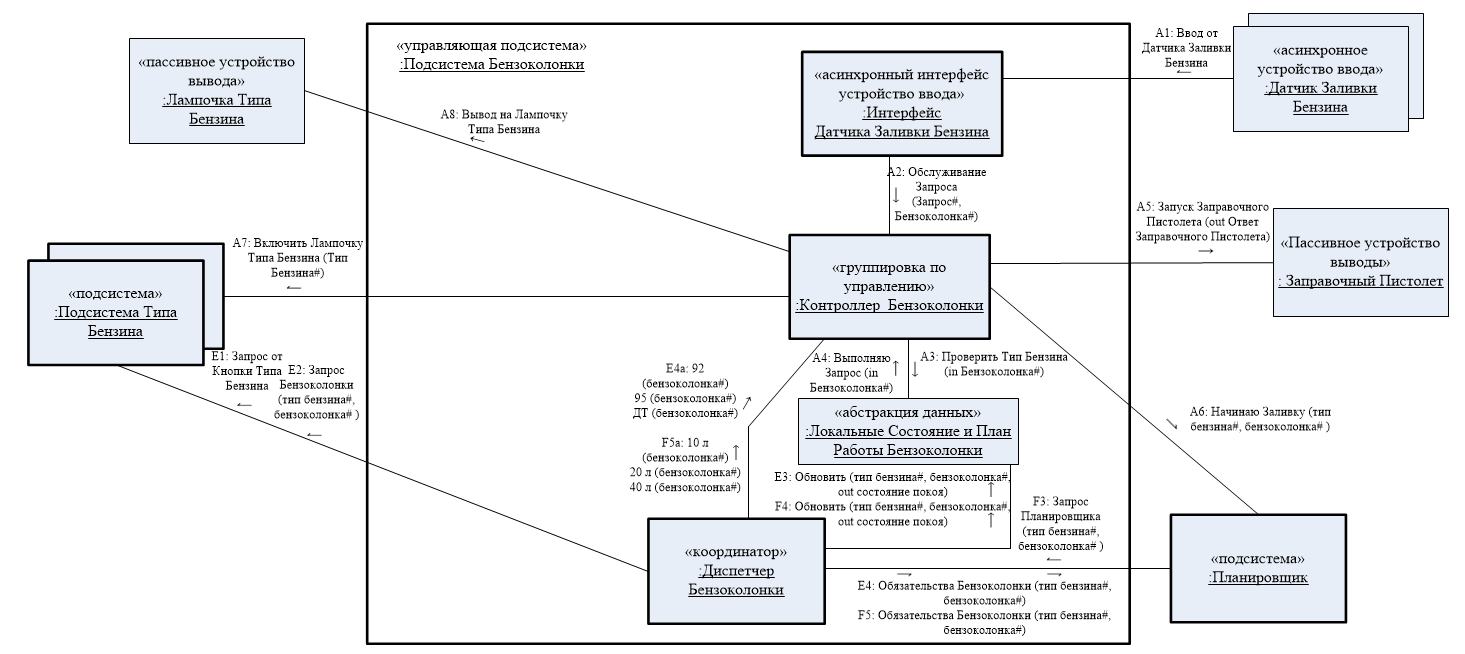


Рис.37. Последовательность событий в распределенной подсистеме бензоколонки

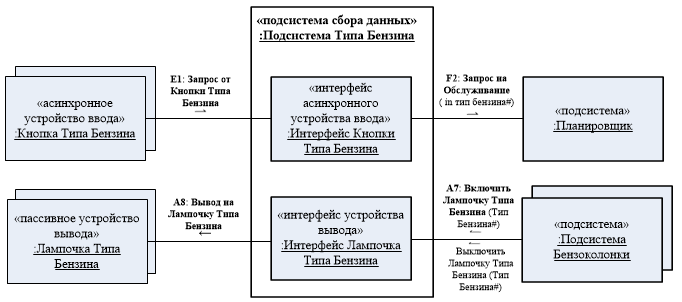


Рис.38. Последовательность событий для подсистемы Типа Бензина

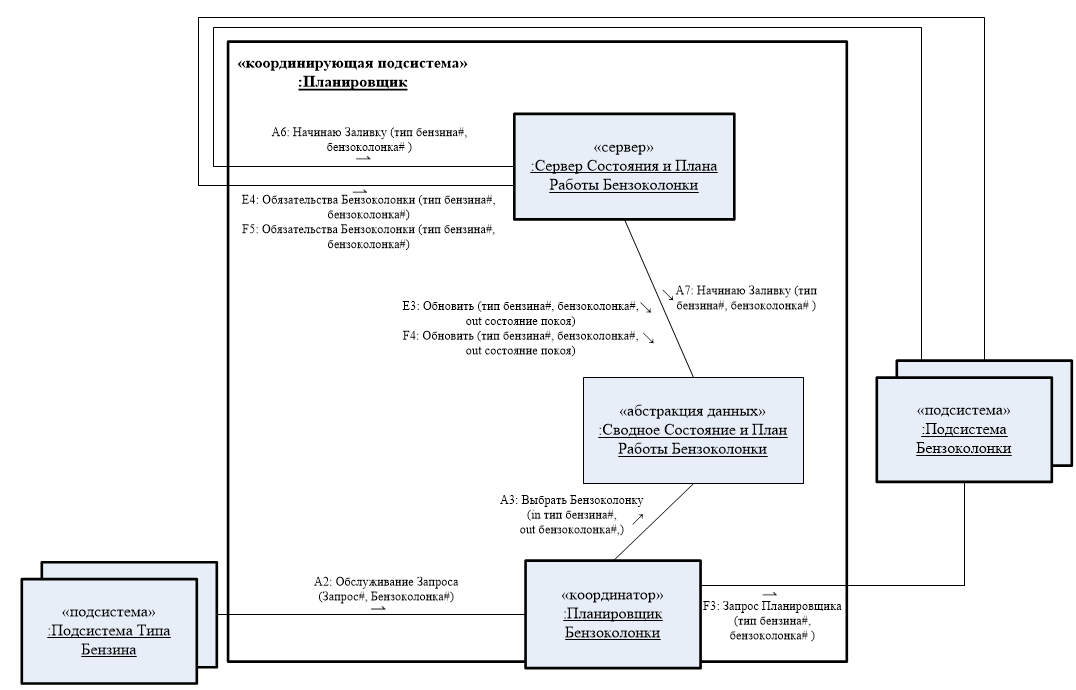


Рис.39. Последовательность событий для подсистемы планировщика

15.2. Планирование в реальном времени для распределенной архитектуры

Рассмотрим три критические последовательности событий для распределенной конфигурации (см. рис.37-39).

**Последовательность событий «Выбор Типа Бензина»** (период = Тb):

Е1: Поступил Запрос Кнопки Типа Бензина Интерфейсу Кнопки Типа Бензина.

Е2: Интерфейс Кнопки Типа Бензина посылает Запрос Бензоколонки Диспетчеру Бензоколонки.

Е3: Диспетчер Бензоколонки просит Состояние и План Работы Бензоколонки добавить запрос в список работы, которые необходимо сделать

Е4: Диспетчер Бензоколонки отправляет Планировщику сообщение Обя­зательство Бензоколонки, информирующее о том, что Бензоколонке необходимо выполнить запрос.

Е4а: Диспетчер Бензоколонки посылает сообщение 92 (95 или ДТ) Контроллеру Бензоколонки, извещающее, какой тип бензина заливать.

**Последовательность событий «Выбор Этажа Назначения»** (период = Тb):

**Последовательность событий «Количество Бензина»** (период = Тс):

F1: Поступил Запрос Кнопки Количества Бензина Интерфейсу Кнопки Количества Бензина.

F2: Интерфейс Кнопки Количества Бензина посылает Запрос на Обслуживание Планировщик.

F3: Планировщик выбирает Бензоколонку и передает Запрос Планировщика Диспетчеру Бензоколонки, входящему в состав выбранной Бензоколонки.

F4: Диспетчер Бензоколонки отправляет сообщение Обновить Со­стоянию и План Движения Бензоколонки, чтобы тот добавил новый запрос в список работы, который необходимо выполнить.

F5: Диспетчер Бензоколонки посылает Планировщику сообщение Обяза­тельство Бензоколонки.

F5a: Если бензоколонка не заливает, то Диспетчер Бензоколонка посылает сообщение 10 (20 или 40 литров) Контроллер Бензоколонкой, извещающее, какое количество бензина заливать.

**Последовательность событий «Заливка Бензина»** (период = Та):

А1: Интерфейс Датчика Заливки Бензина получает информацию от внешнего устройства.

А2: Интерфейс Датчика Заливки Бензина посылает тип бензина в сообщении Обслуживание Запроса Контроллеру Бензоколонкой.

A3: Контроллер Бензоколонкой отправляет сообщение Проверить Тот Ли Тип Бензина Состоянию и План Работы Бензоколонки, который проверяет, какой тип бензина выбран.

А4: Состояние и План Работы Бензоколонки передает сообщение Выполняю Запрос Контроллеру Бензоколонкой. В сообщении содержится тип бензина.

А5: Контроллер Бензоколонкой вызывает операцию Запуск Заправочного Пистолета.

А6: Контроллер Бензоколонкой посы­лает Планировщику сообщение Начинаю Заливку.

А7: Контроллер Бензоколонкой посы­лает Монитору Лампочки Типа Бензина сообщение Включить Лампочку Типа Бензина (с параметром «92», «95» или «ДТ»)

А8: Монитор Лампочек Типа Бензина включает Лампочку Типа Бензина.

15.3. Последовательность событий «Выбор Типа Бензина»

В эту последовательность событий входят только задачи из Подсистемы Бензоколонки и Подсистемы Планировщика (см. рис.37-39). Сначала опишем Подсистему Бензоколонки.

Задачи, входящие в последовательность событий: Интерфейс Датчиков Заливки и Контроллер Бензоколонки. Период последовательности равен Та = 80 мс.

Рассмотрим четыре фактора, вытекающие из обобщенной теории планирова­ния в реальном времени:

– время выполнения задач в последовательности событий. 3 мс для задачи Интерфейс Датчиков Заливки, затем 4 мс для задачи Контроллер Бензоколонки, что в сумме дает Са = 7 мс. Коэффициент использования ЦП за счет выполнения Ua = Сa / Тa = 7 / 50 = 0,0875;

– время вытеснения более приоритетными задачами с меньшими периодами. Поскольку эта последовательность имеет наименьший период, таких задач нет;

– вытеснение более приоритетными задачами с большими периодами. Возмож­но вытеснение на 3 мс задачей Диспетчер Бензоколонки. Время вытеснения Ср = 7 мс, а соответствующий коэффициент использования Up = 7 / 80 = 0,0875;

Теперь легко определить суммарное затраченное время и полный коэффици­ент использования ЦП.

Суммарное затраченное время = время выполнения + время вытеснения + + время блокировки в худшем случае = 7 + 7 = 14 мс, что меньше периода, равного 80 мс.

Полный коэффициент использования = коэффициент использования за счет выполнения + коэффициент использования за счет = Ue + Up = 0,0875 + 0,0875 = 0,175, что меньше верхней границы 0,999.

Согласно обобщенной теореме о верхней границе коэффициента использова­ния и обобщенной теореме о времени завершения, задачи в последовательности событий Выбор Типа Бензина всегда успевают уложиться в срок. В дан­ном случае ограничение по времени жесткое: если задача не завершится вовремя, бензоколонка не зальет бензин. Коэффициент использования ЦП ниже, чем в нераспределенном случае, поскольку теперь в Подсистеме Бензоколонки меньше задач.

15.4. Последовательность событий «Количество Бензина»

В обработке последовательности событий Количество Бензина также принимают участие только задачи из Подсистемы Бензоколонки и Подсистемы Планировщика (см. рис.37-39). Сначала рассмотрим Подсистему Бензоколонки.

**Задачи, входящие в последовательность событий:** Интерфейс Кнопок Типа Бензина, Монитор Лампочек Типа Бензина. Период Тb этой последовательности равен 110 мс.

Рассмотрим четыре фактора, вытекающие из обобщенной теории планирова­ния в реальном времени:

– время выполнения задач в последовательности событий. 3 мс для задачи Интерфейс Датчиков Заливки, затем 4 мс для задачи Контроллер Бензоколонки, что в сумме дает Са = 7 мс. Коэффициент использования ЦП за счет выполнения Ua = Сa / Тa = 7 / 50 = 0,0875;

– время вытеснения более приоритетными задачами с меньшими периодами. Поскольку эта последовательность имеет наименьший период, таких задач нет;

– вытеснение более приоритетными задачами с большими периодами. Возмож­но вытеснение на 3 мс задачей Диспетчер Бензоколонки. Время вытеснения Ср = 7 мс, а соответствующий коэффициент использования Up = 7 / 80 = 0,0875;

Теперь легко определить суммарное затраченное время и полный коэффици­ент использования ЦП.

Суммарное затраченное время = время выполнения + время вытеснения + + время блокировки в худшем случае = 7 + 7 = 14 мс, что меньше периода, равного 80 мс.

Полный коэффициент использования = коэффициент использования за счет выполнения + коэффициент использования за счет = Ue + Up = 0,0875 + 0,0875 = 0,175, что меньше верхней границы 0,999.

Согласно обобщенной теореме о верхней границе коэффициента использова­ния и обобщенной теореме о времени завершения, задачи из Подсистемы Бензоколонки, принимающие участие в последовательности событий Выбор Этажа Назначения удовлетворяют временным ограничениям. И снова коэффициент использо­вания ЦП оказывается меньше, чем для централизованного решения.

15.5. Последовательность событий «Заливка Бензина»

В обработке последовательности событий Заливка принимают участие задачи из нескольких распределенных подсистем (см. рис.37-39). Хотя, как показано выше, полный коэффициент использования ЦП в каждом узле не выходит за допустимые пределы, необходимо удостовериться, что затраченное время не превышает заданного предельного значения. Нужно применить теорию планирования в реальном времени к каждому узлу, учитывая параметры, приведенные в табл.2. Период последовательности событий Тс = 220 мс.

В Подсистеме Бензоколонки возможны следующие задержки:

Рассмотрим четыре фактора, вытекающие из обобщенной теории планирова­ния в реальном времени:

– время выполнения задач в последовательности событий. 3 мс для задачи Интерфейс Датчиков Заливки, затем 4 мс для задачи Контроллер Бензоколонки, что в сумме дает Са = 7 мс. Коэффициент использования ЦП за счет выполнения Ua = Сa / Тa = 7 / 50 = 0,0875;

– время вытеснения более приоритетными задачами с меньшими периодами. Поскольку эта последовательность имеет наименьший период, таких задач нет;

– вытеснение более приоритетными задачами с большими периодами. Возмож­но вытеснение на 3 мс задачей Диспетчер Бензоколонки. Время вытеснения Ср = 7 мс, а соответствующий коэффициент использования Up = 7 / 80 = 0,0875;

Теперь легко определить суммарное затраченное время и полный коэффици­ент использования ЦП.

Суммарное затраченное время = время выполнения + время вытеснения + + время блокировки в худшем случае = 7 + 7 = 14 мс, что меньше периода, равного 80 мс.

Полный коэффициент использования = коэффициент использования за счет выполнения + коэффициент использования за счет = Ue + Up = 0,0875 + 0,0875 = 0,175, что меньше верхней границы 0,999.

Согласно обобщенной теореме о верхней границе коэффициента использова­ния и обобщенной теореме о времени завершения, задачи в последовательности событий Выбор Типа Бензина всегда успевают уложиться в срок. В дан­ном случае ограничение по времени жесткое: если задача не завершится вовремя, бензоколонка не зальет бензин. Коэффициент использования ЦП ниже, чем в нераспределенном случае, поскольку теперь в Подсистеме Бензоколонки меньше задач.

Вывод

В рамках этой курсовой работы была разработана система реального времени – управление бензоколонкой.

При разработке системы были изучены основные этапы создания системы в реальном времени с применением UML, рассмотрены основные характеристики, функции и методы управления, а также общую концепцию системы. Был получен опыт проектирования систем реального времени, их поддержки (модификации) и управления. В результате была разработана аналитическая модель системы реального времени, и была проведена оценка ее эффективности. Также рассмотрены варианты проектирования системы для случаев с распределенной и нераспределенной архитектурой.

Список Литературы

Сулейманова А.М. Проектирование систем реального времени: Учеб. пособие/Уфимск.гос.авиац.ун-т.-Уфа,2004.-292с.

Гома Х. UML. Проектирование систем реального времени, параллельных и распределенных приложений: Пер. с англ. – М.: ДМК Пресс, 2002. – 704 с.