



Практический анализ с использованием RUP

Николай Киреев

ИИТ БГУИР

Цели доклада

1. Ознакомление с методологией унифицированного процесса разработки программного обеспечения (RUP) на конкретном примере промышленной ПС;
2. Обсуждение основных подходов и принципов моделирования

Список сокращений

ООПС – объектно-ориентированная программная система

ПС -- программная система

ПО – предметная область

ВИ – вариант использования, прецедент

RUP – Rational Unified Process

ЭФ – экранная форма

МК – металлический компонент

НМК – неметаллический компонент (кокс, известь, спецкокс)

Общие принципы моделирования



1. Объектно-ориентированный подход. Поведение (функциональность) ОО ПС реализуется взаимодействующими объектами ПС, которые относятся к связанным между собой классам ПС [1].
2. Взаимодействие объектов описывается клиент-серверной моделью: объект «клиент» посылает СООБЩЕНИЕ объекту «сервер» с целью вызова соответствующей операции.
3. Итерационный подход к построению моделей.
4. В моделях в соответствующем представлении фиксируется все, что прямо или косвенно связано с работой ПС.
5. Создаваемые модели должны быть полезны заинтересованным сторонам: пользователям (Заказчикам), проектировщикам и другим разработчикам.
6. Модели должны быть простые и читабельные.
7. Количество создаваемых артефактов должно быть достаточным, чтобы отразить все важные аспекты разрабатываемой ПС и минимальным, чтобы максимально сократить время затрачиваемое на анализ и проектирование.

Аналитическая модель



Аналитическая модель – это точное, четкое представление задачи, позволяющее отвечать на вопросы и строить решения.

На этапе проектирования вы будете ссылаться именно на аналитическую модель, а не на исходную формулировку задачи. [1]

Аналитическая модель включает

- Модель предметной области (domain model);
- Модель программной системы (application model).

Представления аналитической модели

1. **Представление классов (Logical View).** Моделируем: сущности предметной области (business entity), классы анализа (boundary, entity, controll);
2. **Представление процессов & состояний (Process View).** Моделируем: бизнес-процессы, последовательности действий в вариантах использования, алгоритмы операций;
3. **Представление прецедентов (Use Case View).** Моделируем: варианты использования (use case), пользователей (actor), объекты классов анализа, их связи и взаимодействие.

Начало. Организационная структура проекта

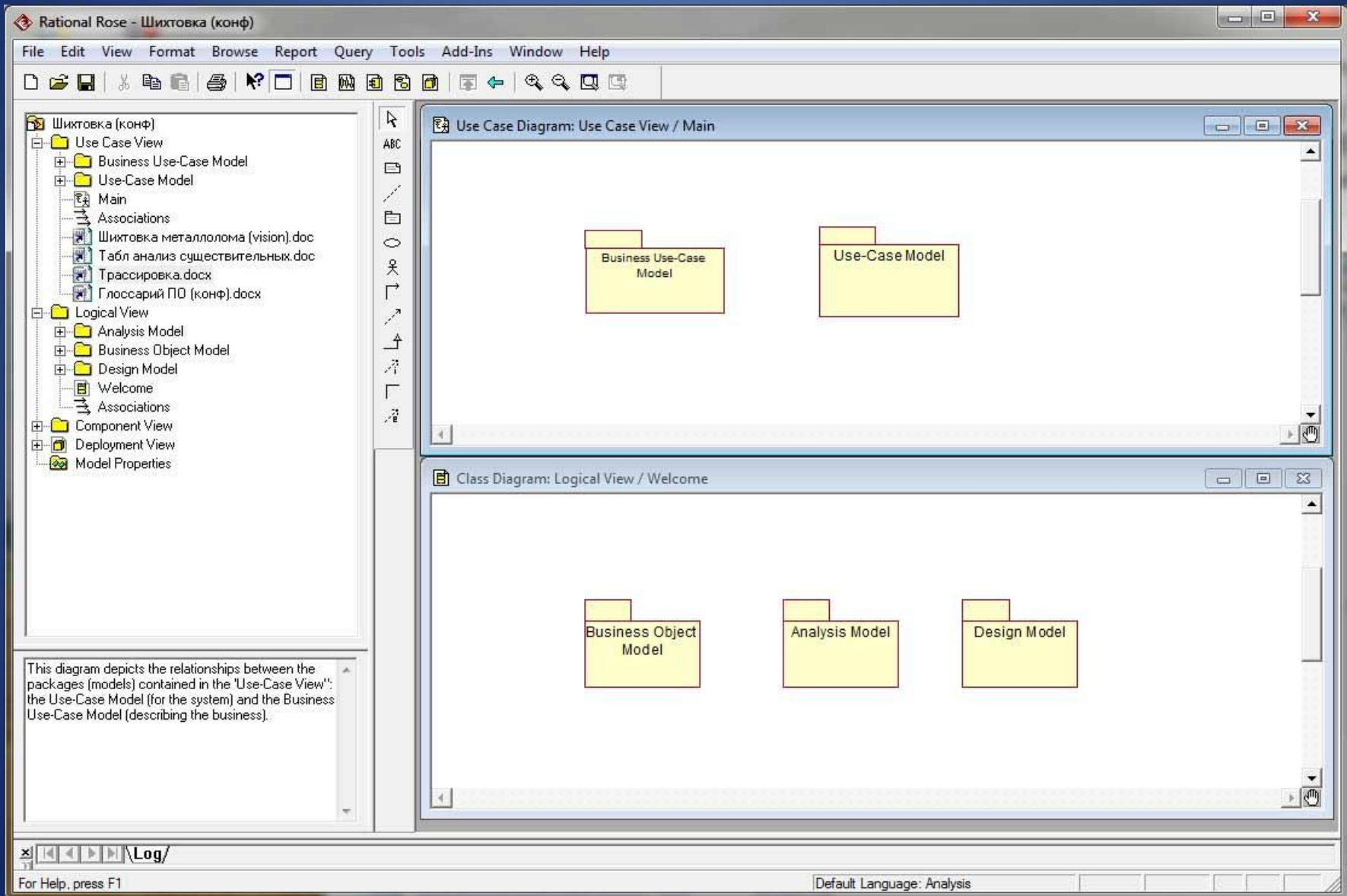


Без четкой организационной структуры любой проект может превратиться в хаос.

Каждый артефакт должен храниться в своей папке и легко находится всеми заинтересованными участниками проекта.

1. В соответствии с принятой аналитической моделью при помощи пакетов создаем организационную структуру проекта. При этом можно использовать предоставляемый в Rational Rose шаблон „rational unified process“, либо создавать в соответствующих представлениях свои папки.
2. Пакеты помещаем на организационных диаграммах соответствующих представлений.
3. К пакету представления Use Case View подсоединяем исходные материалы (концепцию ПС, анкеты, опросные листы и т.д.)

Начало. Организационная структура проекта



The screenshot displays the Rational Rose software interface for a project named "Шихтовка (конф)". The interface is divided into several panes:

- Left Pane (Project Structure):** Shows a hierarchical tree view of the project. The "Use Case View" is expanded, showing sub-packages: "Business Use-Case Model", "Use-Case Model", and "Main". Under "Main", there are "Associations" and several document files: "Шихтовка металлолома (vision).doc", "Табл анализ существительных.doc", "Трассировка.docx", and "Глоссарий ПО (конф).docx". Other views like "Logical View", "Component View", and "Deployment View" are also visible.
- Top Diagram Pane:** Titled "Use Case Diagram: Use Case View / Main", it contains two yellow rectangular boxes representing packages: "Business Use-Case Model" and "Use-Case Model".
- Bottom Diagram Pane:** Titled "Class Diagram: Logical View / Welcome", it contains three yellow rectangular boxes representing packages: "Business Object Model", "Analysis Model", and "Design Model".
- Bottom Left Text:** A descriptive text box states: "This diagram depicts the relationships between the packages (models) contained in the 'Use-Case View': the Use-Case Model (for the system) and the Business Use-Case Model (describing the business)."
- Bottom Status Bar:** Includes a log window showing "\Log/" and the text "For Help, press F1". The default language is set to "Analysis".

Моделирование предметной области (ПО)

1. Моделирование предметной области является в RUP опциональной и выполняется для самих разработчиков, Заказчик является экспертом предметной области и не нуждается в ее моделировании.
2. Целью моделирования предметной области является выработка точной, четкой, доступной для понимания и, наконец, корректной модели реального мира [2].
3. В зависимости от конкретного проекта по усмотрению бизнес-аналитиков при анализе ПО могут создаваться следующие артефакты: глоссарий терминов предметной области, диаграмма бизнес-сущностей (классов ПО), диаграммы взаимодействия объектов ПО (моделирование связей), диаграммы бизнес-процессов или состояний, диаграммы функционеров (бизнес-актеров) и их функциональные обязанности, модели, описывающие состояния и условия перехода из одного в другое.

Артефакты бизнес-анализа

Глоссарий – текстовой документ, разъясняющий разработчикам специфические термины в том числе жаргонные, встречающиеся в данной предметной области.

В глоссарии также документируются синонимы, омонимы и факты смены терминологии.

Глоссарий не должен дублировать информацию введенную в других артефактах. Информация вводится в систему однократно.

Цель глоссария – это НЕ повышение IQ участников проекта, а разъяснение терминов с точки зрения работы ПС.

ПРИМЕР

Шихтовка (процесс шихтовки) – это процесс составления плавильной смеси, в котором обеспечивается заданная пропорция МК и соответствующее набранному суммарному весу МК количество кокса, извести и спецкокса.

Моделирование сущностей ПО

Цель: выявление и документирование бизнес-сущностей (классов ПО) и их атрибутов.

Бизнес-сущность — некий объект ПО или концептуальное понятие ПО, характеризуемое набором данных (существенных признаков), прямо или косвенно связанное с проектируемой программной системой.

Бизнес-сущности — это кандидаты в классы и объекты ПС, которые отвечают за ввод, изменение, удаление и хранение данных (атрибутов).

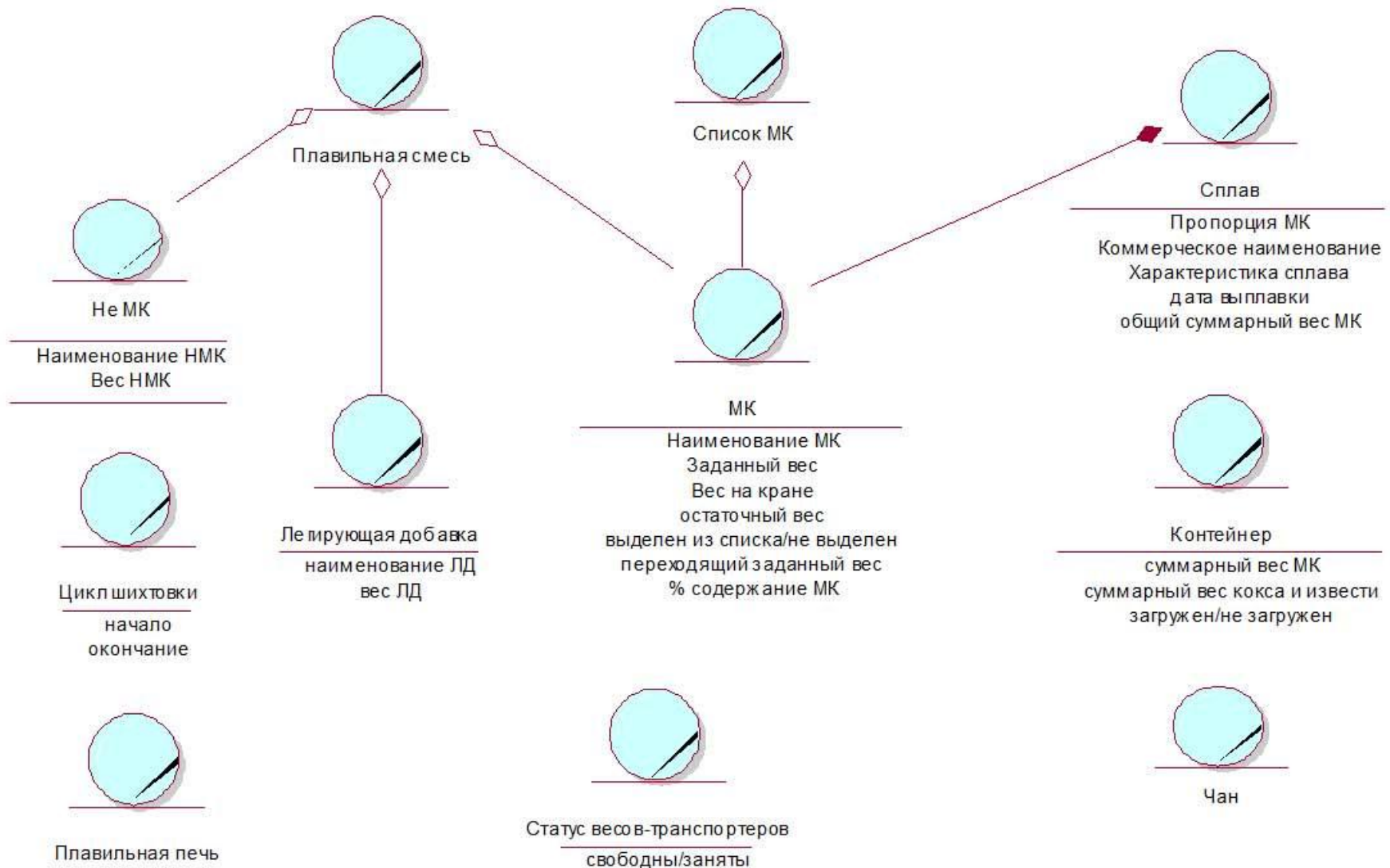
Методика выявления бизнес-сущностей: чтение текста концепции и анализ имен существительных.

ПРИМЕР анализа текста:

Инструментом для этого служит **КРАН**, который забирает **МЕТАЛЛИЧЕСКИЕ КОМПОНЕНТЫ** (МК) своим *магнитом* и потом опускает в дозировочный **КОНТЕЙНЕР**. В соответствии с **ВЕСОМ НАБРАННЫХ МК** подаются **КОКС, ИЗВЕСТЬ, СПЕЦКОКС**. По заполнении контейнера его *содержание* со всеми занесенными в него металлическими компонентами направляется в **ЧАН** для плавления, в который добавляются **ЛЕГИРУЮЩИЕ ДОБАВКИ**. В конце рабочего дня чан направляется в **ПЛАВИЛЬНУЮ ПЕЧЬ**. Так как в процессе плавления должны использоваться определенные *соотношения веса различных МК*, программа должна обеспечить требуемый состав **ПЛАВИЛЬНОЙ СМЕСИ**. Этот *контроль и составление смеси компонентов для плавки* называется **ШИХТОВКОЙ**.

Программная система устанавливается на *рабочем месте* **КРАНОВЩИКА** и используется им для контроля над процессом шихтовки. Кроме того, она позволяет **ТЕХНОЛОГУ** задать определенную **ПРОПОРЦИЮ МК** в конечном **СПЛАВЕ** и осуществлять сохранение *информации* о составе и свойствах полученных сплавов в *базе данных*.

Моделирование сущностей ПО



Моделирование бизнес-актеров



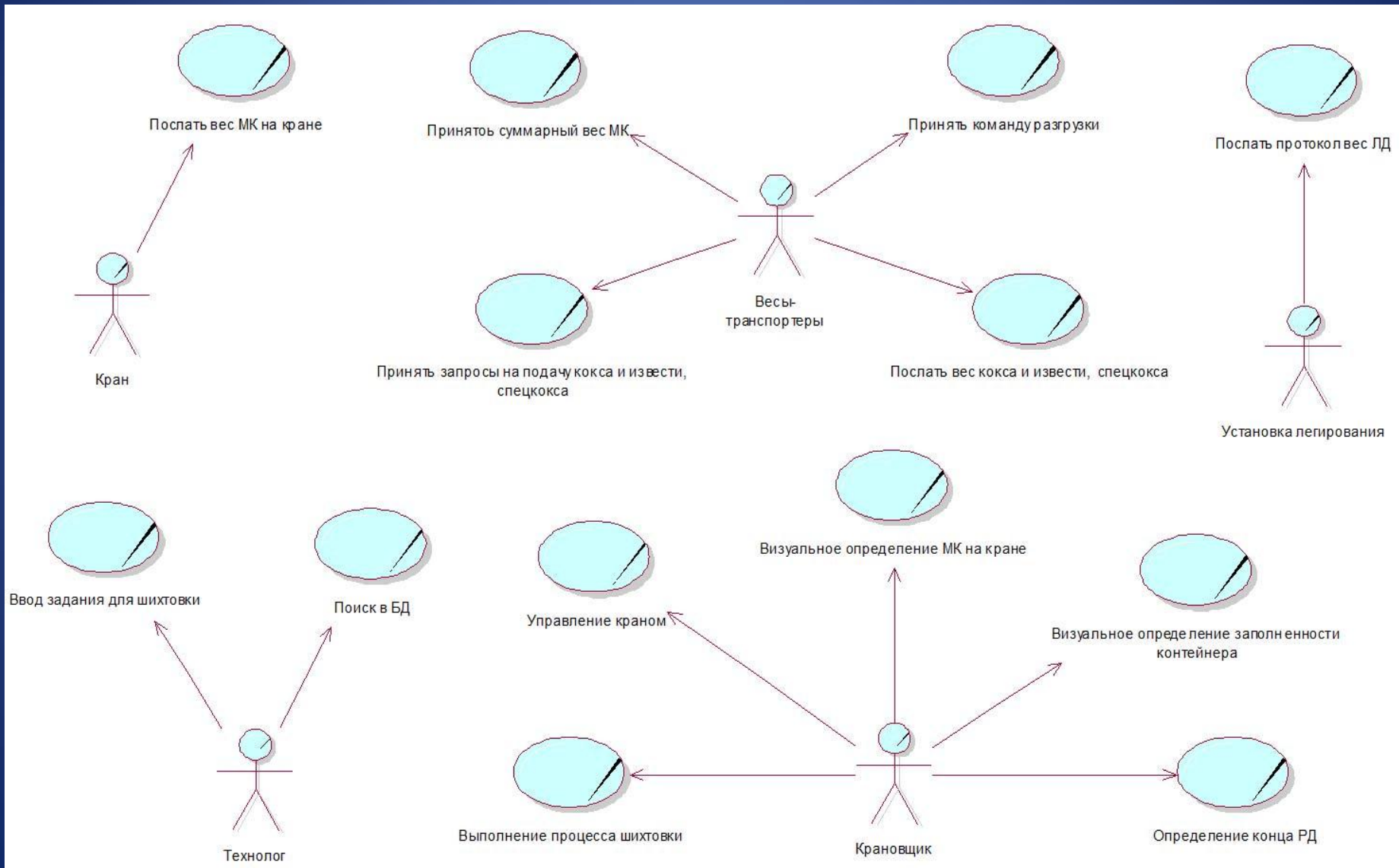
Модель бизнес-актеров и их функциональных обязанностей (бизнес-функций) строится в представлении Use Case View в папке Business Use Case Model (при вызове шаблона rational unified process) на диаграмме ВИ.

Цель моделирования: изучение действующих лиц ПО и их функциональных обязанностей для понимания бизнес-процессов, выявления пользователей системы, уточнения высокоуровневых бизнес-целей.

Business Actor — это штатная единица (действующее лицо, функционер) в предметной области, который взаимодействует с ПС, либо не взаимодействует, но является участником бизнес-процесса. Чаще всего это штатный работник предприятия или должность (генеральный директор, главный инженер, экономист, зам. директора по производству и т.д.), но может быть и обособленной штатной системой, исполняющей определенные функциональные обязанности (например: кран, весы-транспортеры и т.д.).

Business Use Case — это функциональные (должностные) обязанности, возложенные на данного функционера. Фиксируются только те из них, которые прямо или косвенно связаны с работой программной системы.

Моделирование бизнес-актеров

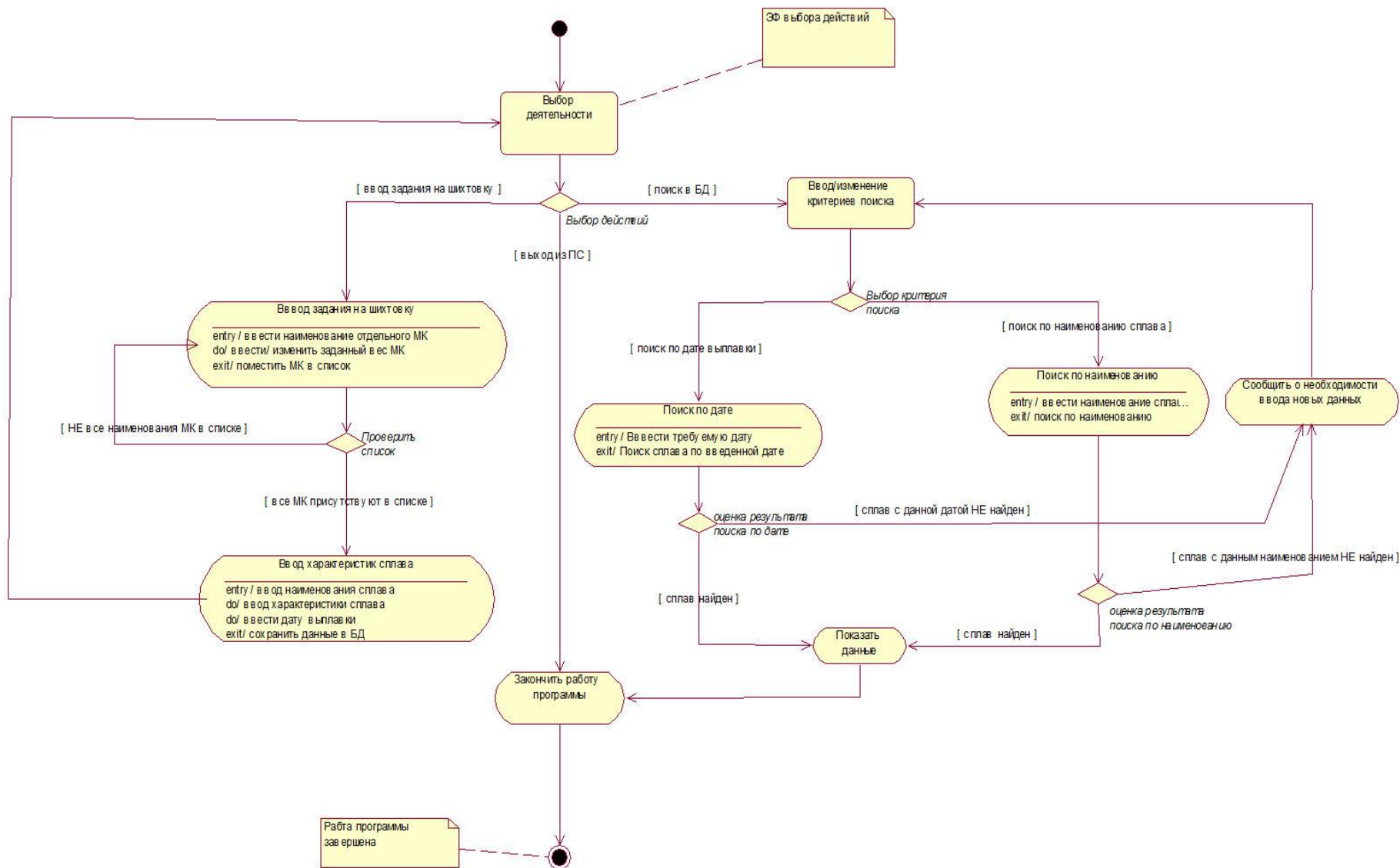


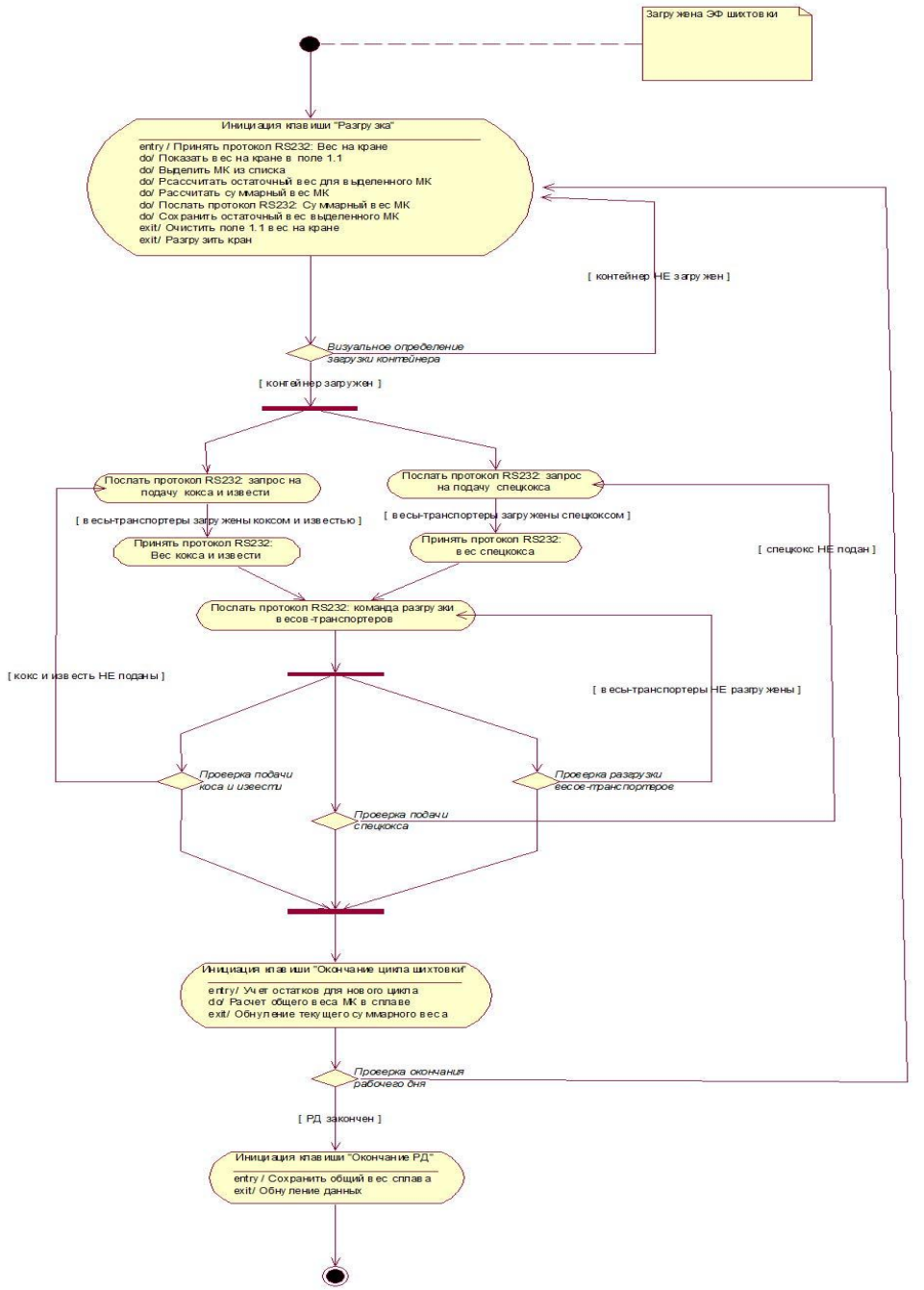
Моделирование бизнес-процессов

Если проектируемая ПС должна обеспечивать некий бизнес-процесс, то его необходимо изучить и смоделировать.

В Rational Rose отсутствует отдельное представление Process View, однако, можно присоединить диаграмму деятельности (activity diagram) к любому элементу в браузере проекта.

Моделирование бизнес-процессов (действия технолога)





Моделирование бизнес-процессов (действия крановщика)

Анализ требований



Выясняем ЧТО должна делать система
и строим ее концептуальную модель

Требование (requirement) -- желательное свойство, характеристика или условие, которым должна удовлетворять система в процессе своей эксплуатации.

Применительно к ПС используется следующая *классификация требований*, которая получила название модели **FURPS+**, что соответствует первым буквам соответствующих категорий требований на английском языке:

- функциональные требования (*Functionality*)
- требования удобства использования (*Usability*)
- требования надежности (*Reliability*)
- требования производительности (*Performance*)
- требования возможности сопровождения (*Supportability*)

Анализ требований

символом "+" обозначены дополнительные условия, к которым относятся:

- проектные ограничения
- требования управления системой
- требования к графическому интерфейсу пользователя
- физические требования
- юридические требования

Функциональное требование -- желаемое поведение системы с точки зрения ее пользователя.

Функциональные требования реализуются функциями ПС.

Функция системы — поведение, которое необходимо реализовать в разрабатываемой программной системе.

Если X действительно является функцией системы, то имеет смысл следующее предложение: система должна выполнять «X»

Формулировка пожеланий (требований) заказчика может быть неконкретной, расплывчатой и не однозначной.

Анализ требований



Формат записи требований по [4] :

<id> <имя системы>должна (shall)<действие, которое должно быть выполнено>

Классификация функциональных требований по [3]:

- *Группа функциональных требований*
 - *Бизнес-требования (Business Requirements)* – определяют высокоуровневые цели организации или клиента (потребителя) – заказчика разрабатываемого программного обеспечения.
 - *Пользовательские требования (User Requirements)* – описывают цели/задачи пользователей системы, которые должны достигаться/выполняться пользователями при помощи создаваемой программной системы. Эти требования часто представляют в виде вариантов использования (Use Cases).
 - *Функциональные требования (Functional Requirements)* – определяют функциональность (поведение) программной системы, которая должна быть создана разработчиками для предоставления возможности выполнения пользователями своих обязанностей в рамках бизнес-требований и в контексте пользовательских требований.

Анализ требований



Каждое функциональное требование необходимо выявить, зафиксировать, уточнить, конкретизировать его смысл и поставить ему в соответствие одну или несколько функций системы. Некоторые требования могут быть абстрактными, т.е. им не будет соответствовать ни одна функция системы и они из дальнейшего рассмотрения исключаются.

Функциональные требования в RUP моделируются при помощи вариантов использования.

Вариант использования (*Use Case, прецедент*) — внешняя спецификация последовательности действий, которые система может выполнять в процессе взаимодействия с действующими лицами (actor) с целью получения значимого для них результата [2].

Прецедент описывает поведение, демонстрируемое системой с целью получения значимого результата для одного или более актеров.

Читая текст концепции (vision), фиксируем в функциональные требования, детализируем их в виде функций (бизнес-логики), каждую функцию отображаем соответствующим вариантом использования и сводим в трассировочную таблицу (матрицу).

Анализ требований



№	Требование Заказчика	№	Функция системы	Use-Case
1	...программа должна обеспечить требуемый состав плавильной смеси		НЕТ (абстрактное требование)	
2	Программная система устанавливается на рабочем месте крановщика и используется им для контроля над процессом шихтовки		НЕТ (абстрактное требование)	
3	она позволяет технологу задать определенную пропорцию МК в конечном сплаве	3.1 3.2 3.3	См. 5.1 См. 5.2 См. 5.3	
4	осуществлять сохранение информации о составе и свойствах полученных сплавов в базе данных	4.1	См. 23.1 – 23.7	
5	Технолог вводит в программу заданный вес по каждому МК, предназначенному для плавки, определяя, таким образом, их пропорцию в сплаве	5.1 5.2 5.3	Ввести наименование МК Ввести заданный вес Добавить в список	Ввод наименования МК Ввод заданного веса Добавить в список
6	... коммерческое название сплава и характеристику сплава	6.1 6.2	Ввести название сплава Ввести характеристику сплава	Ввод названия сплава Ввод характеристики сплава
7	Вес поднятого краном МК определяется автоматически и вводится через последовательный порт (RS232) в компьютер	7.1	Принять протокол RS232 (вес МК)	Принять протокол RS232 (вес МК)
8	Вес кокса и извести обуславливается суммарным весом набранных в контейнер МК, который посылается на весы-транспортёры протоколом RS232	8.1 8.2	Рассчитать суммарный вес МК Послать протокол RS232 (суммарный вес МК)	Рассчитать суммарный вес МК Послать протокол RS232 (суммарный вес МК)
9	Кокс и известь подаются в контейнер смешанными, однако в протоколе их веса представлены по отдельности. Представляет интерес суммарный вес кокса и извести	9.1 9.2	Принять протокол RS232 (вес кокса и извести) Рассчитать суммарный вес кокса и извести	Принять протокол RS232 (вес кокса и извести) Рассчитать суммарный вес кокса и извести
10	крановщик посылает соответствующий запрос на весы-транспортёры (протокол RS232), (команда «дозировать кокс и известь»),	10.1	Послать протокол RS232 (команду «дозировать кокс и известь»)	Послать протокол RS232 («дозировать кокс и известь»)
11	система должна индентифицировать статус весов-транспортёров «загружены».	11.1 11.2	Определить статус весов-транспортёров Показать статус весов-транспортёров	Определить статус весов-транспортёров Показать статус весов-транспортёров
12	крановщик должен послать команду разгрузки на весы-транспортёры, после чего статус весов изменится на «разгружены»	12.1 12.2 12.3	Послать протокол RS232 (команду разгрузки весов-транспортёров) См. 11.1 См. 11.2	
13	Его вес, аналогично весам кокса и извести обуславливается суммарным весом МК	13.1 13.2	См. 8.1 См. 8.2	
14	Ход процесса подачи спецкокса полностью аналогичен подаче кокса и извести	14.1 14.2 14.3 14.4 14.5	Принять протокол RS232 (вес спецкокса) Послать протокол RS232 (команду «дозировать спецкокс») См. 11.1 См. 11.2 См. 12.1	Принять протокол RS232 (вес спецкокса) Послать протокол RS232 («дозировать спецкокс»)
15	показываются в качестве информации без контроля со стороны пользователя	15.1 15.2 15.3 15.4	Принять протокол RS232 (вес ЛД) Показать вес FeSi Показать вес FeCr Показать вес FeMn	Принять протокол RS232 (вес ЛД) Показать вес FeSi Показать вес FeCr Показать вес FeMn
16	В компьютерной программе по каждому МК должен показываться заданный и остаточный веса, а также вес МК на кране	16.1 16.2 16.3 16.4 16.5 16.6 16.7 16.8	см. 5.1 см. 5.2 см. 5.3 см. 18.1 показать вес на кране показать список МК показать заданный вес показать остаточный вес	показать вес на кране показать список МК показать заданный вес показать остаточный вес
17	Крановщик, выделив наименование МК из списка может нажать клавишу «Разгрузка» в программной системе	17.1 17.2 17.3 17.4	Выделить МК в списке См. 18.1 См. 8.1 См. 8.2	Выделить МК в списке

№	Требование Заказчика	№	Функция системы	Use-Case
18	Вес, который был в этот момент на кране должен быть вычтен из величины заданного веса	18.1 18.2	Рассчитать остаточный вес выделенного МК Сохранить остаточный вес МК	Рассчитать остаточный вес МК Сохранить остаточный вес МК
19	оставшиеся остаточные веса должны быть учтены в следующем цикле шихтовки	19.1	Рассчитать заданный вес следующего цикла	Рассчитать заданный вес следующего цикла
20	контейнер, который оказался заполненным, то можно завершать текущий цикл шихтовки	20.1 20.2 20.3 20.4	См. 19.1 Рассчитать вес сплава Сохранить вес сплава Обнулить суммарный вес МК	Рассчитать вес сплава Сохранить вес сплава Обнулить суммарный вес МК
21	должны быть выполнены следующие проверки: • если кокс, известь и спецкокс не поданы, то процесс не может быть завершён; • весы по взвешиванию кокса и извести должны иметь статус «опорожнены»	21.1 21.2 21.3 21.4 21.5	Проверить подачу кокса и извести Проверить подачу спецкокса См. 11.1 См. 10.1 См. 14.2 См. 12.1	
22	должна быть предусмотрена возможность изменения заданного веса МК с рабочего места крановщика	22.1	Изменить заданный выбранного вес МК	Изменить заданный выбранного вес МК
23	Программа должна сохранять в базе данных (БД) наименование сплава, дату выплавки, вес сплава (общий суммарный вес набранных МК), список составляющих МК, их процентное содержание и характеристику сплава	23.1 23.2 23.3 23.4 23.5 23.6 23.7	Сохранить название сплава Сохранить дату выплавки Сохранит вес сплава Сохранить список МК Рассчитать пропорцию МК в % Сохранить пропорцию МК в % Сохранить характеристику сплава	Сохранить название сплава Сохранить дату выплавки Сохранит вес сплава Сохранить список МК Рассчитать пропорцию МК в % Сохранить пропорцию МК в % Сохранить характеристику сплава
24	Программа обеспечивает поиск по дате выплавки и по коммерческому наименованию сплава	24.1 24.2 24.3 24.4 24.5 24.6	Ввести дату Найти сплав по введенной дате Показать сообщение если сплав по введенной дате отсутствует Ввести название сплава Найти сплав по названию Показать сообщение если сплав с данным названием отсутствует	Ввести дату Найти сплав по введенной дате Показать сообщение если сплав по введенной дате отсутствует Ввести название сплава Найти сплав по названию Показать сообщение если сплав с данным названием отсутствует
25	выводиться на экран следующие данные (рабочее место Крановщика) • наименование сплава; • список МК с заданными и остаточными весами; • МК, набираемый в настоящий момент должен высвечиваться другим цветом; • вес МК на кране; • суммарный вес МК в контейнере; • общий суммарный вес набранных МК; • статус весов-транспортёров; • вес кокса, извести, спецкокса, суммарный вес кокса и извести; • количество FeCr, FeMn, FeSi	25.1 25.2 25.3 25.4 25.5 См. 16.5 См. 16.6 См. 16.7 См. 16.8 См. 11.2 См. 15.2 См. 15.3 См. 15.4	Показать название сплава Показать список МК Показать заданный вес каждого МК Показать остаточный вес каждого МК Показать выделенный МК См. 16.5 См. 16.6 См. 16.7 См. 16.8 Показать суммарный вес МК Показать вес сплава См. 11.2 Показать вес кокса Показать вес извести Показать вес спецкокса Показать вес кокса и извести	Показать название сплава Показать список МК Показать заданный вес каждого МК Показать остаточный вес каждого МК Показать выделенный МК Показать суммарный вес МК Показать вес сплава Показать вес кокса Показать вес извести Показать вес спецкокса Показать вес кокса и извести
26	выводиться на экран следующие данные (рабочее место Технолога) • дата выплавки; • наименование сплава; • список МК, вошедших в состав сплава; • содержание каждого МК в сплаве (в %); • характеристика сплава	См. 25.1 См. 25.2 См. 25.2	Показать дату выплавки См. 25.1 См. 25.2 Показать пропорцию % Показать характеристику сплава	Показать дату выплавки Показать пропорцию % Показать характеристику сплава

Анализ требований

Построение модели вариантов использования

Модель вариантов использования (use case model) — это модель, которая описывает функциональные требования к системе в терминах вариантов использования.

Диаграмма вариантов использования (use case diagram) — это диаграмма, на которой изображены отношения, существующие между актерами (actor) и вариантами использования системы (use case).

Актер (actor), синонимы актант, действующее лицо — это абстрактное понятие, которое характеризует внешнего пользователя (или нескольких пользователей), взаимодействующего с системой. Каждый актер соответствует одной роли в которой выступает пользователь, взаимодействуя с системой.

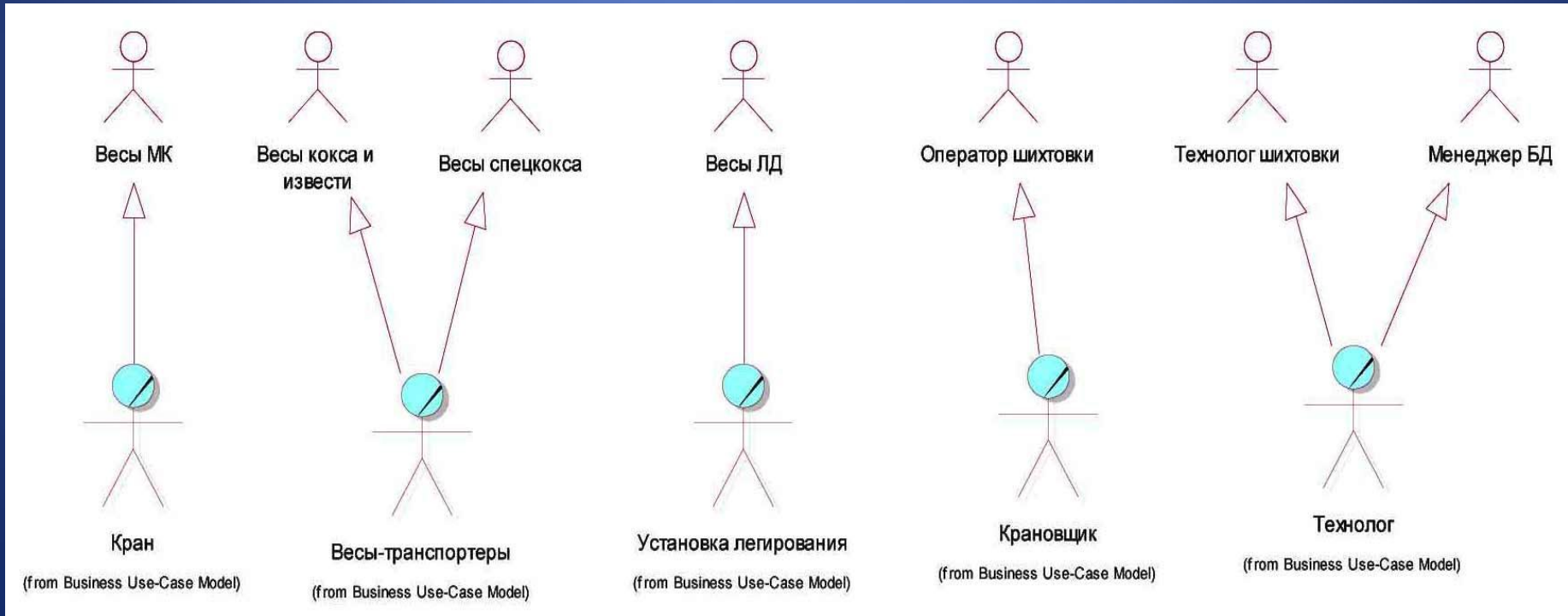
Каждый актер использует свои для него предназначенные сервисы, предоставляемые программной системой.

Анализ требований

Построение модели вариантов использования

1. Определяем актеров

Если некая штатная единица (Business Actor) выполняет хотя бы часть своих функциональных обязанностей с помощью ПС, то она может выступать в одной или нескольких ролях.



Анализ требований

Построение модели вариантов использования

2. Создаем организационную структуру (папки) модели ВИ

- Для каждого актера должна быть предусмотрена отдельная папка;
- Общие для нескольких актеров сервисы помещаются в отдельную папку.

3. Строим модель вариантов использования

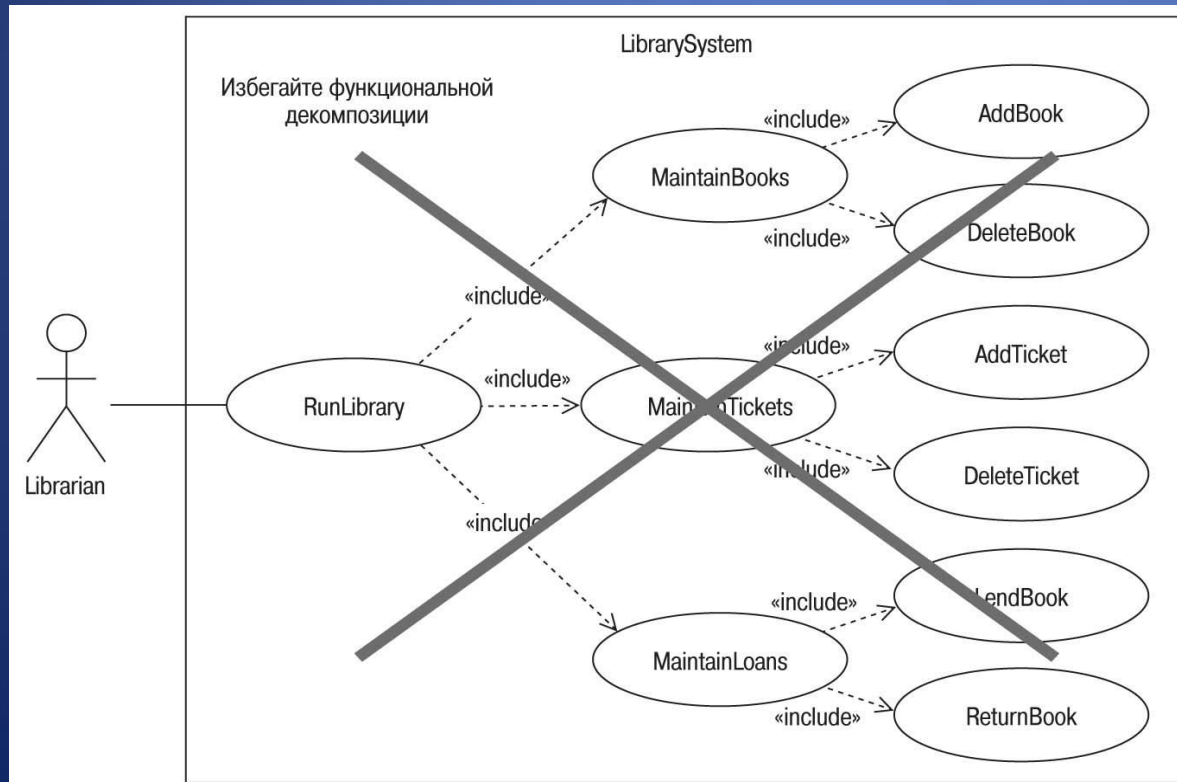
- Используя трассировочную матрицу, модель бизнес-процессов и диаграмму бизнес-актеров для каждого актера определяем набор базовых ВИ, общих сервисов (функциональные или пользовательские требования), которые инициируются непосредственно актерами, а к ним отношениями «включение» («include») и «расширение» («extend») добавляем функции ПС;
- В отдельной папке показываем ВИ, связанные с визуализацией данных.

Анализ требований

Построение модели вариантов использования

Основные правила построения модели ВИ:

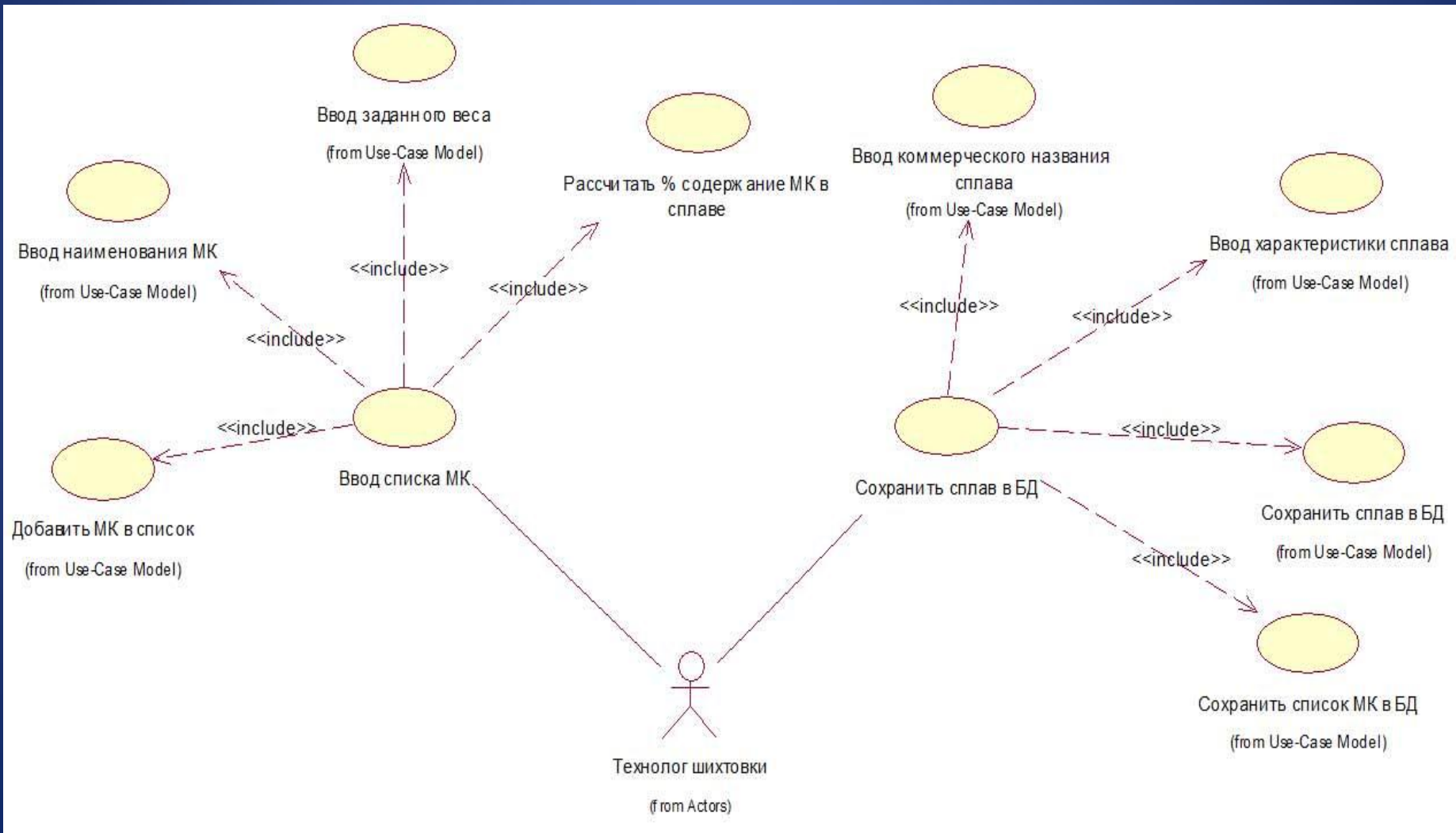
- каждый ВИ должен быть инициирован актером или вступать в отношения «include», «extend» к другим ВИ;
- ВИ должны концентрироваться на том «ЧТО», а не «КАК» должна делать система;



- необходимо избегать функциональной декомпозиции. Она не подходит для модели ВИ [2].
- предпочтительнее простая модель ВИ без отношений «include» и «extend».

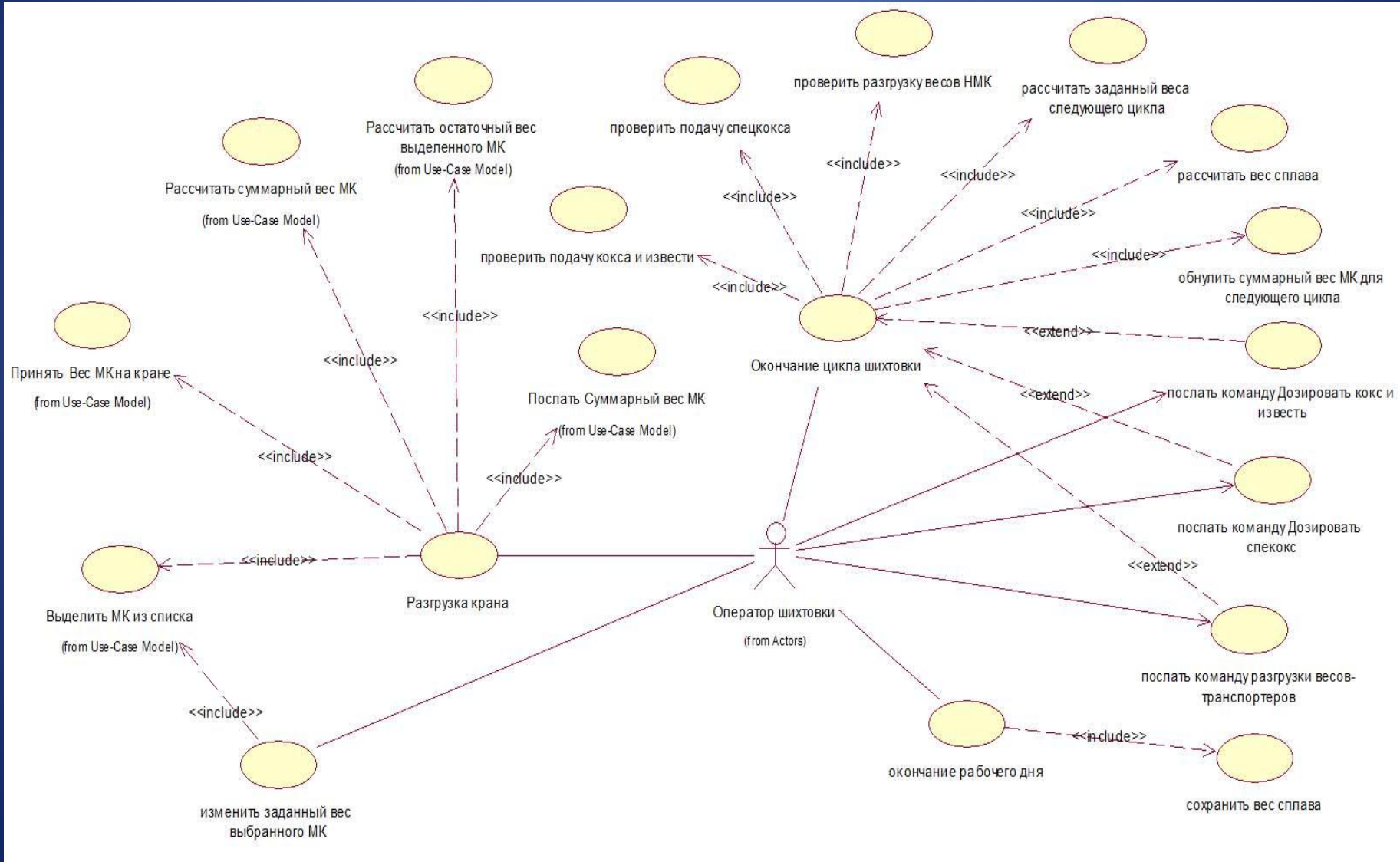
Анализ требований

Построение модели вариантов использования



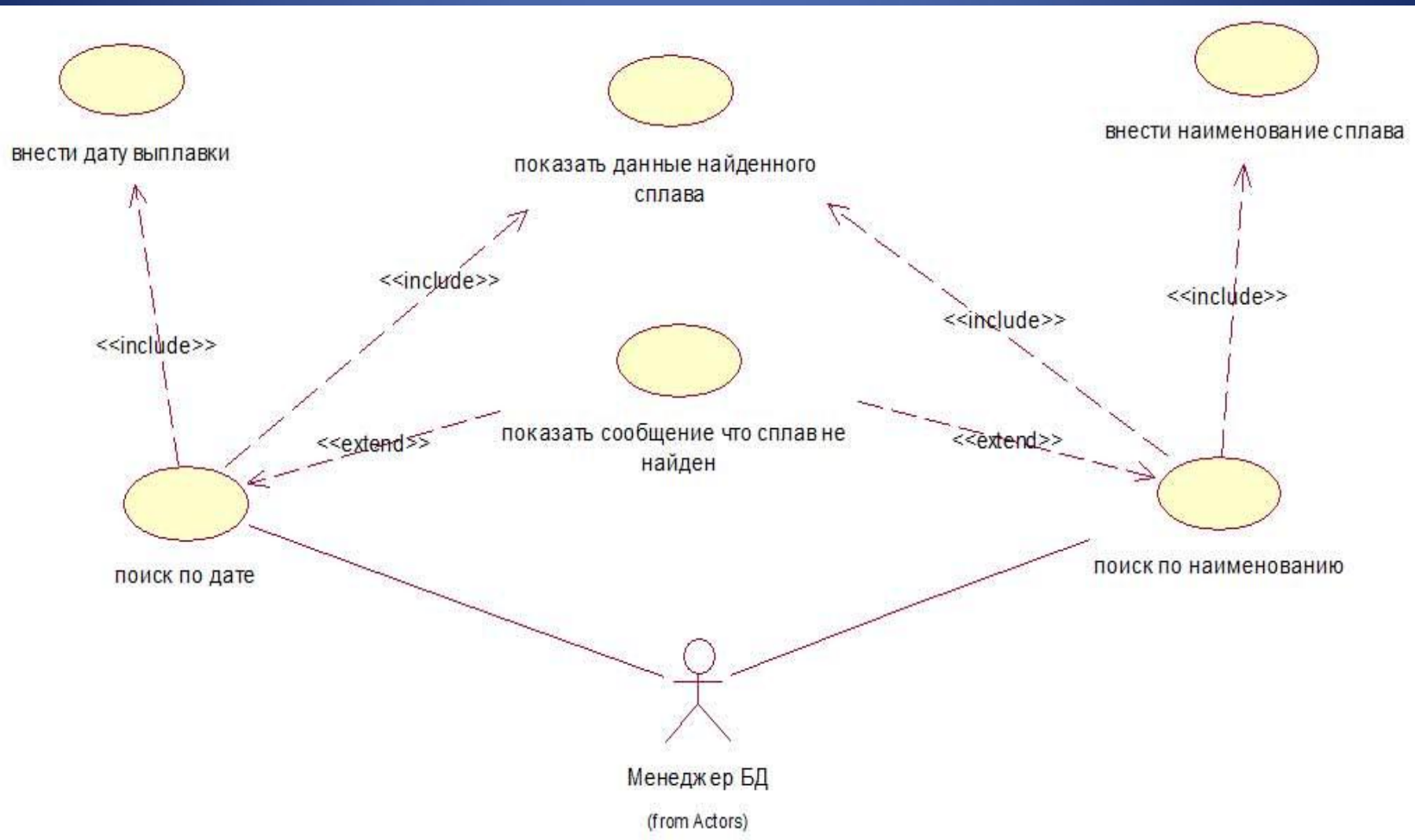
Анализ требований

Построение модели вариантов использования



Анализ требований

Построение модели вариантов использования



Анализ требований

Сценарии

4. Написание сценариев для базовых ВИ

Для базовых ВИ создаются прототипы ЭФ и пишутся сценарии.

Сценарий — это текстовое описание потока событий при выполнении конкретного варианта использования, выражающее некий аспект поведения системы.

Сценарии служат для перехода от вариантов использования к объектам программной системы. Анализируются имена-существительные в тексте сценария. Некоторые из них будут действующими лицами, другие — объектами, а третьи — атрибутами объекта.

Для текстового описания сценария в RUP имеется специальный шаблон. Каждый сценарий начинается главного раздела, далее описываются типовой и альтернативные потоки событий, для актеров – физических лиц должны быть предусмотрены ошибочные потоки событий.

Сценарии с прототипами ЭФ являются первой действующей моделью ПС, которую необходимо согласовать с Заказчиком.

Анализ требований

Прототип ЭФ

СПИСОК МК

Наименование

Зад.вес % сод.

4	5	6

Ввод наименования МК

Ввод заданного веса

1

2

3

Добавить МК
в список

Анализ требований

Сценарии. Главный раздел

Наименование ВИ	Ввод списка МК
Актеры	Технолог шихтовки
Цель исполнения	Ввод списка МК, входящих в сплав и задание их пропорции
Краткое описание	Технолог шихтовки вводит последовательно наименование каждого МК и его заданный вес, затем помещает его в список МК, система по введенным заданным весам рассчитывает процентное содержание каждого МК в сплаве
Тип	Базовый
Начальные условия	На мониторе технолога шихтовки отображается окно ввода начальных данных

Далее, после описания главного раздела описывается типовой поток событий, результат которого и является целью выполнения данного варианта использования.

Анализ требований

Сценарии. Типовой поток событий



Типовой поток событий сценария ВИ "Ввод списка МК"

Действия актеров

Отклик системы

1. Технолог шихтовки вводит в поле 1 экранной формы (ЭФ1, «Ввод данных») наименование МК;
2. Технолог шихтовки вводит в поле 2 ЭФ1 ввода данных заданный вес данного МК;
3. Технолог шихтовки нажимает клавишу 3 (Добавить МК в список) ЭФ ввода данных

4. ПС рассчитает % содержание данного МК и покажет в поле 6 списка МК;
5. Наименование МК переместится в поле 4, заданный вес в поле 5 списка МК,
6. Поля 1 и 2 очистятся

Примечание 1: Действия 1 – 6 повторяются по количеству МК

Конечные условия:

На ЭФ1 заполнены поля списка МК 4, 5, 6. По каждому МК, предназначенному для шихтовки, показываются наименование, заданный вес и % содержание в сплаве

Диаграммы системного анализа

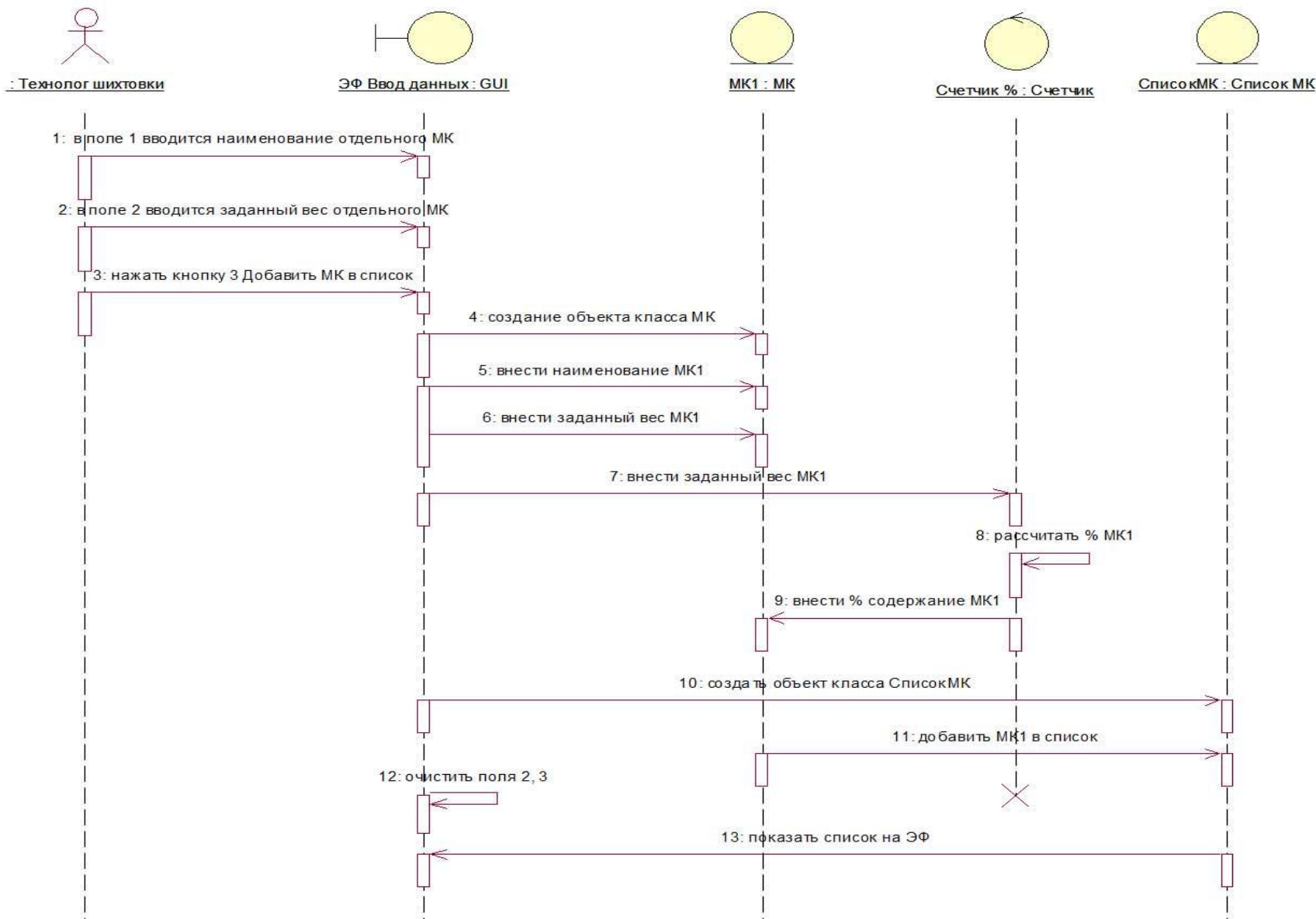
Диаграмма последовательности (sequence diagram) описывает временную последовательность обмена сообщений между объектами ПС одном из потоков событий варианта использования. По сути это последовательность вызова операций.

Все объекты в системном анализе относятся к классам трех типов, каждый из которых обозначается своим стереотипом:

- **класс сущность** (entity) — описывает объекты ПС, отвечающие за хранение данных (значений атрибутов). Объекты сущности не могут инициировать взаимодействия;
- **граничный класс** (boundary) — описывает те объекты ПС, которые являются посредниками между системой и действующими лицами. В случае, если актер физическое лицо, то с помощью объектов-boundary моделируются экранные формы;
- **класс управления** (control) — описывает объекты ПС, которые управляют взаимодействиями, реализуют математические методы и т.д. В этих объектах главное — это их методы.

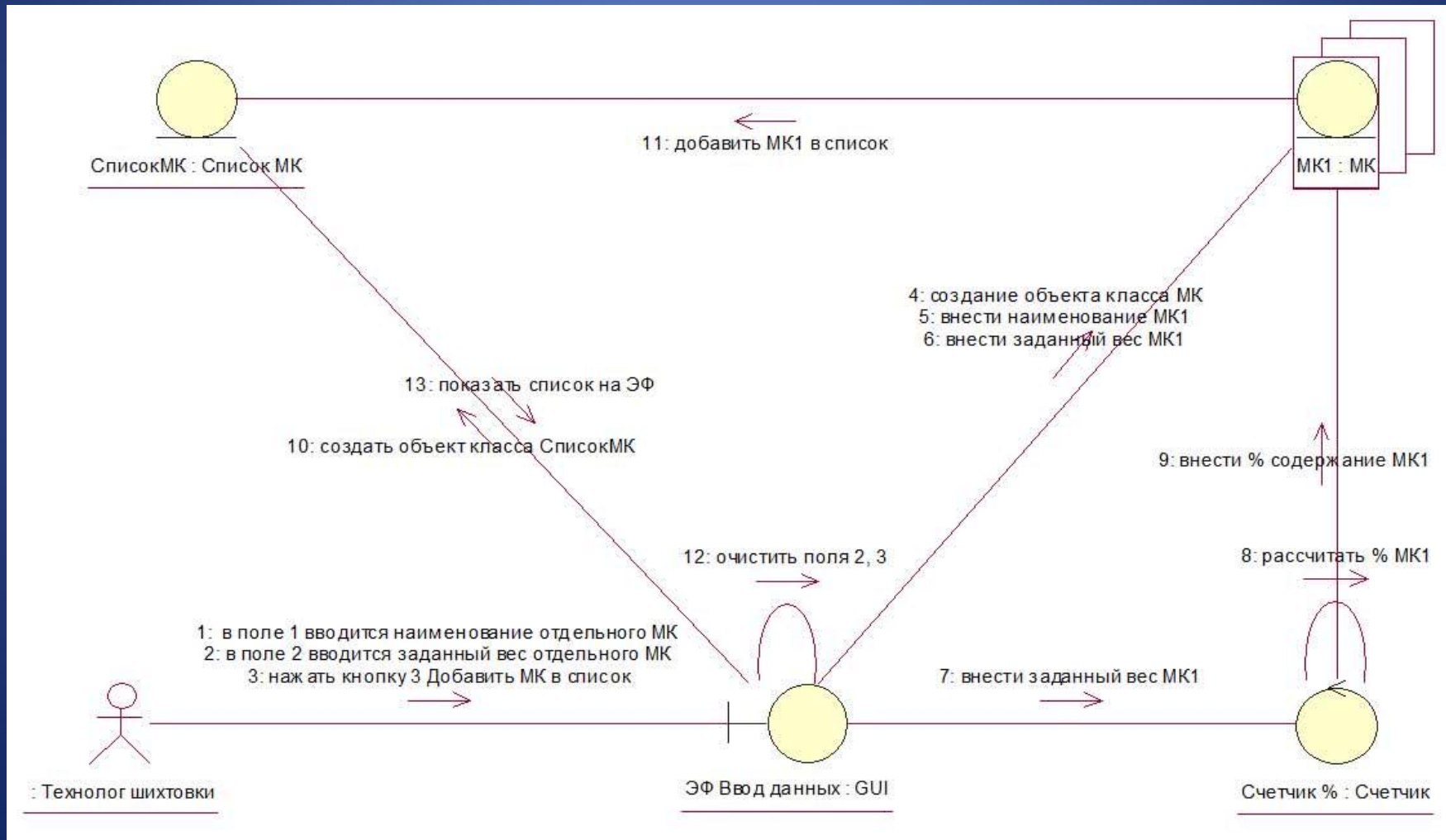
Диаграммы системного анализа

Диаграмма последовательности (sequence diagram)



Диаграммы системного анализа

Кооперативная диаграмма (collaboration diagram) показывает структуру взаимосвязей между объектами программной системы в одном из потоков событий варианта использования.

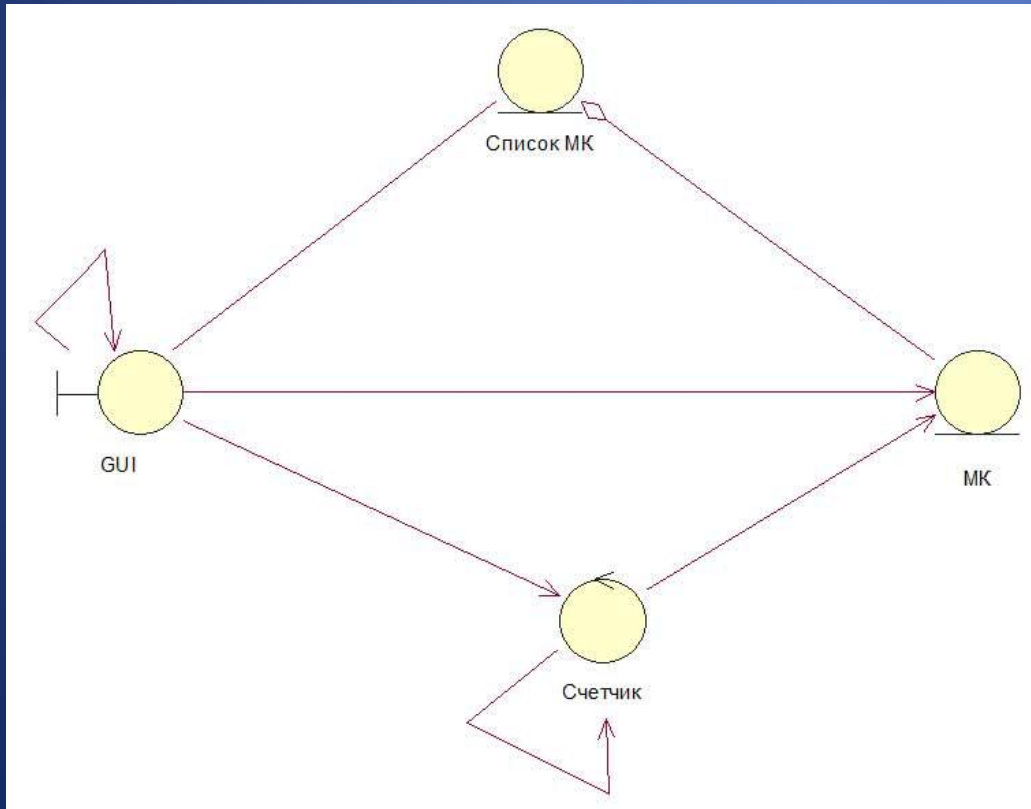


Диаграммы системного анализа

диаграмма классов (class diagram)



Диаграмма классов, отображающая классы и связи между ними для типового потока событий ВИ «Ввод списка МК»



На основе комплекта кооперативных диаграмм (collaboration diagram) строится **диаграмма классов анализа** (class diagram), при этом выявленные связи между объектами и направления послыки сообщений позволят определить связи и их направления на диаграмме классов ПС, а каждое сообщение должно соответствовать методам объекта к которому оно направлено.

Заключение

В данном докладе на примере разработки небольшой промышленной системы показан процесс анализа, выполненный в рамках стандартного RUP, включающий:

- анализ предметной области;
- анализ требований;
- системный анализ.

Строгое следование стандартному процессу не является самоцелью, важнее как можно быстрее получить работающую ПС, поэтому процесс можно изменять и минимизировать количество создаваемых артефактов, оценивая при этом возрастающие риски.

ИСПОЛЬЗУЕМАЯ ЛИТЕРАТУРА

1. Г. Буч и д.р.«Объектно-ориентированный анализ и проектирование с примерами приложений», Москва, «Вильямс», 2008 г., 3-е издание.
2. Дж. Арлоу, А. Нейштадт, «UML-2 и Унифицированный процесс. Практический объектно-ориентированный анализ и проектирование». Санкт-Петербург, «Символ-Плюс», 2007 г., 2-е издание.
3. Основы программной инженерии (по SWEBOK, 2004 г.), перевод С.Орлик, 2004-2010 г.
4. Дж. Рамбо, М. Блаха «UML-2 Объектно-ориентированное моделирование и разработка» Санкт-Петербург, «Питер», 2007 г., 2-е издание.

Спасибо за внимание

Николай Киреев

ИИТ БГУИР

e-Mail: nousy@mail.ru

Skype: nousy123