

Лабораторная работа №4

МОДЕЛИРОВАНИЕ НА ЭТАПЕ АНАЛИЗА ИС ПРИ ОБЪЕКТНО-ОРИЕНТИРОВАННОМ ПОДХОДЕ: ФУНКЦИОНАЛЬНОСТЬ ИС

Цель работы: научиться на основе требований к ИС создавать диаграмму прецедентов UML

Задание.

1. Используя разработанные в лабораторной работе №3 требования к ИС, уточнить действующие лица (акторы) и варианты использования (прецеденты) по индивидуальной теме, привести их описание.
2. Выбрать CASE-средство для построения диаграмм UML (**на выбор: StarUML, BOUML, Rational Rose, Telelogic TAU G2 и другие, которые включают генерирование кода на C ++, Java, Php, Python и MySQL**)
3. Используя выбранное CASE-средство для построения диаграмм UML, разработать диаграмму прецедентов.

КРАТКИЕ ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ

Действующие лица

Вопрос о выделении (или идентификации) действующих лиц при составлении модели - один из самых болезненных. Неудачный выбор действующих лиц может отрицательно повлиять на всю модель в целом. Здесь легко впасть в крайность: объявить, что имеется одно действующее лицо (внешний мир), взаимодействующее со всеми вариантами использования или, наоборот, придумать искусственных действующих лиц для каждого варианта использования. Оба экстремальных варианта являются, по существу, моделью черного ящика и сводят к нулю преимущества моделирования использования, рассмотренные в предыдущем разделе. Формального метода идентификации действующих лиц не существует. Здесь мы перечислим некоторые приемы, которые полезно иметь в виду при выделении действующих лиц и покажем применение этих приемов на нашем примере информационной системы отдела кадров. Для начала укажем более детальное определение действующего лица.

С синтаксической точки зрения действующее лицо - это стереотип классификатора, который обозначается специальным значком. Для действующего лица указывается только имя, идентифицирующее его в системе. Семантически действующее лицо - это множество логически взаимосвязанных ролей.

С прагматической точки зрения главным является то, что действующие лица находятся вне проектируемой системы (или рассматриваемой части системы).

В типовых случаях различные действующие лица назначаются для категорий пользователей (если их удастся выделить естественным образом), внешних программных и аппаратных средств (если система взаимодействует с таковыми).

Выделение вариантов использования - ключ ко всему дальнейшему моделированию. На этом этапе определяется функциональность системы, то есть, что она должна делать.

Нотация для варианта использования очень скудная - это просто имя, помещенное в овал (или помещенное под овалом - такой вариант тоже допустим). Другими словами, функции, выполняемые системой, на уровне моделирования использования никак не раскрываются - им только даются имена.

Семантически вариант использования (use case) - это описание множества возможных последовательностей действий (событий), приводящих к значимому для действующего лица результату.

Каждая конкретная последовательность действий называется сценарием.

Примечания

Третьим типом сущности, применяемым на диаграмме использования, является примечание. Заметим, что примечания являются очень важным средством UML, значение которого часто недооценивается начинающими пользователями. Примечания можно и нужно употреблять на всех типах диаграмм, а не только на диаграммах использования. UML является унифицированным, но никак не универсальным языком - при моделировании проектировщик часто может сказать о моделируемой системе больше, чем это позволяет сделать строгая, но ограниченная нотация UML. В таких случаях наиболее подходящим средством для внесения в модель дополнительной информации является примечание.

В отличие от большинства языков программирования примечания в UML синтаксически оформлены с помощью специальной нотации и выступают на тех же правах, что и остальные сущности. А именно, примечание имеет свою графическую нотацию - прямоугольник с загнутым уголком, в котором находится текст примечания. Примечания могут находиться в отношении соответствия с другими сущностями - эти отношения изображаются пунктирной линией без стрелок. Если пунктирная линия отсутствует, то примечание относится ко всей диаграмме.

Примечания содержат текст, который вводит пользователь - создатель модели. Это может быть текст в произвольном формате: на естественном языке, на языке программирования, на формальном логическом языке, например, OCL и т. д. Более того, если возможности инструмента это позволяют, в примечаниях можно хранить гиперссылки, вложенные файлы и другие артефакты, внешние по отношению к модели.

Примечания могут иметь стереотипы. В UML определены два стандартных стереотипа для примечаний:

- «requirement» - описывает общее требование к системе;
- «responsibility» - описывает ответственность сущности (классификатора).

Примечания первого типа часто присутствуют на диаграммах использования, а примечания второго типа - на диаграммах классов.

Границы системы.

Если посмотреть на модель использования с самой общей точки зрения, то нетрудно заметить, что в модели присутствуют:

- внутренняя моделируемая система, в форме набора вариантов использования, возможно связанных зависимостями и обобщениями;
- внешнее окружение, в форме набора действующих лиц, возможно связанных обобщениями;
- связь между моделируемой системой и внешним окружением в форме ассоциаций между действующими лицами и вариантами использования.

Обычно совершенно ясно, что находится внутри моделируемой системы, а что снаружи. Если это почему-либо неясно, или же требуется увеличить наглядность диаграмм, то можно воспользоваться специальной конструкцией, которая называется "границы системы".

Границы системы (system boundary) - это графический комментарий в форме прямоугольной рамки, применяемый на диаграммах использования и отделяющий внутреннюю часть системы от ее внешнего окружения.

Внутренняя часть, выделяемая границами, имеет в UML конкретное название - субъект.

Субъект (subject) - это классификатор, который реализует поведение, декларируемое вариантами использования.

Диаграммы прецедентов

В соответствии с методологией объектно-ориентированного анализа и проектирования первым этапом является анализ требований, который подразумевает выделение процессов и требований и их формулировку в виде прецедентов.

Прецедент в объектном моделировании (англ. – use case) представляет собой документ, описывающий последовательность событий, связанных с исполнителем (внешним агентом), который для завершения требуемого процесса использует создаваемую систему. Прецеденты являются описанием или вариантами использования системы. С помощью прецедента описывается некоторый процесс.

По результатам анализа прецедентов на первом этапе моделирования предметной области создается диаграмма определения требований к системе Use Case (сценарии поведения). Данная диаграмма позволяет создавать диаграммы поведения объектов системы.

На диаграмме прецедентов иллюстрируется набор прецедентов системы и исполнители, а также взаимосвязи между ними. Прецеденты определяют, как исполнители взаимодействуют с программной системой. В процессе этого взаимодействия исполнителем генерируются события, передаваемые системе, которые представляют собой запросы на выполнение некоторой операции.

Диаграмма прецедентов содержит:

- варианты использования (прецеденты) системы (use case);
- действующее лицо (actor).

Диаграмма отражает взаимодействие вариантов использования и действующих лиц. Она отражает требования к системе с точки зрения пользователя.

Варианты использования системы – описание функций системы на «высоком уровне». Они описывают все, что происходит внутри области действия системы. Варианты использования иллюстрируют, как можно использовать систему. Они заостряют внимание на том, что пользователи хотят получить от системы. Каждый вариант использования представляет собой завершенную транзакцию между пользователем и системой.

Действующее лицо (actor) – все, что взаимодействует с системой, передает или получает информацию от системы.

Исполнитель (actor) является внешним по отношению к системе понятием, которое определенным образом участвует в процессе, описываемом прецедентом. Они описывают все, что находится вне системы. Это пользователи системы, другие системы, взаимодействующие с описываемой, время.

Каждый прецедент должен быть инициирован действующим лицом.

Как правило, отдельные шаги или виды деятельности в виде прецедента не представляются.

Часто для одной системы создается несколько диаграмм вариантов использования. На диаграмме высокого уровня (Main) указываются только пакеты вариантов использования. Другие диаграммы описывают совокупности вариантов использования и действующих лиц.

Цель диаграмм – документирование вариантов использования, действующих лиц и связей между ними.

Разрабатывая диаграммы, придерживаются следующих правил:

1. Не моделируют связи между действующими лицами. По определению они находятся вне сферы действия системы. Связи между ними не относятся к ее компетенции.

2. Не соединяют стрелкой непосредственно два варианта использования (кроме связей использования и расширения). Диаграмма описывает только, какие варианты использования доступны системе, а не порядок их выполнения.

3. Каждый вариант использования должен быть инициирован действующим лицом.

4. Всегда должна быть стрелка, начинающаяся на действующем лице и заканчивающаяся на варианте использования (кроме связей использования и расширения).

5. Думают о БД, как о слое, находящемся под диаграммой. С помощью одного варианта использования можно вводить данные в базу, а получать их – с помощью другого. Не рисуют стрелки от одного варианта использования к другому для изображения потока информации.

6. При создании диаграмм прецедентов вначале определяются исполнители (роли, пользователи).

7. Исполнитель может быть абстрактным, не имеющим экземпляров. Например, есть несколько действующих лиц: служащий с почасовой оплатой, служащий с окладом и т.д. Все они являются разновидностями действующего лица- служащего. Абстрактный исполнитель существует для того, чтобы показать общность между этими типами.

Следующий шаг - идентификация прецедентов.

У каждого прецедента должно быть уникальное имя.

Каждый прецедент должен иметь связанное с ним короткое описание того, что он будет делать. Следует делать описание коротким и к «месту», при этом оно должно определять типы пользователей, выполняющих вариант использования, и ожидаемый ими конечный результат.

Предусловия варианта использования – это такие условия, которые должны быть выполнены, прежде чем вариант начнет свою работу. Например, это может быть выполнение другого варианта использования или наличие у пользователя прав доступа, требуемых для запуска данного варианта использования.

Постусловия – такие условия, которые должны быть выполнены после завершения варианта использования. С помощью постусловий можно вводить сведения о порядке выполнения вариантов использования системы.

Если, скажем, после одного варианта использования должен всегда выполняться другой, это можно описать как постусловие. Такие условия имеются не у каждого варианта использования.

Конкретные детали вариантов использования отражаются в основном и альтернативном потоках событий.

Поток событий поэтапно описывает, что должно происходить во время выполнения заложенной в варианты использования функциональности. Поток событий уделяет внимание тому, что (а не как) будет делать система, причем описывает это с точки зрения пользователя. Первичный и альтернативный потоки событий содержат:

- описание того, каким образом запускается вариант использования,
- различные пути выполнения варианта использования,
- нормальный, или основной, поток событий варианта использования,
- отклонения от основного потока событий (так называемые альтернативные потоки),
- потоки ошибок,
- описание того, каким образом завершается вариант использования.

Поток событий должен быть согласован с определенными ранее требованиями. Избегают детальных обсуждений того, как поток будет реализован.

Можно создать подробную спецификацию для каждого варианта использования. Они помогают документировать такие атрибуты вариантов использования, как имена, приоритеты и стереотипы.

ПОРЯДОК ВЫПОЛНЕНИЯ:

1. Уточнить действующие лица (акторы), выбранные в лабораторной работе №3 и написать их.
2. Уточнить варианты использования (прецеденты), выбранные в лабораторной работе №3 и написать их.
3. Классифицировать прецеденты по категориям (главные, второстепенные, вспомогательные).
4. Описать все прецеденты как высокоуровневые :
 - название;
 - акторы;
 - описание.Для более важных и рискованных прецедентов добавить:
 - основной поток событий (действия акторов и отклик системы, подпотоки);
 - альтернативные потоки;
 - предусловия;
 - постусловия.
5. Построить диаграмму прецедентов (не менее 8 прецедентов), в которой отразить требования к ИС, ее основную функциональность.
6. На диаграмме указать границы ИС.
7. Определить отношения между акторами (обобщение) и проиллюстрировать их на диаграмме.
8. Определить отношения между прецедентами и проиллюстрировать их на диаграмме (ассоциация, обобщение, зависимости включения и расширения).
9. Добавить примечания в диаграмме прецедентов.
10. Оформить отчет.
11. Продемонстрировать работу преподавателю.
12. Ответить на контрольные вопросы.

ПРИМЕР ВЫПОЛНЕНИЯ

(рассматривается ИС «PAsystem» из лабораторной работы №3)

1. Уточненные списки Акторов и прецедентов модели представлены в таблице 6.1.

Таблица 6.1 - Элементы модели функциональных требований

Список Акторов	Список прецедентов	
Пользователь Инструктор Обучающийся Грейдер БД Moodle	Вход в систему Выход из системы Задание сроков отправки и оценивания работ Настройка критериев оценивания Описание критерия Задание баллов оценивания Назначение грейдеров	Случайное распределение Ручное распределение Вычисление итоговых оценок всех обучающихся Экспорт оценок Отправка работы Оценивание работы Просмотр примера Чтение личной итоговой оценки

2. Классифицируем прецеденты по категориям:

– главные:

- 1) вход в систему;
- 2) настройка критериев оценивания;
- 3) отправка работы;
- 4) назначение грейдеров;
- 5) оценивание работы;
- 6) вычисление итоговых оценок всех обучающихся;
- 7) экспорт оценок;
- 8) выход из системы;

– второстепенные:

- 1) задание сроков отправки и оценивания работ;
- 2) чтение личной итоговой оценки;

– вспомогательные.

- 1) просмотр примера.

Прецедент «Задание сроков отправки и оценивания работ» классифицирован как второстепенный в связи с тем, что PAsystem может быть использована обучающимися в демоверсии без ограничений деятельности во времени, чтобы сформировать у них навыки работы с системой.

3. Описываем все прецедент как высокоуровневые (рис. 6.1).

<p><u>Название:</u> Вход в систему.</p> <p><u>Акторы:</u> Пользователь, БД.</p> <p><u>Описание:</u> Пользователь входит в систему с логином и паролем, с которыми он зарегистрирован в LMS Moodle. В случае успешного входа система начинает сеанс работы, назначает Пользователю соответствующие права и сохраняет эту информацию в БД.</p>
<p><u>Название:</u> Выход из системы.</p> <p><u>Акторы:</u> Пользователь, БД.</p> <p><u>Описание:</u> Пользователь явно выходит из системы. Система завершает сеанс работы Пользователя и сохраняет эту информацию в БД.</p>
<p><u>Название:</u> Задание сроков отправки и оценивания работ.</p> <p><u>Акторы:</u> Инструктор, БД.</p> <p><u>Описание:</u> Инструктор задает крайний срок отправки работ обучающимися. Инструктор задает крайний срок завершения оценивания работ грейдерами. Система контролирует, чтобы срок завершения оценивания был позже, чем срок отправки работ. При корректности значений этих параметров система сохраняет их в БД.</p>
<p><u>Название:</u> Настройка критериев оценивания.</p> <p><u>Акторы:</u> Инструктор, БД.</p> <p><u>Описание:</u> Инструктор настраивает критерии парного оценивания работы. Система сохраняет значения настроек в БД.</p>
<p><u>Название:</u> Описание критерия.</p> <p><u>Акторы:</u> Инструктор, БД.</p> <p><u>Описание:</u> Инструктор заполняет для каждого критерия оценивания работы его описание.</p>
<p><u>Название:</u> Задание баллов оценивания.</p> <p><u>Акторы:</u> Инструктор, БД.</p> <p><u>Описание:</u> Инструктор задает баллы оценивания для каждого критерия.</p>
<p><u>Название:</u> Назначение грейдеров.</p> <p><u>Акторы:</u> Инструктор, БД.</p> <p><u>Описание:</u> Инструктор переводит систему в режим распределения грейдеров. Он или система выполняют назначение грейдеров. Выполненное распределение фиксируется в БД.</p>
<p><u>Название:</u> Вычисление итоговых оценок всех обучающихся.</p> <p><u>Акторы:</u> Инструктор, БД.</p> <p><u>Описание:</u> Инструктор переводит систему в режим вычисления итоговых оценок за оценивание работы. Система вычисляет оценку каждого обучающегося на основании оценок, данных им как грейдерами, и полученных от других грейдеров за свою работу. Система сохраняет итоговые оценки всех обучающихся в БД.</p>

Рис. 6.1 – Описание высокоуровневых прецедентов PASystem

<p><u>Название:</u> Экспорт оценок.</p> <p><u>Актеры:</u> Инструктор, Moodle.</p> <p><u>Описание:</u> Инструктор дает команду системе выгрузить консолидированные оценки за работу в платформу LMS. Система экспортирует данные оценки в Moodle и заносит их в журнал оценок курса.</p>
<p><u>Название:</u> Отправка работы.</p> <p><u>Актеры:</u> Обучающийся, БД.</p> <p><u>Описание:</u> Обучающийся выполняет работу и отправляет ее на проверку в LMS Moodle. Система загружает работу, критерии оценивания и данные о студенте из Moodle в БД.</p>
<p><u>Название:</u> Оценивание работы.</p> <p><u>Актеры:</u> Грейдер, БД.</p> <p><u>Описание:</u> Грейдер выбирает работу другого обучающегося для оценивания. Работа и критерии оценивания выгружаются из БД и направляются этому Грейдеру. Грейдер изучает критерии оценивания работы. Грейдер читает работу обучающегося. Грейдер при помощи критериев оценки выставляет оценку данной работе по каждому из предложенных критериев и завершает оценивание. Система вычисляет оценку этого Грейдера за работу обучающегося и сохраняет ее в БД.</p>
<p><u>Название:</u> Просмотр примера.</p> <p><u>Актеры:</u> Грейдер, БД.</p> <p><u>Описание:</u> Грейдер не знает, как применить предложенные критерии при оценивании назначенной ему работы. Он запрашивает помощь в виде примера оценивания. Система показывает Грейдеру пример.</p>
<p><u>Название:</u> Чтение личной итоговой оценки.</p> <p><u>Актеры:</u> Грейдер, Moodle.</p> <p><u>Описание:</u> Грейдер читает свою итоговую оценку (за выполнение работы и парного оценивания) в журнале оценок LMS Moodle.</p>

Окончание рис. 6.1

4. Выделяем наиболее важные и рискованные прецеденты и записываем их в развернутой идеальной форме (рис. 6.2)

<u>Название:</u> Настройка критериев оценивания. <u>Акторы:</u> Инструктор, БД. <u>Описание:</u> Инструктор настраивает критерии парного оценивания работы. Система сохраняет значения настроек в БД. <u>Триггер:</u> Инструктор выбирает вариант количественного парного оценивания. <u>Основной поток:</u>	
Действия акторов	Отклик системы
1. Инструктор выбирает вариант количественного парного оценивания	2. Предоставляет Инструктору инструмент для создания рубрик
3. Инструктор создает рубрику и заполняет все критерии оценивания. При настройке атрибута критерия «Описание» выполняется подпоток S1. При настройке атрибута критерия «Балл» выполняется подпоток S2. Инструктор сохраняет рубрику	4. Прикрепляет только что созданную рубрику к элементу парного оценивания работы и сохраняет в БД
<u>Подпотoki:</u> S1: Описание критерия. Инструктор заполняет для каждого критерия оценивания работы его описание. Система отображает описание критерия. Прецедент продолжается. S2: Задание баллов оценивания. Инструктор задает баллы оценивания для каждого критерия. Система отображает баллы. Прецедент продолжается.	
<u>Название:</u> Назначение грейдеров. <u>Акторы:</u> Инструктор, БД. <u>Описание:</u> Инструктор переводит систему в режим распределения грейдеров. Он или система выполняют назначение грейдеров. Выполненное распределение фиксируется в БД. <u>Триггер:</u> Инструктор переводит систему в режим распределения грейдеров.	
<u>Основной поток:</u>	
Действия акторов	Отклик системы
1. Инструктор анализирует вид оцениваемой работы (статус задания), выполненной обучающимися. При выполнении индивидуальных работ выполняется подпоток S3. При выполнении командных работ выполняется подпоток S4. Инструктор подтверждает распределение грейдеров	2. Сохраняет распределение грейдеров этой работы в БД
<u>Подпотoki:</u> S3: Случайное распределение. Инструктор указывает количество грейдеров каждому из обучающихся. Система по определенному алгоритму назначает уникальных грейдеров для обучающихся. Прецедент продолжается. S4: Ручное распределение. Система отображает Инструктору список обучающихся и список потенциальных грейдеров. Инструктор выбирает обучающегося и назначает ему необходимое количество уникальных грейдеров. Прецедент продолжается.	

Рис. 6.2 – Описание прецедентов ИС PASystem в развернутой форме

<p><u>Название:</u> Оценивание работы.</p> <p><u>Акторы:</u> Грейдер, БД.</p> <p><u>Описание:</u> Грейдер выбирает работу другого обучающегося для оценивания. Работа и критерии оценивания выгружаются из БД и направляются этому Грейдеру. Грейдер изучает критерии оценивания работы. Грейдер читает работу обучающегося. Грейдер при помощи критериев оценки выставляет оценку данной работе по каждому из предложенных критериев и завершает оценивание. Система вычисляет оценку этого Грейдера за работу обучающегося и сохраняет ее в БД.</p> <p><u>Триггер:</u> Грейдер выбирает работу обучающегося для оценивания.</p> <p><u>Основной поток:</u></p>	
Действия акторов	Отклик системы
1. Грейдер выбирает работу обучающегося	2. Работа и критерии оценивания выгружаются из БД и направляются Грейдеру
3. Грейдер изучает критерии оценивания работы. Грейдер читает работу обучающегося. Грейдер при помощи критериев оценки выставляет оценку данной работе по каждому из предложенных критериев (E1) и завершает оценивание (E2)	4. Система вычисляет оценку Грейдера за работу обучающегося и сохраняет ее в БД
<p><u>Альтернативные потоки:</u></p> <p>E1: Просмотр примера. При невозможности применить предложенные критерии при оценивании назначенной ему работы Грейдер запрашивает помощь в виде примера оценивания. Система показывает Грейдеру пример. Прецедент продолжается.</p> <p>E2. Незавершенное оценивание. Грейдер не оценивает работу по каким-либо причинам. Прецедент завершается</p>	

<p><u>Название:</u> Вычисление итоговых оценок всех обучающихся.</p> <p><u>Акторы:</u> Инструктор, БД.</p> <p><u>Описание:</u> Инструктор переводит систему в режим вычисления итоговых оценок за оценивание работы. Система вычисляет оценку каждого обучающегося на основании оценок, данных им как грейдером, и полученных от других грейдеров за свою работу. Система сохраняет итоговые оценки всех обучающихся в БД.</p> <p><u>Триггер:</u> Инструктор переводит систему в режим вычисления итоговых оценок за оценивание работы.</p> <p><u>Основной поток:</u></p>	
Действия акторов	Отклик системы
1. Инструктор анализирует состояние выполнения парного оценивания (E3). При наступлении даты окончания оценивания Инструктор инициирует вычисление итоговых оценок за парное оценивание	2. Система вычисляет оценку каждого обучающегося на основании оценок, данных им как грейдером, и полученных от других грейдеров за свою работу, и отображает результаты
3. Инструктор подтверждает завершение парного оценивания работы	4. Система сохраняет итоговые оценки всех обучающихся в БД
<p><u>Альтернативные потоки:</u></p> <p>E3: Частичное оценивание. Не все Грейдеры выполнили оценивание. Инструктор продлевает срок завершения оценивания. Прецедент завершается</p>	

5. Строим диаграмму прецедентов.

Предварительная диаграмма прецедентов приведена на рисунке 6.3. Граница показывает, что в рамках дисциплины «МИСиТ» моделированием и проектированием БД информационной системы студенты не занимаются. На диаграмме прецеденты размещены по вертикали в порядке их возникновения в ИС.

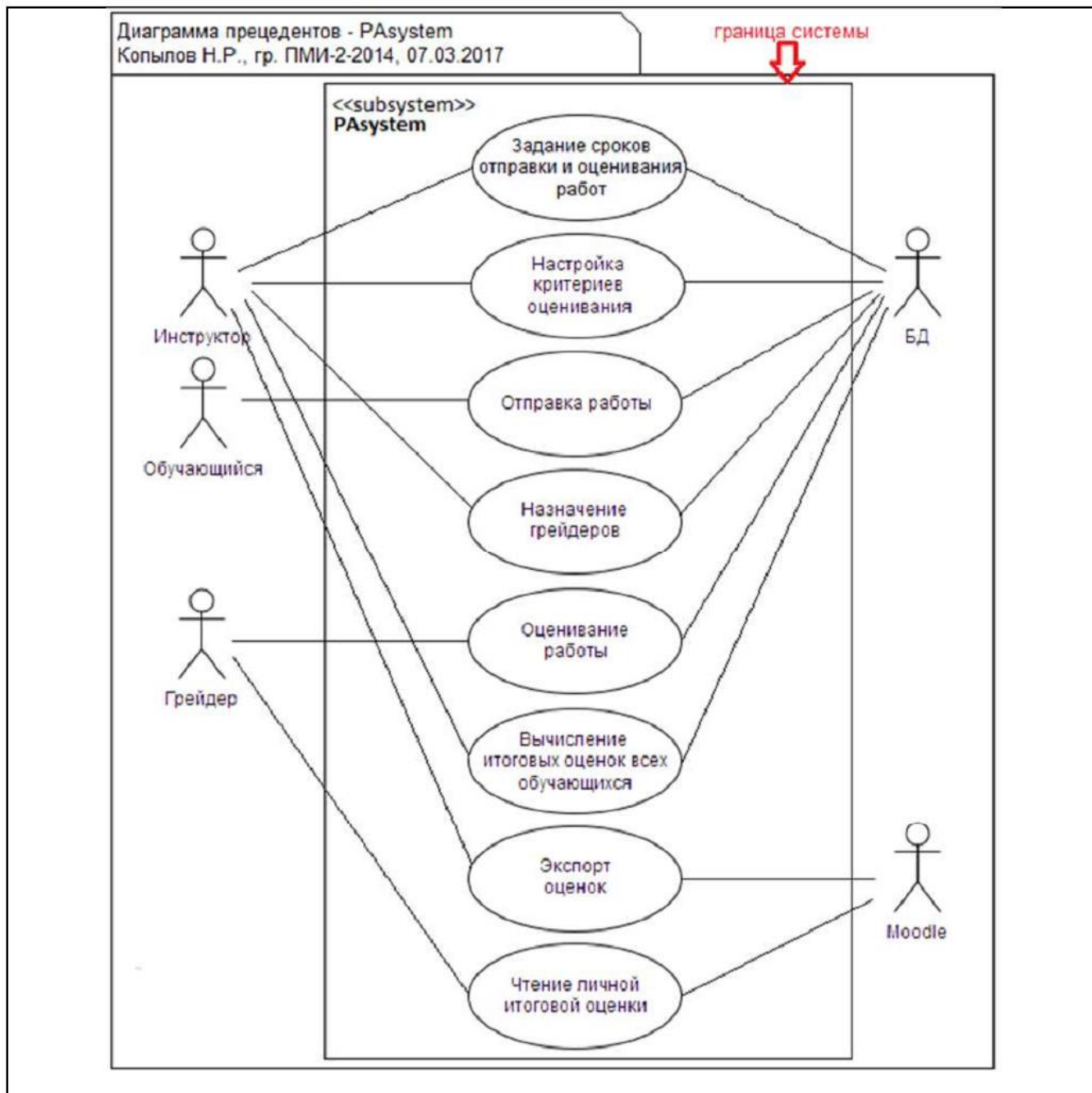


Рис. 6.3 – Предварительная диаграмма прецедентов ИС PASystem

Альтернативным вариантом является создание отдельных диаграмм прецедентов для каждого из Акторов, что иллюстрируют рисунки 6.4–6.5.

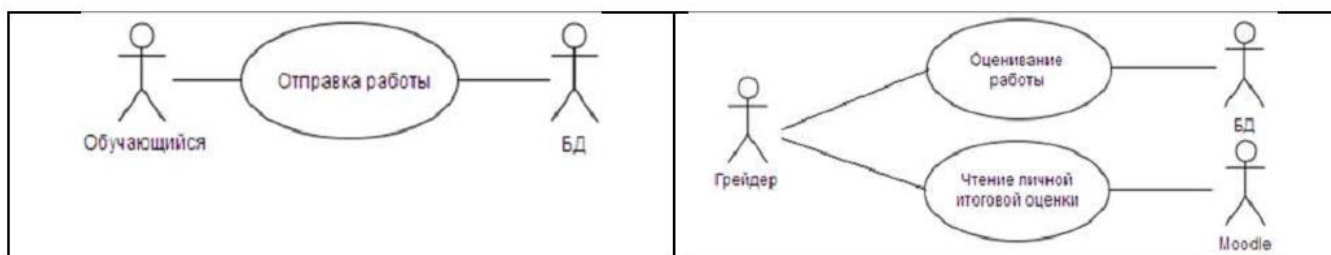


Рис. 6.4 – Диаграмма прецедентов ИС PASystem для Обучающегося и Грейдера

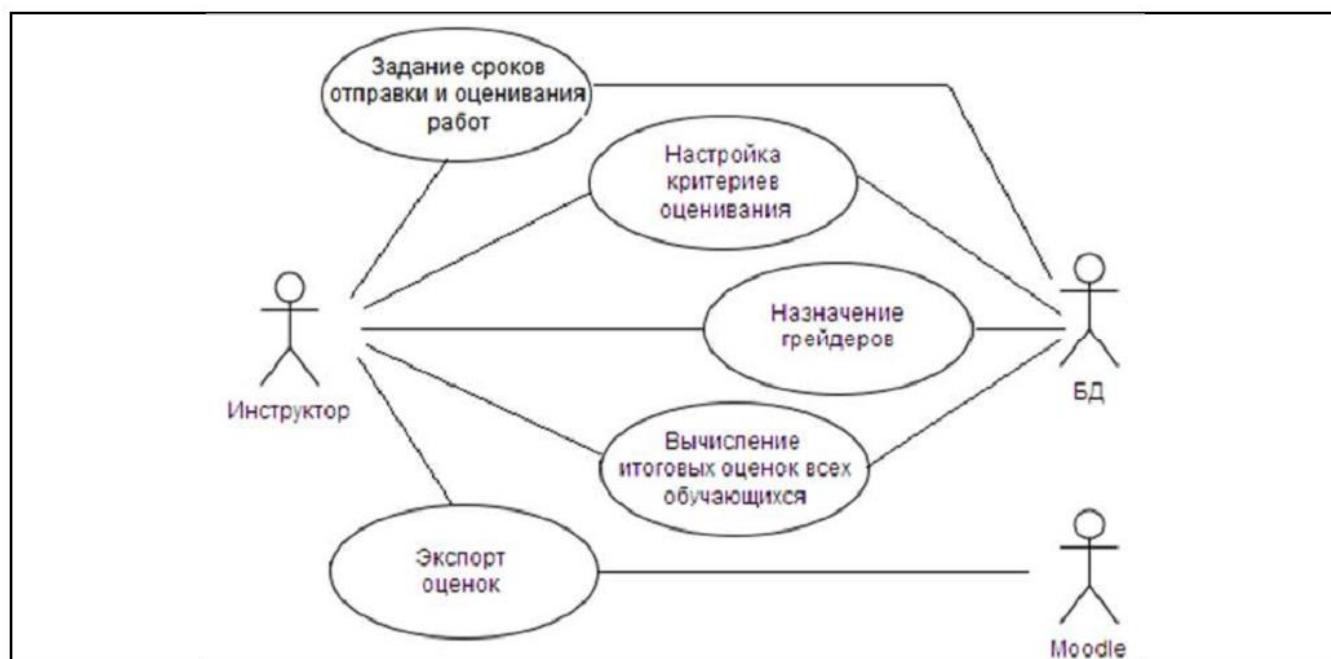


Рис. 6.5 – Диаграмма прецедентов ИС PASystem для Инструктора

7. Определим отношения между акторами (обобщение) и проиллюстрируем их на диаграмме.

7.1. Уточним набор прецедентов ИС PAsystem. Так как разрабатываемая система является веб-приложением с требованиями о безопасности персональных данных пользователей, необходимо кроме основных функциональных требований предусмотреть дополнительно прецеденты «Вход в систему» и «Выход из системы».

7.2. С прецедентами «Вход в систему» должен быть ассоциирован еще один Актор системы – Пользователь, наличие которого не было рассмотрено ранее. После входа в систему выполняется распределение соответствующих ролей пользователям.

7.3. Между Акторами в ИС PAsystem существуют отношения обобщения, которые иллюстрирует рис. 6.6,а. Участие Актора Пользователь в прецедентах иллюстрирует рис. 6.6,б.

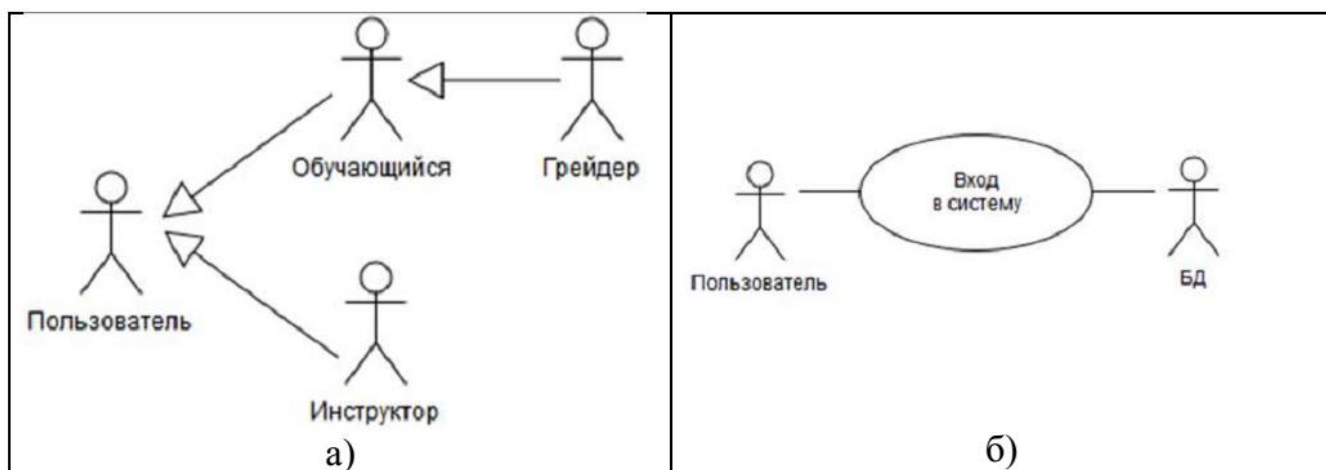


Рис. 6.6. – Отношения между Акторами и прецеденты Пользователя

Чтобы уточнить правильность отношения обобщения между Акторами, проверяем участие Акторов-потомков в прецедентах Актора-родителя:

- входить в систему должны Акторы Инструктор и Обучающийся,
- Актор Грейдер перед оцениванием назначенных ему работ должен отправить свою работу как Обучающийся.

Таким образом, отношение обобщения между Акторами построено корректно.

8. Определим отношения между прецедентами и проиллюстрируем их на диаграмме (ассоциация, обобщение, зависимости включения и расширения).

8.1. Уточним набор прецедентов, связанный с Актором Инструктор. В прецеденте «Назначение грейдеров» возможны варианты случайного и ручного распределения. Первый из них можно применить при оценивании индивидуальных работ, второй – работ, выполненных в группе. Прецедент «Назначение грейдеров» является родительским для прецедентов «Случайное распределение» и «Ручное распределение», так как они используют общую схему взаимодействия Актора Инструктор с ИС, и различаются только в конкретных шагах. Описанное отношение обобщения между прецедентами представлено на рисунке 6.7, а.

8.2. Прецедент «Настройка критериев оценивания» предусматривает выполнение Актором Инструктор таких действий как подробное описание критерия, по которым производится оценивание, и указание баллов, которые показывают степень соответствия данной работы конкретному критерию.

Значит, в модели ИС должны быть прецеденты «Описание критерия» и «Задание баллов оценивания», описывающие деятельность Инструктора при работе с каждым из критериев оценивания. Между прецедентами, рассмотренными в этом пункте, существует зависимость включения, показанная в модели на рисунке 6.7, б, так как базовый вариант «Настройка критериев оценивания» зависит только от результатов выполнения включаемых в него вариантов использования «Описание критерия» и «Задание баллов оценивания», но не от структуры включаемых в него вариантов.



Рис. 6.7 – Отношение обобщения и зависимость включения между прецедентами в ИС PASystem

8.3. При оценивании работы, назначенной Актору Грейдеру, ему может потребоваться помощь, которую он получит при просмотре примера выполнения рецензии. Поэтому между вариантами использования «Оценивание работы» и «Просмотр примера» существует зависимость расширения с точкой расширения «Example», представленная на рисунке 6.8. В этом случае расширяемый прецедент «Оценивание работы» выполняет последовательность действий, которая определяет его поведение. Так как у него имеется точка расширения «Example» на экземпляр другого варианта использования «Просмотр примера», то проверяется логическое условие данного отношения (Grader selected HELP). Если это условие выполняется, исходная последовательность действий по оцениванию работы расширяется посредством включения действий просмотра примера рецензии

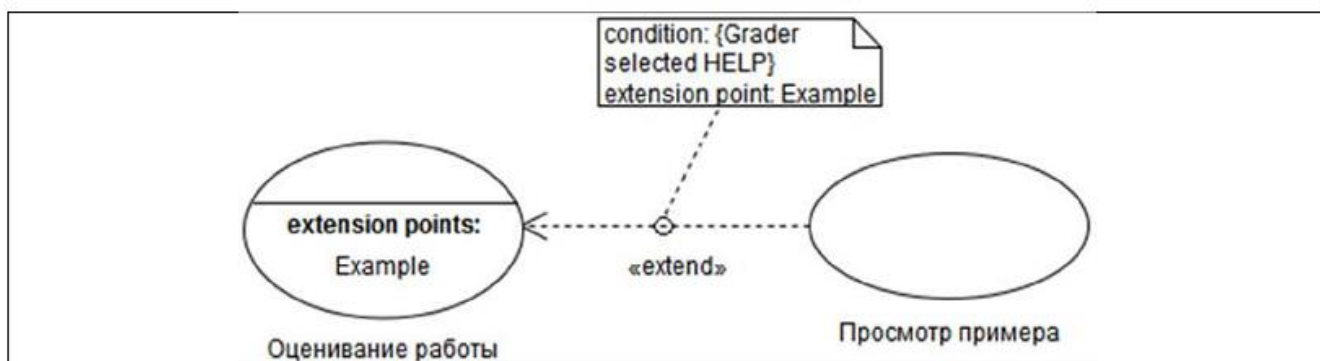


Рис. 6.8 – Зависимость расширения между прецедентами в ИС PASystem

8.4. Названия стереотипов зависимостей включения и расширения могут быть записаны как по-русски, так и по-английски в зависимости от нотации, допустимой в соответствующем инструменте для создания Use Case Diagram.

9. Итоговая модель функциональности на этапе анализа ИС представлена на рисунке 6.9.

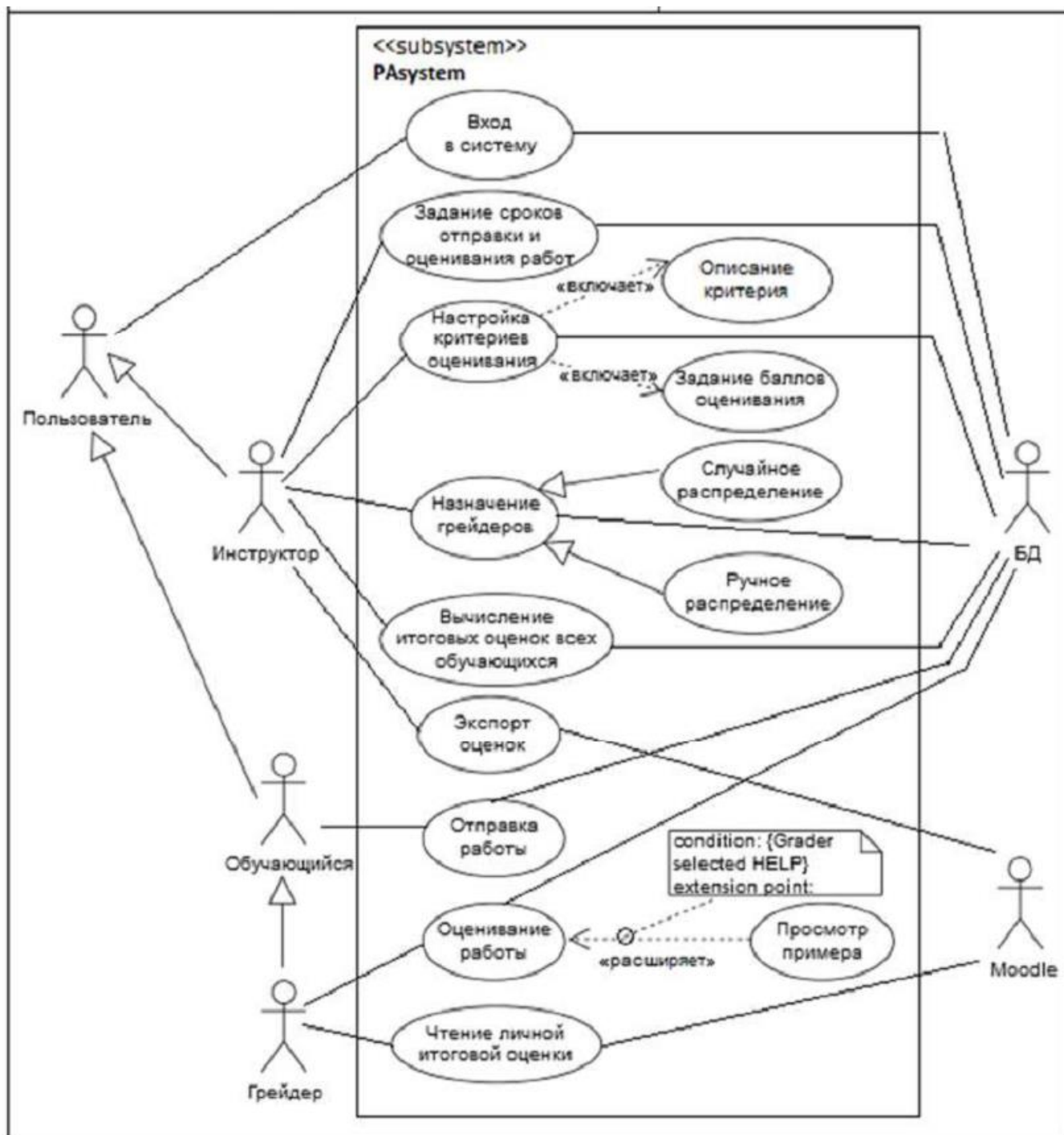


Рис. 6.9 – Итоговая модель функциональности на этапе анализа ИС «PAsystem»

КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Что понимается под прецедентом в объектном моделировании?
2. Что понимается под актором в объектном моделировании?
3. Для чего строится диаграмма прецедентов?
4. Перечислите основные элементы диаграммы прецедентов.
5. Какие типы связей поддерживаются в диаграммах прецедентов?