Лекция 7

Системное моделирование вносит дополнительную формальность в процессы анализа и проектирования. В процессе разработки системы очень часто используются схемы и рисунки, которые помогают наглядно отобразить некоторые аспекты разработки. Системное моделирование формализует это наглядное представление не только с помощью диаграмм, выполненных с использованием стандартных нотаций (синтаксиса), но и обеспечивает среду (средства) для понимания и обсуждения идей, связанных с процессом разработки.

Можно смело утверждать, что искусство моделирования является самой творческой

частью работы системного инженера.

На практике не существует единственного «правильного» решения, поэтому модели меняются и развиваются на протяжении этапов разработки. В большинстве случаев, модели представляют собой некий визуальный ряд, в котором для отображения информации используются взаимосвязанные диаграммы. Новые методы, - такие как, например, объектно-ориентированные методы моделирования, - разумеется, расширяют концепцию моделирования, однако большинство используемых в них подходов, тем не менее, базируются на известных и проверенных временем принципах.

Хорошая модель - это модель, помогающая общению. Модели нужны, для того чтобы можно было обсуждать идеи и внутри команды разработчиков, и во всей организации в целом, подключая к этому процессу и другие заинтересованные стороны. Модели могут применяться для различных целей и покрывать самые разнообразные аспекты разработки системы. Так, например, одна модель может описывать общую структуру взаимодействия внутри всей организации, а другая – отображать всего лишь одно конкретное функциональное требование к этой системе.

Моделирование имеет следующие преимущества:

* Поощряет использование точно определенной терминологии, однозначность которой поддерживается в рамках разработки всей системы.
* Позволяет с помощью диаграмм получить наглядное представление системных спецификаций и архитектуры системы.
* Позволяет рассматривать различные аспекты взаимодействия системы с различных точек зрения.
* Поддерживает системный анализ.
* Позволяет подтвердить достоверность некоторых аспектов поведения системы с помощью динамических моделей.
* Позволяет постоянно совершенствовать систему посредством уточнения архитектуры, поддерживая генерацию тестов и исходного кода.
* Позволяет свободно общаться различным организациям между собой, используя стандартные нотации.

Моделирование позволяет системному инженеру в большей мере проявить свои творческие способности. Данная глава рассказывает о системном моделировании и описывает некоторые методы разработки требований, в которых используется моделирование.

Изображение выглядит как текст, снимок экрана, диаграмма, линия

Контент, сгенерированный ИИ, может содержать ошибки.

Два вопроса:

Что будет делать приложение?

Кто будет пользоваться этим приложением?

Диаграммы использования (или диаграммы прецедентов, англ. Use Case Diagrams) в UML (Unified Modeling Language) Это один из видов диаграмм в UML, который используется для отображения взаимодействий между пользователями (или "актерами") и системой, а также для иллюстрации функциональных требований к системе.

Для того чтобы описать различные группы пользователей и их возможности в будущей программе, создаётся так называемая диаграмма вариантов использования.

Диаграмма вариантов использования

Диаграмма вариантов использования (англ. use-case diagram) – диаграмма, описывающая, какой функционал разрабатываемой программной системы доступен каждой группе пользователей.

В качестве системы выберем информационную систему для школы (можно рассматривать ее как сайт или как отдельное приложение). Пример, разумеется, демонстрационный и не претендует на законченность.

В этой системе можно выделить следующие группы пользователей:

* Обучающиеся
* Преподаватели
* Классные руководители
* Заместители директора

Заместители директора есть, а где же сам директор?

Каждая из групп пользователей может пользоваться нашей системой по-своему.

Обучающиеся могут:

* Смотреть расписание
* Просматривать свои оценки

Преподаватели могут:

* Размещать материалы для уроков
* Выставлять оценки в электронный журнал

Классные руководители могут делать все то же самое, что и преподаватели плюс:

* Составлять расписание родительских собраний

Заместители директора могут:

* Составлять расписание
* Публиковать посты с важной информацией

Кроме того, у системы есть функционал, который доступен всем группам пользователей. В разрабатываемой нами системе актуально будет добавить мессенджер, в котором можно будет быстро связываться с интересующим человеком. Получается, эта функциональность должна быть доступна каждому пользователю.

Значит Все пользователи могут:

* Отправлять сообщения

Построение диаграммы

Каждая группа пользователей на *диаграмме вариантов использования* обозначается человечком, под которым записывается имя группы людей, которую он обозначает. Давайте изобразим группу пользователей "Преподаватели":

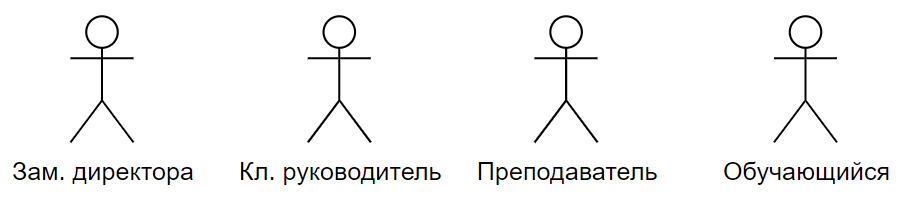


Этот человечек обозначает всех преподавателей, которые будут пользоваться системой

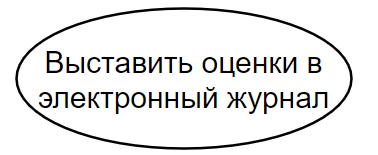
Обратите внимание, что имя группы записывается в единственном числе. Символ человечка уже обозначает *группу* пользователей, поэтому не нужно дополнительно отражать это в имени.

В терминологии UML, этот человечек называется актором (англ. "actor"). В общем случае, актор обозначает любые сущности, использующие систему. Этими сущностями могут быть люди, технические устройства или даже другие системы.

Так же изобразим актёров для оставшихся групп пользователей:

Здесь изображены все группы пользователей, которые могут пользоваться нашей системой

На *диаграмме вариантов использования* функция системы изображается эллипсом, внутри которого записывается имя функции в **форме глагола с пояснительными словами.**



Этот эллипс представляет действие "Выставить оценки в электронный журнал"

В терминологии UML, этот эллипс называется *вариантом использования* (англ. "use-case").

В общем случае, вариант использования – набор действий, который может быть использован актёром для взаимодействия с системой.

Связи между элементами

На диаграммах UML для связывания элементов используются различные соединительные линии, которые называются *отношениями*. Каждое такое отношение имеет собственное название и используется для достижения определённой цели.

Виды отношений:

Отношение ассоциации (англ. "association relationship")

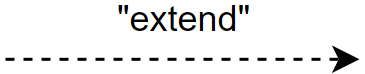
Отношение ассоциации (англ. "association relationship")

Отношение обобщения (англ. "generalization relationship")

Отношение обобщения (англ. "generalization relationship")

Отношение включения (англ. "include relationship")

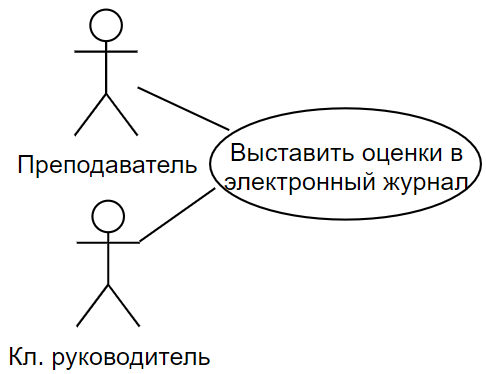
Отношение включения (англ. "include relationship")



Отношение расширения (англ. "extend relationship")

Отношение ассоциации

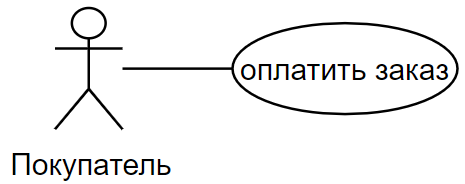
Мы хотим отображать на диаграмме информацию о том, какие варианты использования могут быть использованы каждым актёром. Покажем, что выставлять оценки могут только преподаватели.



Изображаем на диаграмме информацию о том, что преподаватели могут выставлять оценки

Здесь соединили актеров с вариантом использования с помощью сплошной линии без стрелки. Такая линия называется отношением ассоциации.

Отношение ассоциации предназначено **только** для соединения актёров и вариантов использования.

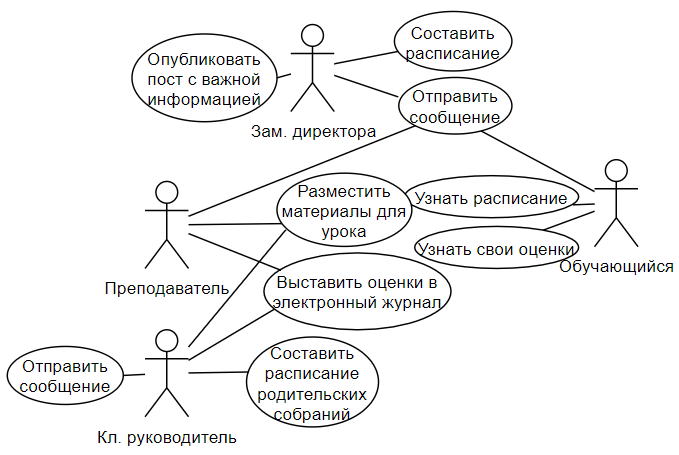


На диаграмме возможность покупателей оплачивать заказы

Если на диаграмме вариантов использования актёр соединен с вариантом использования с помощью *отношения ассоциации*, это означает, что данный актёр может выполнять действия, описанные вариантом использования.

Почему отношение ассоциации называется так и не иначе?

Добавим еще вариантов использования и соединим их с соответствующими актёрами:



Первая версия диаграммы

Отношение обобщения

Заметим, что в нашей системе группы пользователей «Преподаватель» и «Классный руководитель» обладают схожими возможностями. Чтобы изобразить это на диаграмме, мы можем пойти одним из трёх путей:

1. Дублировать варианты использования, чтобы связать их с каждым схожим актёром (очевидно, неудачный вариант).
2. Соединить каждого актёра со всеми нужными вариантами использования. Это может породить множество пересечений линий, что не самым лучшим образом скажется на читаемости диаграммы.
3. Показать с помощью одного из видов отношений, что актёры связаны между собой. Это будет означать, что один из них может пользоваться всеми вариантами использования, с которыми соединён другой актёр.

Последний вариант похож на принцип повторного использования кода при написании программ или на наследование классов в ООП (Объектно-ориентированное программирование). Преимущество этого варианта в том, чтобы уменьшить количество связей на диаграмме.

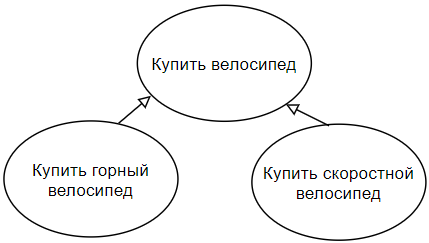
Разумеется, мы воспользуемся третьим путём. В этом нам поможет, так называемое, *отношение обобщения*. Отношение обобщения обозначается сплошной линией с полой треугольной стрелкой.

Изображение выглядит как Шрифт, текст, белый, дизайн

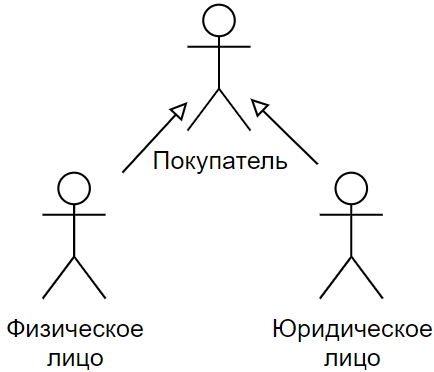
Контент, сгенерированный ИИ, может содержать ошибки.

Отношение обобщения означает, что некоторый актёр (вариант использования) может быть *обобщён* до другого актёра (варианта использования).  Стрелка направлена  от частного случая(специализации) к общему случаю.

Ниже представлены несколько примеров использования отношения обобщения.



Покупка горного и скоростного велосипеда - ЧАСТНЫЙ случай покупки велосипеда



Физическое лицо и юридическое лицо можно ОБОБЩИТЬ до обычного покупателя

Как можно заметить, отношение обобщения используется, чтобы показать, что одно действие является *частным случаем* другого действия или что одну группу людей можно *обобщить*до другой группы.

Вернёмся к нашему основному примеру. Изобразим отношение обобщения от актёра "Кл. руководитель" к актёру "Преподаватель".

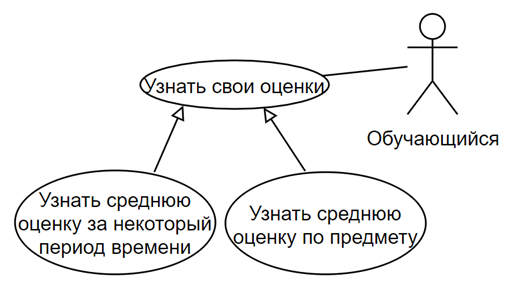
Изображение выглядит как текст, диаграмма, Шрифт, рисунок

Контент, сгенерированный ИИ, может содержать ошибки.

На рисунке сверху сразу видно, насколько понятнее становится диаграмма при использовании отношения обобщения: исчезли все повторы вариантов использования и пересечения линий. Разумеется, это огромный плюс для тех, кто будет читать эту диаграмму в дальнейшем.

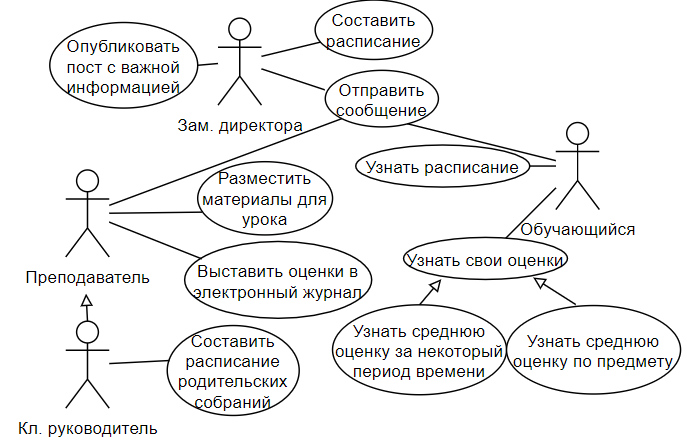
Далее: обратим внимание на действие «Узнать свои оценки». Логично предположить, что обучающиеся захотят не только знать список своих оценок, но и знать свою среднюю оценку за некоторый период времени или среднюю оценку по определённому предмету.

Изобразим это на диаграмме. Для этого создадим два варианта использования "Узнать среднюю оценку за некоторый период времени" и "Узнать среднюю оценку по предмету" и соединим их с *вариантом использования "Узнать свои оценки"* отношением обобщения.



Уточняем на диаграмме, что у обучающихся есть возможность узнать среднюю оценку за некоторый период времени и средний балл по некоторому предмету

Присоединим это к основной диаграмме:

Вторая версия диаграммы

Отношение включения

Для заместителя директора мы отмечали, что ему нужно составлять расписания. *Условно* расписание можно поделить на три категории:

1. Расписание занятий
2. Расписание мероприятий
3. Расписание каникул

Всё это составляется заместителем директора, поэтому покажем это на диаграмме. Для этого будем использовать *отношение включения.*Отношение включения обозначается пунктирной линией с V-образной стрелкой на конце, над стрелкой добавляется надпись “include”.

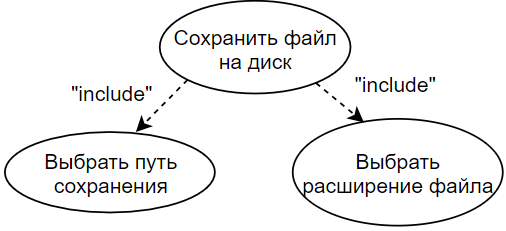
Изображение выглядит как текст, Шрифт, алгебра, дизайн

Контент, сгенерированный ИИ, может содержать ошибки.

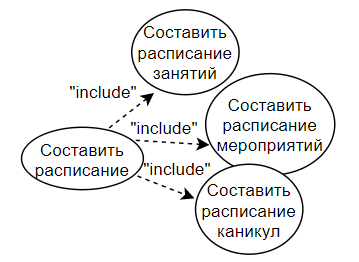
В общем случае, отношение включения используется, чтобы показать, что некоторый вариант использования *включает* в себя другой вариант использования в качестве составной части.

Когда мы используем отношение включения, мы подразумеваем, что составные варианты использования ОБЯЗАТЕЛЬНО входят в состав общего варианта использования.

Пример. Когда пользователь сохраняет результаты своей работы в файл, он указывает место сохранения и расширение файла (например, если он редактировал фотографию в photoshop, он может сохранить ее в различных форматах). Этот процесс можно изобразить на диаграмме вариантов использования следующим образом:

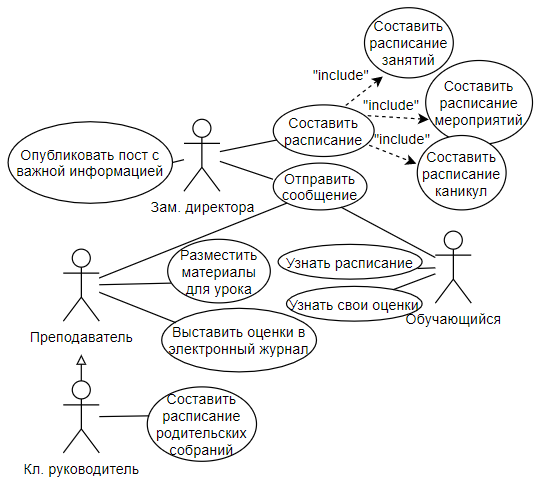
Отношение включения используется для изображения составного действия

Снова вернёмся к нашему основному примеру.



Составление расписания ВКЛЮЧАЕТ в себя составление расписания занятий, мероприятий, каникул(обязательно)

Как итог, наша диаграмма принимает следующий вид:



Третья версия диаграммы

Отношение расширения

Нужно сказать, что в диаграммах вариантов использования применяется ещё один вид связи – *отношение расширения*. Применение отношение расширения несколько специфично, поскольку неправильное его использование может запутать читателя диаграммы. Тем не менее, для полноты картины мы всё равно рассмотрим применение этого отношения на практике. В последний раз модифицируем нашу диаграмму!

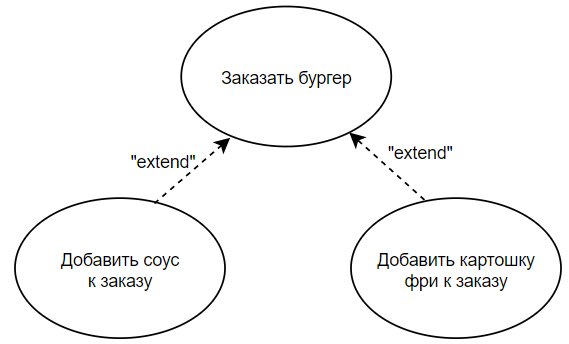
Во время дистанционного обучения школьникам необходимо выполнять домашние задания и присылать их в виде архива или фотографий учителям. Получается, нужно добавить возможность прикреплять файл к сообщению в нашей системе. Чтобы отобразить это на диаграмме мы будем использовать *отношение расширения.* Отношение расширения обозначается пунктирной линией с V-образной стрелкой на конце (похоже на отношение включения), над стрелкой добавляется надпись **“extend ”.**

Изображение выглядит как текст, Шрифт, снимок экрана, белый

Контент, сгенерированный ИИ, может содержать ошибки.З

Зачем над пунктирными линиями добавлять надписи “include” и “extend”?

Чтобы лучше понять этот тип отношений рассмотрим пример. Допустим, вы делаете заказ в сети быстрого питания. Вы хотите заказать бургер, вам предложат *расширить* ваш заказ картошкой фри или соусом. Давайте изобразим процесс заказа на диаграмме вариантов использования.

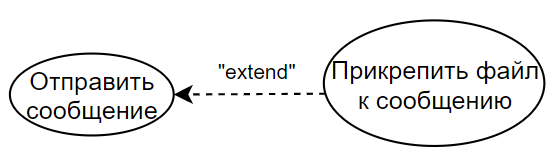
На диаграмме предполагается, что к заказу МОЖЕТ БЫТЬ добавлена картошка фри или соус (необязательно)

Два нижних варианта использования описывают возможные «расширения» для базового варианта использования. Исходя из этого примера, мы можем сделать важное замечание.

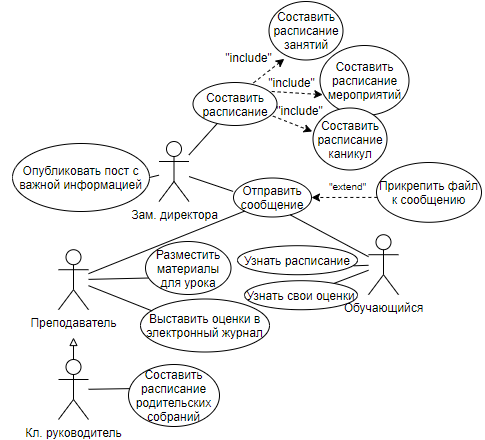
Можно сказать, что отношение расширения - это выборочное отношение включения. Если *отношение включения* обозначает, что элемент **обязательно** включается в состав другого элемента, то в случае *отношения расширения*это включение **необязательно.**

Понимание этого критически важно для грамотного использования этого вида отношений.

Вернёмся к нашему основному примеру. Мы хотим, чтобы действие «прикрепить файл к сообщению» *расширяло*действие «отправить сообщение». На диаграмме это изображается следующим образом:

Расширяем функционал отправки сообщений с помощью функции прикрепления файлов к сообщению (Необязательно прикреплять файл к каждому сообщению)

Как итог, получим такую диаграмму:



Четвёртая версия диаграммы

Изображение выглядит как диаграмма, текст, зарисовка, рисунок

Контент, сгенерированный ИИ, может содержать ошибки.