



Методы и средства проектирования информационных систем и технологий

Лекция 5

Unified Modeling Language

UML (англ. Unified Modeling Language — унифицированный язык моделирования) — язык графического описания для объектного моделирования в области разработки программного обеспечения. UML является языком широкого профиля.

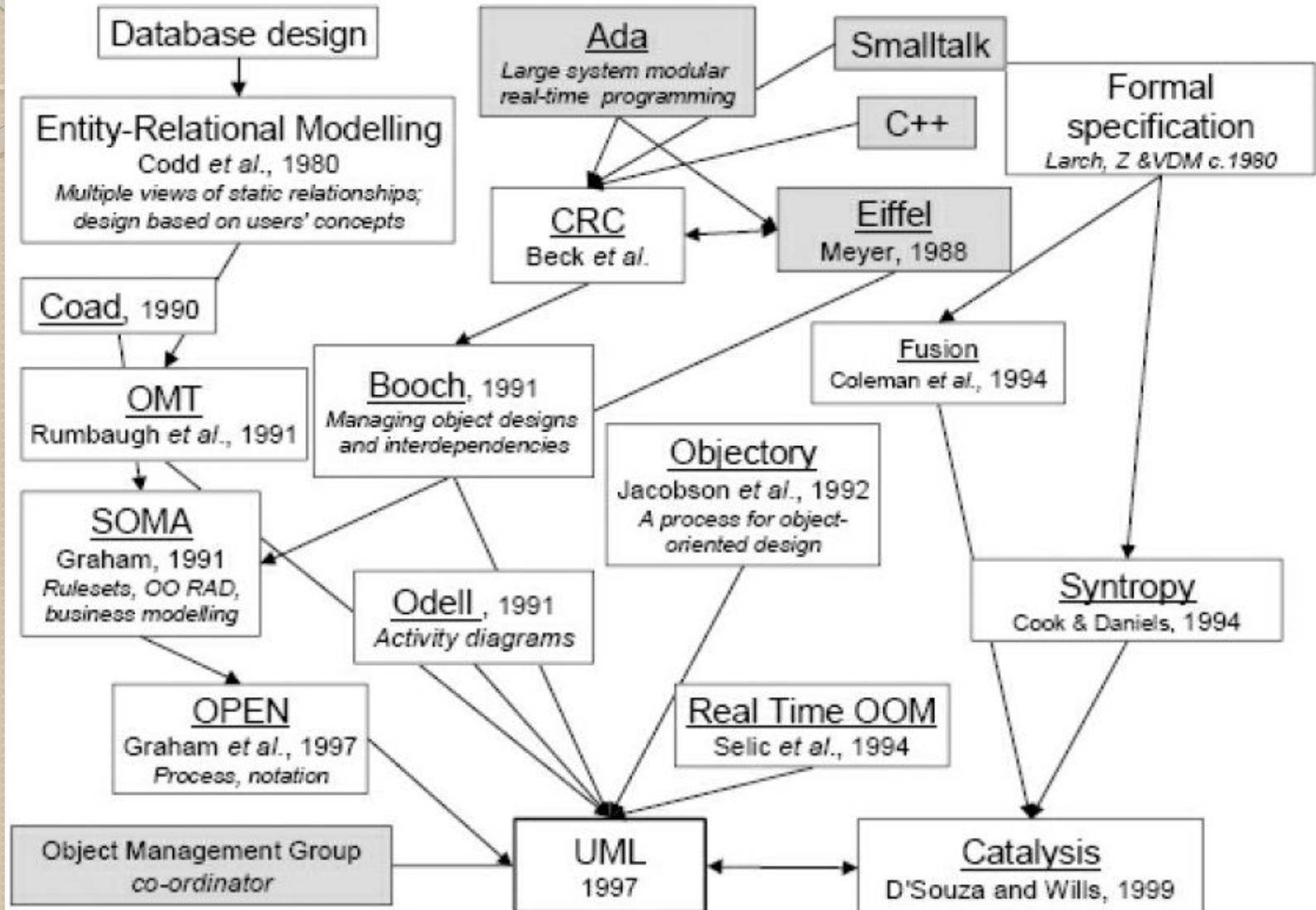
UML это — открытый стандарт, использующий графические обозначения для создания абстрактной модели системы, называемой UML-моделью.

UML был создан для определения, визуализации, проектирования и документирования, в основном, программных систем.

UML не является языком программирования, но на основании UML-моделей возможна генерация кода.

История UML

UML собрал в себе черты нотаций Грейди Буча (Grady Booch), Джима Румбаха (Jim Rumbaugh), Айвара Якобсона (Ivar Jacobson) и многих других.



Структура UML





Диаграммы UML

Диаграммы делятся на следующие категории:

- структурные;
- поведенческие.

Структурные диаграммы используют для отражения физической организации сущностей в системе, то есть отношений между объектами.

Поведенческие диаграммы сконцентрированы на поведении элементов системы.

Диаграммы вариантов использования (модели прецедентов)

Поведение разрабатываемой системы описывается с помощью функциональной модели, которая отображает:

- Системные прецеденты;
- Системное окружение (действующих лиц или актеров);
- Связи между прецедентами и актерами.

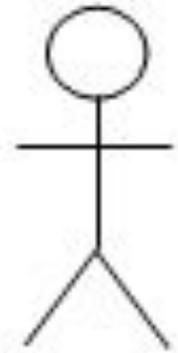
Основная задача диаграммы: представлять собой единое средство, дающее возможность заказчику, конечному пользователю и разработчику совместно обсуждать функциональность и поведение системы.

Актеры (кто-то и что-то, что должно взаимодействовать с системой) могут:

- Только снабжать информацией систему;
- Только получать информацию из системы;
- Снабжать информацией и получать информацию из системы.

Нотации UML

Актер представляет собой любую внешнюю по отношению к моделируемой системе сущность, которая взаимодействует с системой и использует ее функциональные возможности для достижения определенных целей. При этом актеры служат для обозначения согласованного множества ролей, которые могут играть пользователи в процессе взаимодействия с проектируемой системой. Каждый актер может рассматриваться как некая отдельная роль относительно конкретного варианта использования. Стандартным графическим обозначением актера на диаграммах является **фигурка человечка**, под которой записывается имя актера.



Актер

Актеры делятся на 3 основных типа:

- пользователи системы
- другие системы, взаимодействующие с данной
- время

Примерами актеров могут быть сущности, имеющие отношение к концептуальной модели соответствующей предметной области.

С помощью прецедентов моделируется диалог между актером и системой.

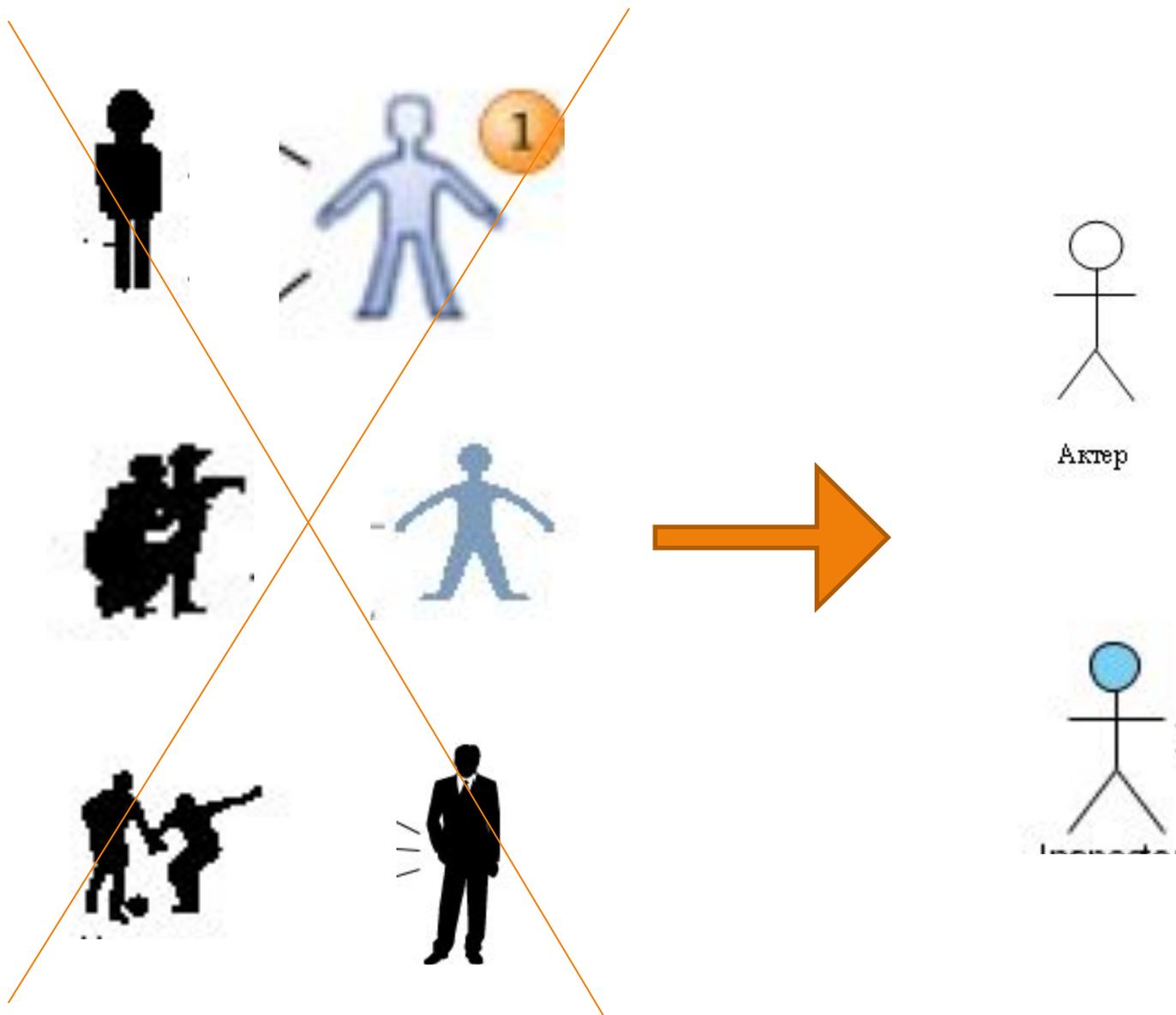
Между актером и прецедентом может существовать ассоциативное отношение.

Ассоциативная связь может быть либо **двусторонней** (от актера к прецеденту и от прецеденту к актеру) либо **односторонней**.

Направление связи показывает кто является ее инициатором (обозначается стрелками на линии связи).

Так как в общем случае актер всегда находится вне системы, его внутренняя структура никак не определяется. Для актера имеет значение только то, как он воспринимается со стороны системы.

Варианты представления актеров



Нотации UML

Отдельный **вариант использования**, **прецедент** это описание множества последовательных событий, выполняемых компьютерной системой, которые приводят к наблюдаемому актером результату.

Графически прецедент изображается в виде ограниченного непрерывной линией эллипса, обычно содержащего только имя прецедента.



Прецедент
(use case)

Нотации UML

Цель варианта использования заключается в том, чтобы определить законченный аспект или фрагмент поведения некоторой сущности без раскрытия её внутренней структуры.

Каждый вариант использования соответствует отдельному сервису, который предоставляет моделируемая сущность по запросу актера, то есть определяет способ применения этой сущности.

Сервис, который инициализируется по запросу актера, представляет собой законченную неделимую последовательность действий. Это означает, что после того как система закончит обработку запроса, она должна возвратиться в исходное состояние, чтобы быть готовой к выполнению следующих запросов.

Нотации UML

Варианты использования могут применяться как для спецификации внешних требований к проектируемой системе, так и для спецификации функционального поведения уже существующей системы.

Множество вариантов использования в целом должно определять все возможные стороны ожидаемого поведения системы. Кроме этого, варианты использования неявно устанавливают требования, определяющие, как актеры должны взаимодействовать с системой, чтобы иметь возможность корректно работать с предоставляемыми сервисами. Для удобства множество вариантов использования может рассматриваться как отдельный пакет.

Примерами вариантов использования могут являться следующие действия: проверка состояния текущего счета клиента, оформление заказа на покупку товара, получение дополнительной информации о кредитоспособности клиента, отображение графической формы на экране монитора и другие действия.

Построение диаграммы

Диаграммы вариантов использования описывают функциональное назначение системы или то, что система должна делать.

Разработка диаграммы преследует следующие цели:

- определить общие границы и контекст моделируемой предметной области;
- сформулировать общие требования к функциональному поведению проектируемой системы;
- разработать исходную концептуальную модель системы для ее последующей детализации в форме логических и физических моделей;
- подготовить исходную документацию для взаимодействия разработчиков системы с ее заказчиками и пользователями.

Суть диаграммы вариантов использования состоит в том, что проектируемая система представляется в виде множества сущностей или актеров, взаимодействующих с системой с помощью вариантов использования. При этом актером (actor) или действующим лицом называется любая сущность, взаимодействующая с системой извне. Это может быть человек, техническое устройство, программа или любая другая система, которая может служить источником воздействия на моделируемую систему так, как определит сам разработчик. Вариант использования служит для описания сервисов, которые система предоставляет актеру. Диаграмма вариантов использования может дополняться пояснительным текстом, который раскрывает смысл или семантику составляющих ее компонентов.

Построение диаграммы

Актеры взаимодействуют с системой посредством обмена сообщениями с вариантами использования.

Сообщение представляет собой запрос актером определенного сервиса системы и получение этого сервиса. Это взаимодействие может быть выражено посредством **ассоциаций** между отдельными актерами и вариантами использования или классами.

С актерами могут быть связаны интерфейсы, которые определяют, каким образом другие элементы модели взаимодействуют с этими актерами.

Два и более актера могут иметь общие свойства (взаимодействовать с одним и тем же множеством вариантов использования одинаковым образом).

Такая общность свойств и поведения представляется в виде **отношения обобщения** с другим, возможно, абстрактным актером, который моделирует соответствующую общность ролей.

Нотации UML

Интерфейс (interface) служит для спецификации параметров модели, которые видимы извне, без указания их внутренней структуры.

Применительно к диаграммам вариантов использования, интерфейсы определяют совокупность операций, которые обеспечивают необходимый набор сервисов для актеров.

Графический символ отдельного интерфейса соединяется на диаграмме сплошной линией или пунктирной линией со стрелкой с тем вариантом использования, который его поддерживает.

Сплошная линия указывает, что связанный с интерфейсом вариант использования **должен реализовывать все необходимые для него сервисы**.

Пунктирная линия со стрелкой означает, что **вариант использования предназначен для спецификации** только того сервиса, который необходим для реализации данного интерфейса.

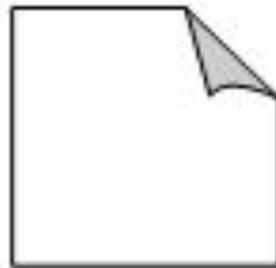


Нотации UML

Примечания (notes) предназначены для включения в модель произвольной текстовой информации, имеющей непосредственное отношение к контексту разрабатываемого проекта.

В качестве такой информации могут быть комментарии разработчика (например, дата и версия разработки диаграммы или ее отдельных компонентов), ограничения (например, на значения отдельных связей или экземпляры сущностей) и помеченные значения.

Если в примечании указывается ключевое слово «constraint», то оно является ограничением, налагаемым на соответствующий элемент модели.



Примечание

Нотации UML

Между элементами диаграммы вариантов использования могут существовать различные **отношения**, которые описывают взаимодействие экземпляров актеров и вариантов использования.

В языке UML существует несколько стандартных видов отношений между актерами и вариантами использования:

- **ассоциации** (association relationship);
- **расширения** (extend relationship);
- **обобщения** (generalization relationship);
- **включения** (include relationship).

В метамодели отношение расширения является направленным и указывает, что применительно к отдельным примерам некоторого варианта использования должны быть выполнены конкретные условия, определенные для расширения данного варианта использования. Так, если имеет место отношение расширения от варианта использования **A** к варианту использования **B**, то это означает, что свойства экземпляра варианта использования **B** могут быть дополнены благодаря наличию свойств у расширенного варианта использования **A**.

Отношение расширения отмечает тот факт, что один из вариантов использования может присоединять к своему поведению некоторое дополнительное поведение, определенное для другого варианта использования. Данное отношение включает в себя некоторое условие и ссылки на точки расширения в базовом варианте использования. Чтобы расширение имело место, должно быть выполнено определенное условие данного отношения. Ссылки на точки расширения определяют те места в базовом варианте использования, в которые должно быть помещено соответствующее расширение при выполнении условия.

Семантика отношения расширения определяется следующим образом. Если экземпляр варианта использования выполняет некоторую последовательность действий, которая определяет его поведение, и при этом имеется точка расширения на экземпляре другого варианта использования, которая является первой из всех точек расширения исходного варианта, то проверяется условие данного отношения. Если условие выполняется, исходная последовательность действий расширяется посредством включения действий экземпляра другого варианта использования.

Отношение обобщения

Отношение **обобщения** служит для указания того факта, что некоторый вариант использования **A** (потомок) может быть обобщен до варианта использования **B** (родитель). В этом случае вариант **A** будет являться специализацией варианта **B**.

Потомок наследует все свойства и поведение своего родителя, а также может быть дополнен новыми свойствами и особенностями поведения.

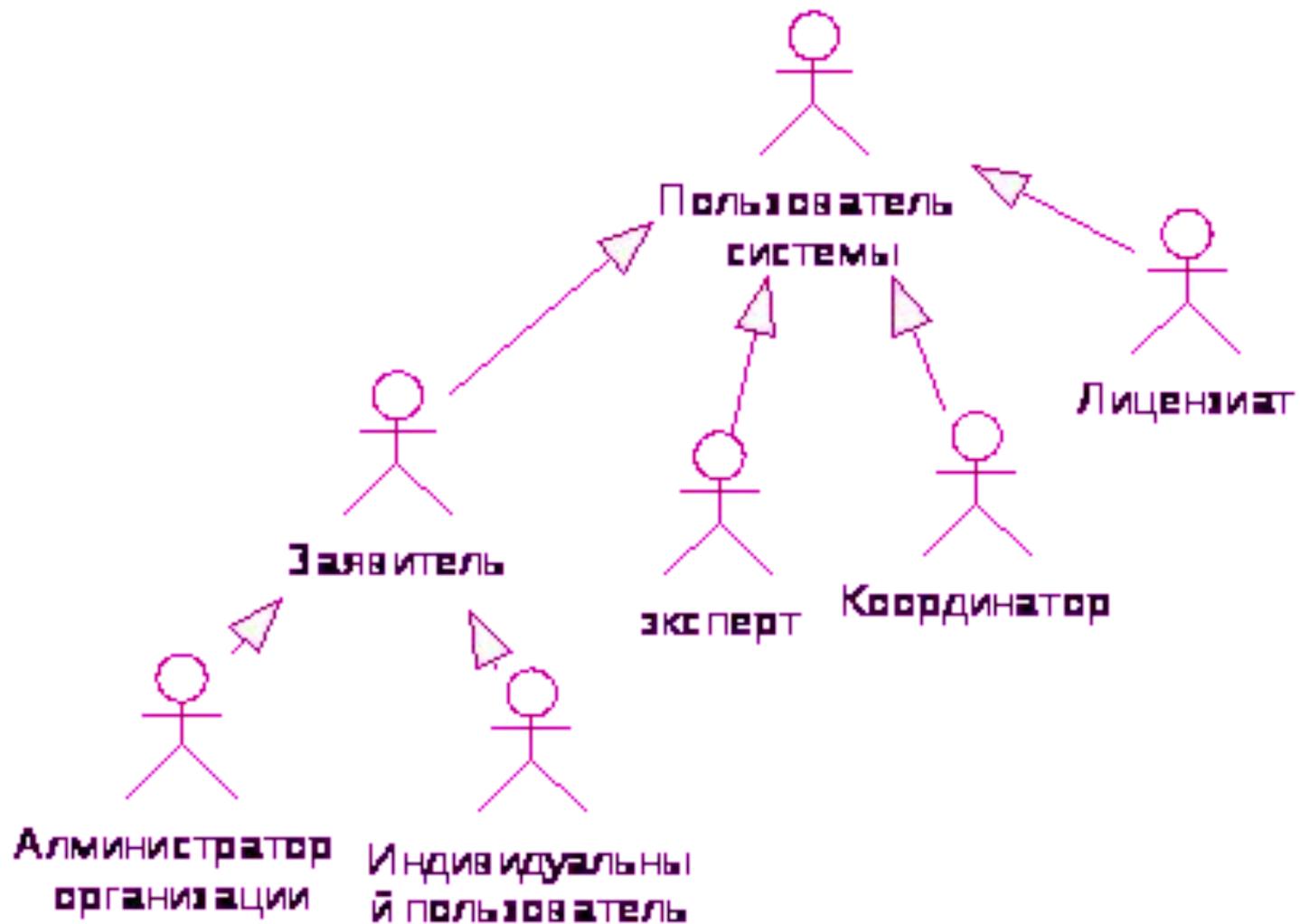
Графически данное отношение обозначается сплошной линией со стрелкой, которая указывает на родительский вариант использования.

Отношение обобщения между вариантами использования применяется в том случае, когда необходимо отметить, что дочерние варианты использования обладают всеми атрибутами и особенностями поведения родительских вариантов.

Между отдельными актерами также может существовать отношение обобщения. Данное отношение является направленным и указывает на факт специализации одних актеров относительно других.



Обобщения



Отношение включения

Отношение **включения** между двумя **вариантами использования** указывает, что некоторое заданное поведение для одного варианта использования **включается** в качестве составного компонента в последовательность поведения другого варианта использования.

Один вариант использования может быть включен в несколько других вариантов, а также включать в себя другие варианты.

Включаемый вариант использования может быть независимым от базового варианта (он предоставляет ему некоторое инкапсулированное поведение, детали реализации которого скрыты и могут быть перераспределены между несколькими включаемыми вариантами использования).

Базовый вариант может зависеть только от результатов выполнения включаемого в него поведения, но не от структуры включаемых в него вариантов.



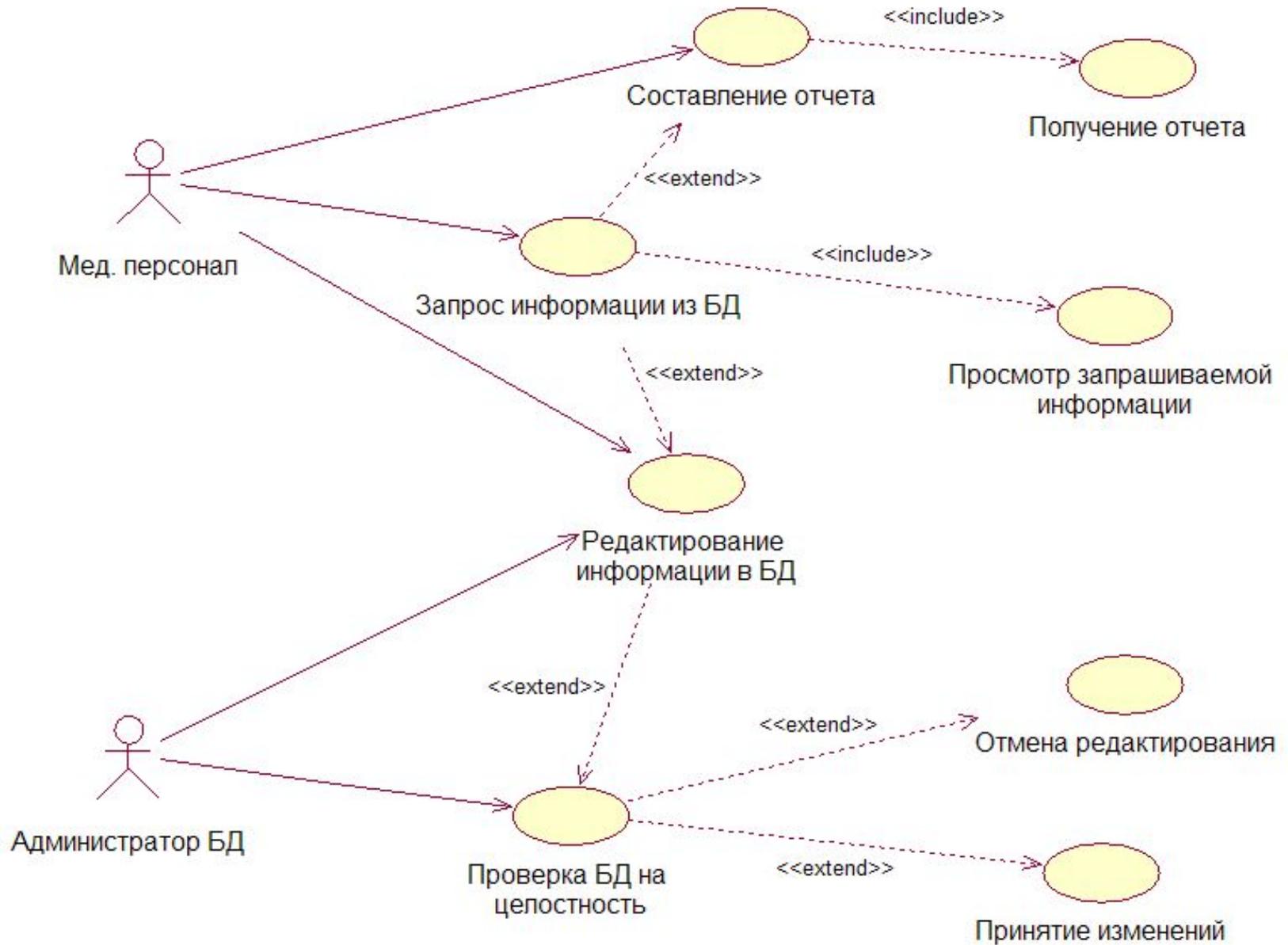
«include»

1. В метамодели отношение расширения является направленным и указывает, что применительно к отдельным примерам некоторого варианта использования должны быть выполнены конкретные условия, определенные для расширения данного варианта использования. Так, если имеет место отношение расширения от варианта использования **A** к варианту использования **B**, то это означает, что свойства экземпляра варианта использования **B** могут быть дополнены благодаря наличию свойств у расширенного варианта использования **A**.
2. Отношение расширения отмечает тот факт, что один из вариантов использования может присоединять к своему поведению некоторое дополнительное поведение, определенное для другого варианта использования. Данное отношение включает в себя некоторое условие и ссылки на точки расширения в базовом варианте использования. Чтобы расширение имело место, должно быть выполнено определенное условие данного отношения. Ссылки на точки расширения определяют те места в базовом варианте использования, в которые должно быть помещено соответствующее расширение при выполнении условия.
3. Семантика отношения расширения определяется следующим образом. Если экземпляр варианта использования выполняет некоторую последовательность действий, которая определяет его поведение, и при этом имеется точка расширения на экземпляр другого варианта использования, которая является первой из всех точек расширения исходного варианта, то проверяется условие данного отношения. Если условие выполняется, исходная последовательность действий расширяется посредством включения действий экземпляра другого варианта использования.

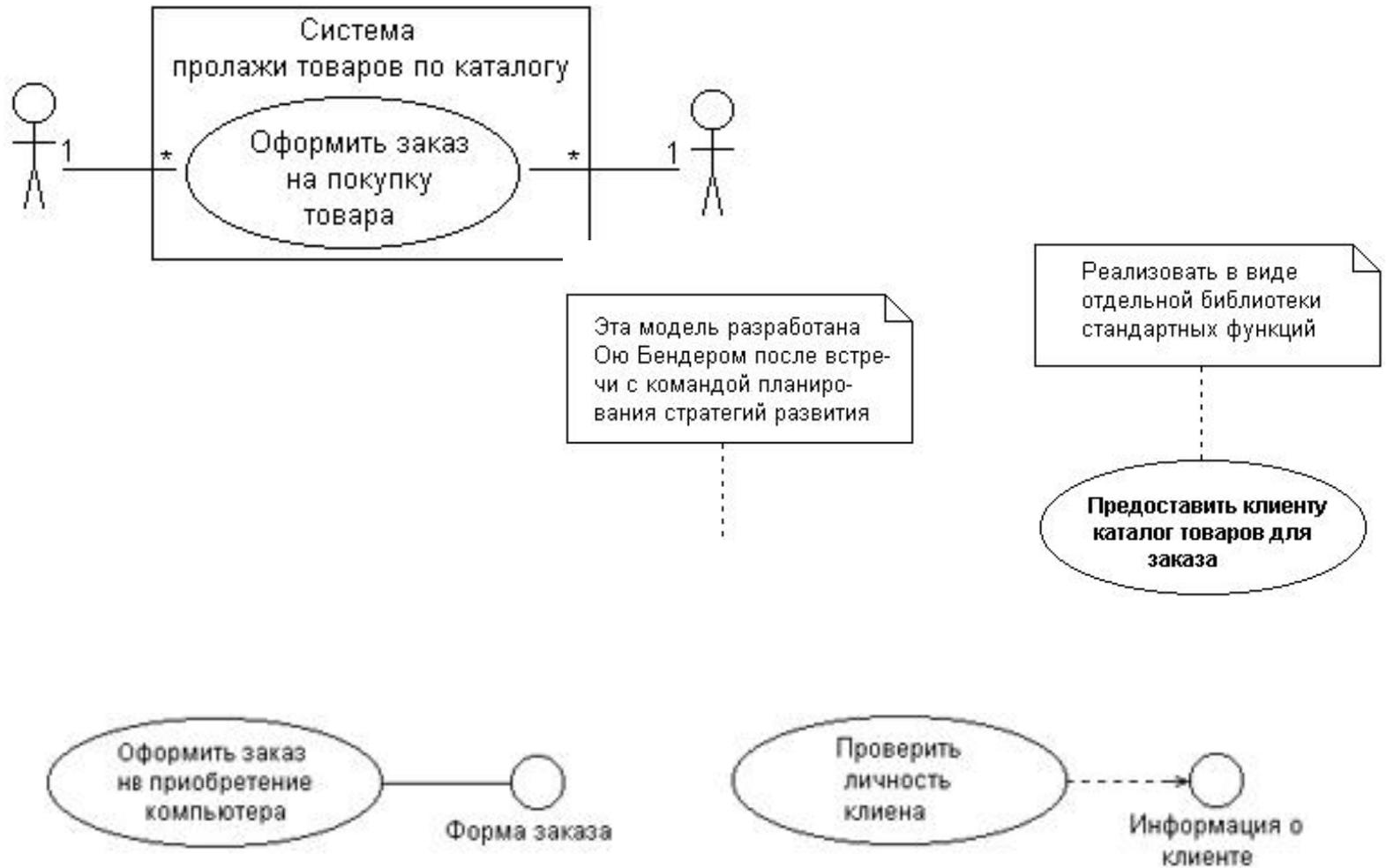
Пример диаграммы вариантов использования для банковского автомата



Пример диаграммы вариантов использования



Пример диаграммы вариантов использования



Диаграммы использования

Построение диаграммы вариантов использования является самым первым этапом процесса объектно-ориентированного анализа и проектирования.

Цель - представить совокупность требований к поведению проектируемой системы.

Спецификация требований к проектируемой системе в форме диаграммы вариантов использования представляет собой самостоятельную модель.

Большая часть вариантов использования формируется на этапе исследования проекта, однако в ходе дальнейшей работы можно обнаружить дополнительные особенности.

Каждый вариант использования является потенциальным требованием к системе, и пока оно не выявлено, не возможно перейти к планированию его реализации.

Диаграммы использования

Некоторые разработчики вначале:

- составляют перечень вариантов использования и обсуждают их,
- приступают к моделированию.

Концептуальное моделирование с участием пользователей помогает выявить варианты использования.

- Важно помнить, что варианты использования служат *внешним представлением системы.*

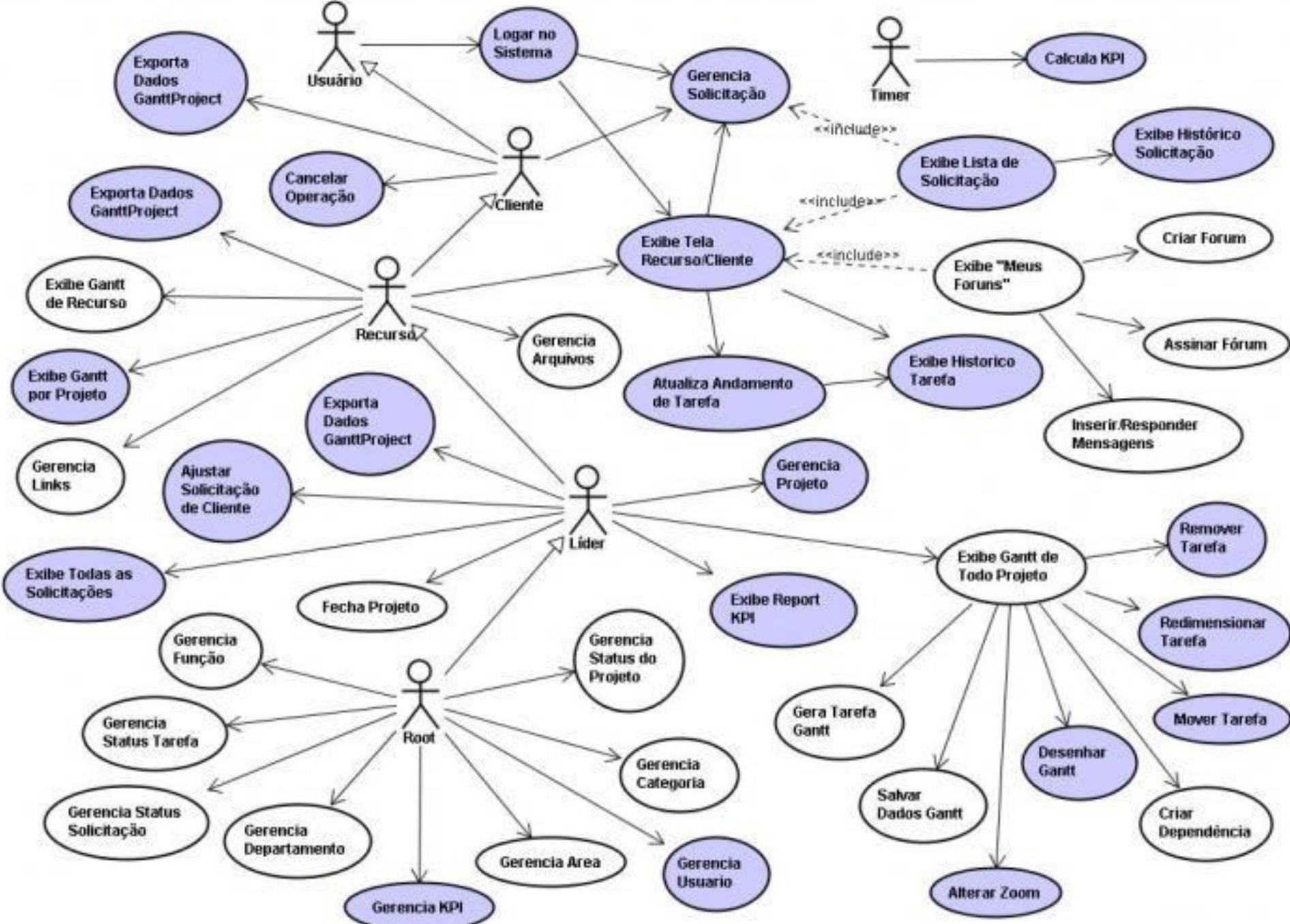
Диаграммы использования

Сколько вариантов использования следует сформировать?

- Некоторые эксперты по вариантам использования утверждали, что для проекта с трудоемкостью 10 человеко-лет необходимо около **дюжины** вариантов использования.

Но это только базовые варианты использования; каждый из таких вариантов использования может заключать в себе несколько сценариев и альтернативных вариантов использования.

- Также встречаются проекты аналогичной трудоемкости с более чем **сотней** отдельных вариантов использования. (Если подсчитать количество альтернативных вариантов использования для дюжины базовых вариантов использования, то их общее количество как раз и будет около 100.).
- Рекомендуемое общее количество **актеров** в модели - не более **20**, а **вариантов использования** - не более **50**. В противном случае модель теряет свою наглядность и, возможно, заменяет собой одну из некоторых других диаграмм



Диаграммы использования

В случае, когда для представления иерархической структуры проектируемой системы используются подсистемы, система может быть определена в виде вариантов использования на всех уровнях.

Отдельные **подсистемы** или **классы** могут выступать в роли таких **вариантов использования**.

При этом **вариант**, соответствующий некоторому из этих элементов, в последующем может уточняться множеством более мелких **вариантов использования**, каждый из которых будет определять сервис элемента модели, содержащийся в сервисе исходной системы.