

Анализ и проектирование на UML

Хлопотов М.В.,
Университет ИТМО

Моделирование поведения

- Модель поведения — это описание алгоритма работы системы.
- В UML предусмотрено несколько различных средств для описания поведения.
- Выбор того или иного средства диктуется типом поведения, которое нужно описать.
- Для моделирования поведения используются диаграмма состояний, диаграмма деятельности, диаграммы взаимодействия.

Диаграммы взаимодействия

- Диаграммы взаимодействия предназначены для моделирования поведения путем описания взаимодействия объектов для выполнения некоторой задачи или достижения определенной цели.
- Взаимодействие происходит путем обмена сообщениями между экземплярами различных классификаторов

Диаграммы взаимодействия

- Данный тип диаграмм позволяет описывать не только взаимодействие программных объектов (экземпляров классов), но и взаимодействие экземпляров иных классификаторов: действующих лиц, вариантов использования, подсистем, компонентов, узлов.
- Диаграммы взаимодействия графически изображаются в двух формах: диаграммы последовательности и диаграммы кооперации.

Диаграмма последовательности

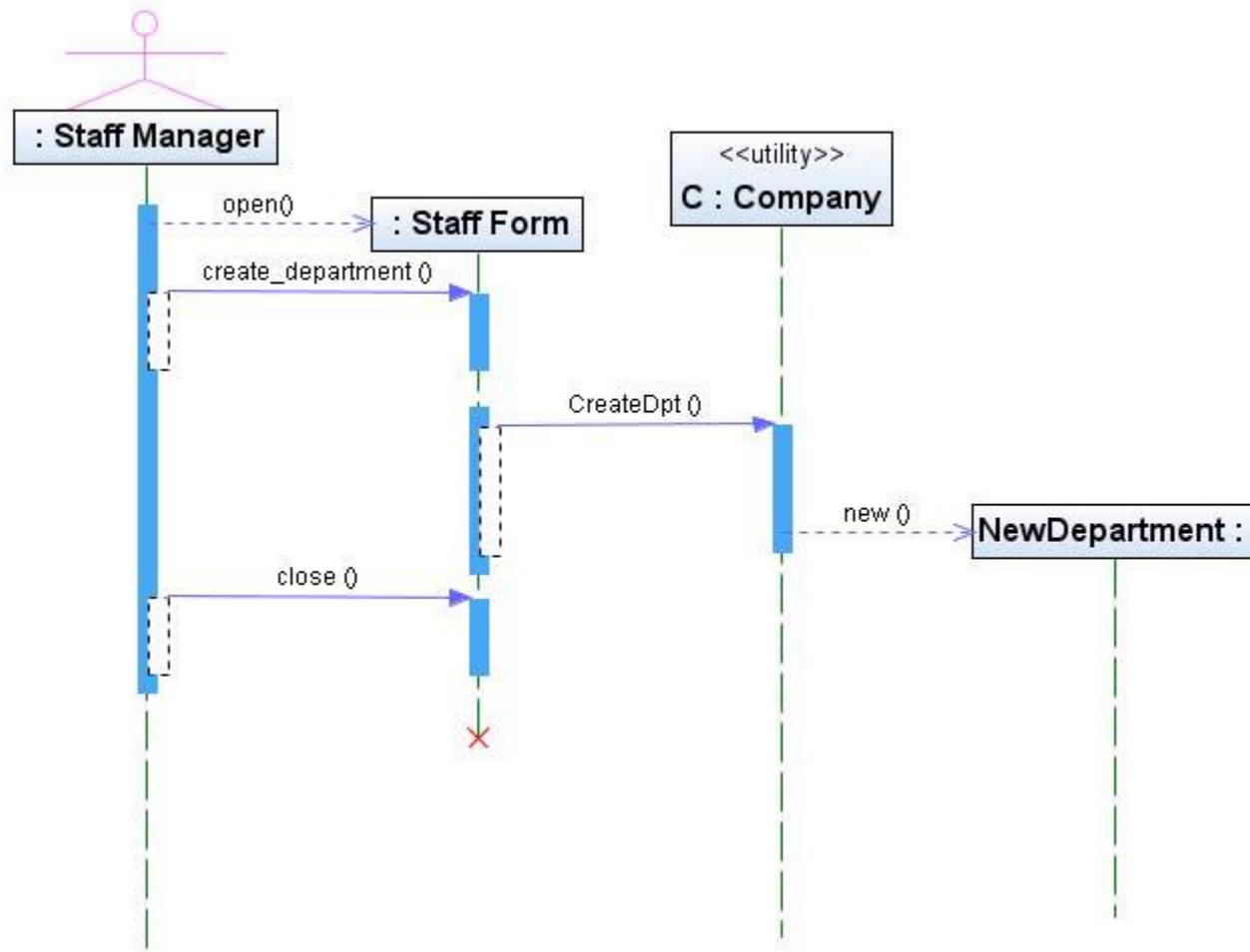
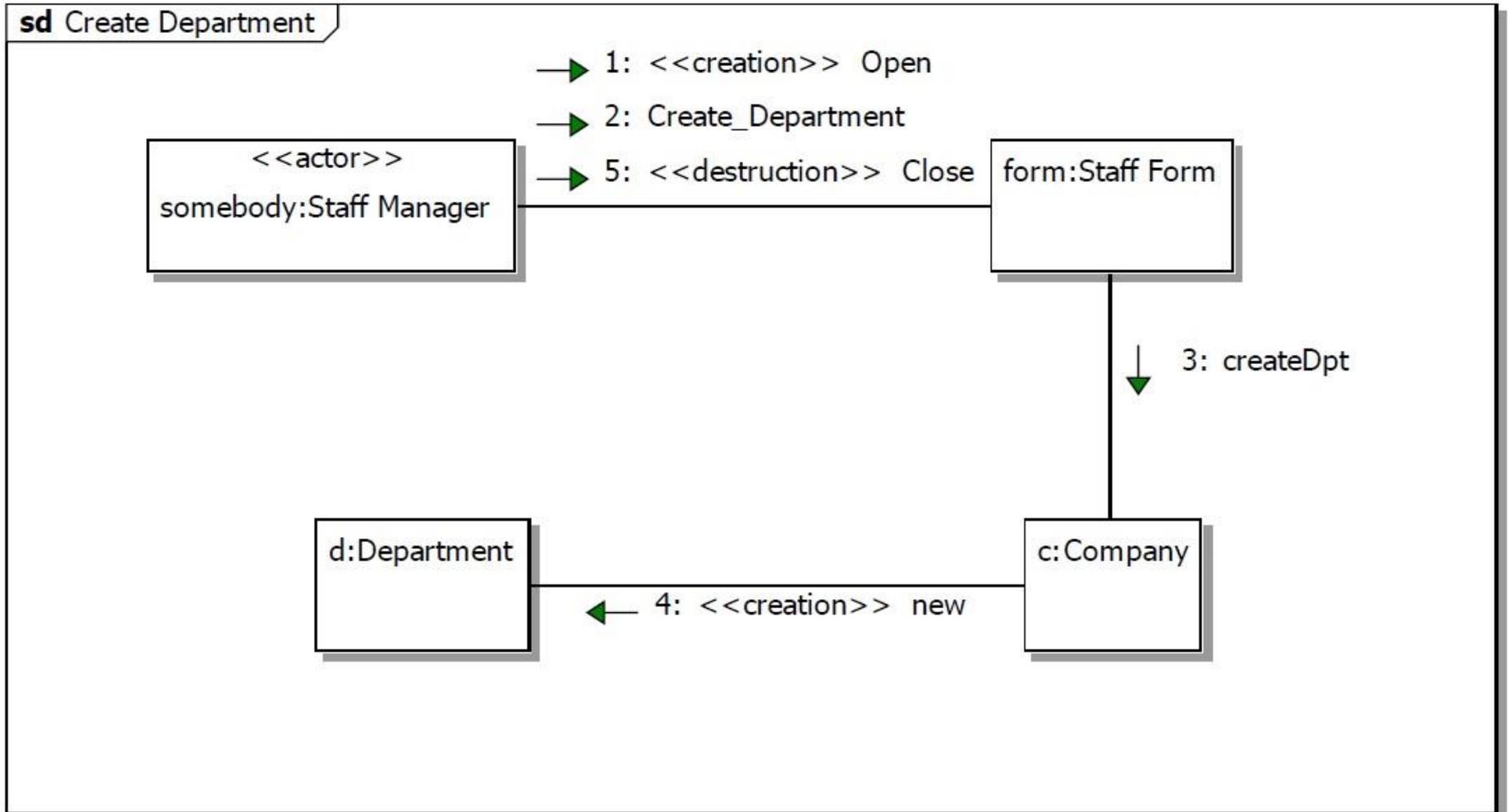


Диаграмма кооперации

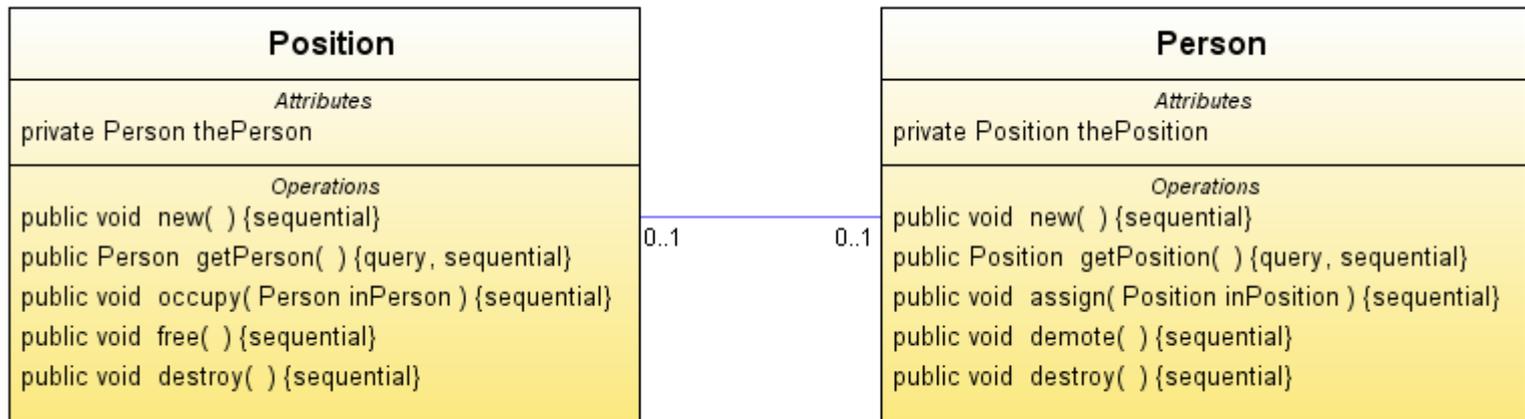


Связь между диаграммами

- Рассмотрим пример моделирования поведения взаимодействующих процессов, которое должно обеспечивать определенную функциональность, т. е. изменения состояния системы.

Связь между диаграммами

- Пусть у нас определены два класса: Position (должность) и Person (сотрудник)



Связь между диаграммами

- У объектов этих классов имеются по два состояния:
- должность может быть вакантна (Vacant) или занята определенным сотрудником (Occupied) и сотрудник может быть не занят (Free) или назначен на определенную должность (Assigned).

Связь между диаграммами

- У каждого из классов есть конструктор (new) и деструктор (destroy) и по паре операций, которые ответственны за изменение состояния объекта: у класса Position операции называются occupy (занять должность) и free (освободить должность), а у класса Person — assign (назначить на должность) demote (освободить от должности).

Связь между диаграммами

- У каждого класса есть скрытый атрибут, предназначенный для хранения ссылки на объект другого класса (т. е. между этими классами существует ассоциация) и предусмотрена операция без побочного эффекта, возвращающая значение данного (`getPerson` и `getPosition`, соответственно).

Связь между диаграммами

- Рассмотрим теперь, как должна выполняться операция назначения сотрудника на должность.
- Мы оставим в стороне вопрос о том, в каком классе разумно определить данную операцию (на самом деле это совершенно не важно).
- Положим, что операция назначения сотрудника на должность имеет два параметра — сотрудника и должность:
- `assignP2P (person:Person, position:Position)`

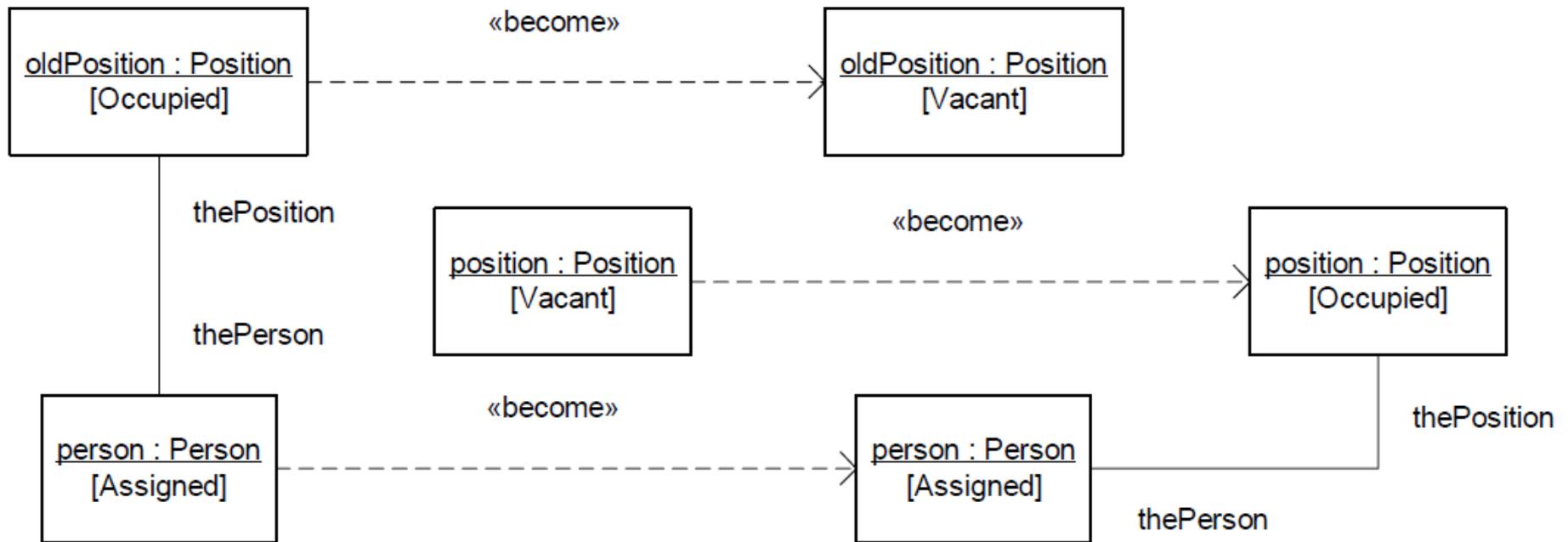
Связь между диаграммами

- Допустим, что требуется обеспечить элементарный порядок в учете кадров (целостность данных): если сотрудник А назначен на должность Б, то и в должности Б должно быть записано, что ее занимает сотрудник А и наоборот.
- Другими словами, занятые должности и сотрудники должны взаимно однозначно соответствовать друг другу, а свободные должности и сотрудники должны быть действительно свободны и не должны содержать неадекватных ссылок друг на друга.

Связь между диаграммами

- Требуемое поведение операции `assignP2P` можно описать с помощью диаграммы объектов, на которой показано, как должны измениться связи между объектами в результате выполнения операции. В данном описании контекста рассматривается типичный сценарий, в котором до выполнения операции сотрудник занимает некоторую должность, а целевая должность вакантна.

Связь между диаграммами



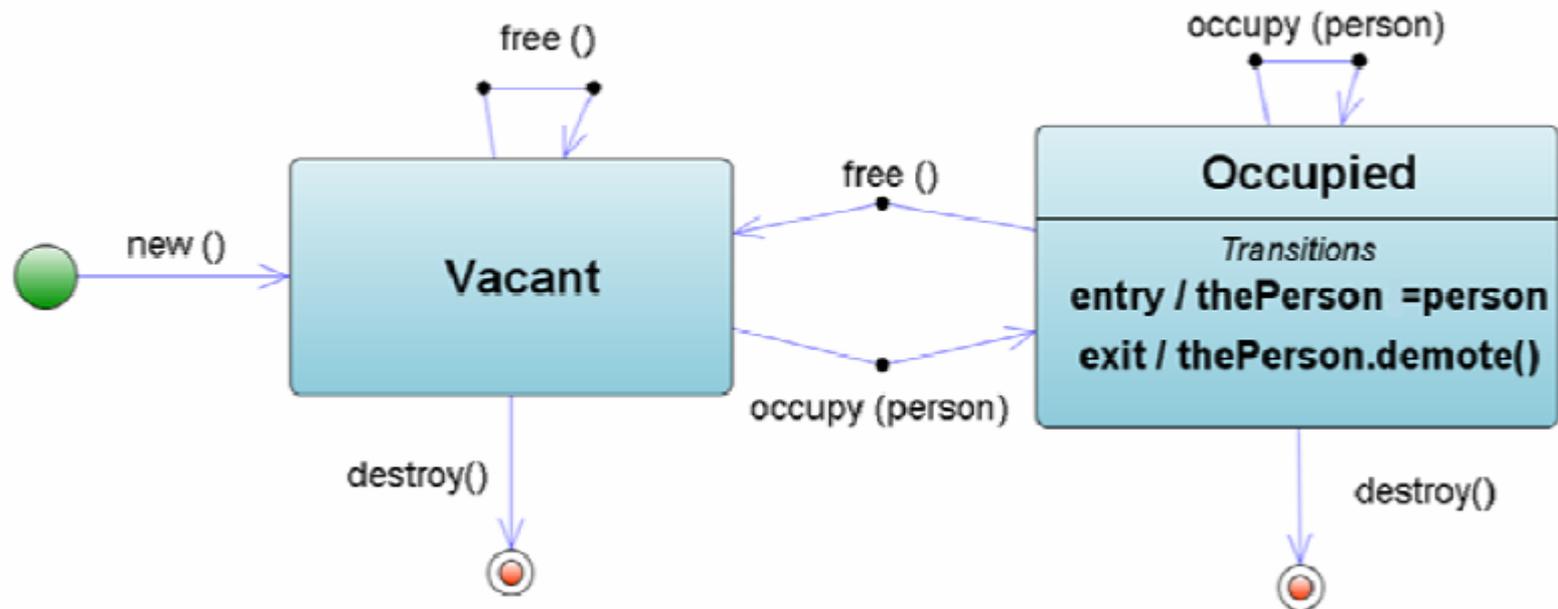
Связь между диаграммами

- Мы видим, что при назначении сотрудника на должность задействованы три объекта: Требуемое поведения может быть обеспечено за счет взаимодействия автоматов, реализующих поведение каждого из этих объектов.

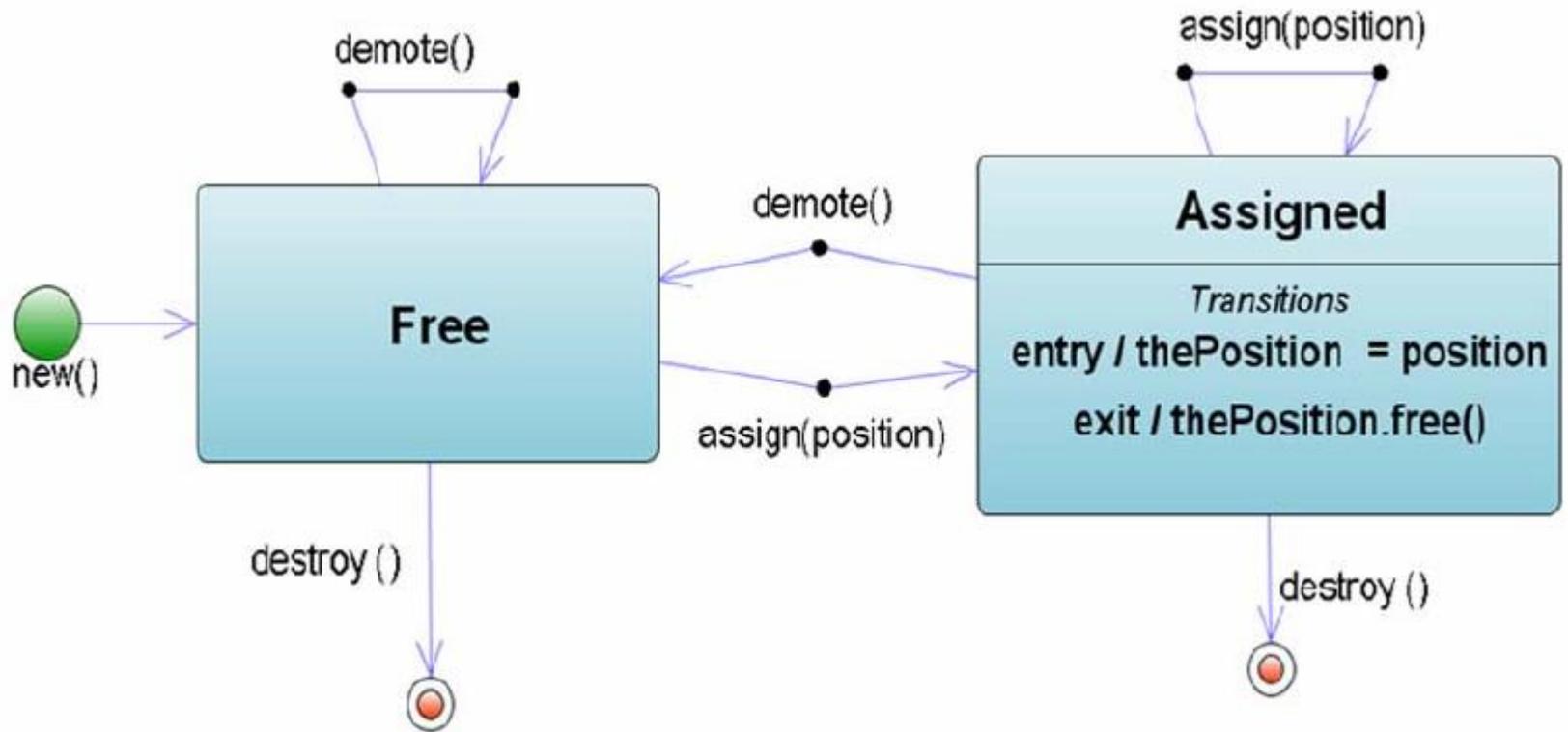
Связь между диаграммами

- Рассмотрим диаграммы машин состояний для `Position` и `Person`, соответственно. Классы `Position` и `Person` в нашей модели совершенно равноправны и их поведение по отношению друг к другу совершенно симметрично.

Связь между диаграммами



Связь между диаграммами



Связь между диаграммами

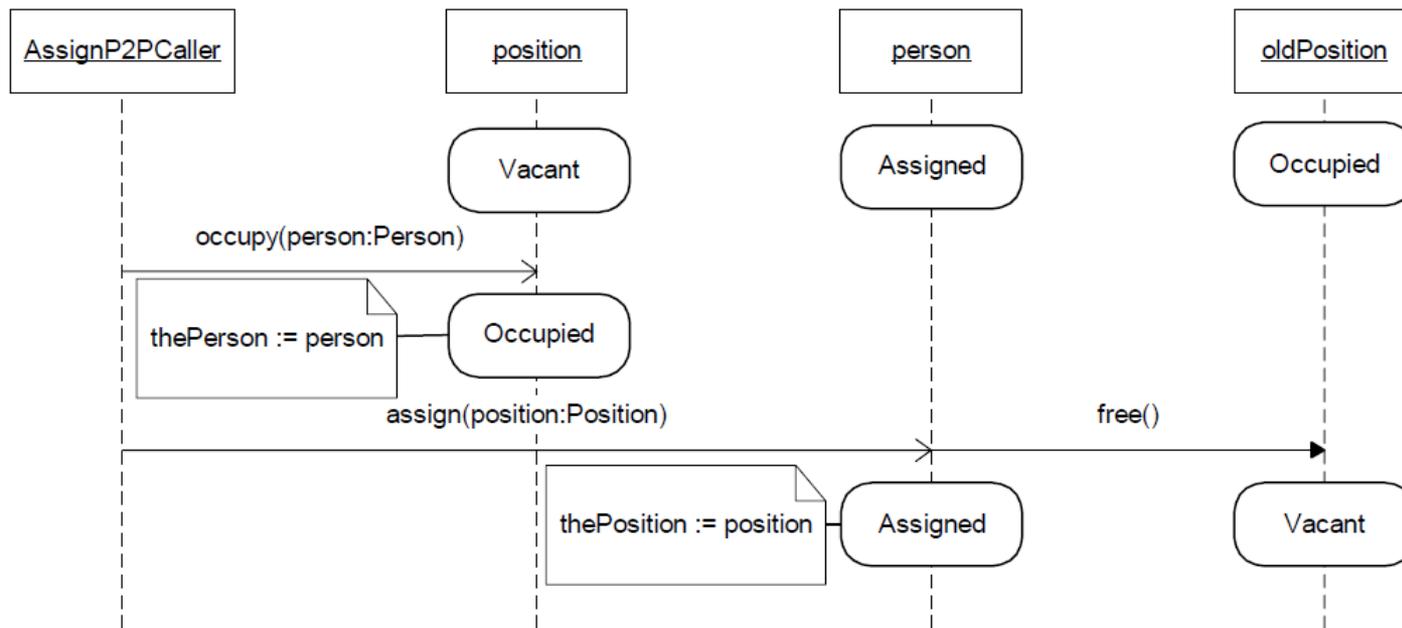
- В таком случае, требуемая операция назначения сотрудника на должность может быть реализована двумя вызовами операций объектов, являющихся аргументами операции:
 - `position.occupy(person)`
 - `person.assign(position)`

Связь между диаграммами

- Указанные две операции можно вызвать в любом порядке или *параллельно*, более того, их можно вызывать с ожиданием возврата управления или без ожидания — в любом случае взаимодействие автоматов обеспечит требуемое поведение (если только в процесс обмена сообщениями не вмешается "посторонний" вызов операции одного из этих объектов).

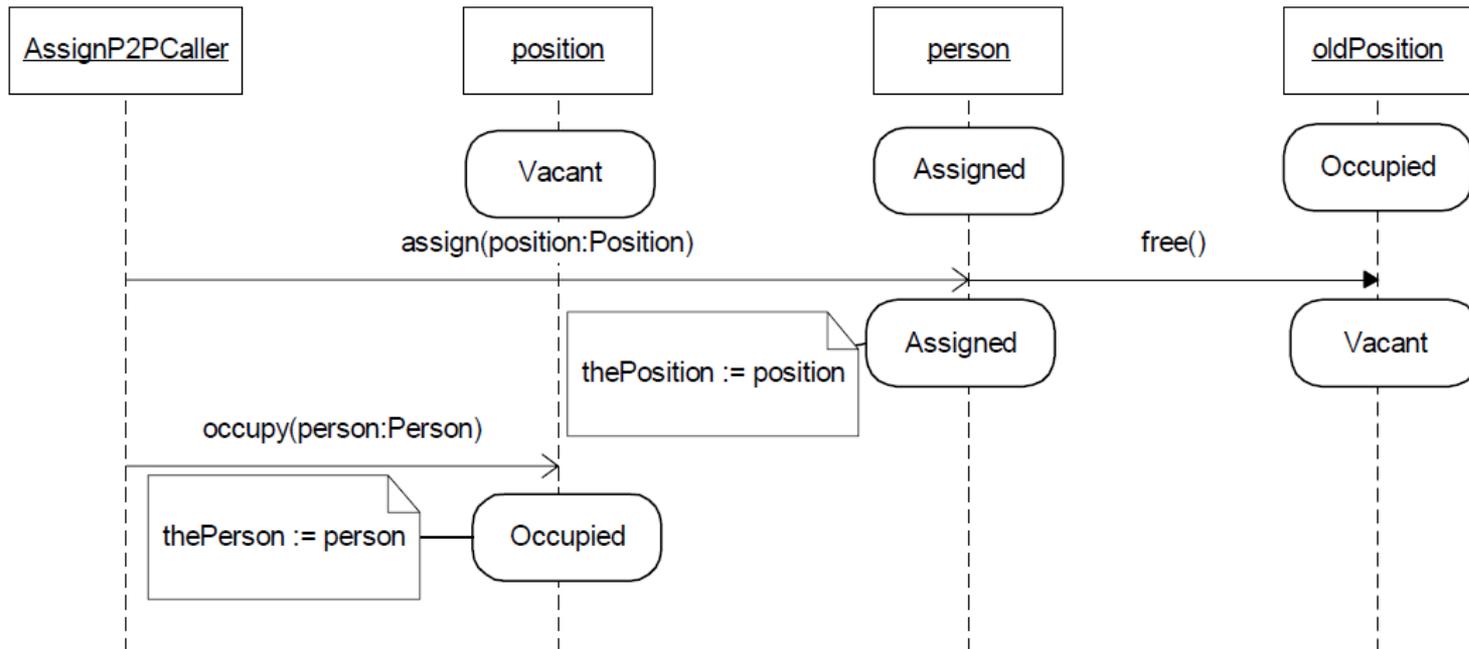
Связь между диаграммами

- Диаграммы последовательности, описывающие возможные протоколы взаимодействия при выполнении операции `assignP2P`.



Связь между диаграммами

- Диаграммы последовательности, описывающие возможные протоколы взаимодействия при выполнении операции `assignP2P`.



Выводы

- Диаграммы взаимодействия связывают диаграмму вариантов использования, диаграмму классов и диаграмму состояний.
- Диаграммы для моделирования поведения позволяют связать диаграммы структуры и диаграммы использования.
- По сути в процессе моделирования поведения создаётся **модель** системы.