Моделиране на изискванията

- **Моделиране на изискванията: Сценарии, Данни и Аналитични класове**

Slide Set to accompany

Software Engineering: A Practitioner's Approach, 8/e

by Roger S. Pressman

Лектор: Доц. д-р Ася Стоянова-Дойчева

Модел на данните

- Изучаване на данните независимо от обработката
- Фокусиране върху домейна на данните
- Създаване на модел на потребителско ниво на абстракция
- Определяне как обектите от данни са свързани един с друг

Какво е обект на данните?

- Представяне на почти всяка съставна информация, която трябва да бъде разбрана от софтуера.
  - *Съставна информация—нещо, което има характеристики или атрибути*
- може да бъде external entity - нещо, което произвежда или консумира информация)
- Описанието на обектите от данни обекта от данни и всичките му атрибути.
- Обекта от данни капсулира само данни – няма връзка от данните към операциите които действат върху данните.

Data Objects and Attributes

Обекта от данни съдържа множество от атрибути

object: automobile

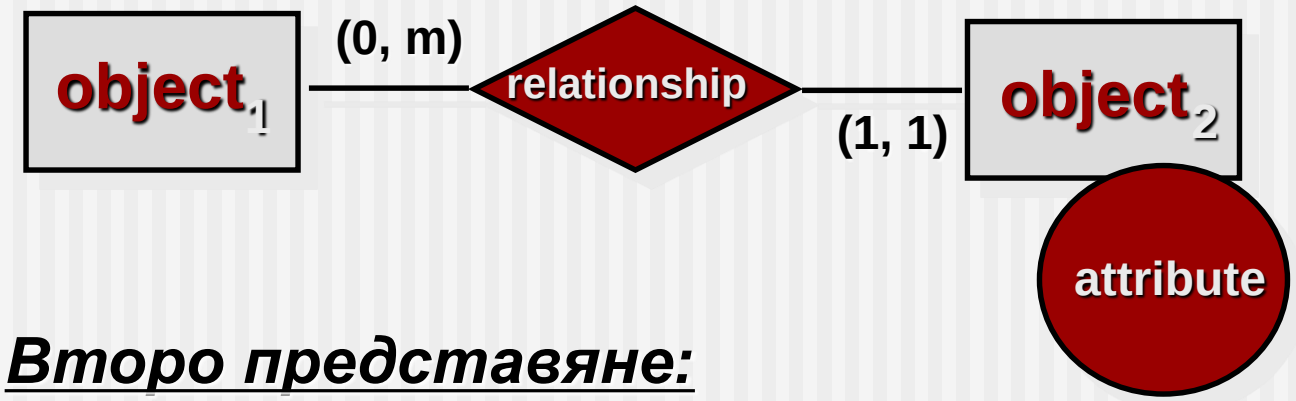
attributes:
- **make**
- **model**
- **body type**
- **price**
- **options code**

Какво е връзката

- Обектите от данни са свързани един с друг по различен начин.
  - Има връзка между **човек** и **кола,** защото двата обекта са свързани
    - Човека притежава кола
    - Човека има право да управлява кола
- Връзките *притежава* and *има право да управлява* дефинират съответни връзки между обектите човек и кола.
- Могат да съществуват няколко инстанции на връзките
- Обектите могат да бъдат свързани по много различни начини

ERD нотация

Първо представяне:

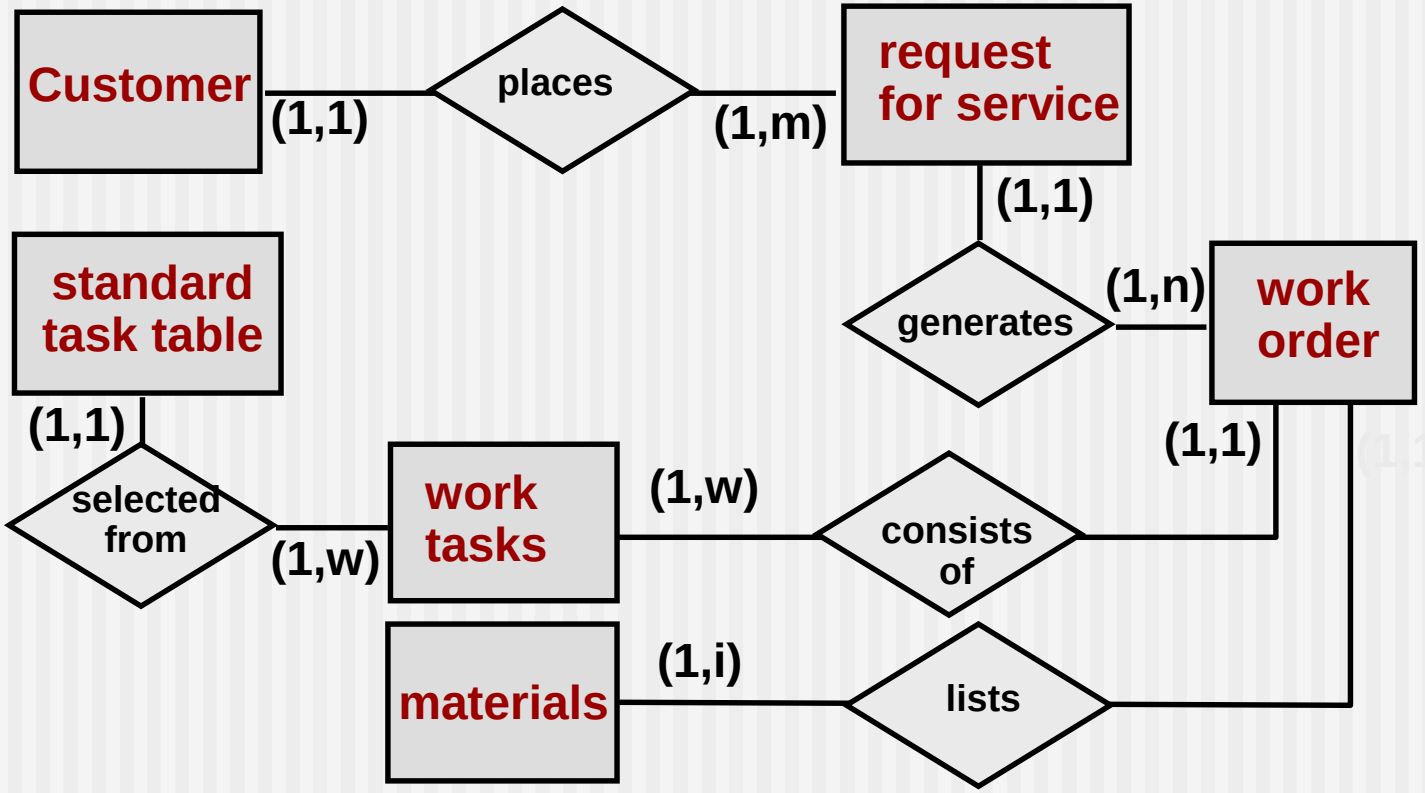


Второ представяне:

Разработване на ERD

- *Ниво1*—моделиране на всички обекти от данни(entities) и връзките им един с друг
- *Ниво2*—моделиране на всички entities и връзки
- *ниво3*—моделиране на всички entities, връзки и атрибути

ERD: Пример



Базирано на класове моделиране

- Базираното на класове моделиране включва:
  - Обекти които системата ще манипулира
  - Операции (също наречени методи или услуги) , които ще бъдат приложени върху обектите
  - Връзки (йерархични) между обектите
  - Взаимодействия който се случват между дефинираните класове.
- Елементите на базирания на класове модел включва класове, обекти, атрибути, операции, CRC модели collaboration диаграми и пакети

Идентифициране на аналитичните класове

- Проучване на потребителските сценарии, разработени като част от модела на изискванията.
- Това са съществителни имена или фрази със съществителни имена, за които говори крайният потребител.
- Трябва да извадим всички потенциални такива.

Проявяване на аналитичните класове

- *Аналитичните класове* проявяват себе си по един от следните начини:
  - *External entities* (други системи, устройства, хора), които произвеждат или използват информация
  - *Things* (справки, показвания, писма, сигнали), които са част от информационния домейн за проблема
  - *Occurrences or events* които се случват в контекста на системните операции
  - *Roles* изпълнявани от хора, които си взаимодействат със системата
  - *Organizational units, които са свързани с приложението*
  - *Places, които определят контекста на проблема и общата функция*
  - *Structures* които дефинират клас обекти или свързани класове от обекти

Дефиниране на атрибути

- *Атрибутите* описват клас, който е включен в аналитичния модел.

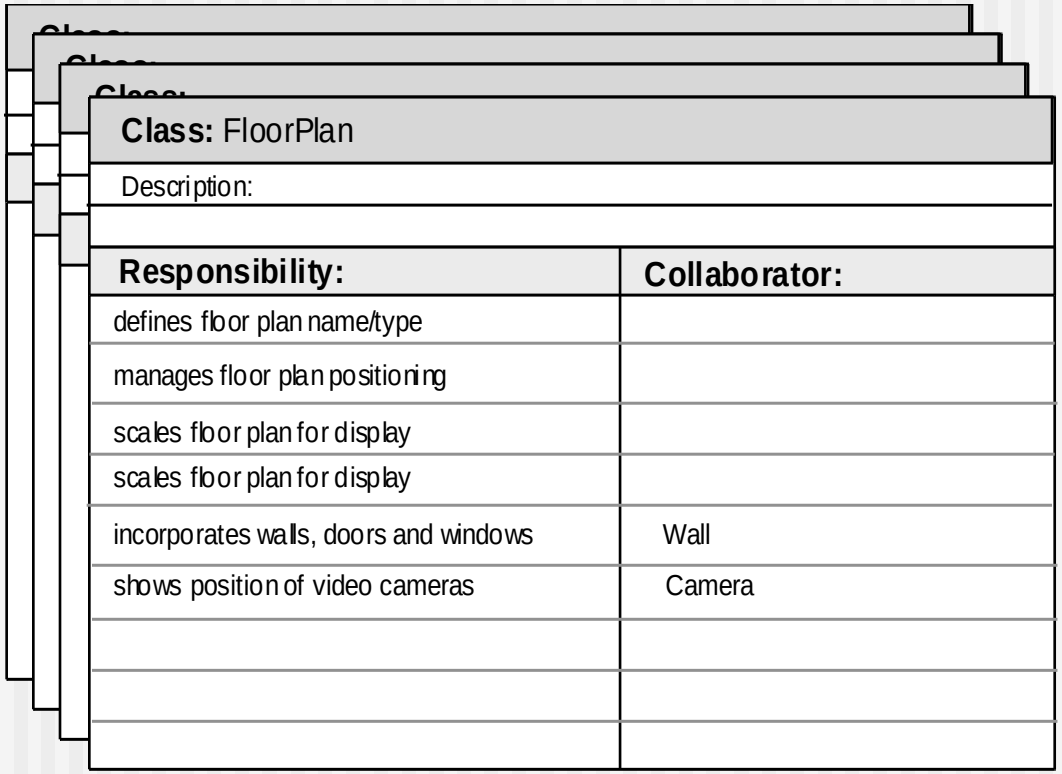
Дефиниране на операции

- Правим граматичен разбор на потребителските изисквания и търсим глаголите
- Операциите могат да бъдат разделени в 4 категории:
  - (1) операции, които манипулират данни (добавяне, изтриване, редактиране, избор)
  - (2) операции, които извършват изчисления
  - (3) операции, които отговарят за състоянието на обектите
  - (4) операции, които наблюдават обектите за появяване на контролни събития

CRC Модели

- *Class-responsibility-collaborator (CRC) моделирането предоставя прост начин за идентифициране и организиране на класовете, които съответстват на системата или изискванията към нея.* Ambler [Amb95] описваCRC моделирането по следния начин:
  - A CRC модел реално е колекция от стандартни индекс карти, които представят класовете. Картите са разделени на три секции. В най-горната секция записвате името на класа. В тялото на картата изброявате отговорностите на класа в лявата страна и сътрудничества в дясната.

CRC Моделиране



Типове Класове

- *Entity classes*, наречени още *model* или *business* classes, извлечени са от проблемния домейн (FloorPlan and Sensor).
- *Boundary classes* се използват за създаване на интерфейси, които потребителя вижда и взаимодейства, при използване на софтуера.
- *Controller classes* управлява“unit of work” [UML03] от началото до края. Controller classes могат да бъдат проектирани да управляват:
  - Създаване и обновяване на entity objects; ;
  - Сложна комуникация между множество от обекти;
  - Валидация на данни при комуникация между обекти или между потребител и приложението.

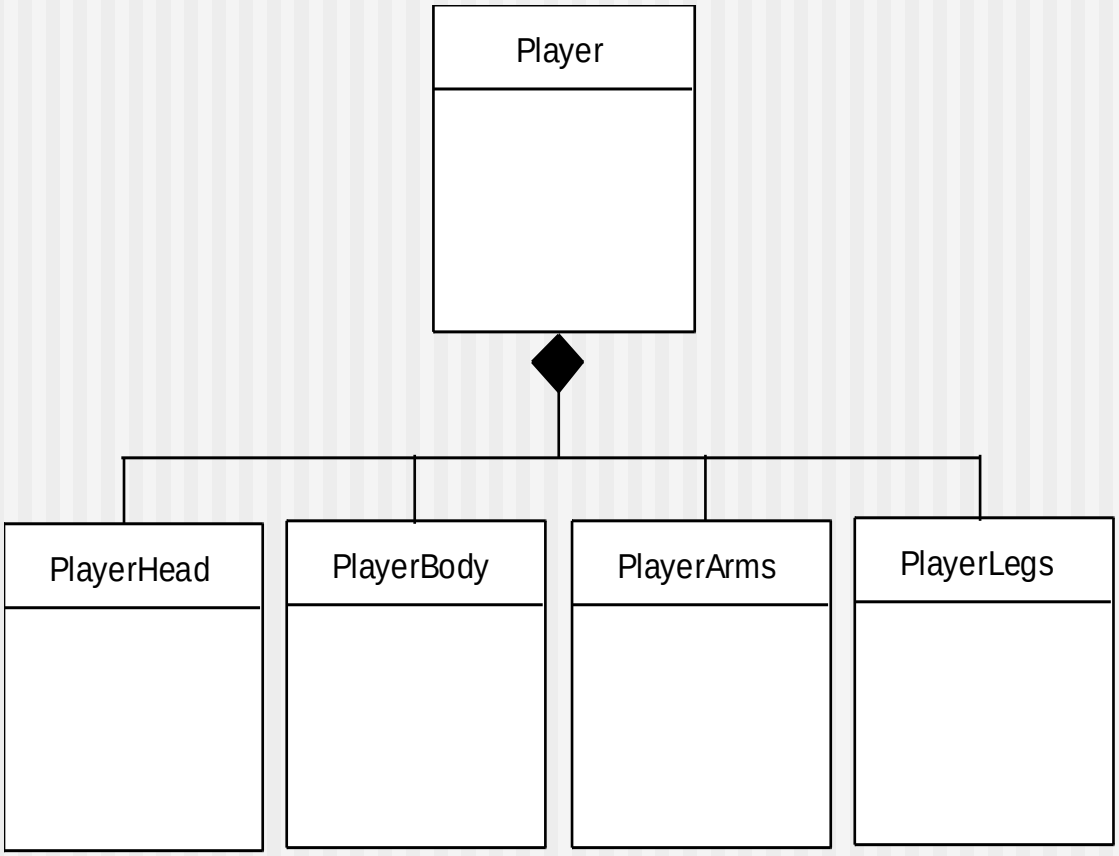
Отговорности

- Интелигентността на системата трябва да бъде разпределена между класовете, за да се достигнат решаване на проблема
- Всяка отговорност трябва да се посочи колкото е възможно по-просто
- Информацията и поведението трябва да са в един клас
- Информация за едно нещо трябва да бъде в един клас не разпределена между множество класове
- Отговорностите трябва да бъдат разпределени между свързани класове, когато е подходящо

Сътрудничество

- Класовете изпълняват техните отговорности по един от двата начина:
  - Класа може да използва собствени операции, за да манипулира собствени атрибутиA class can use its own operations to manipulate its own attributes като по този начин изпълнява собствена отговорност или
  - класа може да си сътрудничи с други класове
- Сътрудничествата идентифицират връзките между класовете
- Сътрудничествата са идентифицирани чрез определяне дали класа може да изпълни всяка от отговорностите си
- Три различни връзки между класове [WIR90]:
  - *is-part-of*
  - *has-knowledge-of*
  - *depends-upon*

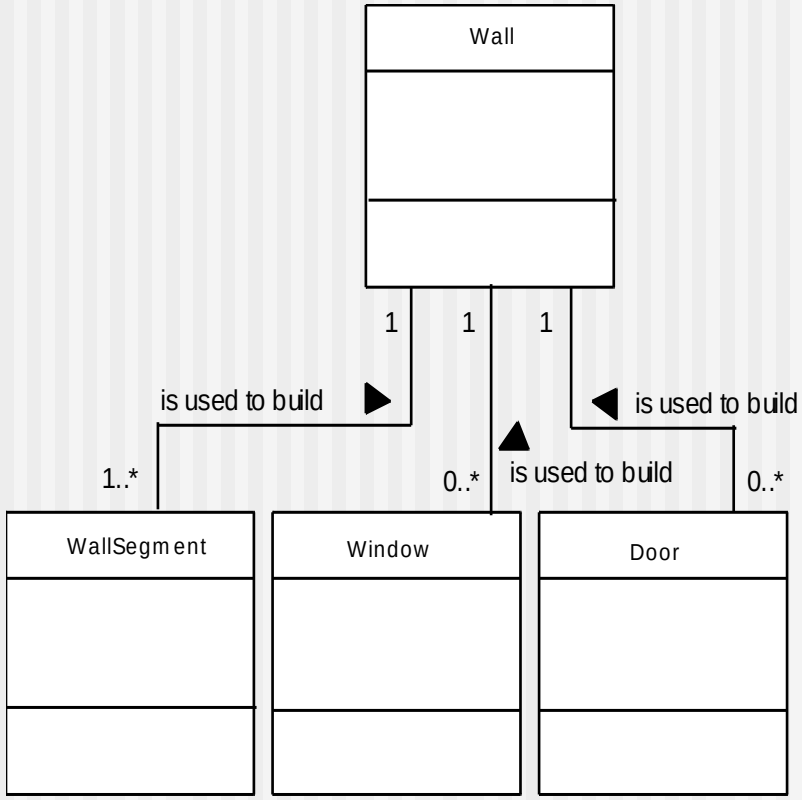
Composite Aggregate Class



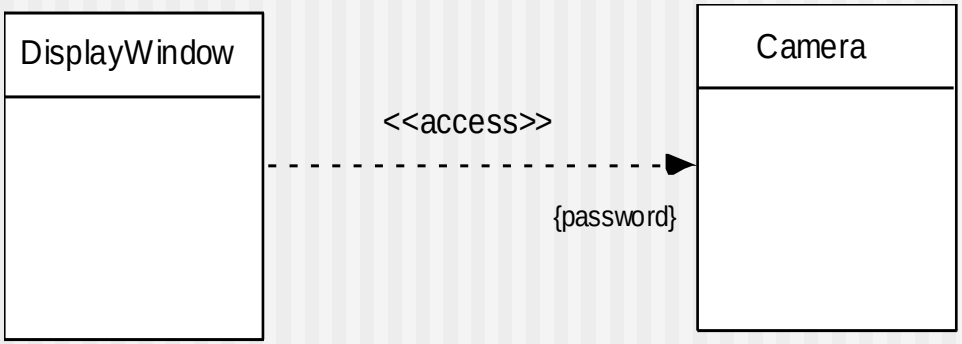
Associations и Dependencies

- Два аналитични класа често са свързани по някакъв начин
  - В UML тези връзки се наричат *associations*
  - Associations могат да бъдат усъвършенствани чрез *multiplicity* (*cardinality* се използва в моделирането)
- В много случаи съществуват клиент-сървър връзки между два аналитични класа.
  - В такива случаи клиент-класа зависи от сървър-класа по някакъв начин и се установява *dependency връзка*

Multiplicity



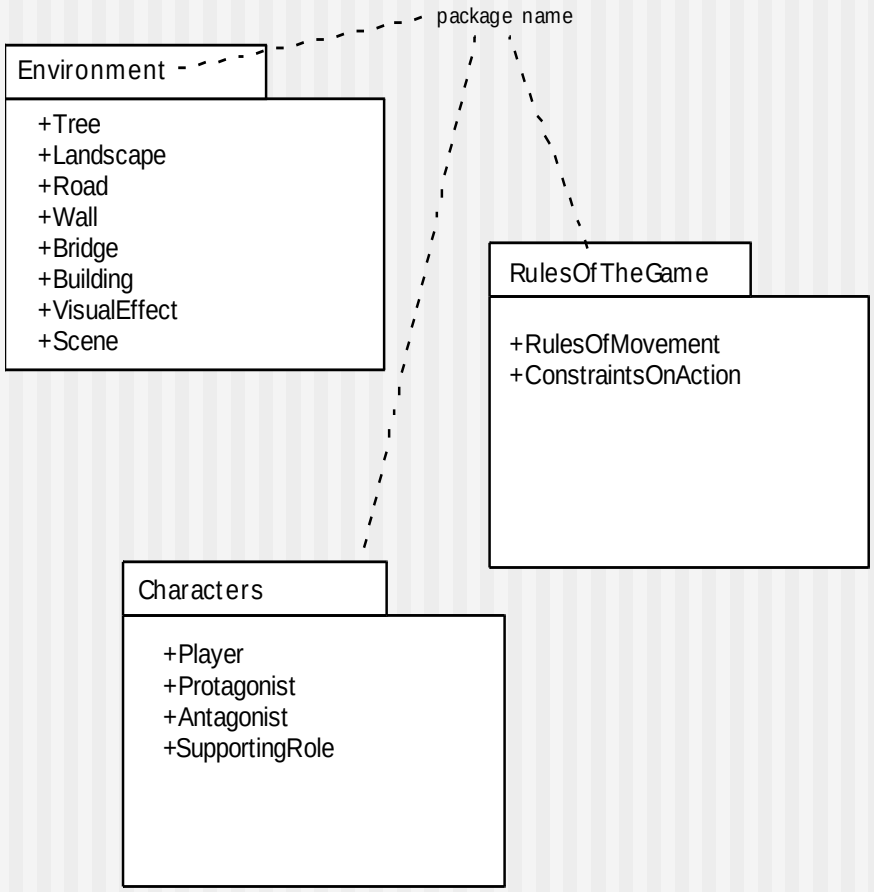
Dependencies



Аналитични Пакети

- Различните елементи на аналитичния модел (use cases, аналитични класове) са категоризирани по начин, който ги пакетира като група.

Analysis Packages



Преглед на CRC модела

- На всички участници в прегледа на модела са дадени множество CRC индексни карти.
  - Картите които си сътрудничат трябва да бъдат отделени (две сътрудничещи си карти не трябва да са в един участник).
- Всички use case сценарии (и съответните use-case diagrams) трябва да бъдат организирани в категории.
  - Когато водещия на прегледа стигне до име на обект, той символично минава към човека , който държи съответната клас индекс карта.
- Когато се намира на определен обект, този който държи карата е помолен да опише отговорностите отбелязани на картата.
  - Групата определя дали една или повече от отговорностите удовлетворяват изискванията на use case.
- Ако отговорностите и сътрудничествата отбелязани в индекс картата не могат да удовлетворят use case, се правят модификации на картата.
  - Това може да включва дефиниции на нови класове (и съответни CRC карти), или спецификации на нови или прегледани отговорности или сътрудничества на съществуваща карта.