Requirements engineering: OOA with Type Diagrams

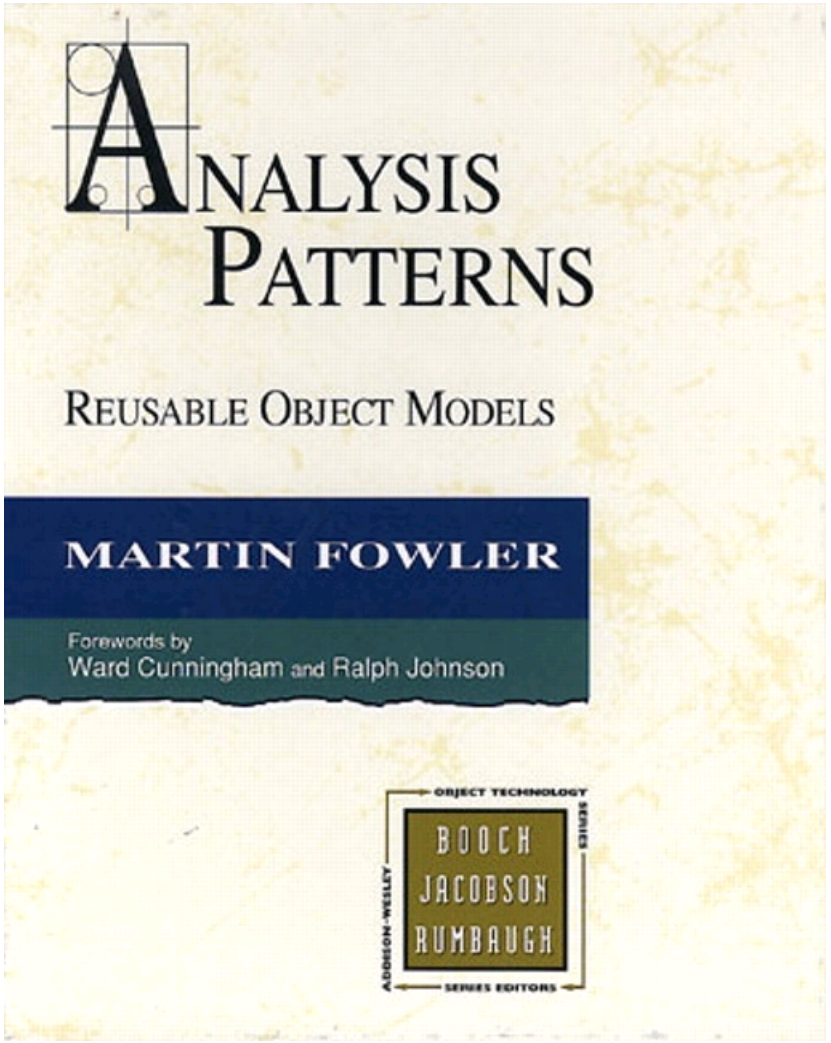
- Object-oriented analysis example
- Modelling techniques: type diagrams
- Analysis patterns

CE202 Software Engineering, Autumn term

Dr Michael Gardner, School of Computer Science and Electronic Engineering, University of Essex

OOA with Type Diagrams

- M. Fowler (1997). *Analysis Patterns*. Addison-Wesley.
  - Mainly *Chapter 3: Observations and Measurements*
  - Also Page 313 – Appendix A



Analysis of the problem domain

- Models produced from analysis and design are deliberately similar
  - The emphasis in analysis is on understanding the **problem**
  - The emphasis in design is on understanding the **solution**
- Last week we looked at requirements in general and how use case diagrams and descriptions can be used to analyse the requirements
- In analysis you may need to look behind the surface requirements (rather than just listing requirements in use-cases)

Example: understanding Snooker

Write a simulation of the game of snooker

- Could write a set of use-cases which describe the **surface features**
    - eg. “The player hits the white ball so that it travels at a certain speed; it hits the red ball at a certain angle, and the red ball travels a certain distance and direction”

- In order to understand the problem, you need to look behind the surface to understand the laws of motion that relate mass, velocity, momentum, etc.

- Unlike snooker, in most enterprises these foundations/laws are not well understood.
    - So we need to create **conceptual models** that allow us to understand and simplify the problem.

How can we develop conceptual models as reusable patterns?

- Analysis patterns:
  - Groups of concepts that represent a common construction in business modeling.
  - Relevant to one or more domains
  - Can be re-used for different enterprises that share the same problem
  - Overall principle:
    - Patterns are a starting point, not a destination
    - Models are not right or wrong, they are more or less useful

Modeling technique: the TYPE diagram

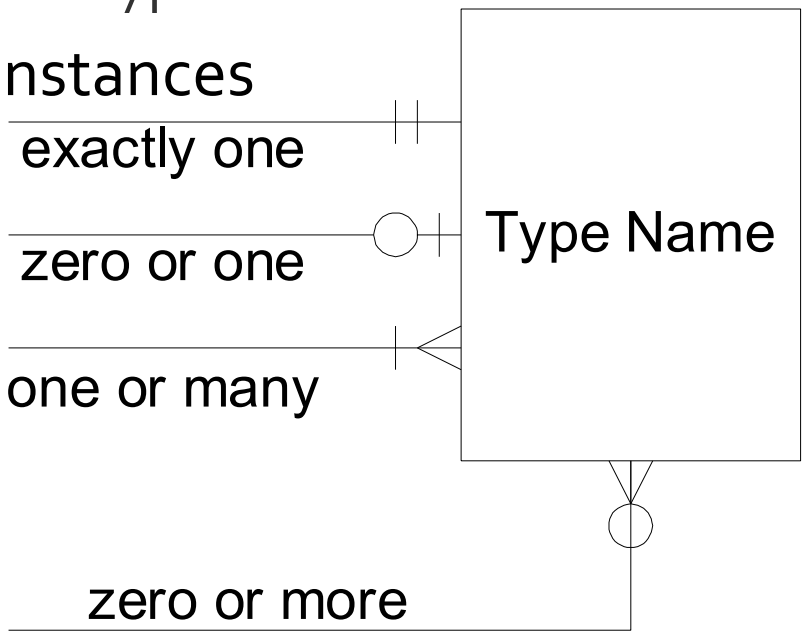
- Shows a structural view of a system
- Describes the **types of objects** and **relationships** that exist amongst them
- Based on Odell's notation (see earlier reference)
- Not based on UML

Requirements modelling technique: Type Diagrams

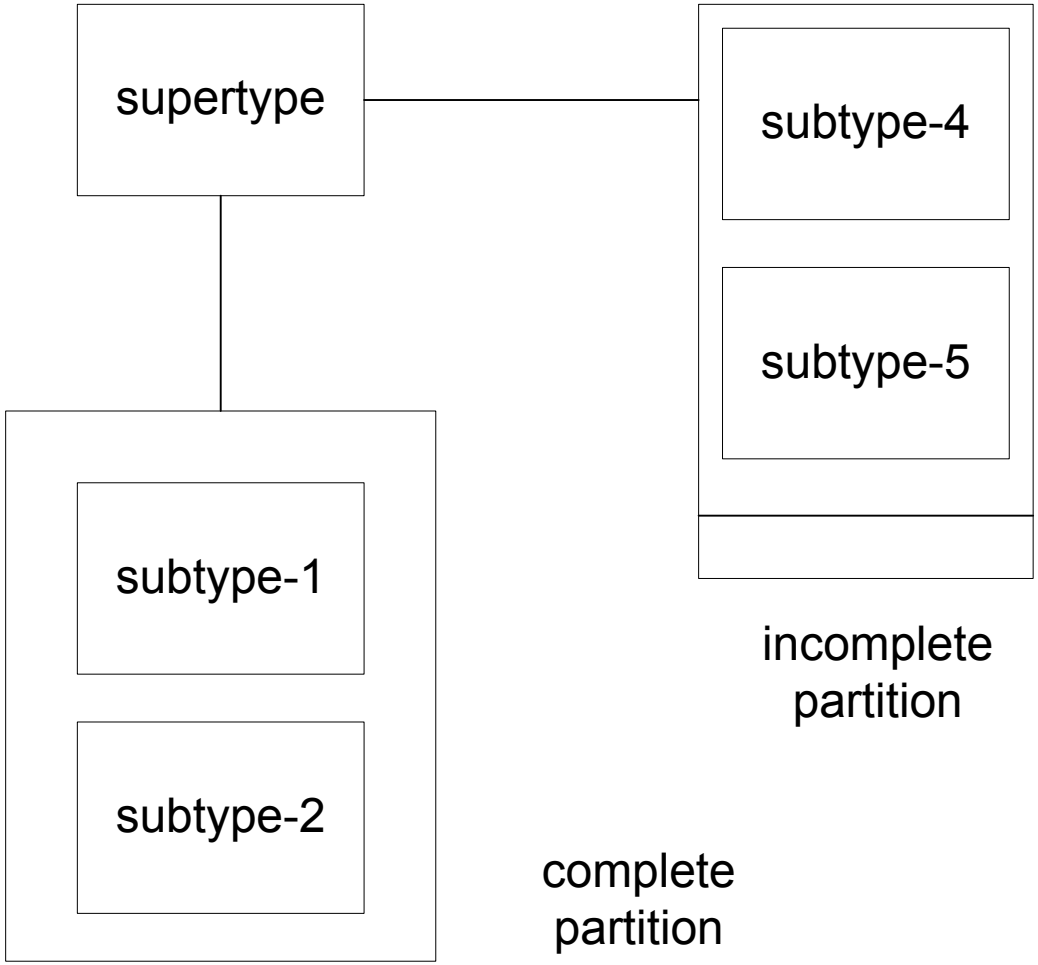
- Basic notion: Type
  - Stands for a set of instances
  - Describes the visible *interface* of a class
  - A *Type* can be implemented by many classes
  - Similar to ER diagrams with added 'types'
- Cardinality: relation between instances

Type Name

[**attribute**: Value type
attribute: Value type
...]



Type Diagrams II: subtypes



Complete partition: all instances of supertype must be instances of either subtype

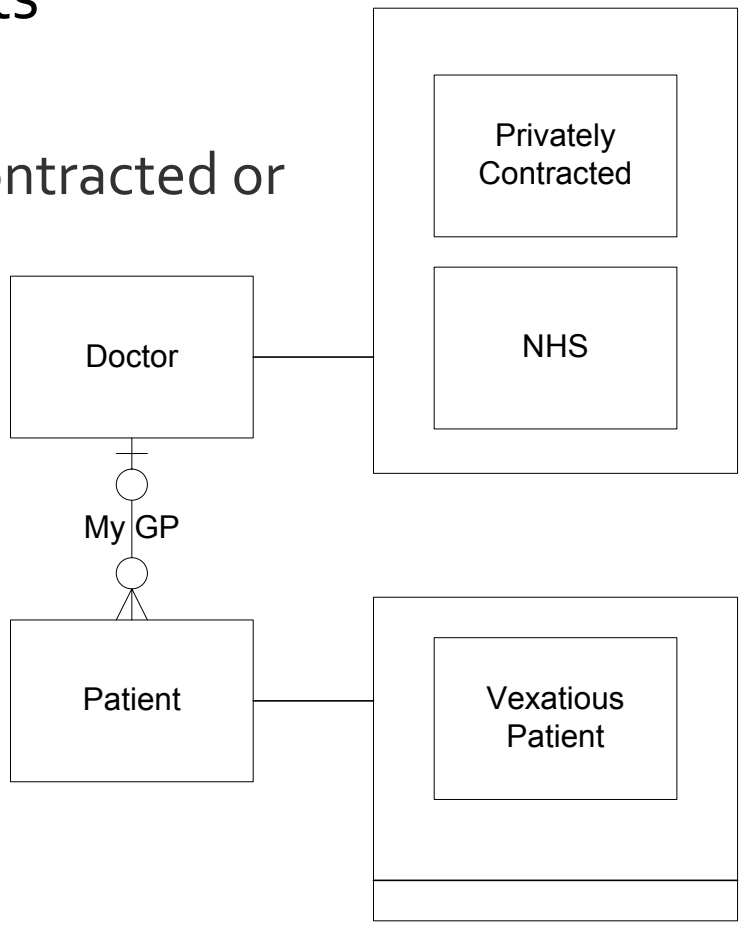
Incomplete partition: instances of supertype can be instances of either subtype OR neither

Type diagrams

- Example: Suppose a patient has one GP at the most, but a doctor can have many patients
  - Some patients are 'vexatious'
  - All doctors are either privately contracted or NHS-employed

Complete partition = all instances of supertype must be instances of at least one of the subtypes

Incomplete partition = instances of supertype can be instances of either subtype or neither



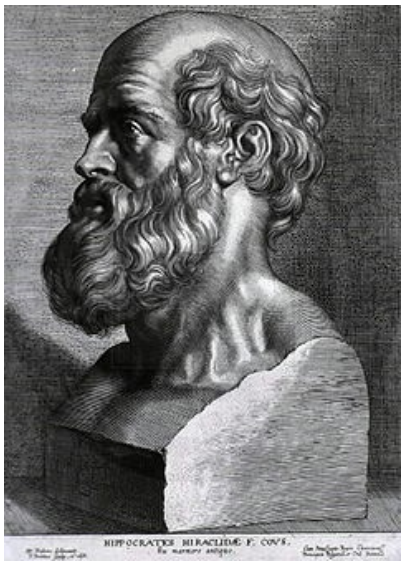
Types and objects

- A **type** (of objects) specifies a domain of objects together with the operations applicable to them.
  - An object can be a member of several types.

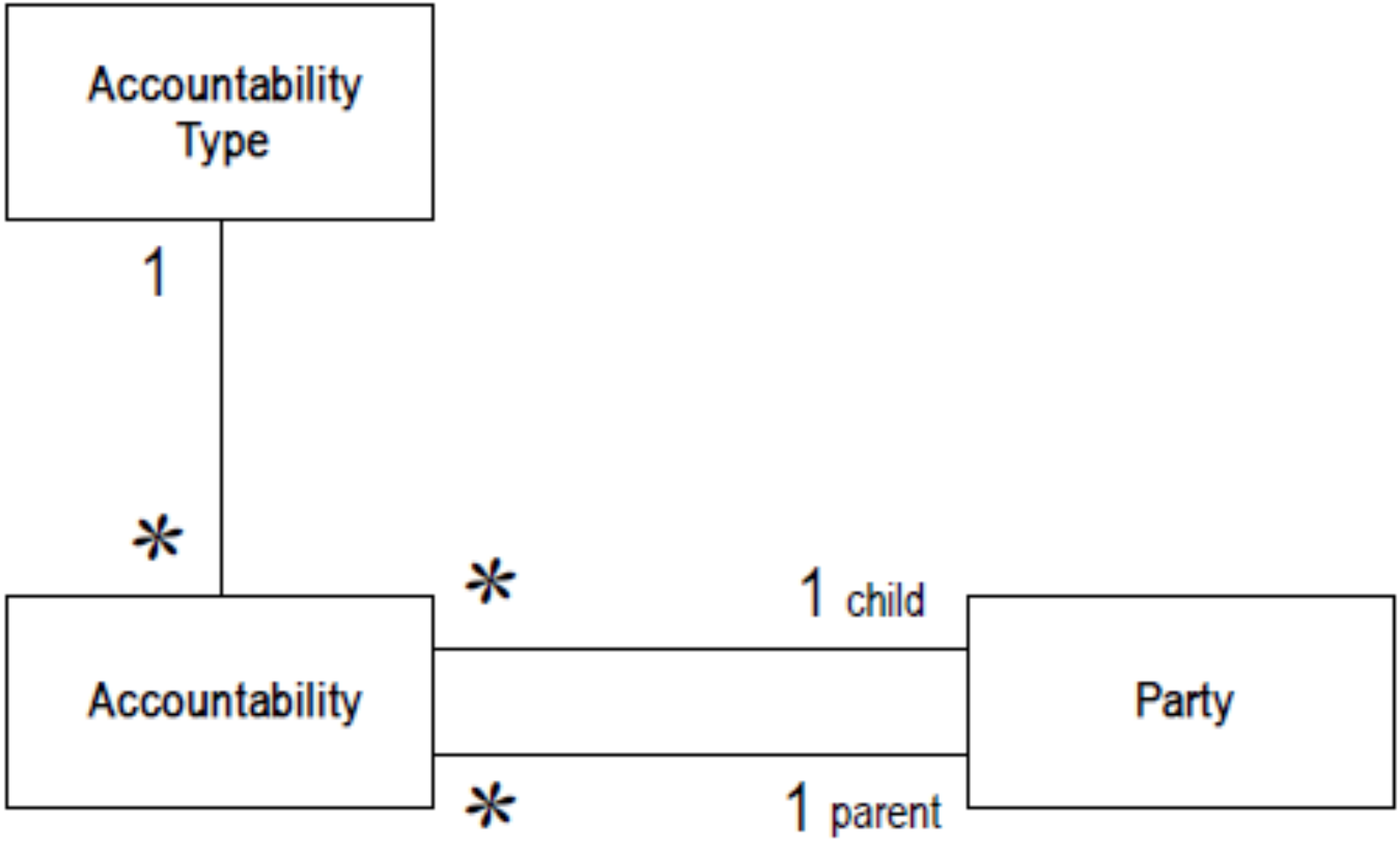
Patient

```
height
weight
blood glucose level
```

Doctor



We can use Type diagrams to create a conceptual model for a given domain



Example (pattern): observations and measurements

- (see ch. 3 of Fowler)
- Most systems record information about objects in the real world
- Typical solution is to record this using an attribute to an object
  - Eg. Person object with an attribute weight with a value of 185
- Problems with this solution
- We will see how a pattern/model can be developed to handle both quantitative and qualitative information

Example: *Quantity*

- Problem: How to record measurements?
- Context: Registering measurements performed by hospital staff
- Issues:
  - Different units may be used
  - Accuracy is essential
  - Record exactly what was measured (use original units delivered)
    - Motivation: Converting to a different unit a wrong value in one order of magnitude ("Her temperature is 3.71 degrees") reduces the chance to trace the error.

Quantity: wrong solution

- Model the measurement using an attribute of the object involved; either
  - Use only numbers:
    - height: int
  - Introduce the unit in the attribute name:
    - height_in_inches: int
- What's wrong with this solution?
  - Low level of abstraction
    - Mistakes not easily discovered
  - What does it mean to say my height is 6 or that my weight is 185? We need units.
  - One solution is to introduce the unit into the attribute name eg. 'weight in pounds'. But what happens if someone gives me their weight in kilos?

Person

```
height: int
weight: float
blood glucose level: float
```

Quantity: solution

- Combine unit and amount in a single concept *Quantity*:

Person

```
height: Quantity
weight: Quantity
blood glucose level: Quantity
```

Quantity

```
amount: long
units: Unit
+ * - / = < >   (allowed ops.)
```

- **Modeling Principle:** "When multiple attributes interact with behavior that might be used in several types, combine the attributes into a new fundamental type."

(This also maps well to an implementation in an OO language, such as Java)

Quantity: sample object

- Representing John's attributes:
  - A weight of 185 pounds:
    - amount: 185
    - unit: pound
  - 15 American dollars:
    - amount: 15
    - unit: USD

Person

```
height: Quantity
weight: Quantity
blood glucose level: Quantity
```

Example continued: *Quantity Conversion*

- Context:
  - Measurements are recorded in different units
  - Different measurements are combined in a single operation
- Issues
  - Do not affect the original measurement
- For example: 17 mm + 1.3 inch

Conversion: wrong solution

- Convert everything to a single unit
- For example:
  - Measurement of temperature: Celsius or Fahrenheit
  - At 7am Nurse Betty measured Alice's temperature: 37.4° Celsius
  - at 8am Nurse John measured Alice's temperature: 102.2° Fahrenheit
  - Ask the application to calculate the 'spike' (increase) in Alice's temperature between the measurements
- Procedure:
  - Convert 102.2° Fahrenheit → 39° Celsius
  - Represent in records only the converted temperature 39° Celsius
- What's wrong with this solution?
  - forces uses to only take (or convert them to) measurements in a single unit.

Conversion : Solution

- Solution: Introduce 'Conversion' type
  - Converts objects from one unit to another

Example 1. We can convert between inches and feet by defining a conversion ratio from feet to inches with the number 12.
Example 2. We can convert between inches and millimeters by defining a conversion ratio from inches to millimeters with the number 25.4. We can them combine this ratio with the conversion ratio from feet to inches to convert from feet to millimeters

Conversion : sample objects

- Quantity 1: Nurse Betty's measurement of Alice's temperature:
  - Unit: Celsius
  - Number: 37.4
- Quantity 2: Nurse John's measurement of Alice's temperature
  - Unit: Fahrenheit
  - Number: 102.2
- Conversion:
  - $C = (5/9)\times(F-32)$
  - $F= (9/5) \times C + 32$

Example 3: *Measurement*

- Context: There are thousands of quantities of different phenomenon types
  - Height, weight, blood glucose level, heart rate, blood pressure, liver performance, lung capacity, white blood cell count, red blood cell count, …
- Question: How to represent all the different measurements of a single patient?

Measurement: wrong solution

- Possible solution: A very large type
- What's wrong with this solution?
  1. Missing data: who measured? when?
  2. Different measurements at different times
  3. Inflexible to adding new measuring equipment

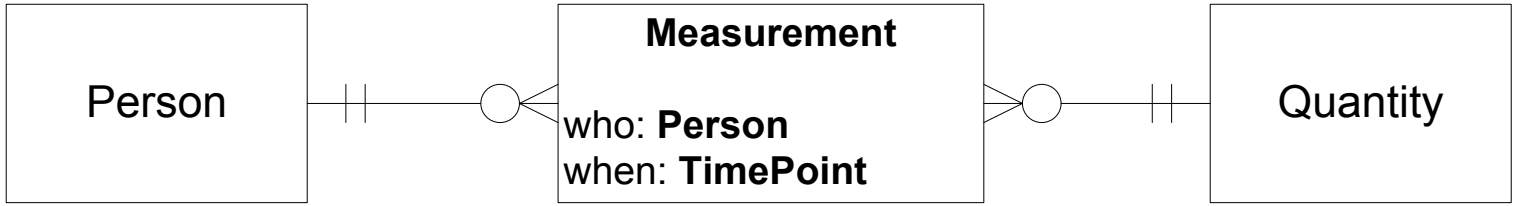
This is part of the analysis problem of when to model something as an attribute or an association

Person

```
height: quantity
weight: quantity
blood glucose level: quantity
heart rate: quantity
blood pressure: quantity
liver performance: quantity
lung capacity: quantity
white blood cell count: quantity
red blood cell count: quantity
...
```

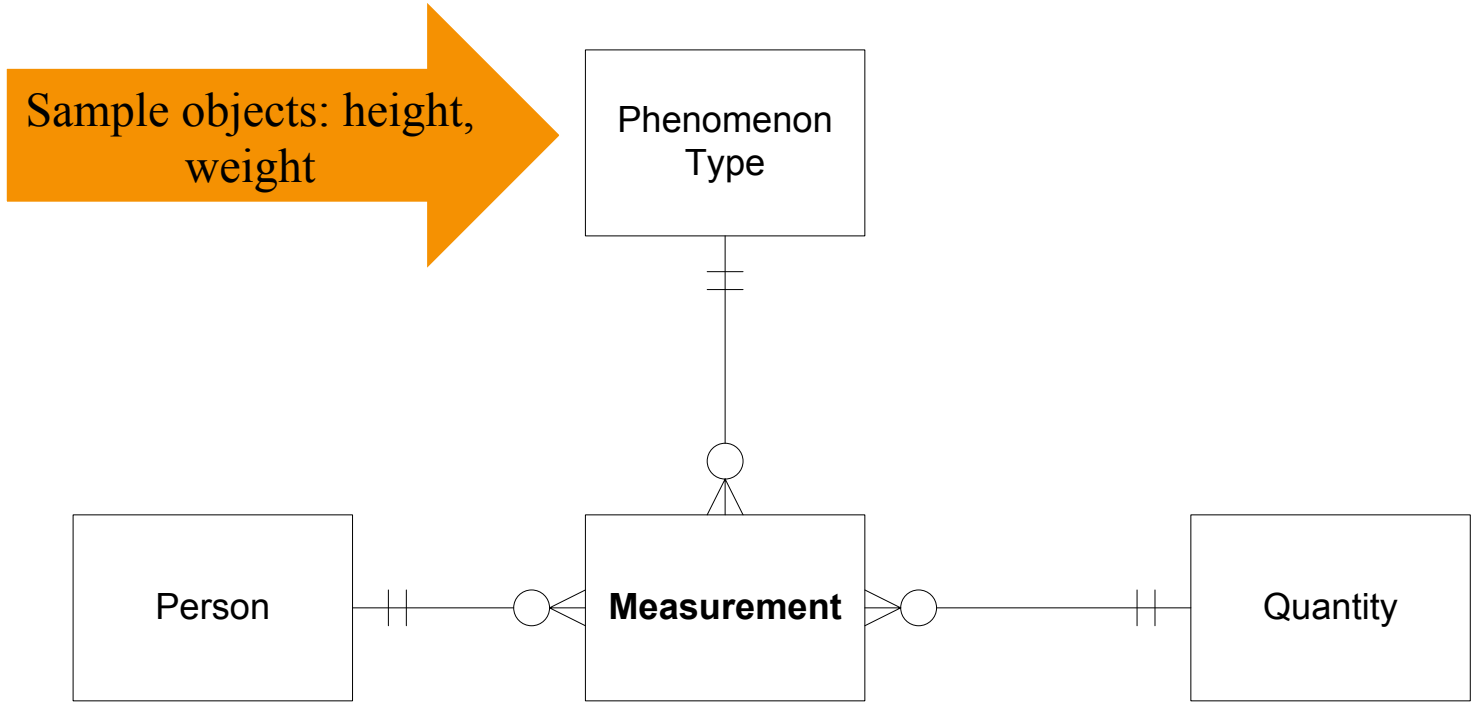
Measurement: solution, part 1

- Introduce a Measurement type
- Contains: details of the act of measurement
  - who performed it
  - when it took place
  - which patient
  - …



Measurement: Solution, part 2

- Identify measurement types with a PhenomenonType



This model/pattern is useful if a large number of possible measurements would make person too complex. The phenomenon types are things we know we can measure. Such knowledge is at the **analysis/knowledge** level of the problem being investigated.

Measurement: Sample objects

- "John Smith is 6 feet tall": measurement of
  - person: John Smith
  - phenomenon type: height
  - quantity:
    - amount: 6
    - unit: feet
- "John Smith has PEF rate of 180 liters per minute": measurement of
  - person: John Smith
  - phenomenon type: "peak expiatory flow"
  - quantity:
    - amount: 180
    - unit: compound unit
      - direct: liter
      - inverse: minute

Modeling principles for this pattern

- Measurements are created as part of the day to day operation and will change frequently (operational level)
- Phenomenon types (eg. blood pressure) are created infrequently (knowledge level)

- **Principle 1:** The operational level has those concepts that change on a day to day basis. Their configuration is constrained by a knowledge level that changes much less frequently
- **Principle 2:** If a *type* has many similar associations, make all these associations objects of a new type. Create a knowledge level type to differentiate between them.

Handling Observations

- Based on principle 2:
  - **Principle 2:** If a ***type*** has many similar **associations**, make all these associations objects of a **new type**. Create a knowledge level type to differentiate between them.
    - A **person** has many similar **measurements**, therefore make a new type '**Measurement**' to differentiate between them.

- In the class we will look at how this pattern can be extended to also handle qualitative information such as observations

Summary

- In analysis you may need to look behind the surface requirements (rather than just listing requirements in use-cases)
- So we need to create conceptual models that allow us to understand and simplify the problem
- Type diagrams are one way of creating these conceptual models
- We looked at examples of using Type diagrams to model observations and measurements
- These models can be reused as analysis patterns for other similar problems

OOA with Type Diagrams

- M. Fowler (1997). *Analysis Patterns*. Addison-Wesley.
  - Mainly *Chapter 3: Observations and Measurements*
- Martin, J & Odell J (1995). *Object-oriented methods: A foundation. Prentice-Hall.*