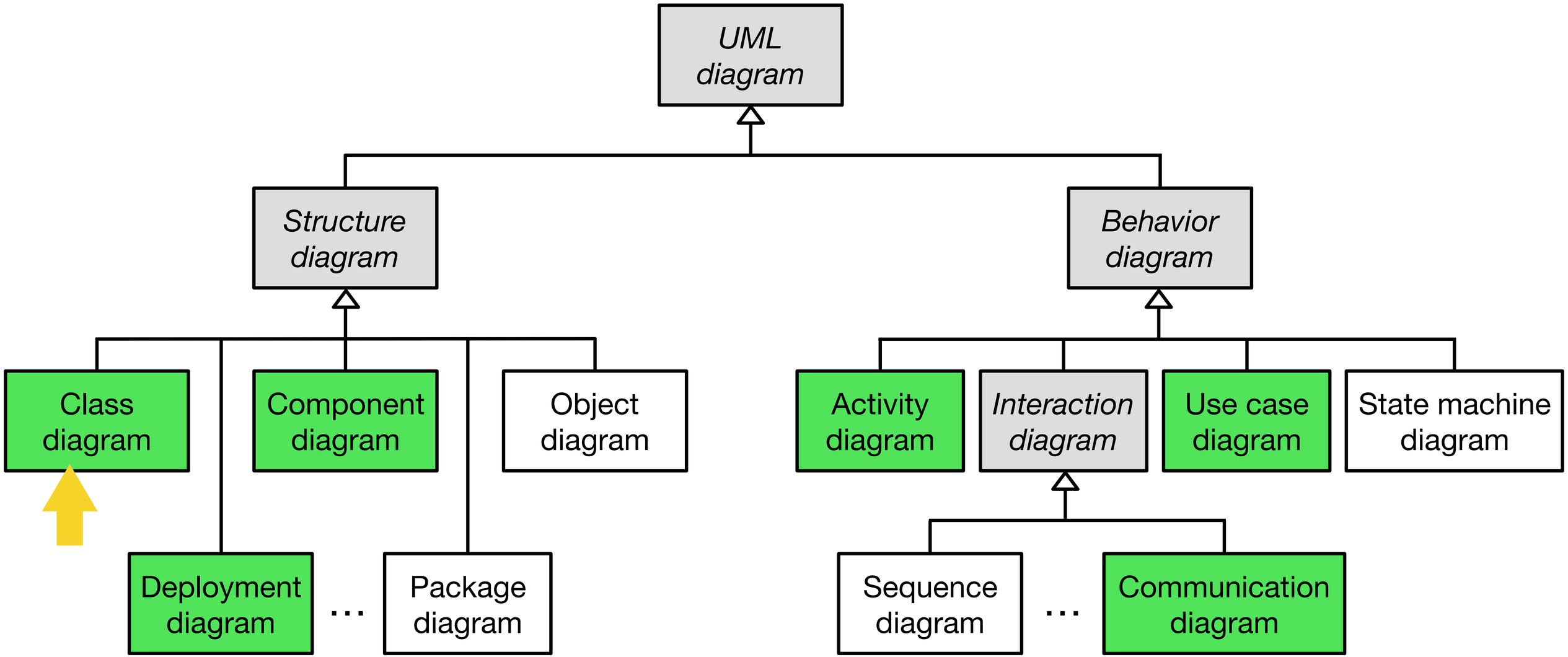
Software Engineering Essentials

Analysis Object Model

Bernd Bruegge, Stephan Krusche, Andreas Seitz, Jan Knobloch
Chair for Applied Software Engineering — Faculty of Informatics

UML diagrams covered in this course



Purpose of class diagram

Contains the structure of objects and their dependencies to communicate the main concepts of a software system

- **Classes**: Abstract representation of an object which defines its structure and its functionality
- **Associations**: Define the relationships between objects and their corresponding dependencies / hierarchies

Can also be used for forward- and reverse- engineering and during object design

Purpose of the analysis object model

Models the individual concepts of the **application domain** that are manipulated by the system, their properties and their relationships

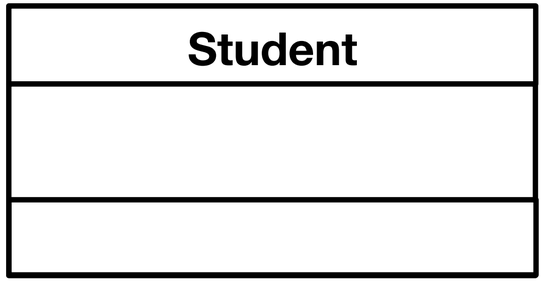
Visual dictionary of the main concepts visible to the user

Does **not** contain concepts of the **solution domain** which are included in system and object design

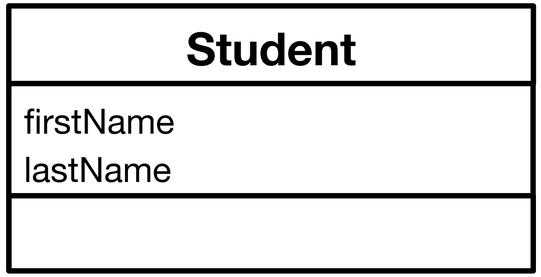
- Does intentionally not include method signatures, visibility of attributes and methods and types of attributes

Class diagram elements

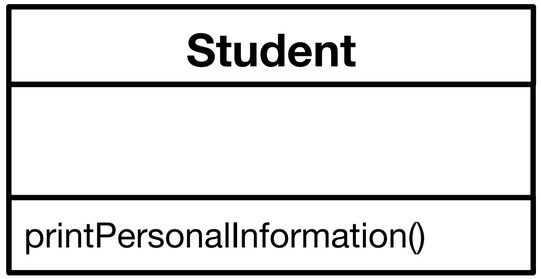
Class:



Attributes:

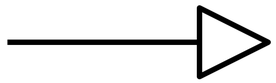


Methods:



Associations:

Inheritance:



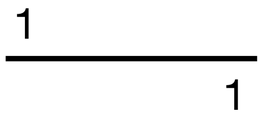
Aggregation:

Composition:

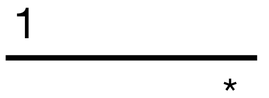
Reference:

Multiplicities:

one to one:

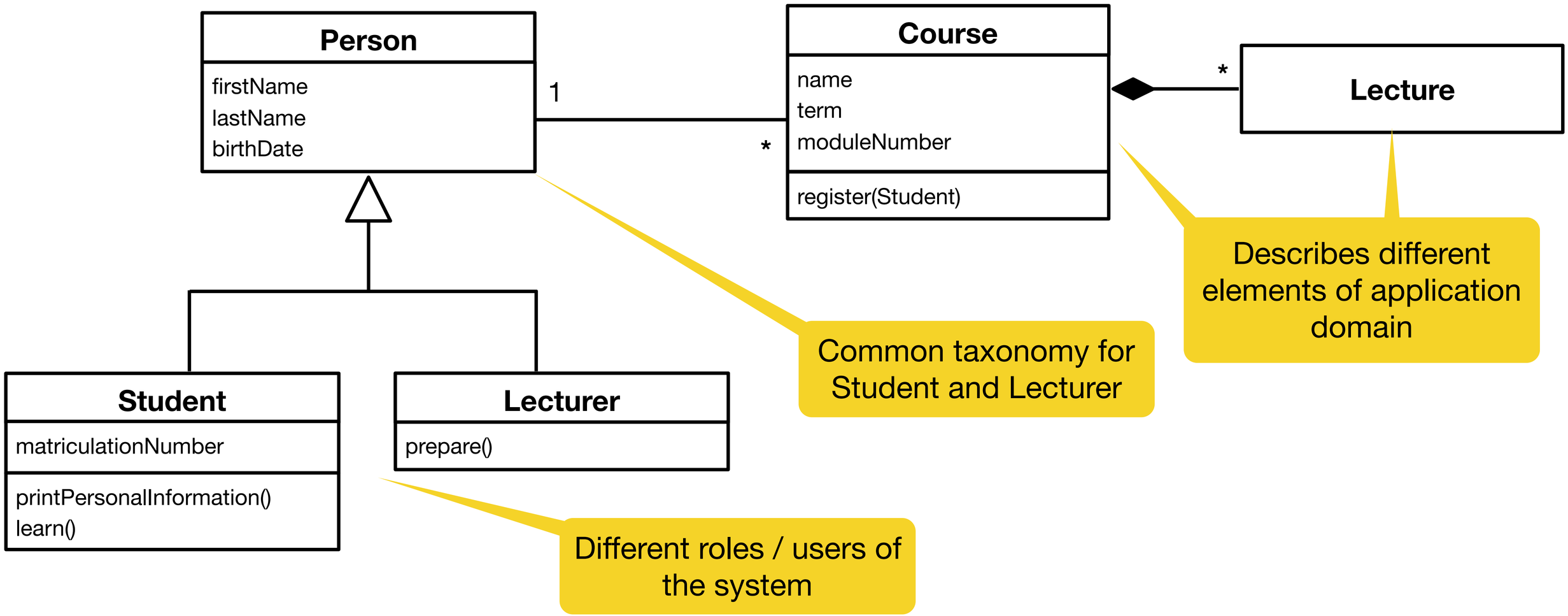


one to many:



many to many:

Example of an analysis object model



Software Engineering Essentials

Analysis Object Model

Bernd Bruegge, Stephan Krusche, Andreas Seitz, Jan Knobloch
Chair for Applied Software Engineering — Faculty of Informatics