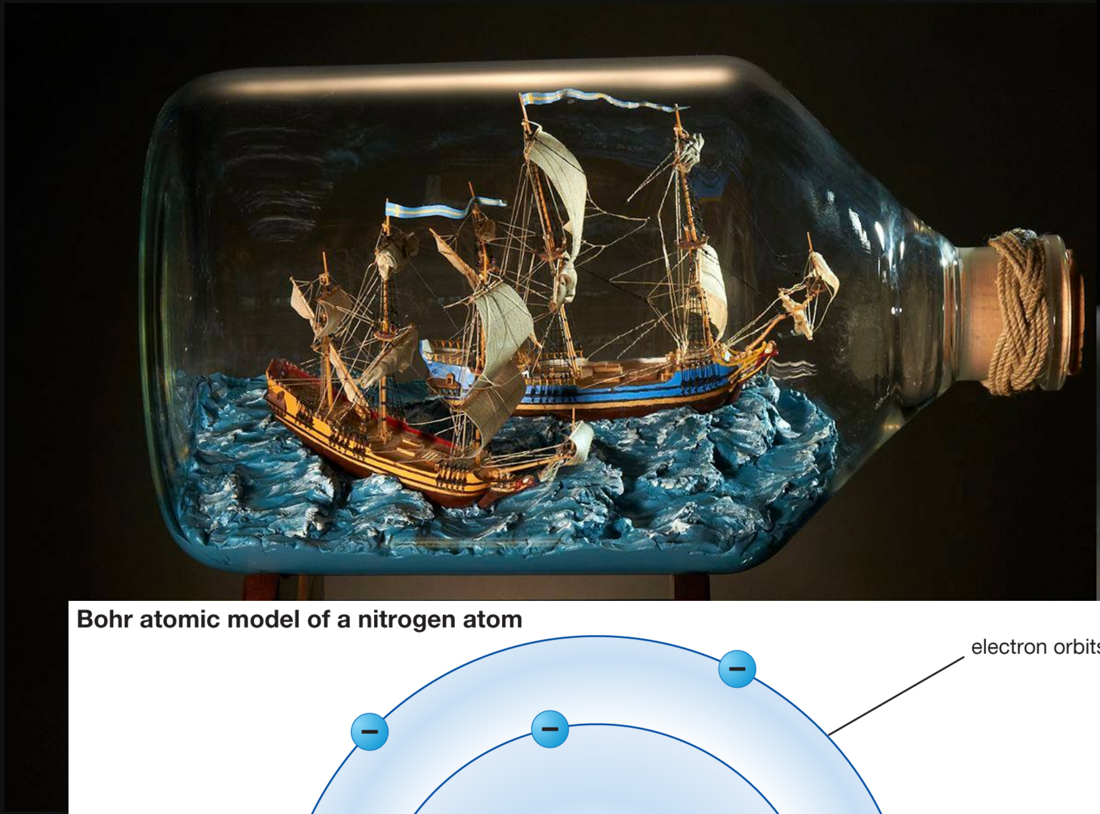
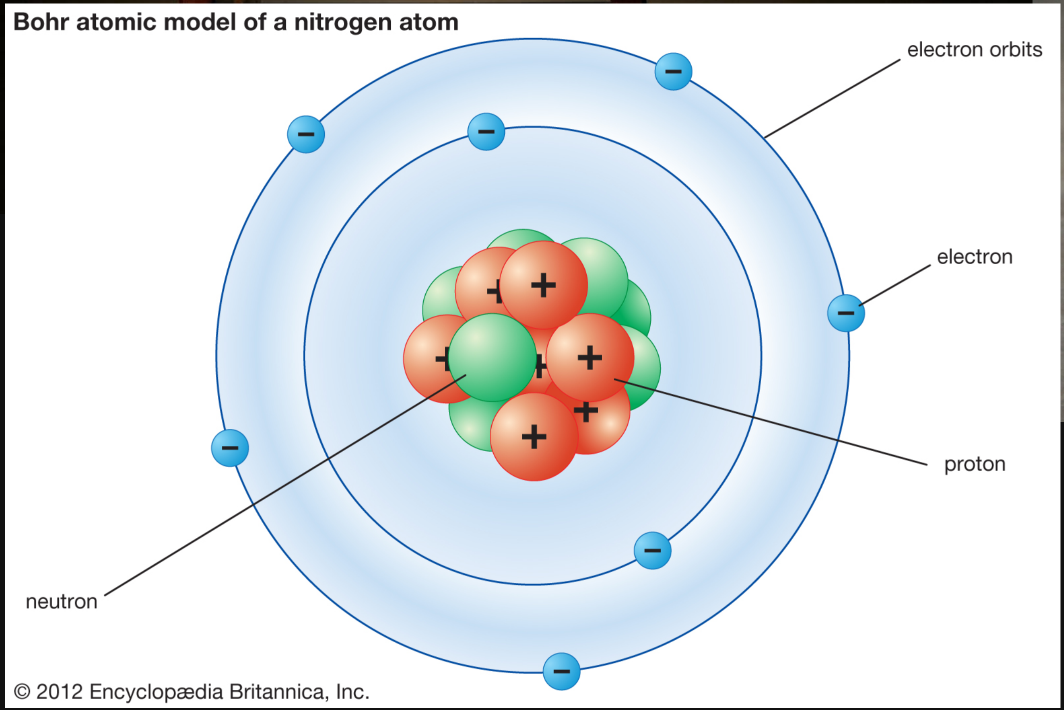
COMP1531

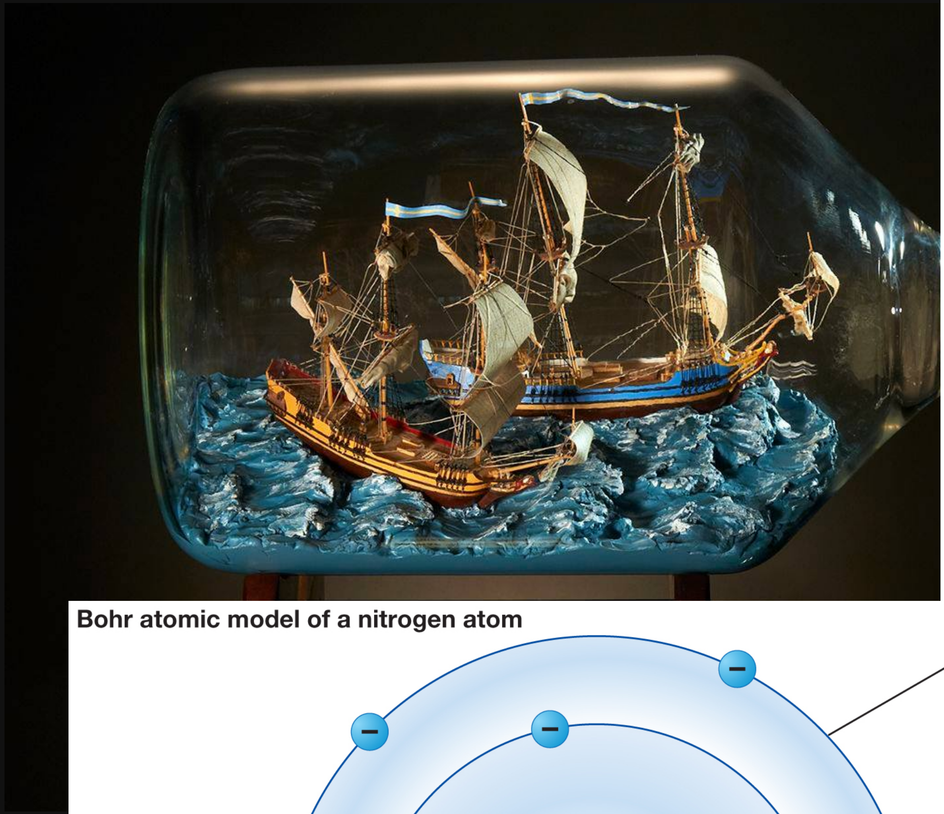
7.3 Conceptual Modelling

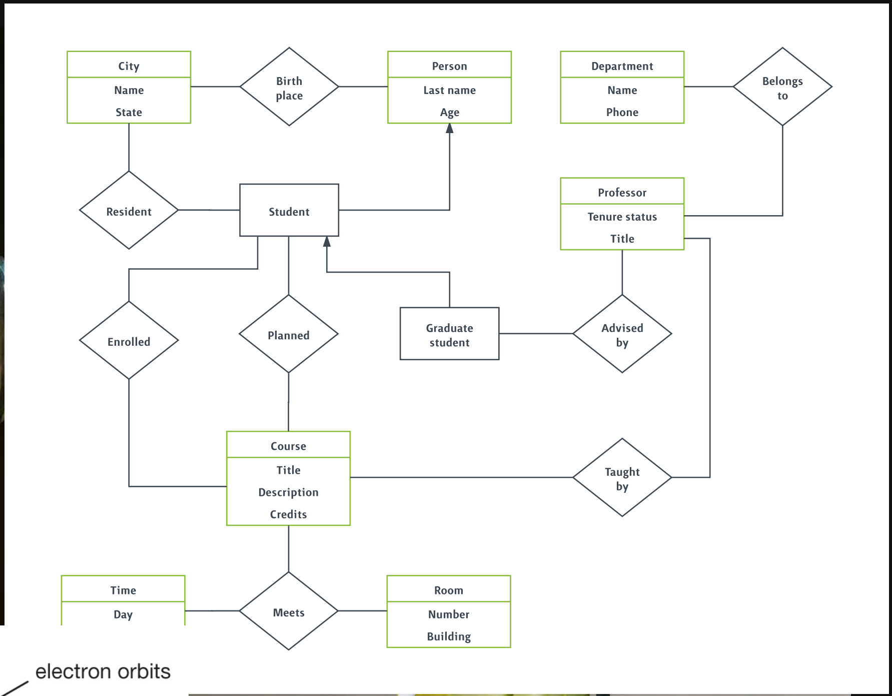
What's a model?

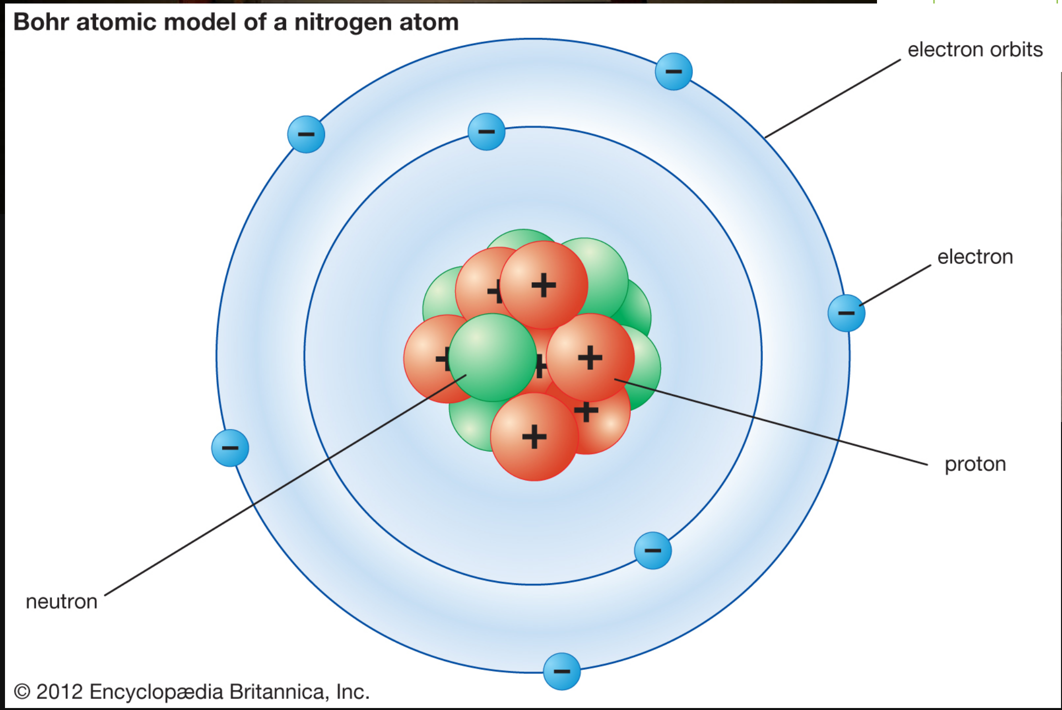

Kalmar Nyckel & Fogel Grip

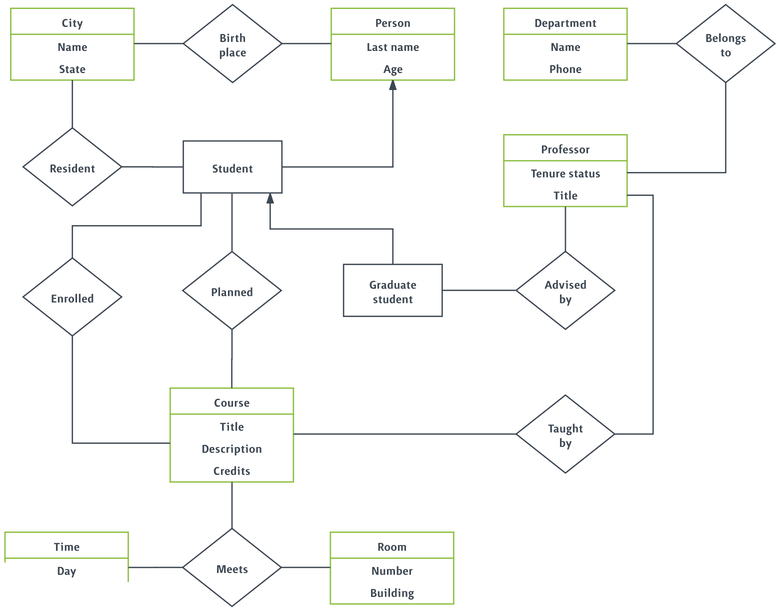




Bohr atomic model of a nitrogen atom
electron orbits
electron
proton
neutron
© 2012 Encyclopædia Britannica, Inc.

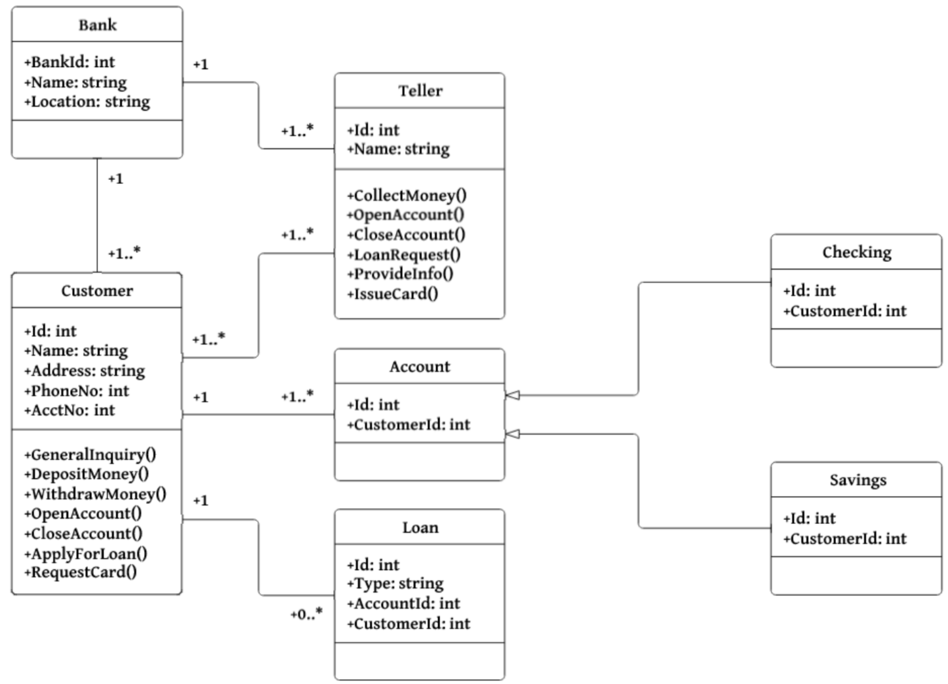



City
Name
State
Birth place
Person
Last name
Age
Department
Name
Phone
Belongs to
Resident
Student
Professor
Tenure status
Title
Enrolled
Planned
Graduate student
Advised by
Course
Title
Description
Credits
Taught by
Time
Day
Meets
Room
Number
Building


Bohr atomic model of a nitrogen atom
electron orbits
electron
proton
neutron
© 2012 Encyclopædia Britannica, Inc.



n orbits



ctron

oton

Figure 1: Mathematical Model

$$\frac{dC}{dt} = \frac{s_c C}{1+(R_b+N_b+T_b+L_b)/b_{c\infty}} + \frac{k_{cp}}{v_m}(P-C) - (k_{crc}+k_{cnc}+k_{ctc}+k_{clc})C - \mu_c C$$

$$\frac{dP}{dt} = \frac{k_{cp}}{v_b}(C-P) - \mu_p P$$

$$\frac{dR_c}{dt} = \frac{s_{rc}R_c}{1+R_b/b_{r\infty}} + k_{crc}C - k_{rcm}R_c - \mu_{rc}R_c \qquad \frac{dR_m}{dt} = k_{rcm}R_c - \frac{k_{rmb}}{v_m}R_m - \mu_{rm}R_m$$

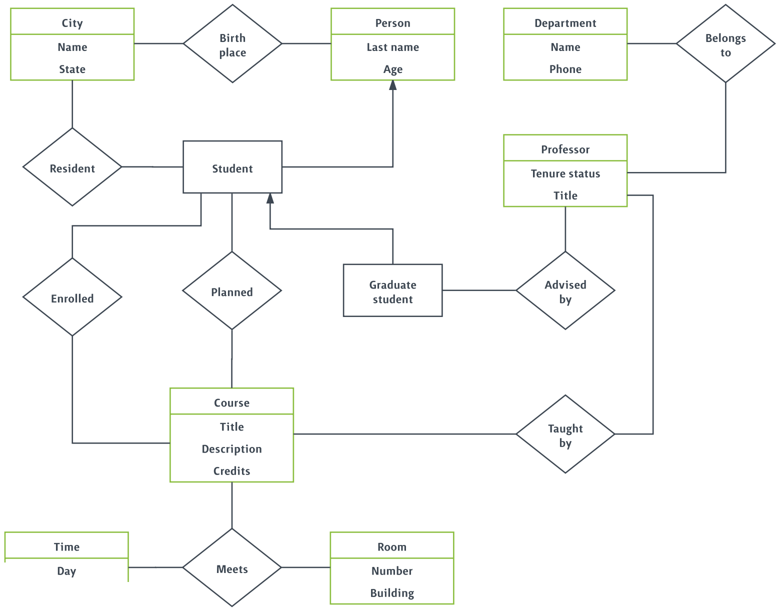
$$\frac{dN_c}{dt} = \frac{s_{nc}N_c}{1+N_b/b_{n\infty}} + k_{cnc}C - k_{ncm}N_c - \mu_{nc}N_c \qquad \frac{dN_m}{dt} = k_{ncm}R_c - \frac{k_{nmb}}{v_m}N_m - \mu_{nm}N_m$$

$$\frac{dT_c}{dt} = \frac{s_{tc}T_c}{1+T_b/b_{t\infty}} + k_{ctc}C - k_{tcm}T_c - \mu_{tc}T_c \qquad \frac{dT_c}{dt} = k_{tcm}T_c - \frac{k_{tmb}}{v_m}T_c - \mu_{tc}T_c$$

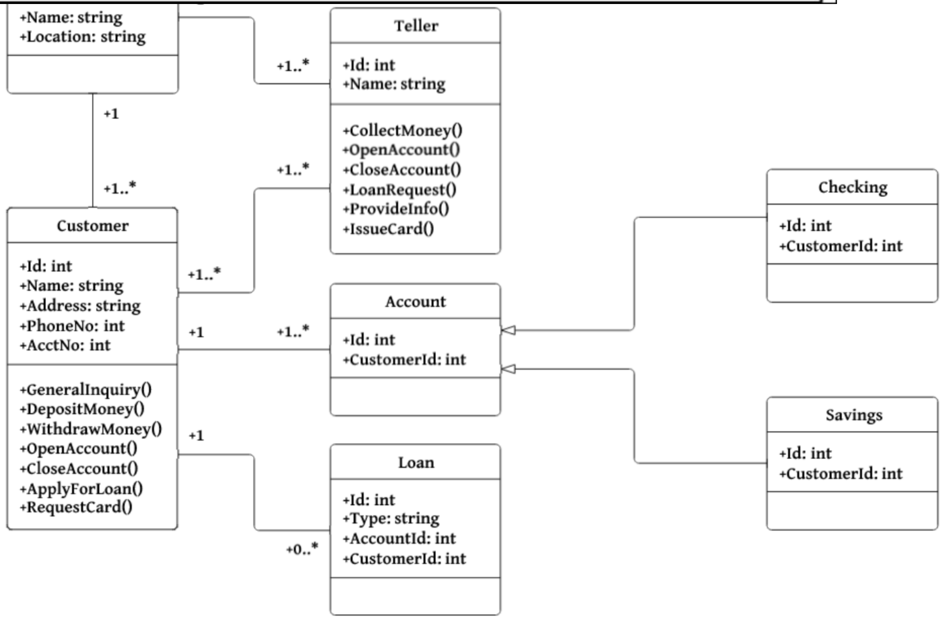
$$\frac{dL_c}{dt} = \frac{s_{lc}L_c}{1+L_b/b_{l\infty}} + k_{clc}C - k_{lcm}L_c - \mu_{lc}L_c \qquad \frac{dL_c}{dt} = k_{lcm}L_c - \frac{k_{lmb}}{v_m}L_c - \mu_{lc}L_c$$

$$\frac{dR_b}{dt} = \frac{k_{rmb}}{v_m}R_m - \mu_{rb}R_b \qquad \frac{dT_b}{dt} = \frac{k_{tmb}}{v_m}T_m - \mu_{tb}T_b$$

$$\frac{dN_b}{dt} = \frac{k_{nmb}}{v_m}N_m - \mu_{nb}N_b \qquad \frac{dL_b}{dt} = \frac{k_{lmb}}{v_m}L_m - \mu_{lb}L_b$$



n orbits



ctron

oton

Figure 1: Mathematical Model

$$\frac{dC}{dt} = \frac{s_c C}{1+(R_b+N_b+T_b+L_b)/b_{c\infty}} + \frac{k_{cp}}{v_m}(P-C) - (k_{crc}+k_{cnc}+k_{ctc}+k_{clc})C - \mu_c C$$

$$\frac{dP}{dt} = \frac{k_{cp}}{v_b}(C-P) - \mu_p P$$

$$\frac{dR_c}{dt} = \frac{s_{rc}R_c}{1+R_b/b_{r\infty}} + k_{crc}C - k_{rcm}R_c - \mu_{rc}R_c \qquad \frac{dR_m}{dt} = k_{rcm}R_c - \frac{k_{rmb}}{v_m}R_m - \mu_{rm}R_m$$

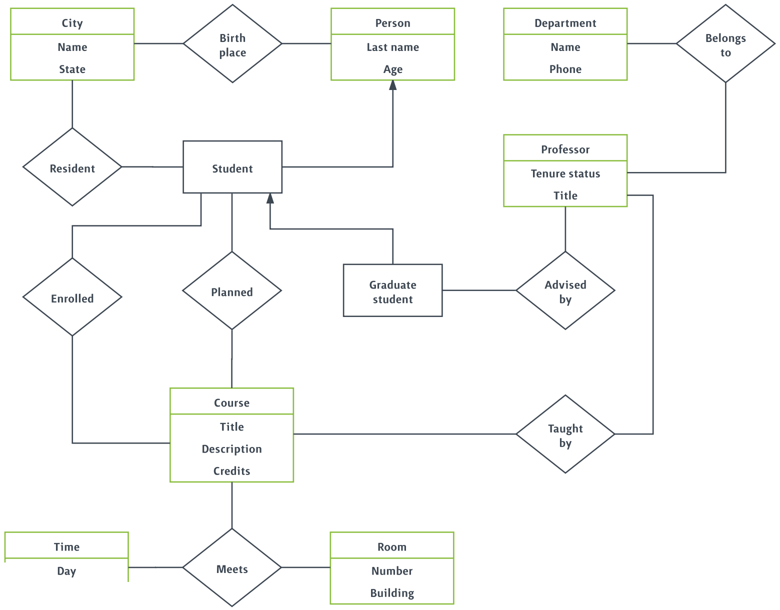
$$\frac{dN_c}{dt} = \frac{s_{nc}N_c}{1+N_b/b_{n\infty}} + k_{cnc}C - k_{ncm}N_c - \mu_{nc}N_c \qquad \frac{dN_m}{dt} = k_{ncm}R_c - \frac{k_{nmb}}{v_m}N_m - \mu_{nm}N_m$$

$$\frac{dT_c}{dt} = \frac{s_{tc}T_c}{1+T_b/b_{t\infty}} + k_{ctc}C - k_{tcm}T_c - \mu_{tc}T_c \qquad \frac{dT_c}{dt} = k_{tcm}T_c - \frac{k_{tmb}}{v_m}T_c - \mu_{tc}T_c$$

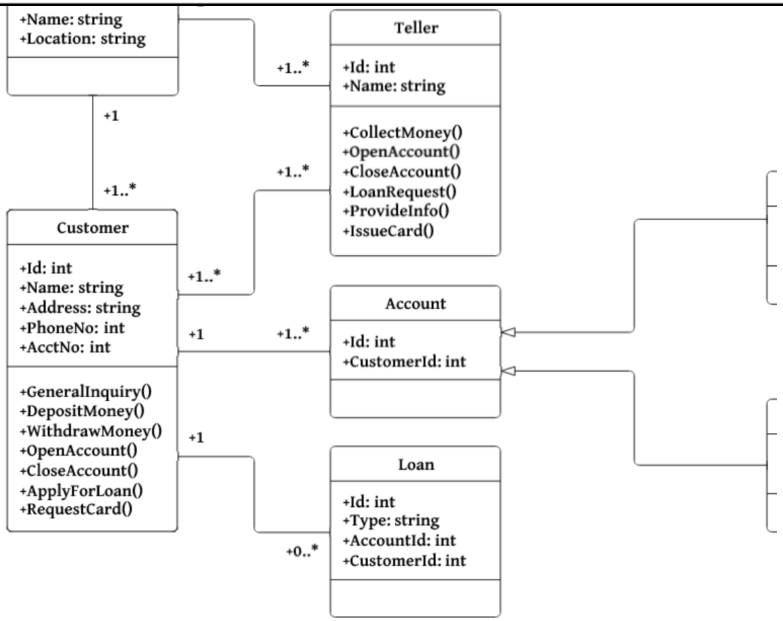
$$\frac{dL_c}{dt} = \frac{s_{lc}L_c}{1+L_b/b_{l\infty}} + k_{clc}C - k_{lcm}L_c - \mu_{lc}L_c \qquad \frac{dL_c}{dt} = k_{lcm}L_c - \frac{k_{lmb}}{v_m}L_c - \mu_{lc}L_c$$

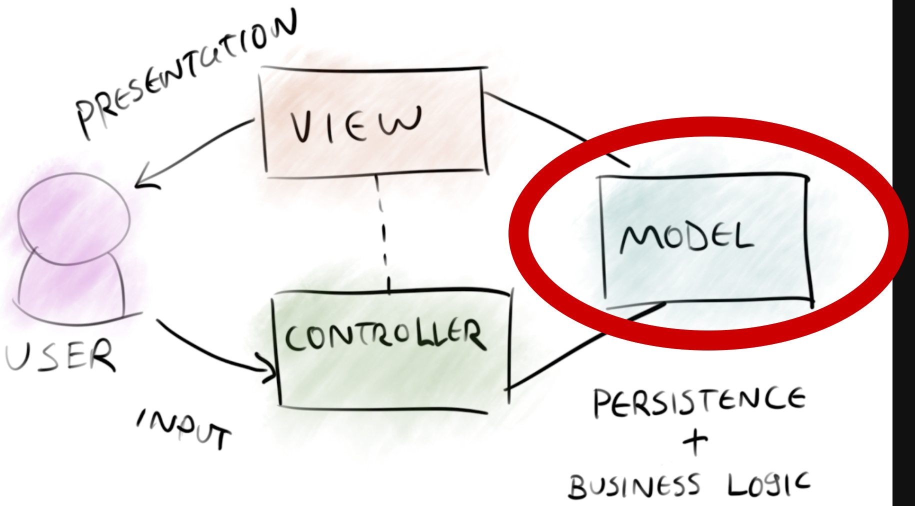
$$\frac{dR_b}{dt} = \frac{k_{rmb}}{v_m}R_m - \mu_{rb}R_b \qquad \frac{dT_b}{dt} = \frac{k_{tmb}}{v_m}T_m - \mu_{tb}T_b$$

$$\frac{dN_b}{dt} = \frac{k_{nmb}}{v_m}N_m - \mu_{nb}N_b \qquad \frac{dL_b}{dt} = \frac{k_{lmb}}{v_m}L_m - \mu_{lb}L_b$$



on orbits





Conceptual Modelling

- *A model that is conceptual*
  - *... with a real world correspondence*
  - *... without a real world correspondence*
- *A model of a concept*

Conceptual models software engineers care about

- Data models (last week)
- Mathematical models
- Domain models (today)
- Data flow models
- State transition models

How models are used

- To predict future states of affairs.
- Understand the current state of affairs.
- Determine the past state of affairs.
- **To convey the fundamental principles and basic functionality of systems (communication)**

Communicating models

- Four fundamental objectives of communicating with a conceptual model:
  1. Enhance an individual's understanding of the representative system
  2. Facilitate efficient conveyance of system details between stakeholders
  3. Provide a point of reference for system designers to extract system specifications
  4. Document the system for future reference and provide a means for collaboration

Kung and Solvberg (1986)

Domain Modelling

- A conceptual model of a domain that incorporates both **data** and **behaviour**
- Typically used as part of a *design* process

What is a domain?

- *Domain* – A sphere of knowledge particular to the problem being solved
- *Domain expert* – A person expert in the domain
- For example, in the domain of cake decorating, cake decorators are the domain experts

Problem

- A motivating example:
  - Tourists have schedules that involve at least one and possibly several cities
  - Hotels have a variety of rooms of different grades: standard and premium
  - Tours are booked at either a standard or premium rate, indicating the grade of hotel room
  - In each city of their tour, a tourist is booked into a hotel room of the chosen grade
  - Each room booking made by a tourist has an arrival date and a departure date
  - Hotels are identified by a name (e.g. Melbourne Hyatt) and rooms by a number
  - Tourists may book, cancel or update schedules in their tour

Ubiquitous language

- **Things in our design must represent real things in the domain expert's mental model**
- For example, if the domain expert calls something an "order" then in our domain model (and ultimately our implementation) we should have something called an Order
- Similarly, our domain model should not contain an OrderHelper, OrderManager, etc.
- Technical details do not form part of the domain model as they are not part of the design.

Noun/verb analysis

- Finding the ubiquitous language of the domain by finding the nouns and verbs in the requirements
- The nouns are possible entities in the domain model and the verbs possible behaviours

Problem

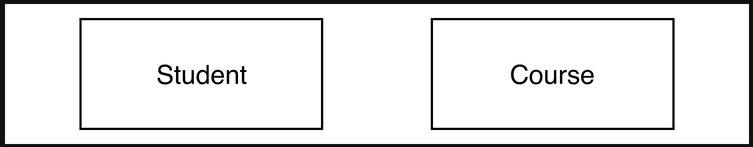
- The **nouns** and verbs:
  - **Tourists** have **schedules** that involve at least one and possibly several **cities**
  - **Hotels** have a variety of **rooms** of different **grades**: standard and premium
  - **Tours** are booked at either a standard or premium rate, indicating the **grade** of hotel **room**
  - In each **city** of their **tour**, a **tourist** is booked into a hotel **room** of the chosen **grade**
  - Each room **booking** made by a tourist has an arrival **date** and a departure **date**
  - **Hotels** are identified by a **name** (e.g. Melbourne Hyatt) and **rooms** by a **number**
  - **Tourists** may book, cancel or update **schedules** in their **tour**

Representing the domain

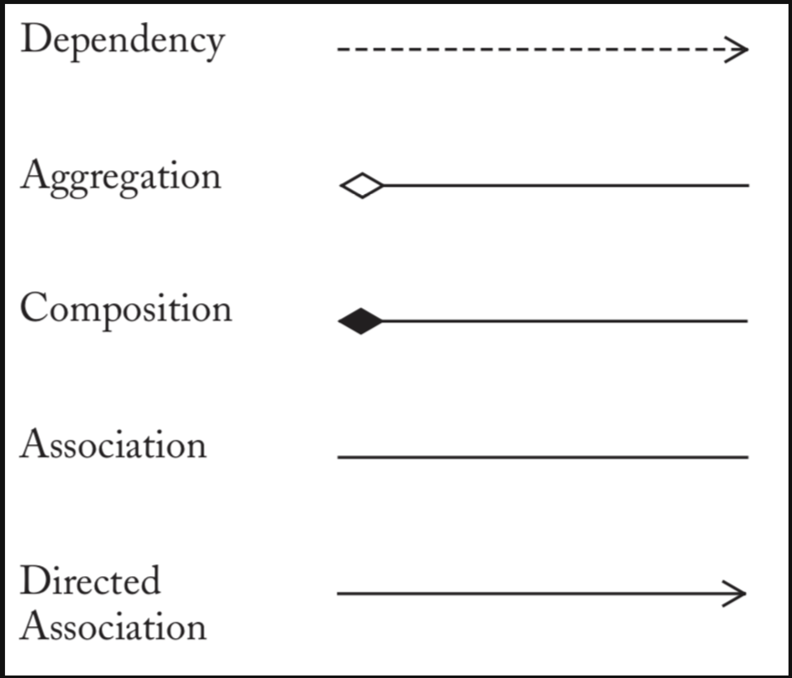
- Diagram formats
  - Mind maps
  - Context maps
  - UML class diagrams
- Because they are commonly occuring and you'll need them for future courses, we'll be using **UML class diagrams**

UML Class diagrams

- Classes



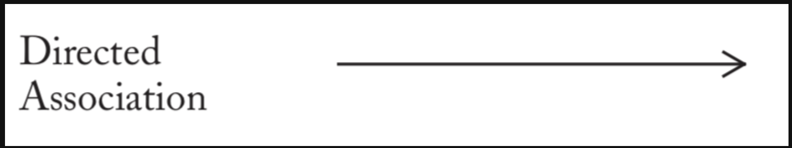
- Relationships



UML Class diagrams

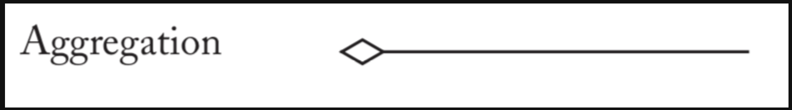
The loosest form of relationship. A class in some way *depends* on another.

A class "uses" another class in some way. When undirected, it is not yet clear in what direction dependency occurs.

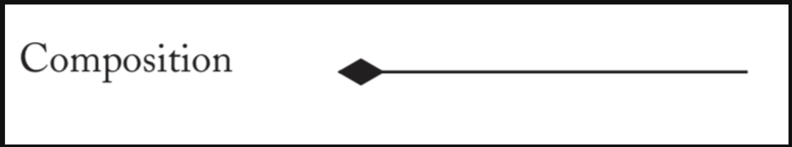


Refines association by indicating which class has knowledge of the other

UML Class diagrams



A class contains another class (e.g. a course contains students). Note that the diamond it at the end with the *containing* class.



Like aggregation, but the contained class is integral to the containing class. The contained class cannot exist outside of the container (e.g. the leg of a chair)

Aggregation vs Composition

- There is little consensus on the precise difference between the two relationships
- The definition above is imprecise
- Choosing not to use composition at all is a valid approach

From a domain model to an implementation

- Add methods to class diagram?
- Refine relationships
- Directly implement in code
- Adjust diagram if necessary

References

- A very detailed description of UML
  - https://www.uml-diagrams.org/
- Books that go into detail on Domain Driven Design
  - *Domain-Driven Design: Tackling Complexity in the Heart of Software* by Eric Evans
  - *Domain Modeling Made Functional: Tackle Software Complexity with Domain-Driven Design and F#* by Scott Wlaschin