

# COMP1531

## Design - Conceptual Modelling

### Lecture 7.2

Author(s): Hayden Smith



[\(Download as PDF\)](#)

# In This Lecture

- **Why?** 🤔
  - A critical element of software design is to be able to translate complex system ideas into something high level and understandable
- **What?** 📄
  - Conceptual Model
  - State Diagrams



# What Is A Model?



# What Is A Model?



# What Is A Model?





# What Is A Model?



# What Is A Model?

Figure 1: Mathematical Model

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

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

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

$$\frac{dR_m}{dt} = k_{rcm} R_c - \frac{k_{rmb}}{v_m} R_m - \mu_{rm} R_m$$

$$\frac{dN_c}{dt} = \frac{s_{nc} N_c}{1 + N_b / b_{n\infty}} + k_{cnc} C - k_{ncm} N_c - \mu_{nc} N_c$$

$$\frac{dN_m}{dt} = k_{ncm} R_c - \frac{k_{nmb}}{v_m} N_m - \mu_{nm} N_m$$

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

$$\frac{dT_m}{dt} = k_{tcm} T_c - \frac{k_{tmb}}{v_m} T_m - \mu_{tm} T_m$$

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

$$\frac{dL_m}{dt} = k_{lcm} L_c - \frac{k_{lmb}}{v_m} L_m - \mu_{lm} L_m$$

$$\frac{dR_b}{dt} = \frac{k_{rmb}}{v_m} R_m - \mu_{rb} R_b$$

$$\frac{dT_b}{dt} = \frac{k_{tmb}}{v_m} T_m - \mu_{tb} T_b$$

$$\frac{dN_b}{dt} = \frac{k_{nmb}}{v_m} N_m - \mu_{nb} N_b$$

$$\frac{dL_b}{dt} = \frac{k_{lmb}}{v_m} L_m - \mu_{lb} L_b$$



# What Is A Model?





# What Is A Model?

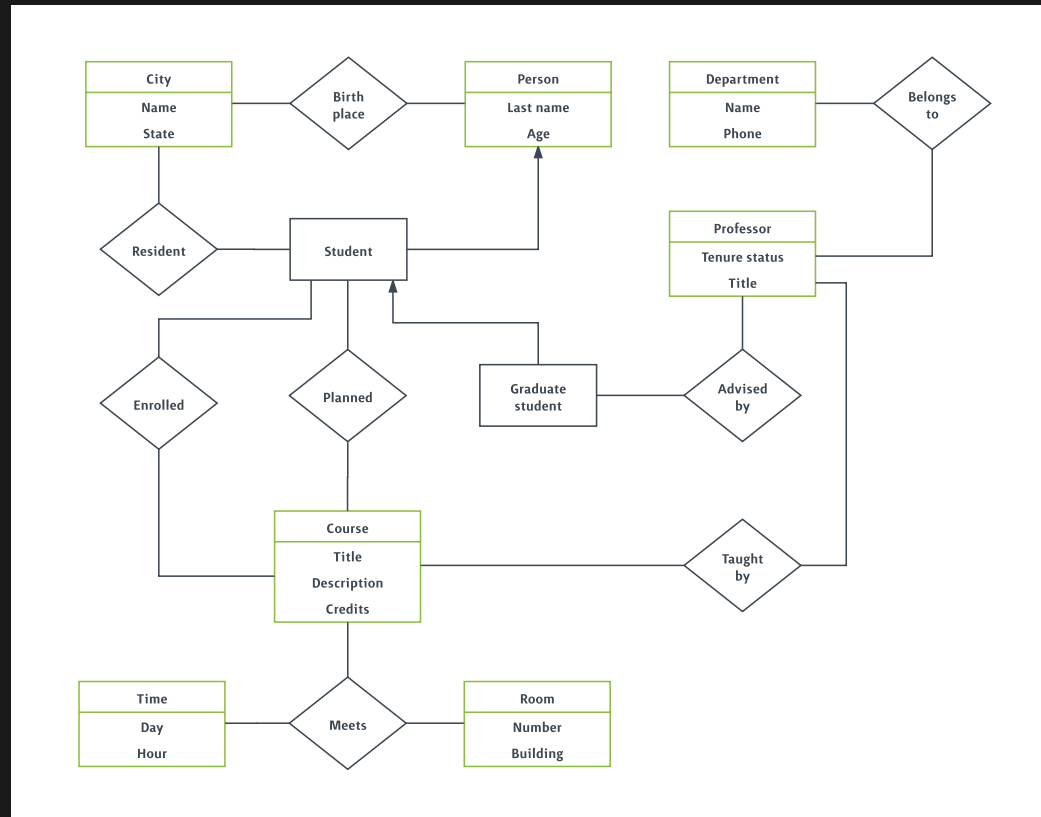




# What Is A Model?



# What Is A Model?

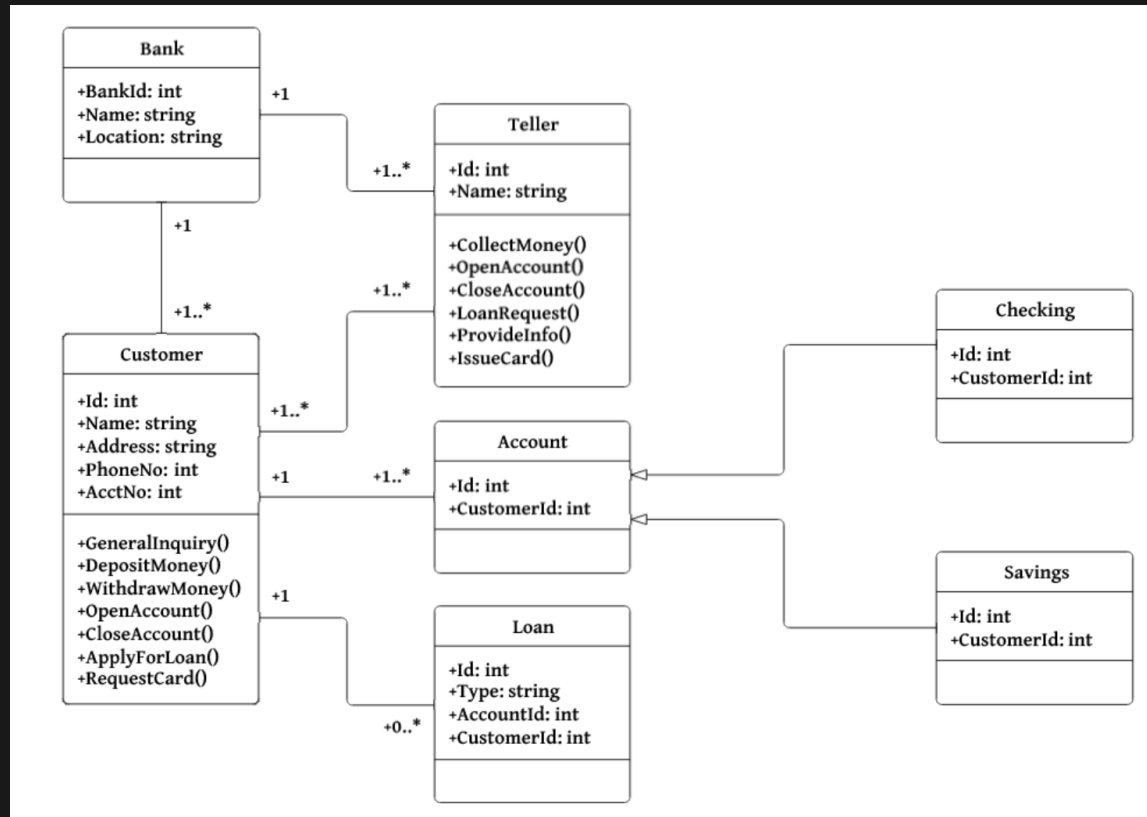




# What Is A Model?



# What Is A Model?



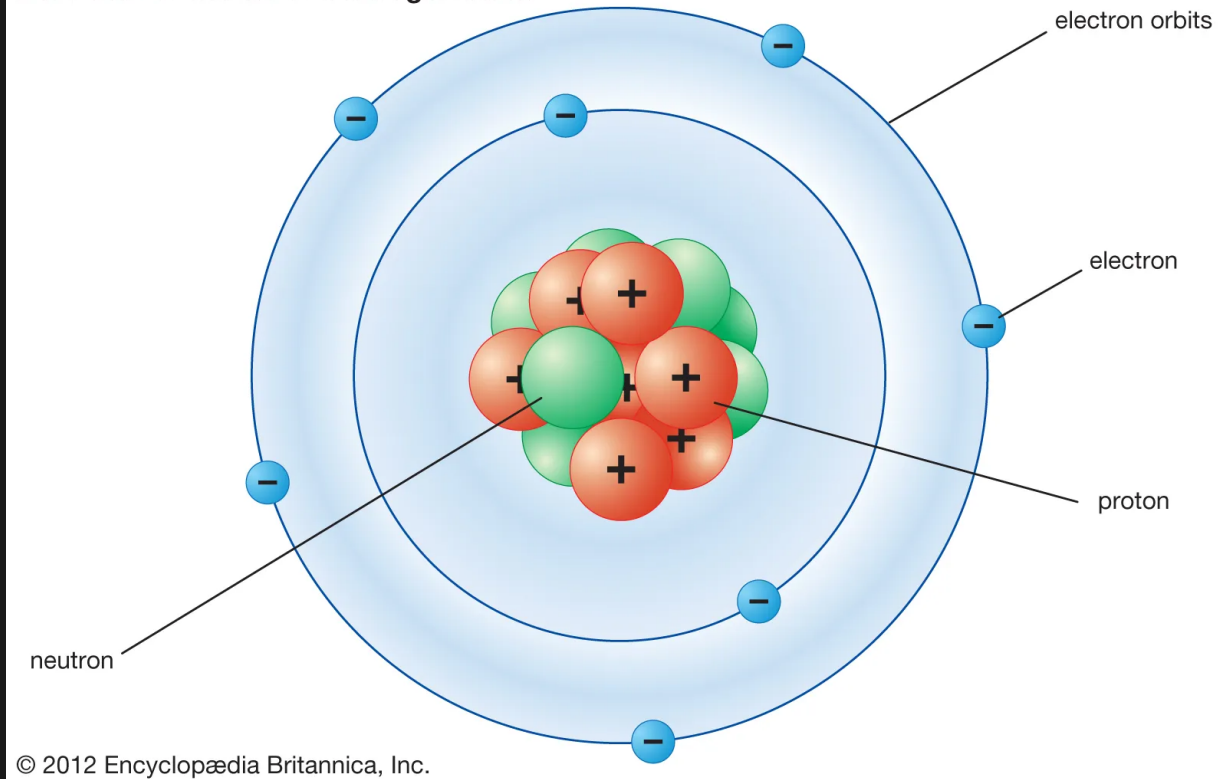


# What Is A Model?



# What Is A Model?

Bohr atomic model of a nitrogen atom





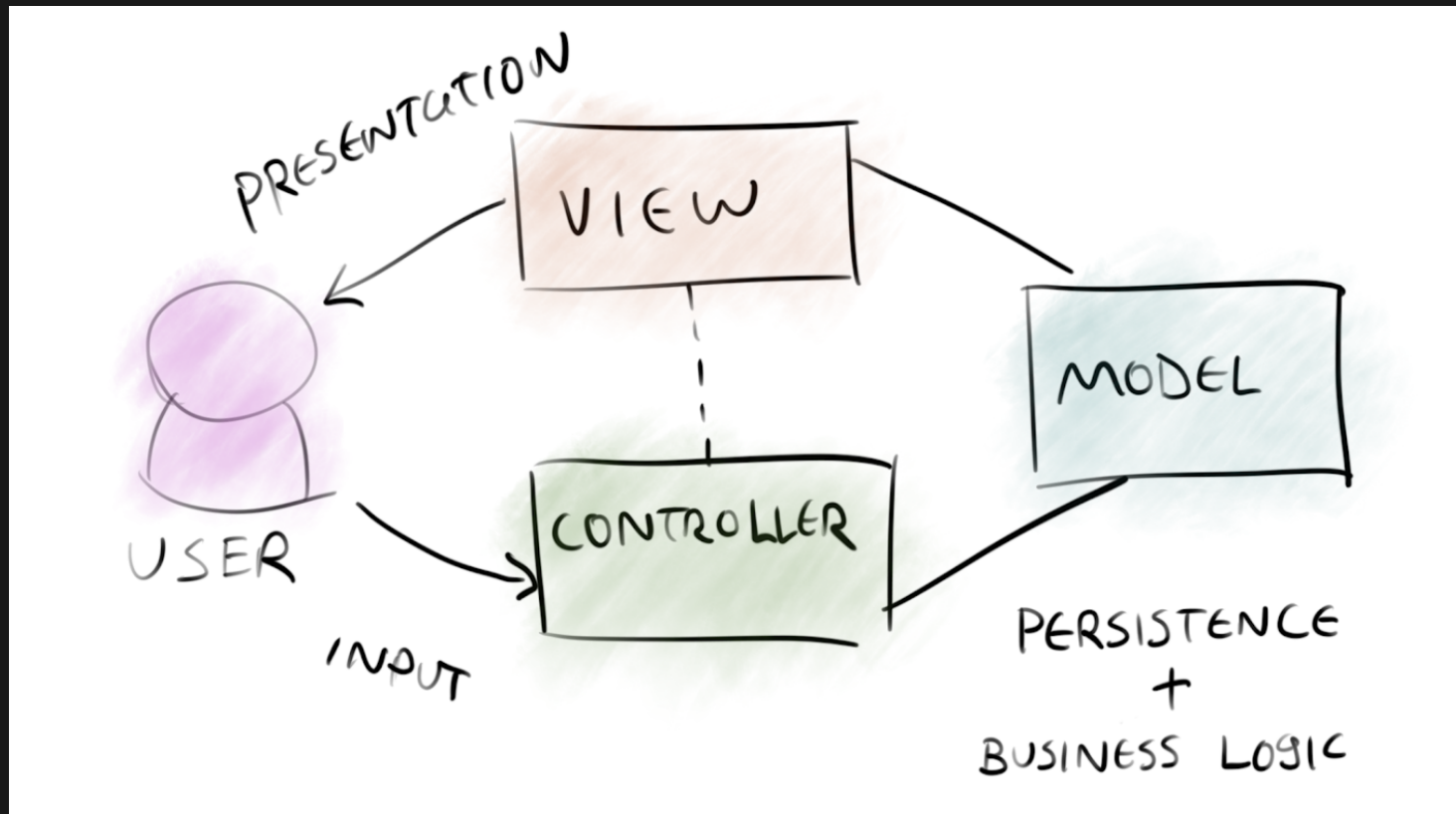
# What Is A Model?

}} />





# What Is A Model?



}} />



# What Is A Model?

A **model** is a simplified representation to assist in understanding something more complex.

This covers everything from mathematics through to model aeroplanes.



# Conceptual Modelling

A **conceptual model** is a type of model that captures a system in a conceptual way rather than a physical way.

They tend to be **diagrammatic** or **visual**.



# Conceptual Modelling

Examples include:

- **Mathematical models**
- **Data models:** To model databases
- **Domain models:** To model components of a software system
- **State transition models:** To model how a program operates at different stages



# Conceptual Modelling

Generally conceptual models in software will be one of two types:

- Structural – Emphasise the static structure of the system. Examples include:
  - UML class diagrams (object oriented programming)
  - ER diagrams (database design)
- Behavioural - Emphasise the dynamic behaviour. Examples include:
  - State diagrams (state machines)
  - Use case diagram (user flows)

We will discuss some of these, and explore **state diagrams** in detail.



# What Are Models Used For?

- 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 - i.e. help communicate



# Communicating Models

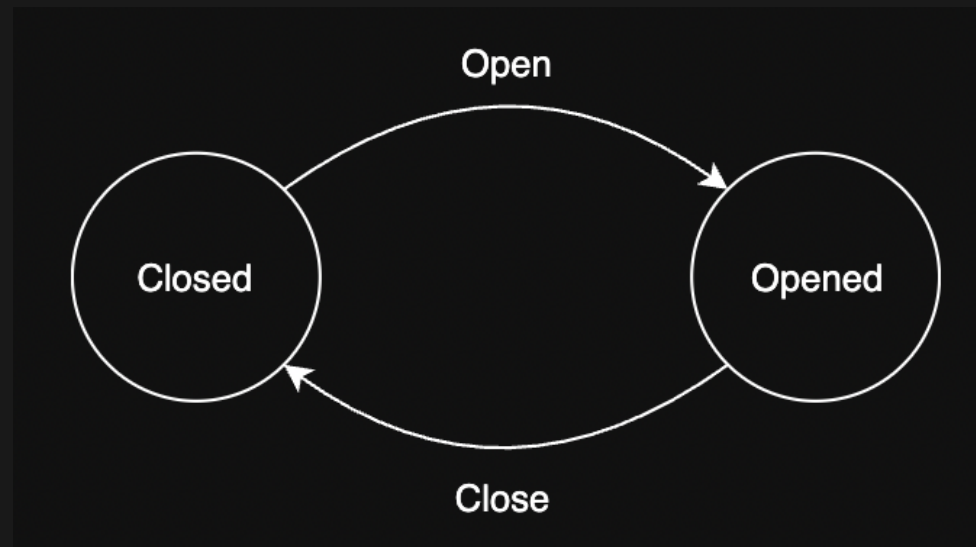
- Four fundamental objectives of communicating with a conceptual model:
  - Enhance an individual's understanding of the representative system
  - Facilitate efficient conveyance of system details between stakeholders
  - Provide a point of reference for system designers to extract system specifications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

[Kung and Solvberg \(1986\)](#)

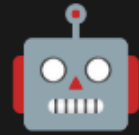


# State Diagrams

- A diagrammatic representation of a state.
- Simple example: a door
- Some variation in notation, though typically: states are circles, transitions are labelled arrows connecting them







# State Machines

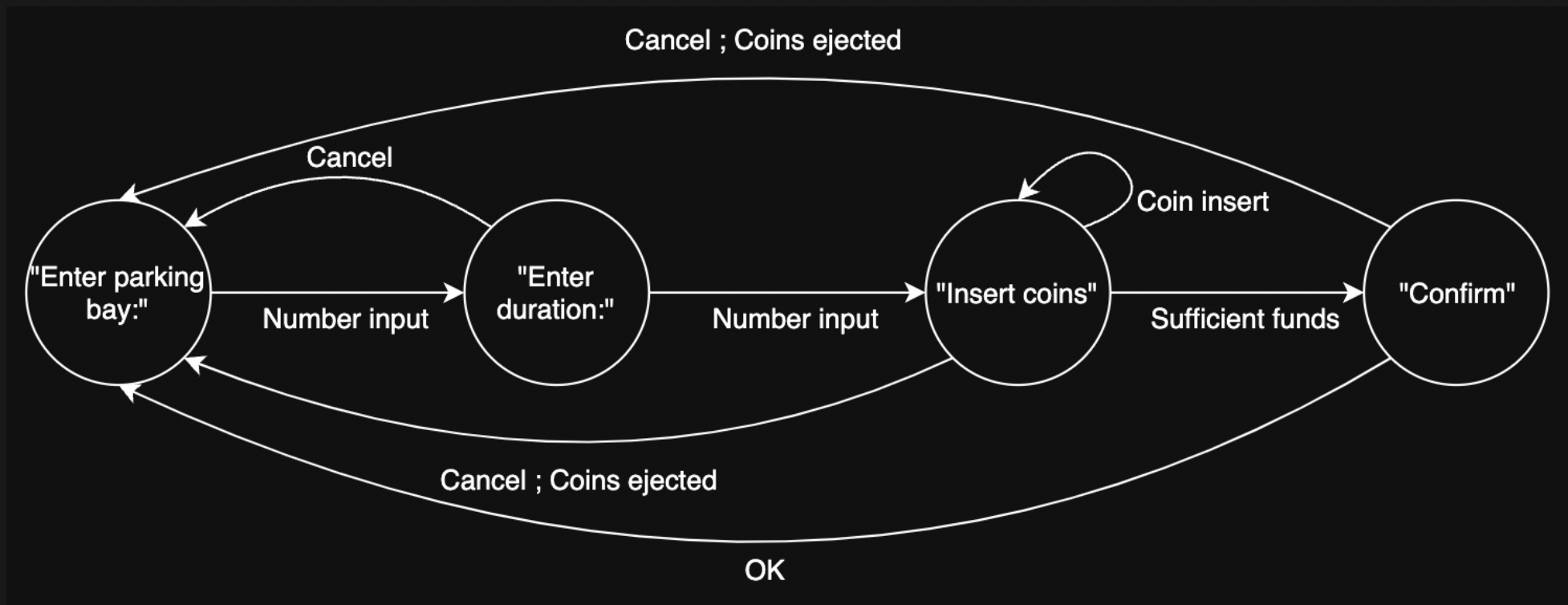
State diagrams are used to describe **state machines**.

State machines (i.e. software) are made up of a finite number of states.

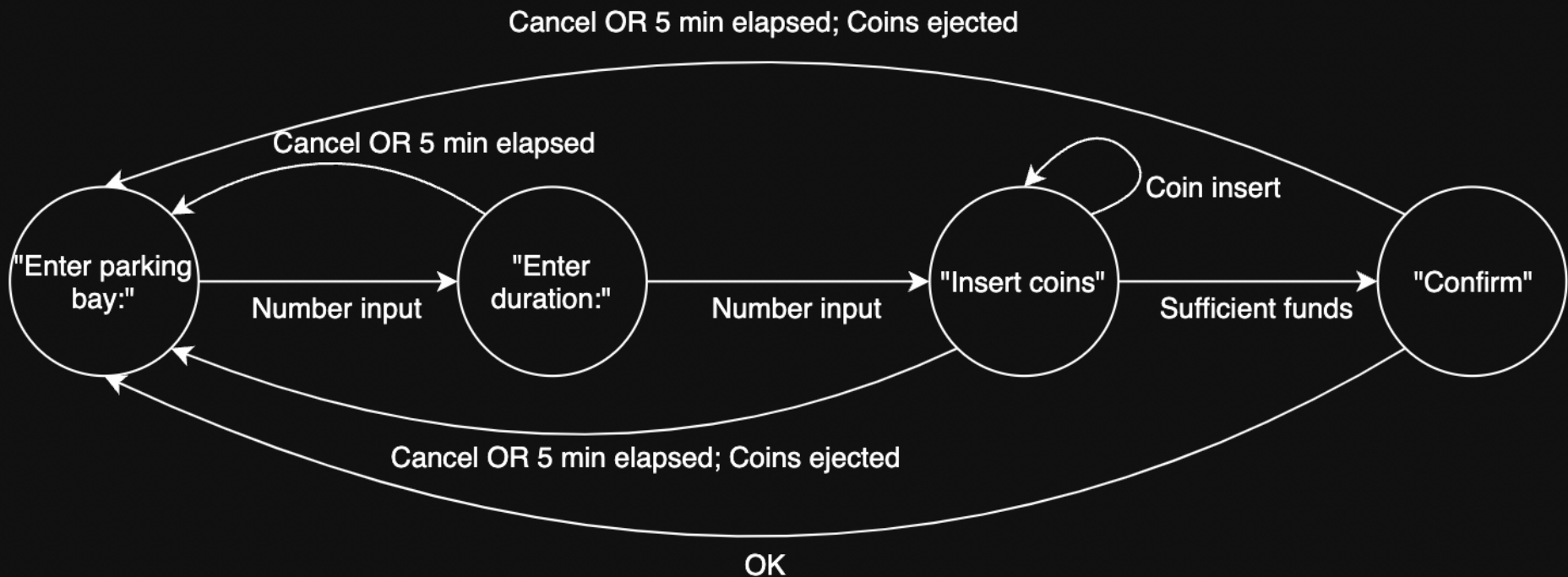
The machine (i.e. software) can be transitioned from one state to another through an  
action

- For example:
  - UIs with different screens
  - Network protocols
  - Conversational interfaces

# Parking Meter Example



# Parking Meter Example





# Quick Activity

Can we model the opal card system as a state machine?



# For Fun: A Complex Conceptual Model

- In 2015 a UNSW student wrote a conceptual model "Gallium" based off of [previous research](#) to represent the energy usage of a solar car as it [drove from Darwin to Adelaide](#)
- The model was written in python and modeled the physical system of the vehicle and it's energy consumption over a fixed distance in response to a dynamic environmental and physical characteristics

The model calculated the energy usage across a number of fixed-distance intervals (e.g. a 200 metre stretch of road) over a 3000km journey across Australia.

# Feedback



Or go to the [form here](#).

