

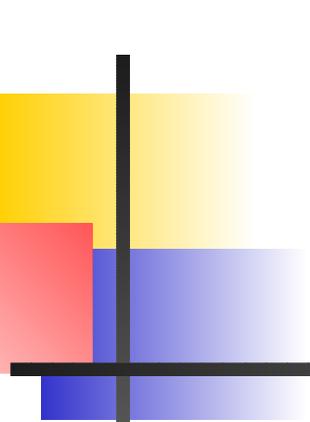
# Use Case Diagrams & Sequence Diagrams

Md Nour Hossain

Department of Computing and Software

McMaster University

Hamilton, Ontario, Canada



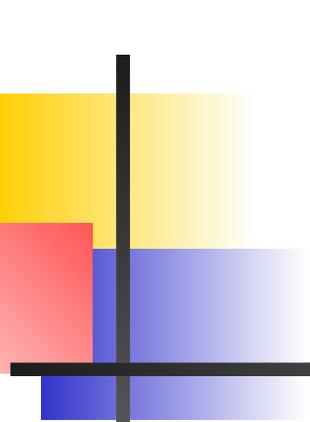
# Outline

---

1. UML Use Case Diagrams
2. UML Sequence Diagrams
3. Questions ?

# Dynamic Behaviour

1. To model a system the most important aspect is to capture the dynamic behaviour.
2. Dynamic behaviour means the behaviour of the system when it is running /operating.
3. UML Class Diagrams are not able to represent the dynamic behaviour of a model.
4. UML Use Case Diagrams & Sequence Diagrams.



# Use Case Diagrams

---

1. Use Case
2. Actor
3. Boundary
4. Connection
5. Include relationship
6. Extend relationship

# Use Case Diagrams

1. Use Case
2. Actor
3. Boundary
4. Connection
5. Include relationship
6. Extend relationship

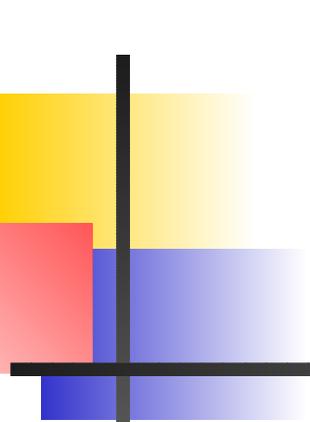
A use case is a single unit of meaningful work. The individual use cases will be represented by ovals within the rectangular system boundary.

# Use Case Diagrams

1. Use Case
2. Actor
3. Boundary
4. Connection
5. Include relationship
6. Extend relationship

A use case is a single unit of meaningful work. The individual use cases will be represented by ovals within the rectangular system boundary.

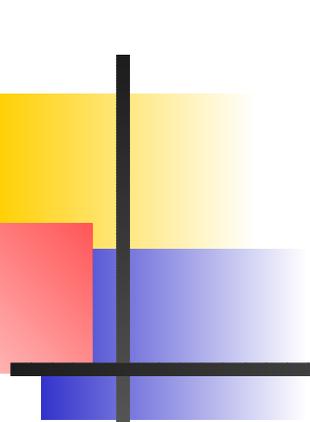
Actors are portrayed in a use case diagram as a stick figure and represent external factors that will provide interaction with the system.



# Use Case Diagrams

---

A boundary is the dividing line between the system and its environment. It is usual to display use cases as being inside the system and actors as being outside the system.



# Use Case Diagrams

---

A boundary is the dividing line between the system and its environment. It is usual to display use cases as being inside the system and actors as being outside the system.

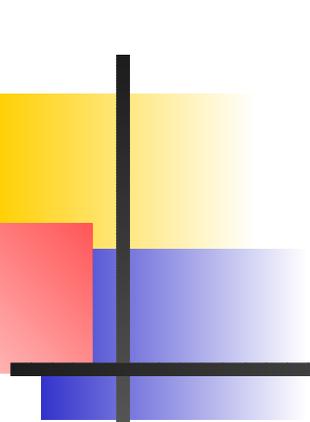
Use cases may contain the functionality of another use case as part of their normal processing.

# Use Case Diagrams

A boundary is the dividing line between the system and its environment. It is usual to display use cases as being inside the system and actors as being outside the system.

Use cases may contain the functionality of another use case as part of their normal processing.

One use case may be used to extend the behavior of another;



# Sequence Diagrams

---

1. Lifelines
2. Execution Occurrence
3. Messages
4. Lifeline Start and End

# Sequence Diagrams

---

1. Lifelines
2. Execution Occurrence
3. Messages
4. Lifeline Start and End

A thin rectangle running down the lifeline denotes the execution occurrence.

# Sequence Diagrams

1. Lifelines
2. Execution Occurrence
3. Messages
4. Lifeline Start and End

A thin rectangle running down the lifeline denotes the execution occurrence.

Messages are displayed as arrows. Messages can be complete, lost or found; synchronous or asynchronous.

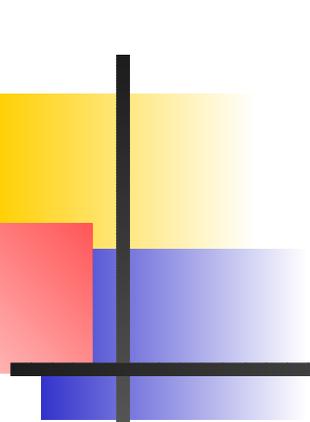
# Sequence Diagrams

1. Lifelines
2. Execution Occurrence
3. Messages
4. Lifeline Start and End

A thin rectangle running down the lifeline denotes the execution occurrence.

Messages are displayed as arrows. Messages can be complete, lost or found; synchronous or asynchronous.

A lifeline may be created or destroyed during the timescale represented by a sequence diagram. In the latter case, the lifeline is terminated by a stop symbol, represented as a cross.



# Questions??

---