CHRISTIAN HUEMER
MARION SCHOLZ

Object-Oriented Modeling with UML

State Machine Diagram

Explanation of Exercise Examples

Christian Huemer and Marion Scholz

Presented by Nicholas Bzowski

State Machine Diagram

Example: Alarm System

Christian Huemer and Marion Scholz

Presented by Nicholas Bzowski

Example: Alarm System

Model the following situation:

After initial start-up, the alarm system is in the deactivated state. If you now enter the numerical code and it is correct, the alarm system is set to the activated state and the LED flashes once. From now on, all sensors are continuously checked and the building is protected against intruders. However, if an incorrect numerical code is entered, the current state is retained. To deactivate the alarm system, enter the numerical code again. If the code is correct, the state changes to deactivated; if the code is incorrect, the state remains activated. If a door is broken open or a window smashed, the alarm system switches to the triggered state, the break-in is reported to the police and a continous alarm siren sounds. The alarm system can be switched back to the deactivated state by entering the correct numerical code. If no code is entered, the system automatically switches to the deactivated state after 10 minutes.

State Machine Diagram
Example: DVD Player

Christian Huemer and Marion Scholz

Presented by Nicholas Bzowski

Example: DVD Player

Model a very simple DVD player using a UML state diagram.

The following buttons are included:

- **Play:** If the DVD player is switched on and this button is pressed, the DVD is played (provided there is a DVD in the player). If the DVD player has only just been switched on, playback starts at chapter 0; if the pause button was previously pressed, playback continues with the chapter that was playing when the pause button was pressed.
- **Pause:** Pauses playback.
- **Stop:** Stops playback. If the Play button is then pressed, playback starts at chapter 0.
- **Forward:** If a DVD is currently being played, this button can be used to skip forward one chapter.
- **Backward:** If a DVD is currently being played, this button can be used to skip backwards one chapter.

The following operations are available:

- **ButtonPressed(Button):** This operation is always required when a button is pressed.
- **Play(Chapter):** This can be used to start playing the DVD, beginning at a specific chapter.
- **PausePlaying( ):** The operation to pause playback.

Model the states and state transitions of this DVD player.

State Machine Diagram

Example: Library Book

Christian Huemer and Marion Scholz

Presented by Nicholas Bzowski

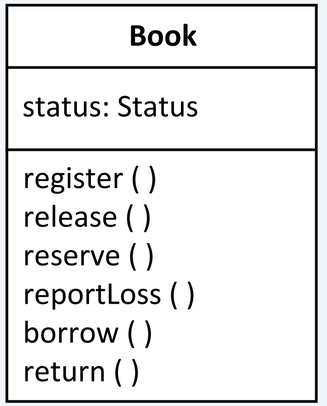
Example: Library Book

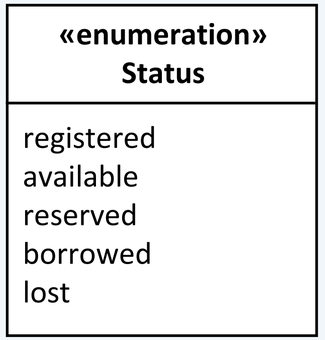
Model the states of a book in a library using a UML state diagram.

The following part of a class diagram is given, showing all the methods of the class Book. Furthermore, a book has a status - the individual possible statuses are also shown:

Using the class diagram, create a corresponding state diagram that reflects the various states and state transitions of a book. Model the status changes of the book explicitly in the state diagram.

Every book is new at the beginning. For the book to exist in the system, it must first be registered. Once this has been done, the book is available in the system and can be released for borrowing. Books that are available can be borrowed or reserved. If a book has been reserved but not picked up within a week, the reservation expires and someone else can borrow it. To keep the example simple, assume that only available books can be reserved (i.e. borrowed books cannot be reserved). If someone reports the loss of a borrowed book, the status of the book is set to lost.





State Machine Diagram

Example: Ticket Machine

Christian Huemer and Marion Scholz

Presented by Nicholas Bzowski

Example: Ticket Machine

Model a UML state diagram that depicts the states of a ticket vending machine (e.g. in a tram) from the vending machine's point of view.

The following information is available:

At the beginning, the ticket machine waits for a new input. The customer selects the desired ticket and is shown the outstanding balance. The customer then inserts coins into the machine until the price of the ticket is reached or exceeded. If the customer has inserted more coins than required, the change is returned to them and they can then take the printed ticket. If the customer has paid exactly the amount due, the ticket is printed immediately. This concludes the ticket purchase and the machine is ready for a new ticket purchase. If the customer has not yet paid the outstanding balance in full, they can press the cancel button at any time. This cancels the ticket purchase, the amount inserted so far is dispensed and the machine waits for the next customer. The purchase is also canceled if the customer does not insert a new coin for 3 minutes.

State Machine Diagram

Example: Travel Request

Christian Huemer and Marion Scholz

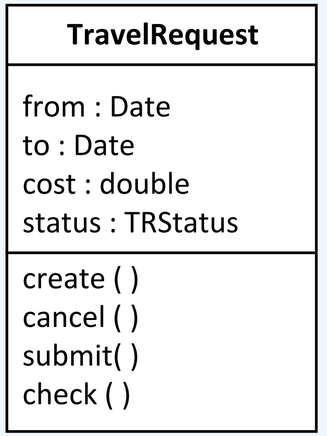
Presented by Nicholas Bzowski

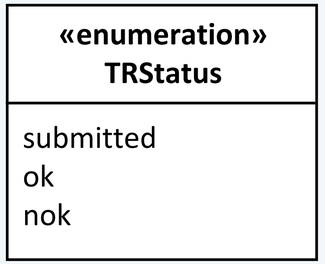
Example: Travel Request

You have the following information about a travel request in an HR administration system:

The new travel request must first be created (create()). Once it has been created, it is filled out. The request can then be submitted using submit(). The status of the travel request is now submitted. It is then checked. If it is ok, the status is set to ok and the request is considered approved. If it is not ok, the status is set to nok and the request is considered rejected. An approved application is deemed to have been carried out as soon as the date of return is reached. As long as the application has not been checked, the submitted application can be canceled at any time. This deletes the object.

Model a UML state diagram that depicts the states of a travel request. Do not forget to explicitly model the method calls, if known.





State Machine Diagram

Example: eBook

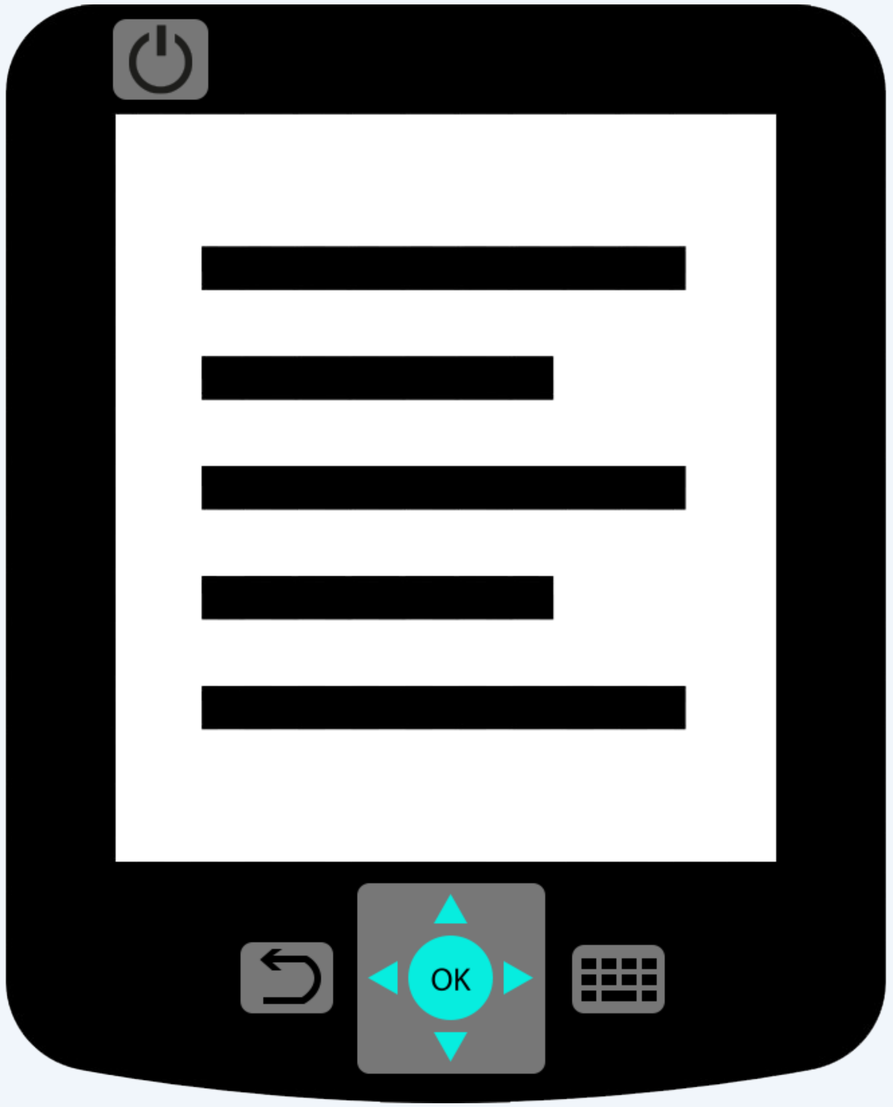
Christian Huemer and Marion Scholz

Presented by Nicholas Bzowski

Example: eBook

Model a UML state diagram that depicts the states and state transitions from the perspective of the eBook reader. Make reasonable assumptions. Functions that are not described in the instructions can be ignored.

- **On/Off button:** To switch on and off.
- **Select eBook:** After switching on, you are in the main menu and can see all titles in the library. Use the navigation buttons to select the eBook. Pressing up or down moves the list of books up or down. If you press left or right, you can scroll one page forwards or backwards in the list. You can confirm your selection by pressing the OK button in the middle and the selected eBook will be opened.
- **Read eBook:** When the selected eBook is displayed, you can use the navigation buttons to scroll forwards and backwards.
- **Exit eBook:** Use the back button to return to the previous menu. The device remembers the last page read and automatically displays it again the next time the title is opened.
- **Search/text input:** Pressing the keyboard button shows or hides the on-screen keyboard.



State Machine Diagram

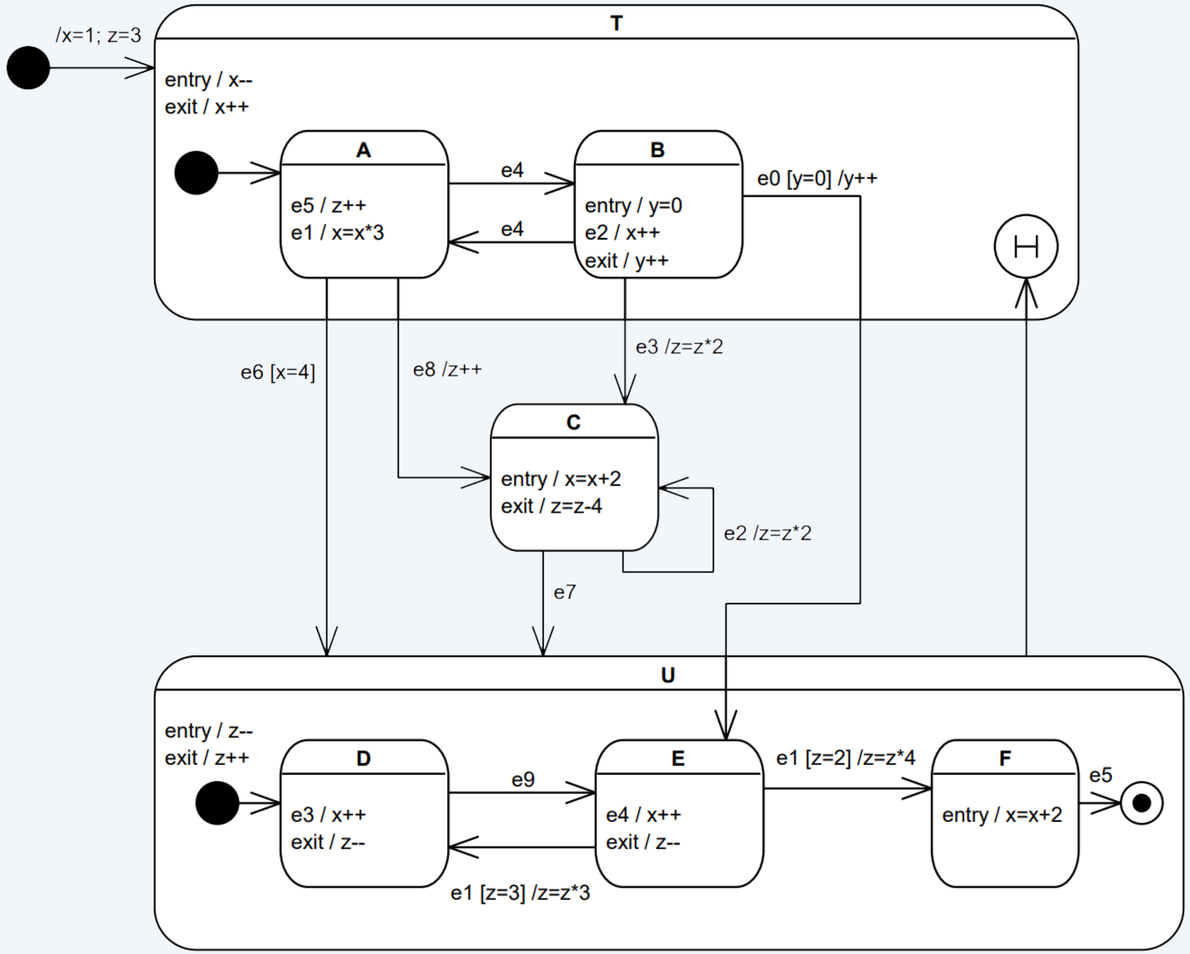
Example: Event Sequence

Christian Huemer and Marion Scholz

Presented by Nicholas Bzowski

Example: Event Sequence

The following state machine diagram is given:



Complete the following table to illustrate which states and actions occur in the following sequence of events.

		Assignment of Variables		
Event	Entered State	x	y	z
Start				
e4				
e0				
e1				
e5				
e3				
e2				
e7				
e3				
e9				
e5				