

Session 6

State chart diagrams

A state chart describes the life of one object: the states it can be in and the events that move it between them. Choose an object that genuinely changes state, such as an exam attempt or a fund transfer.

Objectives

× Do not copy. Read for understanding and the viva

- Complete questions 14 to 15 of the manual: state chart diagrams
- Prepare the deliverable before the lab and finish it during the session
- Be ready to explain every step in the viva

Questions Covered

× Do not copy. Read for understanding and the viva

Question	Requirement	Status
Q14	State Chart Diagram for Online Examination System	Complete
Q15	State Chart Diagram for Online Fund Transfer through Netbanking Banking System	Complete

Preparation

× Do not copy. Read for understanding and the viva

- The manual asks for a problem description of 300 to 500 words and a list of assumptions before every diagram. Write both first; they fix the scope the evaluator marks you against.
- Pick the one class whose objects have the richest life cycle and list its states before drawing.
- Label every transition as event [guard] / action.

- Include the initial and final pseudo-states and check that every state is reachable.

Question 14

Problem Statement

■ Write in lab record

State Chart Diagram for Online Examination System.

Solution

■ Write in lab record

Assumptions

The Online Examination System conducts university examinations over the web. Faculty schedule an examination with a date, a time window and a question paper that mixes objective questions, marked automatically from an answer key, and descriptive questions, marked by a human evaluator. A student who is enrolled for the paper logs in during the window, starts the examination, answers questions while a countdown timer runs, and submits. If the timer reaches zero the server submits whatever answers are saved. If the student's connection drops the timer keeps running; the student may reconnect and continue as long as time remains. A proctor may terminate an attempt for malpractice. After submission the attempt is evaluated on two tracks, objective and descriptive, the marks are totalled, the result is published to the student, and the student may ask for a re-check within a fixed period. Old attempts are archived after the retention period.

A state chart shows the life of one object, so the first task is to choose the object. Student, Question and Examination have almost no life cycle: a Student is created and stays active, a Question is authored and stays authored. The object that genuinely changes state is the ExamAttempt, the record that links one student to one examination. It is created when the student enrolls, it is the thing that is started, answered, submitted, evaluated, published, re-checked and archived, and the rules of the system are rules about what may happen to it in each state: answers may be saved only while it is in progress, marks may be entered only after it is submitted, a re-check may be requested only after publication. Every such rule becomes a transition with an event, an optional guard and an action.

The diagram therefore has one initial pseudo-state, ten states, one final pseudo-state and the transitions between them. The InProgress state carries entry, do and exit activities for the timer. The self-transition on InProgress records that saving an answer does not change state. Three states are terminal in meaning (Expired, Terminated, Published after archival) and each reaches the final pseudo-state, so every state is reachable from the initial state and can reach the final state.

- One ExamAttempt exists per student per examination; a second attempt is a new object.
- The examination window is fixed; an attempt that is never started before the window closes expires.
- The timer is server-side and keeps running while the student is disconnected.
- Auto-submission on timeout submits the answers saved so far.
- Objective marking is done by the server as soon as the attempt is submitted; descriptive marking is done by one evaluator; the attempt is Evaluated only when both are complete.
- A re-check may be requested once, within 7 days of publication, and returns the attempt to Published with possibly changed marks.
- An attempt terminated for malpractice is never evaluated.
- Archival happens after the university's retention period and ends the object's life.

Steps

1. Choose Model, Add Diagram, Statechart Diagram and name it "ExamAttempt State Chart".
2. In the Toolbox choose Initial State and click once on the canvas. Choose Simple State and place ten states with the names in the table.
3. Select InProgress and in the Editor use Add, Entry Activity, Do Activity and Exit Activity to type start timer, accept and save answers, stop timer.
4. Choose Transition and drag from source to target for every row of the transition table. Draw the self-transition by dragging from InProgress back to itself.
5. Select each transition and fill Triggers (the event), Guard and Effect (the action) in the Editor. StarUML renders the label as event [guard] / action.
6. Choose Final State and place one at the bottom. Draw transitions into it from Expired, Terminated and Published.
7. Check every state has at least one incoming and one outgoing transition, then export the diagram as PNG.

Diagram

State	Meaning	Internal activities
Registered	Student enrolled, attempt record exists, paper not started	entry / create attempt record
InProgress	Student is answering	entry / start timer; do / accept and save answers; exit / stop timer
Disconnected	Connection lost, timer still running	do / keep timer running
Submitted	Answers locked, waiting for evaluation	entry / lock answers
Evaluating	Objective and descriptive marking in progress	entry / auto-grade objective answers
Evaluated	All marks entered, total computed	entry / compute total, set pass or fail
Published	Result visible to the student	entry / notify student
UnderReview	Re-check in progress	do / re-mark disputed answers
Expired	Window closed without a start	none
Terminated	Ended for malpractice, never evaluated	entry / flag attempt

From	Event	Guard	Action	To
initial	student enrolls		create attempt record	Registered
Registered	startExam	window open	load paper	InProgress
Registered	windowClosed	not started		Expired
InProgress	answerSaved		save answer	InProgress
InProgress	connectionLost			Disconnected
Disconnected	reconnect	time left	resume	InProgress
Disconnected	timeUp		auto-submit saved answers	Submitted
InProgress	submit		lock answers	Submitted
InProgress	timeUp		auto-submit saved answers	Submitted
InProgress	malpracticeDetected		lock answers, flag attempt	Terminated
Submitted	evaluationStarted		auto-grade objective	Evaluating
Evaluating	allMarksEntered	objective and descriptive done	compute total, set pass or fail	Evaluated
Evaluated	resultReleased		notify student	Published
Published	recheckRequested	within 7 days		UnderReview

From	Event	Guard	Action	To
UnderReview	recheckCompleted		update marks, notify student	Published
Published	archived	retention period over		final
Expired				final
Terminated				final

- Lab record: every tab is one file of the answer. Write all of them.

Sketch

```

1      (*)
2      | student enrolls / create attempt record
3      v
4  +-----+ windowClosed [not started] +-----+
5  | Registered |----->| Expired |----> (X)
6  +-----+ +-----+
7      | startExam [window open] / load paper
8      v
9  +-----+ answerSaved / save answer
10 | InProgress |<-----+
11 | entry / start timer |----- +
12 | do / accept, save answers |
13 | exit / stop timer | connectionLost +-----+
14 +-----+----->| Disconnected |
15 | | ^ +-----+
16 | | +--- reconnect [time left] -----+ |
17 | | | timeUp / aut
18 | | malpracticeDetected / lock, flag +-----+ |
19 | +----->| Terminated |→ (X)
20 | +-----+ |
21 | submit / lock answers |
22 | timeUp / auto-submit |
23 v |
24 +-----+<----- --+
25 | Submitted |
26 +-----+
27 | evaluationStarted / auto-grade objective
28 v
29 +-----+ allMarksEntered [objective and descriptive done]
30 | Evaluating |-----+
31 +-----+ | / compute total, set pa
32 v
33 +-----+
34 | Evaluated |
35 +-----+
36 | resultReleased / notify
37 v
38 +-----+ recheckRequested [within 7 days] +-----+
39 | UnderReview |<-----| Published |
40 +-----+----->+-----+
41 recheckCompleted / update marks | archived [retenti

```

42

v

43

(X)

Explanation

The chart is a finite state machine for one ExamAttempt object. A state is a period during which the object waits for events and obeys a fixed set of rules; a transition is the response to one event. Reading the chart answers the questions the system must enforce. Can a student save an answer after submitting? No: the answerSaved self-transition exists only on InProgress. Can an evaluator enter marks for an attempt that is still running? No: Evaluating is reachable only from Submitted. Can a re-check be requested twice? No: UnderReview returns to Published and the guard `within 7 days` limits the window; a second request would need a second transition that the chart does not offer. The entry, do and exit activities on InProgress put the timer logic in one place: it starts on entry from Registered, keeps running in Disconnected because that state has its own do activity, and stops on any exit. Two different events, submit and timeUp, lead to the same Submitted state and that is normal; what matters is that from Submitted the rest of the life cycle is identical.

Question 15

Problem Statement

■ Write in lab record

State Chart Diagram for Online Fund Transfer through Netbanking Banking System.

Solution

■ Write in lab record

Assumptions

Online fund transfer through net banking lets a logged-in customer move money from one of their accounts to a beneficiary account, at the same bank or another bank, using a web browser. The customer fills in a transfer form with the beneficiary, the amount and a remark, and confirms it with the transaction password. The bank validates the request: the beneficiary must be active, the amount must not exceed the available balance or the daily transfer limit, and the transaction password must be right. The bank then sends a one-time password (OTP) to the customer's

registered mobile number and waits up to 3 minutes for it. Once the OTP is verified, core banking debits the sender's account and sends a credit instruction to the beneficiary's bank over IMPS, NEFT or an internal book transfer. If the credit is confirmed the transfer is complete and the customer receives an SMS and an email. If the debit itself fails the transfer fails and nothing has moved. If the debit succeeded but the beneficiary bank rejects the credit (closed account, invalid IFSC, timeout), the bank reverses the debit and informs the customer.

The object whose life this describes is the Transfer, the record created the moment the customer submits the form. Customer, Account and Beneficiary are stable; the Transfer is the thing that is initiated, authenticated, waits for an OTP, is processed, and finally succeeds, fails or is reversed. The state chart therefore has seven states: Initiated, Authenticated, OtpPending, Processing, Completed, Failed and Reversed, with one initial and one final pseudo-state. The last three are terminal for the business: once a transfer is Completed, Failed or Reversed nothing further happens to it except archival, and all three lead to the final pseudo-state. Guards on the transitions encode the bank's rules: retries of the transaction password and of the OTP are counted, the OTP has a time limit, and the balance and daily limit are checked before the OTP is sent, so that the customer is not asked for an OTP on a transfer that cannot go through.

- The customer is already logged in; the login session is a separate state machine.
- The beneficiary was registered and activated earlier.
- Up to three attempts are allowed for the transaction password and three for the OTP; the third failure fails the transfer.
- The OTP is valid for 3 minutes; the timer starts when the OTP is sent and the timeout fails the transfer.
- Debit and credit are two steps: the debit is inside the bank's core system, the credit is a message to the beneficiary bank (or an internal book entry for same-bank transfers) that can be rejected after the debit has happened.
- Reversal re-credits the sender's account in full; there is no partial reversal.
- Completed, Failed and Reversed are final business states; archival is not modelled as a separate state.

Steps

1. Choose Model, Add Diagram, Statechart Diagram and name it "Transfer State Chart".
2. Place an Initial State, seven Simple States named as in the table, and one Final State.
3. Select Processing and add the entry activity debit sender account and the do activity send credit to beneficiary bank. Add the entry activities listed for the other states the same way.

4. Draw the transitions from the transition table, including the self-transitions on Initiated (password retry) and OtpPending (OTP retry and resend).
5. Fill Triggers, Guard and Effect for every transition in the Editor.
6. Draw the three transitions into the Final State from Completed, Failed and Reversed.
7. Export the diagram as PNG.

Diagram

State	Meaning	Internal activities
Initiated	Form submitted, request being validated	entry / validate beneficiary, amount, limit
Authenticated	Transaction password accepted, balance and limit confirmed	entry / generate OTP
OtpPending	OTP sent, waiting for the customer	entry / send OTP, start 3 minute timer
Processing	OTP verified, money is moving	entry / debit sender account; do / send credit to beneficiary bank
Completed	Credit confirmed	entry / send SMS and email, update statement
Failed	Transfer rejected before any money moved	entry / release hold, notify customer
Reversed	Debit done, credit rejected, debit undone	entry / re-credit sender, notify customer

From	Event	Guard	Action	To
initial	submitTransferForm		create Transfer record	Initiated
Initiated	passwordVerified	beneficiary active and amount within balance and daily limit		Authenticated
Initiated	passwordWrong	attempts below 3	show error	Initiated
Initiated	passwordWrong	attempts equal 3		Failed
Initiated	validationFailed	beneficiary inactive or amount above balance or above daily limit		Failed
Authenticated	otpGenerated		send OTP, start timer	OtpPending
OtpPending	otpEntered	OTP matches and timer running		Processing
OtpPending	otpEntered	OTP wrong and attempts below 3	show error	OtpPending
OtpPending	resendOtp	resends below 3	send new OTP, restart timer	OtpPending
OtpPending	otpEntered	OTP wrong and attempts equal 3		Failed
OtpPending	timeout			Failed
Processing	creditConfirmed		send SMS and email,	Completed

From	Event	Guard	Action	To
			update statement	
Processing	debitFailed			Failed
Processing	creditRejected		re-credit sender	Reversed
Completed				final
Failed				final
Reversed				final

- Lab record: every tab is one file of the answer. Write all of them.

Sketch

```

1      (*)
2      | submitTransferForm / create Transfer record
3      v
4      +-----+ passwordWrong [attempts below 3] / show error
5      | Initiated |<-----+
6      +-----+-----+
7      | |
8      | | validationFailed [beneficiary inactive or amount above balance or
9      | | passwordWrong [attempts equal 3]
10     | +-----+
11     | passwordVerified [beneficiary active and amount within limit] |
12     v                                                                |
13     +-----+                                                        |
14     | Authenticated |                                                |
15     +-----+                                                        |
16     | otpGenerated / send OTP, start timer                          |
17     v                                                                |
18     +-----+ otpEntered [wrong and attempts below 3] / show error|
19     | OtpPending |<-----+                                          |
20     |           |-----+                                          |
21     +-----+ resendOtp [resends below 3] / send new OTP            v
22     | |                                                                +-----+
23     | | otpEntered [wrong and attempts equal 3]                    | Failed |--
24     | | timeout                                                    +-----+
25     | +-----+-----^ ^
26     | otpEntered [OTP matches and timer running]                  |
27     v                                                                |
28     +-----+-----+ debitFailed                                    |
29     | Processing          |-----+
30     | entry / debit sender account |
31     | do / send credit to beneficiary bank | creditRejected / re-credit sen
32     +-----+-----+-----+
33     | creditConfirmed / send SMS and email, update statement
34     v
35     +-----+
36     | Completed |---> (X)
37     +-----+

```

Explanation

Each state answers one question about the money. In Initiated and Authenticated nothing has moved and the bank is still deciding whether it may move. In OtpPending the bank has decided

yes but is waiting for the customer's proof of possession of the phone; the timer and the retry counter live here, which is why the state has two self-transitions and two exits to Failed. Processing is the only state in which money is in flight: its entry activity debits the sender, its do activity sends the credit, and the three exits describe the three things that can happen to a debit followed by a credit. Failed and Reversed are different states on purpose. Failed means the sender's balance was never changed; Reversed means it was debited and then re-credited, and the statement shows both entries. Merging them would hide that difference from the customer-facing history. The same event, otpEntered, appears on three transitions out of OtpPending with three guards that are mutually exclusive and exhaustive: matches, wrong with attempts remaining, wrong with no attempts remaining. That is the state chart equivalent of the complete guards required on an activity diagram decision.

Viva Questions

× Do not copy. Read for understanding and the viva

Q: How do you choose the object for a state chart? **A:** Pick the class whose objects pass through distinct modes with different rules in each. ExamAttempt and Transfer change state many times; Student and Beneficiary barely change at all.

Q: What do entry, do and exit activities mean? **A:** entry runs once when the state is entered, do runs for as long as the object stays in the state and can be interrupted, exit runs once when the state is left, before the transition's action.

Q: What is the full form of a transition label? **A:** event [guard] / action. The event triggers it, the guard must be true for it to fire, and the action runs during the transition.

Q: Why is a self-transition different from staying in the state? **A:** A self-transition leaves and re-enters the state, so exit and entry activities run again. In the ExamAttempt chart answerSaved is a self-transition only because InProgress has no activities that must not repeat; if the timer restarted on entry, save would have to be an internal transition instead.

Q: What is the difference between Failed and Reversed in the Transfer chart? **A:** Failed means the sender was never debited. Reversed means the sender was debited, the beneficiary bank rejected the credit, and the debit was undone.

Q: Why do three transitions leave OtpPending on the same event otpEntered? **A:** Their guards are mutually exclusive: OTP correct, wrong with attempts remaining, wrong with attempts exhausted. Exactly one fires for any OTP entry.

Q: Can a state chart have more than one final pseudo-state? **A:** Yes, but one is enough; several transitions may enter it. The chart has one final state reached from Completed, Failed and Reversed.

Q: How does a state chart relate to the class diagram? **A:** The states are usually the values of a status attribute of the class, and the transition events are its operations or messages it receives from the sequence diagram.

Common Mistakes

✗ Do not copy. Read for understanding and the viva

- Drawing states for the whole system (Login screen, Dashboard) instead of for one object; that is an activity diagram in disguise.
- Labelling transitions with actions only, for example “debit account”, and leaving out the event that triggers them.
- Guards that overlap, such as `attempts below 3` and `attempts below 5`, so that two transitions could fire on one event.
- Forgetting the initial pseudo-state or leaving a state with no way out that is not marked as reaching the final state.
- Using Processing for a state where nothing is happening; state names should describe a waiting condition or an ongoing activity.
- Putting the OTP timer in the Authenticated state; the timer belongs where the object waits for the OTP.

Session Summary

■ Write in lab record

- Question 14: state chart for the ExamAttempt object with ten states, entry, do and exit activities on InProgress, guarded transitions for window, timer, malpractice and re-check; PlantUML source `exam_attempt_state.puml`
- Question 15: state chart for the Transfer object with states Initiated, Authenticated, OtpPending, Processing, Completed, Failed, Reversed and retry self-transitions; PlantUML source `fund_transfer_state.puml`

- Problem description (300 to 500 words) and assumptions for every diagram, with the choice of object justified
- State table, transition table (event, guard, action) and ASCII sketch for each diagram, ready to redraw in StarUML

 [Edit this page](#)

[< Previous](#)
Session 5

Session 7 [Next >](#)