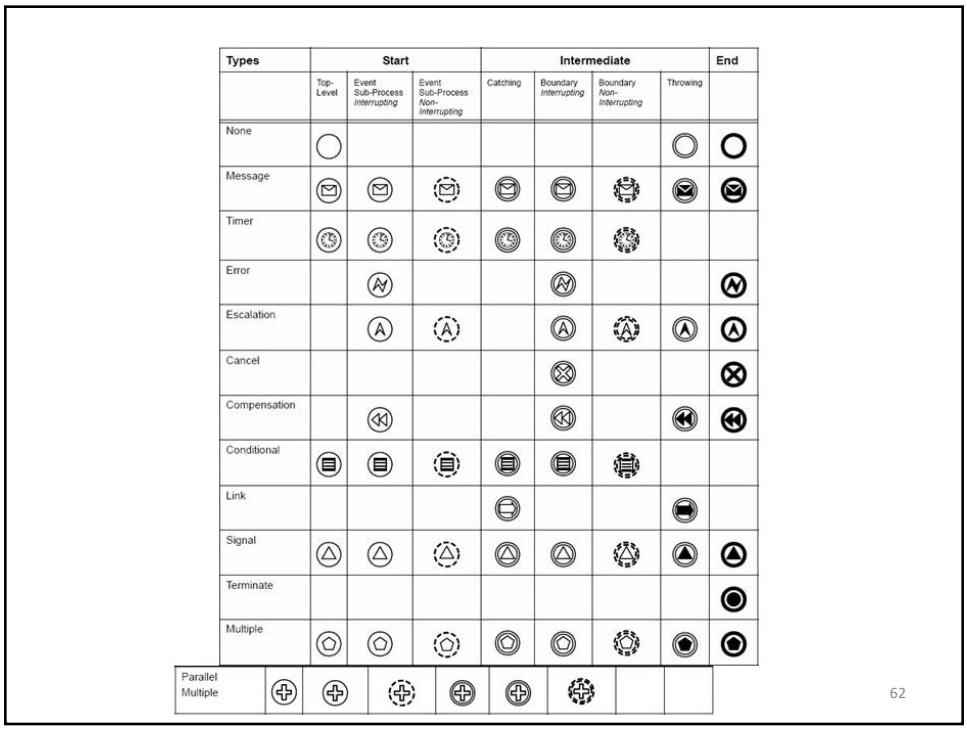
ANALYTIC BPMN

60

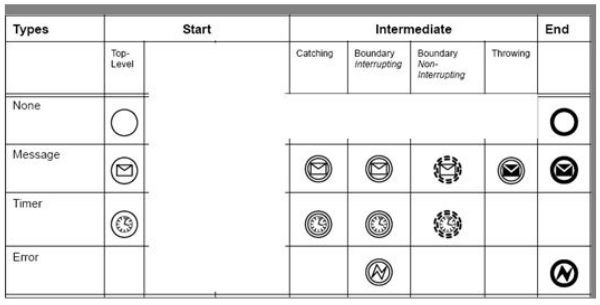
BPMN Events

- Level 1 events:
  - Start: None, Message, Timer
  - End: None, Message, Terminate
- Level 2 events:
  - Intermediate events
  - Additional Triggers
  - Main: Timer, Message, Error
  - Others: Escalation, Signal, Conditional, Link
  - Also: Cancel, Compensation

61


Types
Start
Intermediate
End
Top-Level
Event Sub-Process Interrupting
Event Sub-Process Non-Interrupting
Catching
Boundary Interrupting
Boundary Non-Interrupting
Throwing
None
Message
Timer
Error
Escalation
Cancel
Compensation
Conditional
Link
Signal
Terminate
Multiple
Parallel Multiple
62

62


Types
Start
Intermediate
End
Top-Level
Catching
Boundary Interrupting
Boundary Non-Interrupting
Throwing
None
Message
Timer
Error
63

63

Throwing vs Catching

Catching Message event in sequence flow

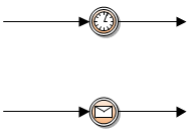
Throwing Message event in sequence flow

64

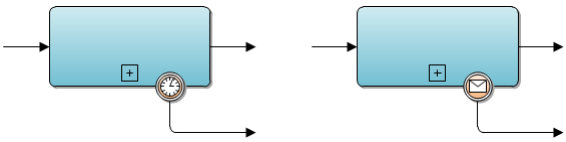
64

Sequence Flow vs Boundary

- Sequence flow:
  - Catching event waits



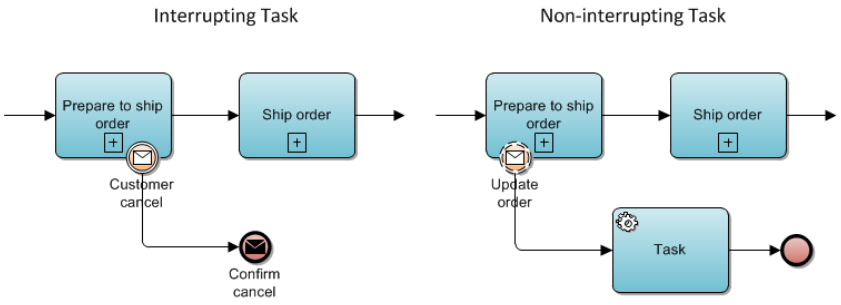
- Boundary:
  - Catching event listens



65

65

Boundary Events: Interrupting vs Non-interrupting

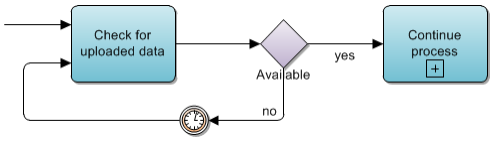


66

66

Timer Events

- Catching == delay
  - Wait for some duration



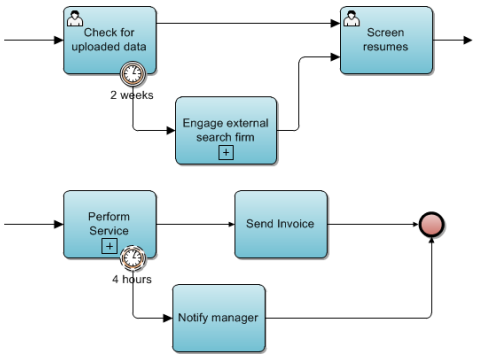
  - Wait until specified time

67

67

Timer Events

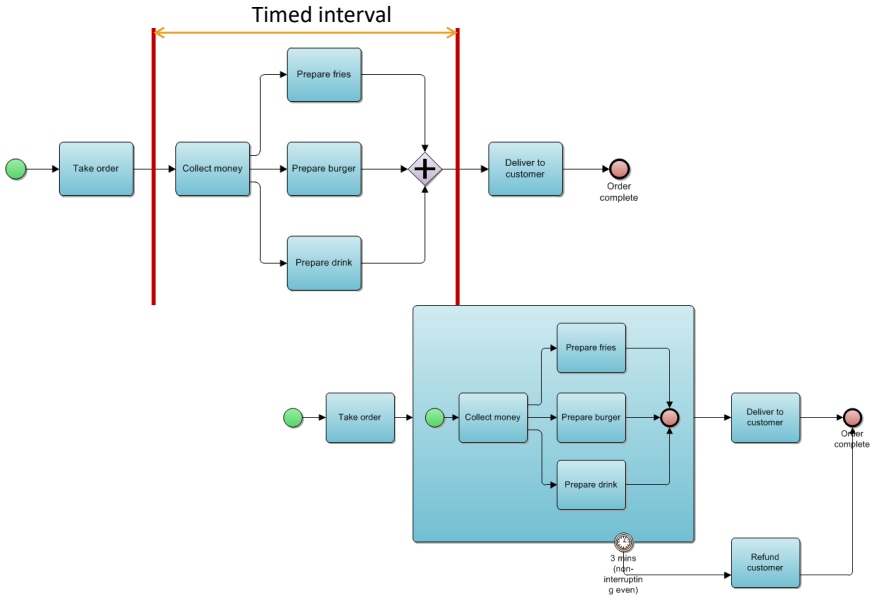
- Timer Boundary Event
  - Time to completion
  - Starting when sequence flow arrives



68

68

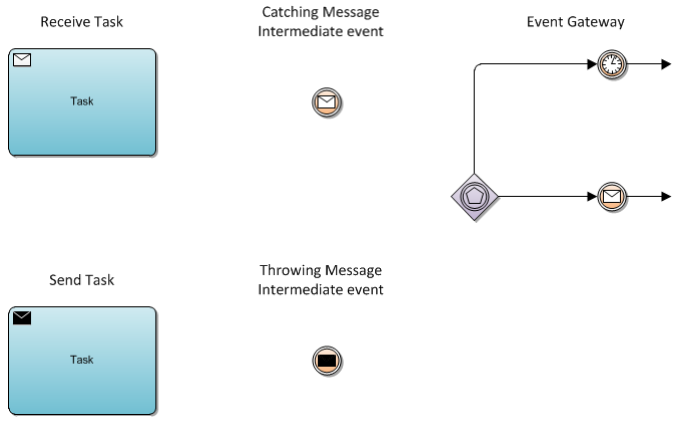
Timed Interval



69

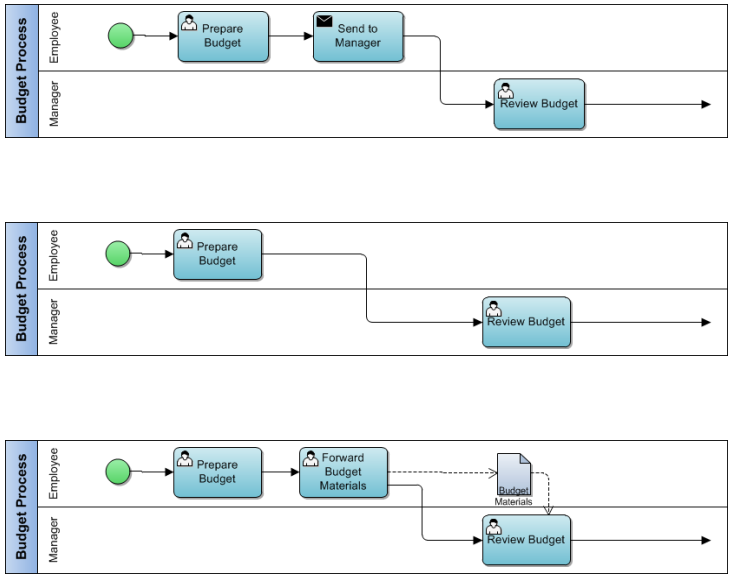
Message Send & Receive

- Possibility vs certainty of message send



70

70



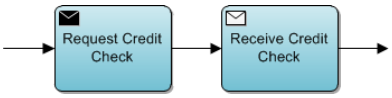
71

71

Synchronous vs Asynchronous Messaging

- Synchronous

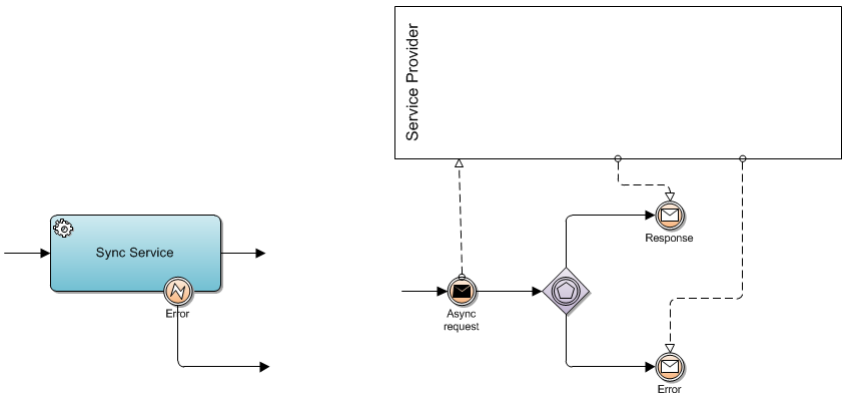
- Asynchronous



72

72

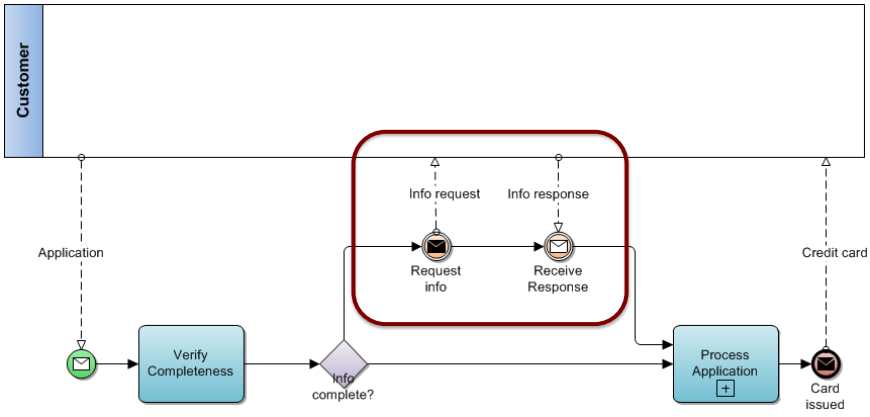
Synchronous and Asynchronous



73

73

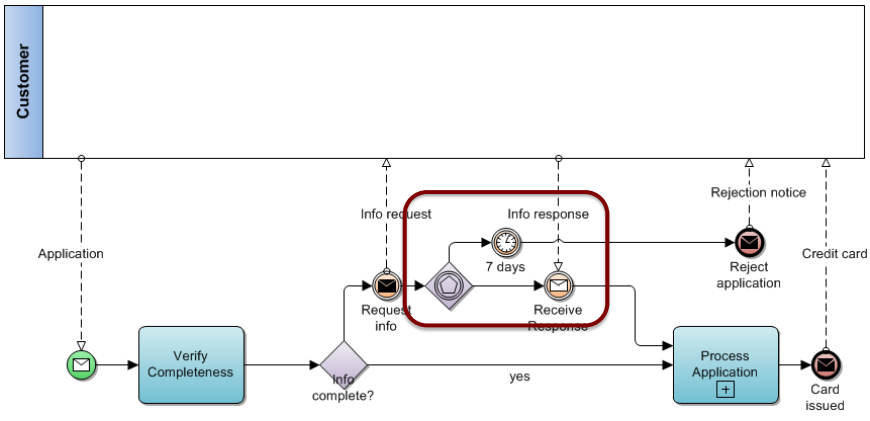
Throwing and Catching



74

74

Event Gateway

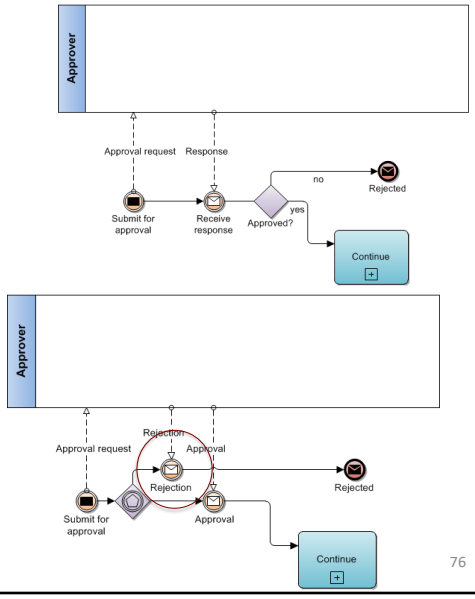


75

75

Internal vs External Choice

- Exclusive OR
  - Choice based on internal condition
- Event Gateway
  - Choice based on first event that fires

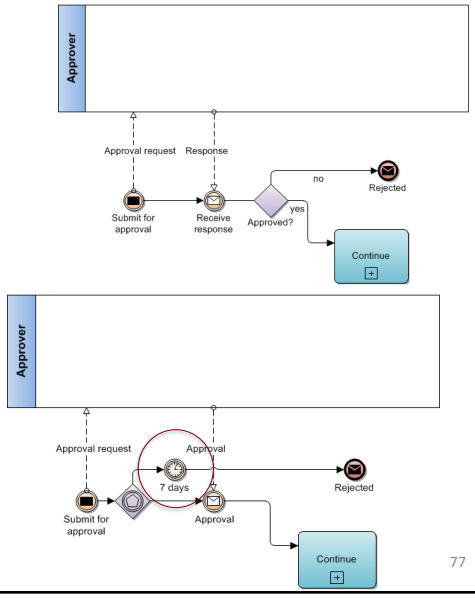


76

76

Internal vs External Choice

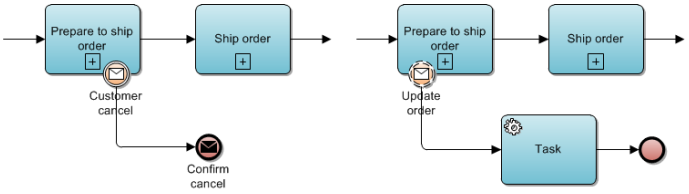
- Exclusive OR
  - Choice based on internal condition
- Event Gateway
  - Choice based on first event that fires



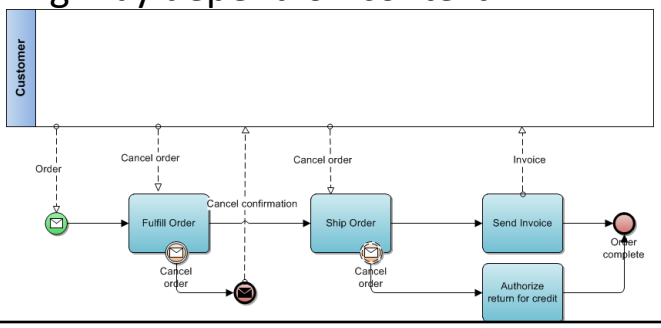
77

77

Message Boundary Event



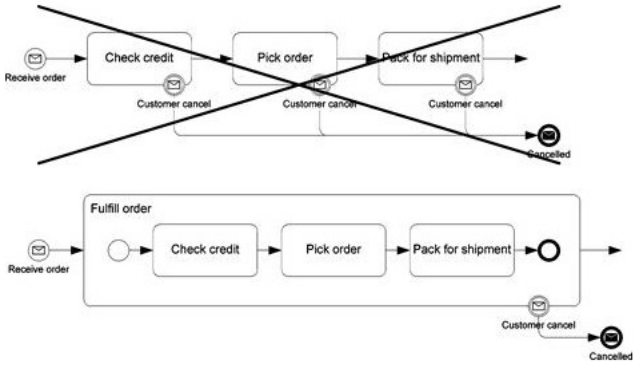
- Handling may depend on context



78

78

Message Boundary Event

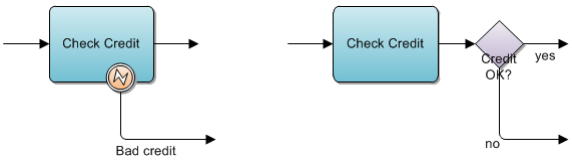


79

79

Error Event

- Two flavors
  - Interrupting Error boundary event
  - Error end event



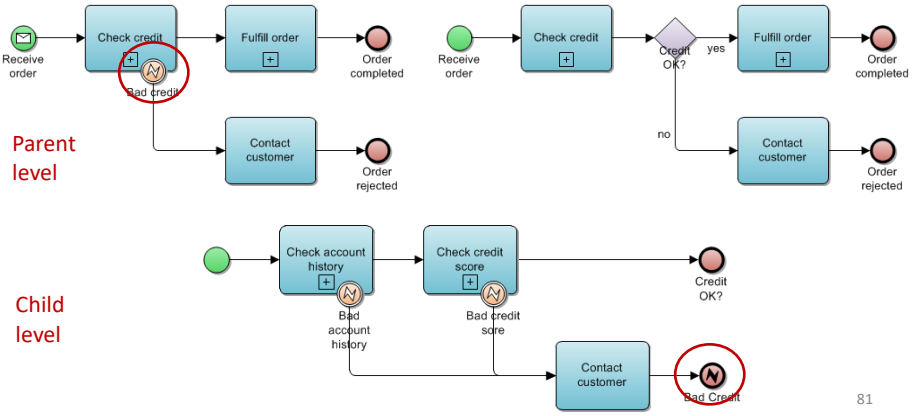
- Subprocess:
  - Error boundary event must have matching end state in expansion

80

80

Error Throw-Catch Pattern

- Similar to gateway end state test
  - Parallel paths ⇒ end state = termination

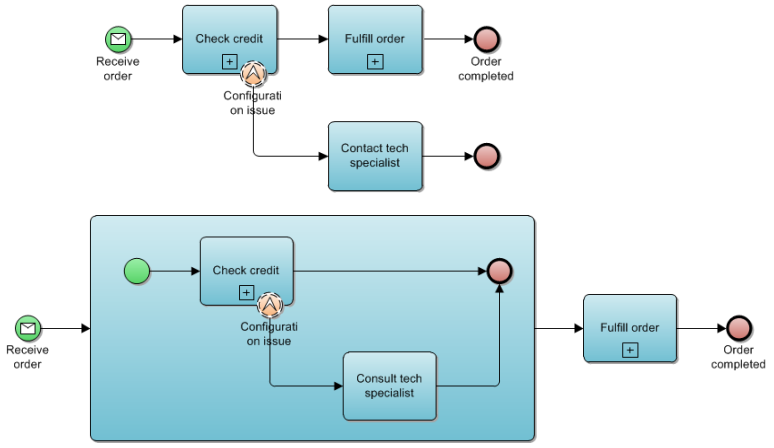


81

81

Escalation Event

- Non-interrupting exception



82

82

Escalation Events

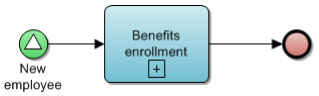
- Boundary escalation event is non-interrupting
- Does not imply an error
  - Just additional processing required
- Thrown from end event or throwing intermediate event
- Can *only* be caught in boundary event

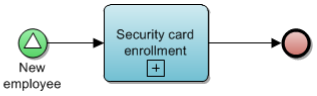
83

83

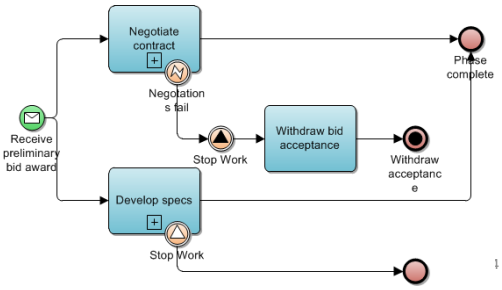
Signal Event

- Two applications:
  - Intra-process signaling (unicast!)
  - Publish-subscribe integration (multicast!)





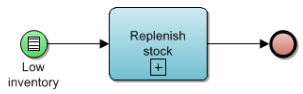
- More flexible than Terminate



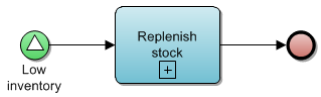
84

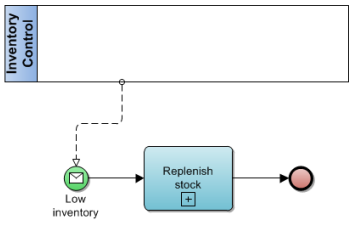
Conditional Event

- Triggered by monitored data



- External trigger:

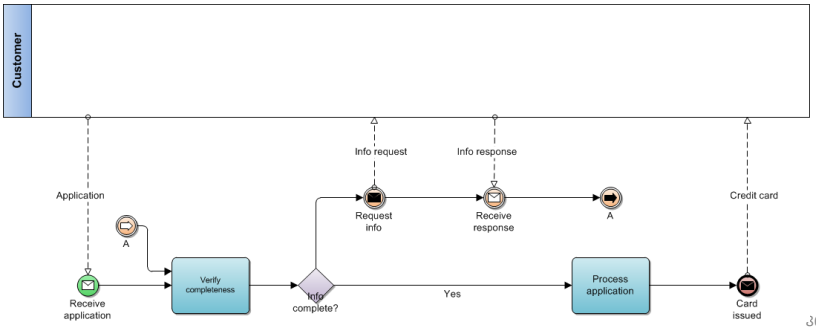




85

Link Events

- Throwing and catching intermediate events
- Off-page connectors
  - Flat model
- On-page connectors



36

86

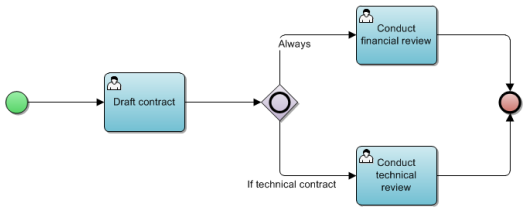
PROCESS SPLITTING AND MERGING

87

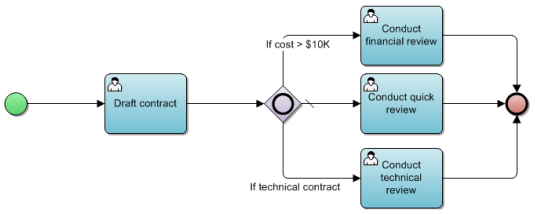
87

OR Gateway Split

- Inclusive gateway
  - Boolean conditions are independent
  - Parallel execution



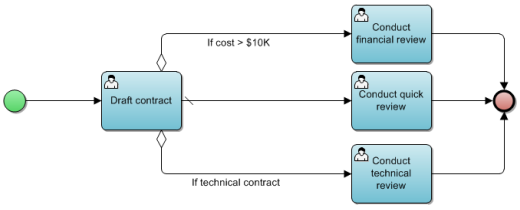
  - Default flow

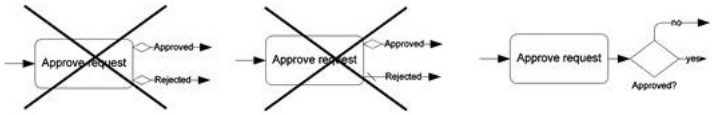


88

Conditional Sequence Flow

- No gateway
- Only for tasks
- Reserve for conditional parallel flow
  - Not XOR, exclusive choice



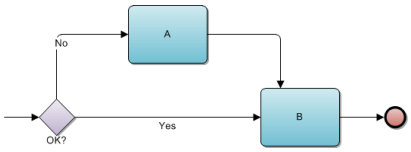


89

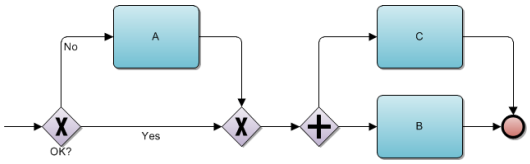
89

Merging Sequence Flows

- Exclusive OR Split
- Merging Alternative Paths
  - Merge into an activity



  - Merge into XOR gateway

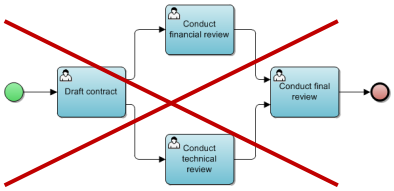


90

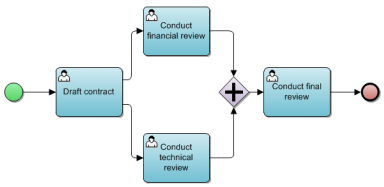
90

Merging Sequence Flows

- Parallel Split
- AND Gateway Join
  - Do NOT merge into an activity
    - Downstream activities triggered multiple times
    - Multi-merge



  - DO use AND gateway
    - Wait for all paths to complete

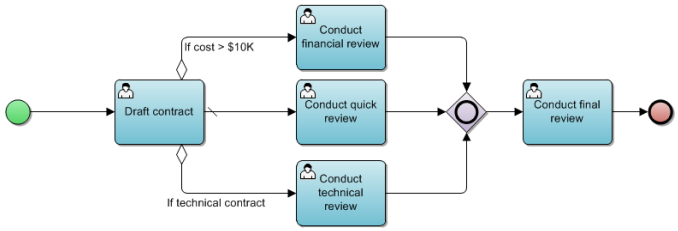


91

91

OR Gateway Join

- Use Case #1: Join conditional sequence flows
  - May be parallel

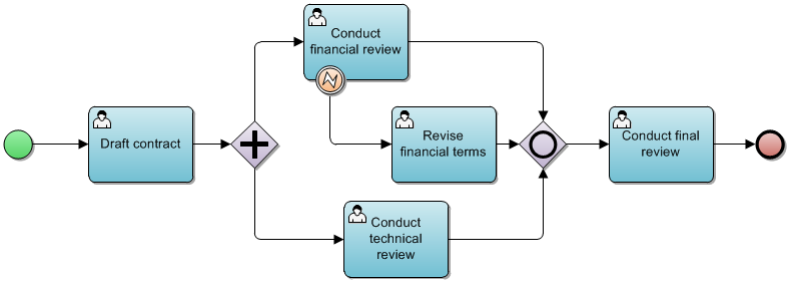


92

92

OR Gateway Join

- Use Case #2: Join while ignoring “dead” parallel flows



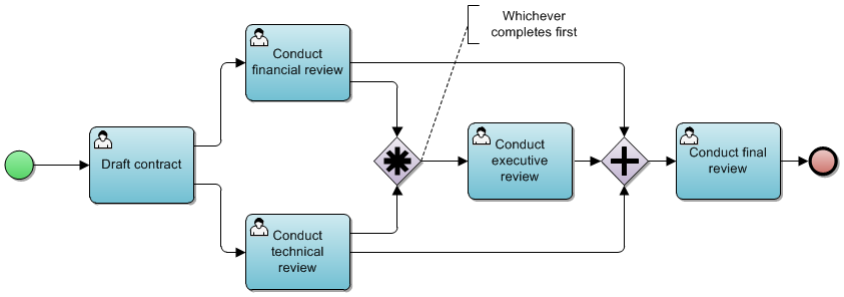
- Use Case #3: Join exception flow path from non-interrupting boundary event

93

93

Discriminator Pattern

- Complex Gateway
- Discriminator Pattern:
  - Accept first path
  - Block all others



94

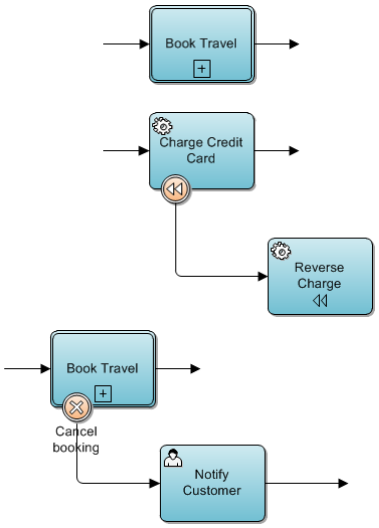
TRANSACTIONS AND COMPENSATIONS

95

95

Transactions

- ACID Transactions
  - Short-lived
- Business Transactions
  - Long-lived
- Compensations
- Cancel Event
  - Internal Error event
  - Compensations for nested activities

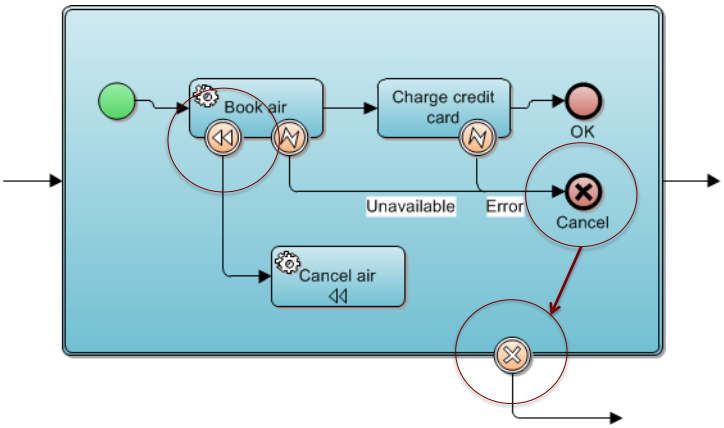


96

96

Cancellation

- Simple case: Cancel airline reservation
  - Cause: credit card failure

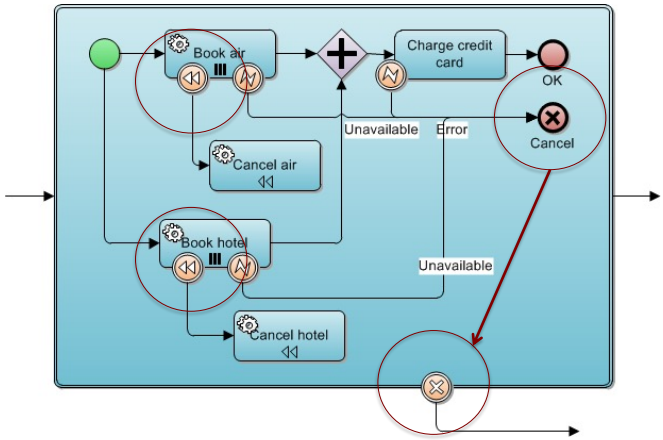


97

97

Cancellation

- Complex case: Multiple possible causes of failure



98

98

Compensation Throw-Catch

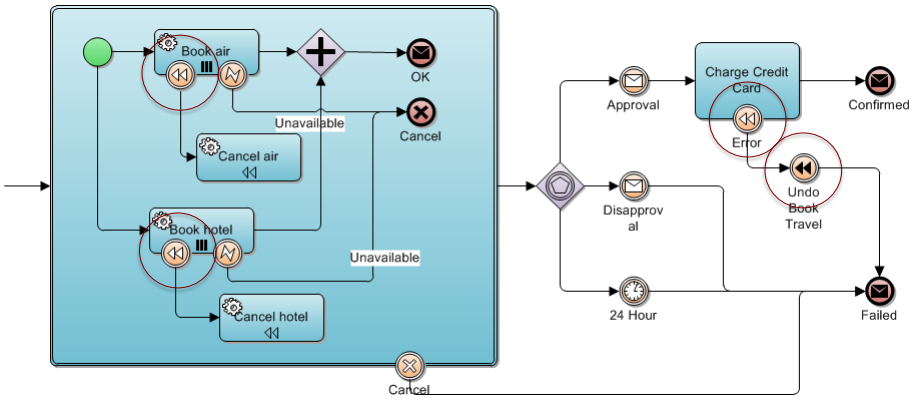
- Throwing Compensation event
  - Intermediate event
  - End event
- Target: activity to be compensated
  - *Not* a boundary event!
- Use case: undo transaction after completion

99

99

Compensation Throw-Catch

- Undo top-level transaction after completion
  - No Cancel event
  - Compensation does not handle exception



100

Summary

- BPMN: Standardized workflow language
- Levels
  - Descriptive
  - Analytic
  - Executable
- Semantics?
  - BPMN: Event-based (pi-calculus)
  - Workflow languages: Flow-based (Petri nets)

101

101