Object-Oriented Design

Lecturer: Raman Ramsin

Lecture 12:
Activity Diagrams – Part 2
Analysis Workflow: *Analyze a Use Case*

- The *analysis workflow* consists of the following activities:
  - Architectural analysis
  - **Analyze a use case**
    - Outputs:
      - analysis classes
      - *use case realizations*
  - Analyze a class
  - Analyze a package
Connectors

![Diagram of Connectors]
Interruptible Activity Regions

- interrupted when a token traverses an interrupting edge.
- all flows in the region are aborted when it is interrupted.
- interrupting edges are drawn as a zigzag arrow or as a normal arrow with a zigzag icon above it.
Exception Handling

- Exception pins:
  - output an exception object from an action;
  - are indicated with an equilateral triangle.

- Protected nodes:
  - have an interrupting edge leading to an exception handler;
  - abort when an exception of the right type is raised, and flow passes to the exception handler node.
Expansion Nodes

- Represent a collection of objects flowing into or out of an expansion region.
- The region is executed once per input element.
- Constraints:
  - the type of the output collection must match the type of the input collection;
  - the type of object held in the input and output collections must be the same.
- Modes:
  - Iterative - process each element of the input collection sequentially;
  - Parallel - process each element of the input collection in parallel;
  - Stream - process each element of the input collection as it arrives at the node;
  - there is no default mode.
Expansion Nodes: Example

Diagram showing the process of grading students with iterative expansion nodes.
Sending Signals and Accepting Events

- Signals:
  - information that is passed asynchronously between objects;
  - class stereotyped «signal»;
  - the information is held in the attributes.
Sending Signals and Accepting Events: Action Nodes

- **Send Signal** action node:
  - starts when there is a token on all input pins;
  - executes - a signal object is constructed and sent;
  - then ends and offers control tokens on its output edges.

- **Accept Event** action node:
  - started by an incoming control edge or if no incoming edge, when its owning activity starts;
  - waits for an event of the specified type:
  - outputs a token that describes the event;
  - continues to accept events while the owning activity executes;
  - for a signal event, the output token is a signal.
Sending Signals and Accepting Events: Examples
Streaming

- Option 1:
  - Read mouse port
  - MouseEvent
  - Handle MouseEvent
  - Indicates streaming

- Option 2:
  - Read mouse port
  - MouseEvent
  - Handle MouseEvent
  - Streaming on normal pins

- Option 3:
  - Read mouse port
  - MouseEvent
  - Handle MouseEvent
  - Indicates streaming

- Option 4:
  - Read mouse port
  - MouseEvent
  - Handle MouseEvent
  - Streaming on object nodes (stand-alone pins)
Advanced Object Flows

- input and output effects show the effects an action has on its input and output objects:
  - write the effect in braces close to the pin;

- selection - a condition on an object flow that causes it to accept only those objects that satisfy the condition:
  - put the selection condition in a note stereotyped «selection» attached to the object flow;

- transformation - transforms objects in an object flow to a different type:
  - put the transformation expression in a note stereotyped «transformation» attached to the object flow.
Input and Output Effects

```
{transformation} Order.toReceipt() : Receipt

Record Transaction

Accept Payment

Transaction {create}

Order [Paid]

Send Receipt

{timestamp}

Order [{!Paid}]

Send Reminder

«selection» (now - Order.date) > 28 days

input effect

output effect
```
Multicast and Multireceive

- Multicast sends an object to many receivers:
  - stereotype the object flow «multicast».

- Multireceive receives objects from many senders:
  - stereotype the object flow «multireceive».
Parameter Sets

- Parameter sets allow an action to have alternative sets of input and output pins:
  - input parameter sets contain input pins;
  - output parameter sets contain output pins;
  - only one input parameter set and one output parameter set may be used per execution of the action.

Diagram:

Authenticate User

- Choose authentication method
  - Input: [password]
  - Choose authentication method
  - Output: [password]

Get UserName and Password

- Input: [passphrase]
- Output: [UserName and Password]

Get UserName and Passphrase

- Input: [card]
- Output: [UserName and Passphrase]

Get Card and PIN

- Input: [card]
- Output: [User [Authenticated]]

Authenticate Password

- Input: [Password]
- Output: [User [Authenticated]]

Authenticate

- Input: [UserName AND Password] XOR [UserName AND Passphrase] XOR [Card AND PIN]
- Output: [User [Authenticated]] XOR [User [!Authenticated]]
Central Buffer Nodes

- Central buffer node - object nodes that are used specifically as buffers:
  - stereotype the object node «centralBuffer».

![Diagram of Central Buffer Nodes]
Interaction Overview Diagrams

- Interaction overview diagrams show flow between interactions and interaction occurrences.

Diagram details: Interaction overview diagrams illustrate the flow between lifelines, interactions, and interaction occurrences. The diagram includes lifelines such as Registrar, RegistrationManager, and Course, with interactions like LogOn, GetCourseOption, AddCourse, RemoveCourse, and FindCourse. The diagram also shows inline and interaction occurrence interactions.
Reference