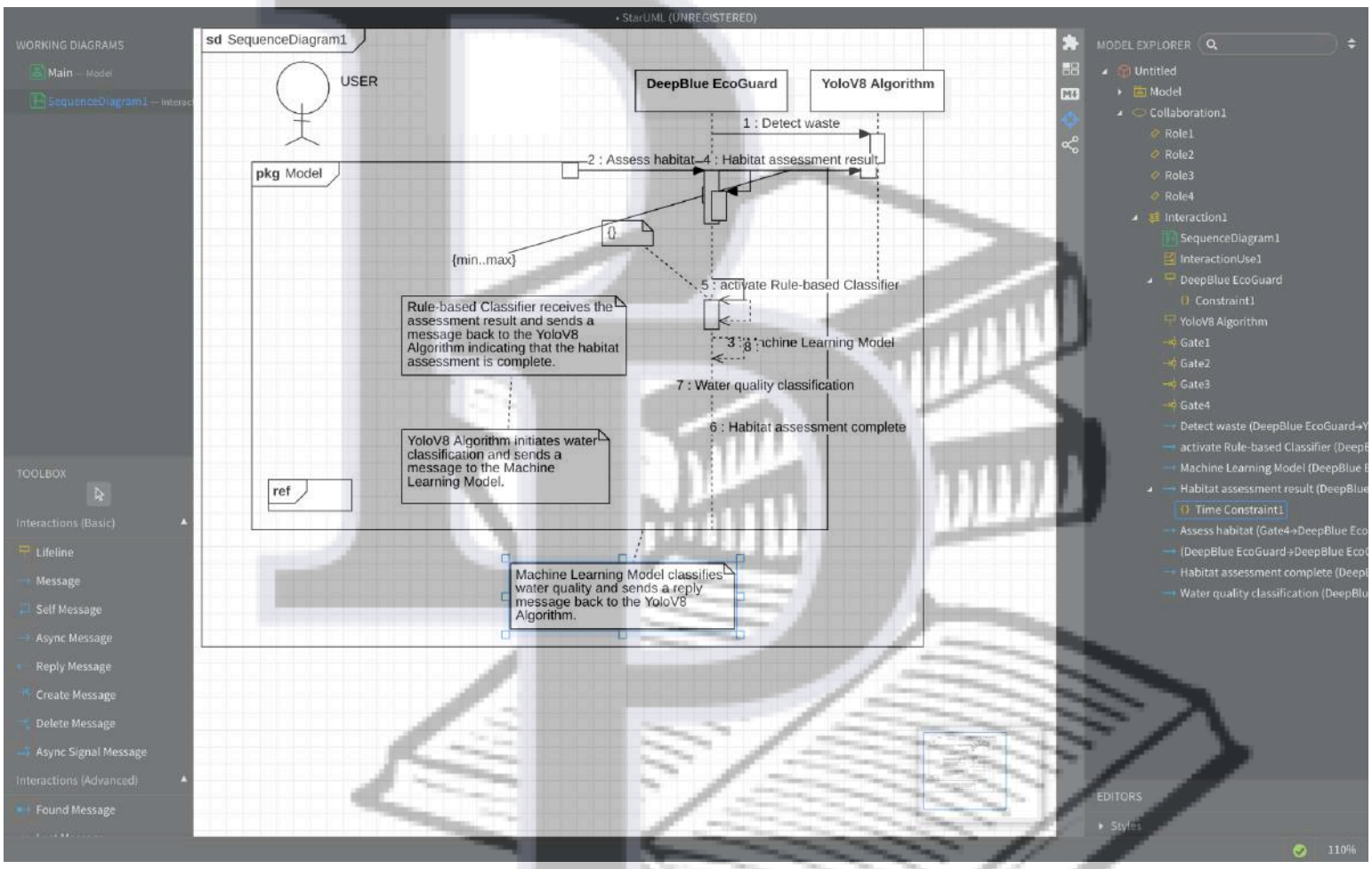
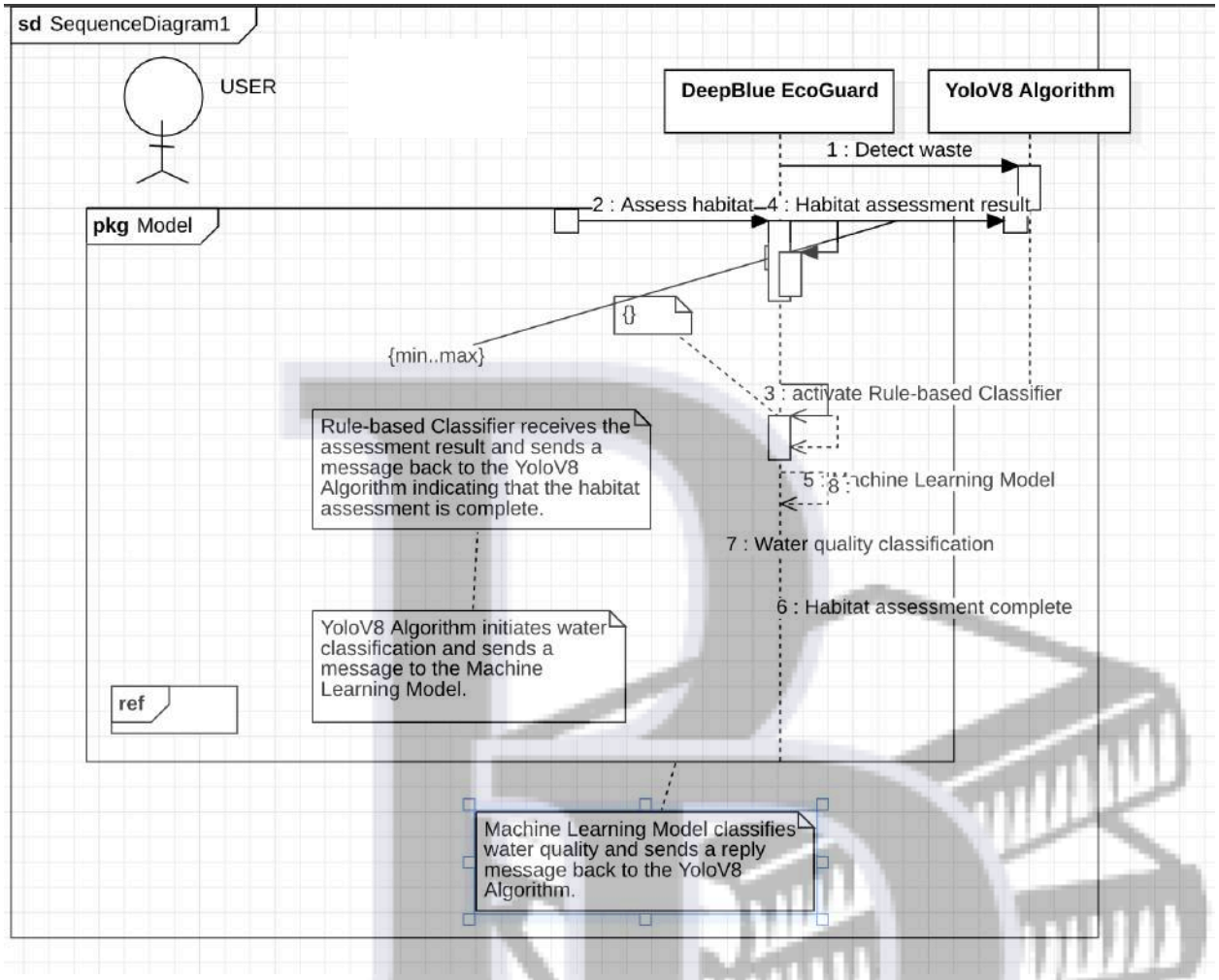
Software Engineering Lab
Assignment 4

SLOT: L21+L22
COURSE CODE: BCSE301P

1. Sequence Diagram



PAJAMA PADHAI



Creating Sequence Diagram using StarUML:

To create a Sequence Diagram:

1. Select first an element where a new Sequence Diagram to be contained as a child.
2. Select **Model | Add Diagram | Sequence Diagram** in Menu Bar or select **Add Diagram | Sequence Diagram** in Context Menu.

You can show or hide sequence numbers of messages. To show or hide sequence numbers of message:

1. Check or Uncheck `showSequenceNumber` property of **Sequence Diagram** or **Communication Diagram**.

PAJAMA PADHAI

To create a Lifeline:

1. Select **Lifeline** in **Toolbox**.
2. Drag on the diagram as the size of Lifeline.

To create a Lifeline from a Classifier (Class, Interface, etc.) by Drag-and-Drop:

1. Drag a Classifier from **Explorer**.
2. Drop on the diagram.

You can use **QuickEdit** for Lifeline by double-click or press `Enter` on a selected Lifeline.

- **Lifeline Expression** : Edit lifeline expression.

  Syntax of Lifeline Expression

```
lifeline ::= [ '<<' stereotype `>>` ] [ visibility ] name [ '[' selector ']' ] [ ':' t
stereotype ::= (identifier)
visibility ::= '+' | '#' | '-' | '~'
name ::= (identifier)
selector ::= (string)
type ::= (identifier)
```

- **Visibility** : Change visibility property.
- **Add Note** : Add a linked note.
- **Add Constraint** : Add a constraint.
- **Select Type** : Select a type of the lifeline.
- **Create Type** : Create a Class as a type of the lifeline.
- **Add Message with Lifeline** : Add a message with a lifeline.
- **Add Create Message with Lifeline** : Add a create message with a lifeline.
- **Add Self Message** : Add a self message.
- **Add Found Message** : Add a found message.
- **Add Lost Message** : Add a lost message.
- **Add Message from Gate** : Add a message from a gate.
- **Add Message to Gate** : Add a message to a gate.

PAJAMA PADHAI

To create a Message (or Self Message):

1. Select **Message** (or **Self Message**) in **Toolbox**.
2. Drag from a Lifeline and drop on another Lifeline. (Just click on a Lifeline if you want to create a self message.)

You can change the kind of message by setting `messageSort` property in **Property Editor**:

- `synchCall` : Synchronous Call
- `asynchCall` : Asynchronous Call
- `asynchSignal` : Asynchronous Signal
- `createMessage` : Create Message
- `deleteMessage` : Delete Message
- `reply` : Reply Message

You can use custom sequence numbers instead of auto-generated integer sequence numbers.

1. Enter sequence number for each Message's `sequenceNumber` property.
2. Change `sequenceNumbering` property of **Sequence Diagram** or **Communication Diagram** to `custom`.

You can use **QuickEdit** for Message by double-click or press `Enter` on a selected Message.

- **Message Expression** : Edit message expression.

  Syntax of Message Expression

```
message ::= [ '<<' stereotype '>>' ] [ visibility ] [ target '=' ] name [ '(' ar
stereotype ::= (identifier)
visibility ::= '+' | '#' | '-' | '~'
target ::= (identifier)
name ::= (identifier)
arguments ::= (string)
```

- **Visibility** : Change visibility property.
- **Add Note** : Add a linked note.
- **Add Constraint** : Add a constraint.
- **Select Operation** : Select an operation as a signature of the message.
- **Create Operation** : Create an operation as a signature of the message.
- **Select Signal** : Select a signal as a signature of the message.
- **Create Signal** : Create a signal as a signature of the message.
- **Add Reply Message** : Add a reply message.

PADAMA PADHAI

Endpoint

To create an Endpoint:

1. Select **Endpoint** in **Toolbox**.
2. Click at the position on the diagram.

Gate

To create a Gate:

1. Select **Gate** in **Toolbox**.
2. Click at the position on the diagram.

State Invariant

To create a State Invariant:

1. Select **State Invariant** in **Toolbox**.
2. Click on a Lifeline where the State Invariant to be attached.

You can use **QuickEdit** for State Invariant by double-click or press `Enter` on a selected State Invariant.

- **Invariant** : Edit invariant property.

To create a Continuation:

1. Select **Continuation** in **Toolbox**.
2. Drag on the diagram as the size of Continuation.

PAJAMA PADHAI

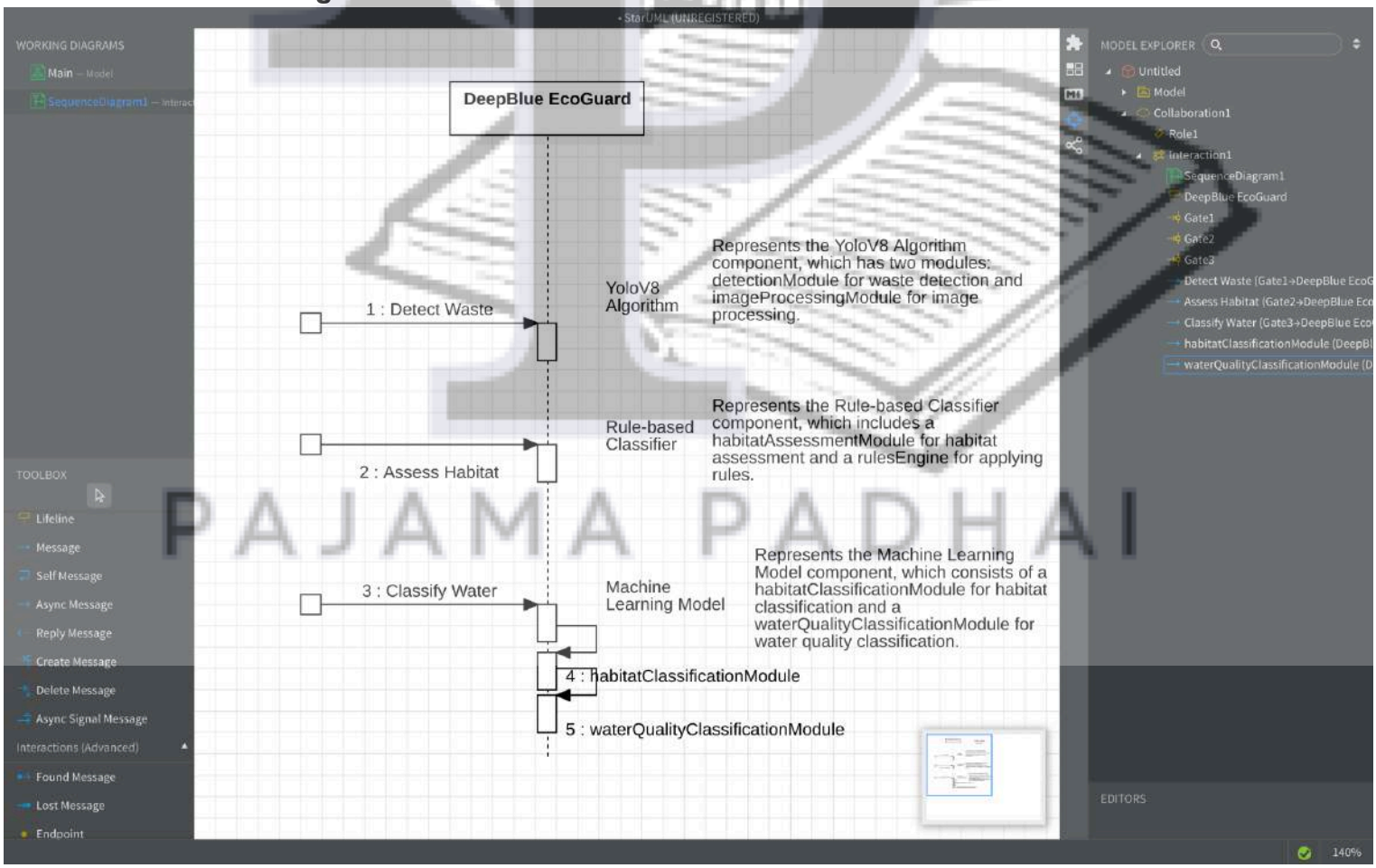
To create a Combined Fragment:

1. Select **Combined Fragment** in **Toolbox**.
2. Drag on the diagram as the size of Combined Fragment.

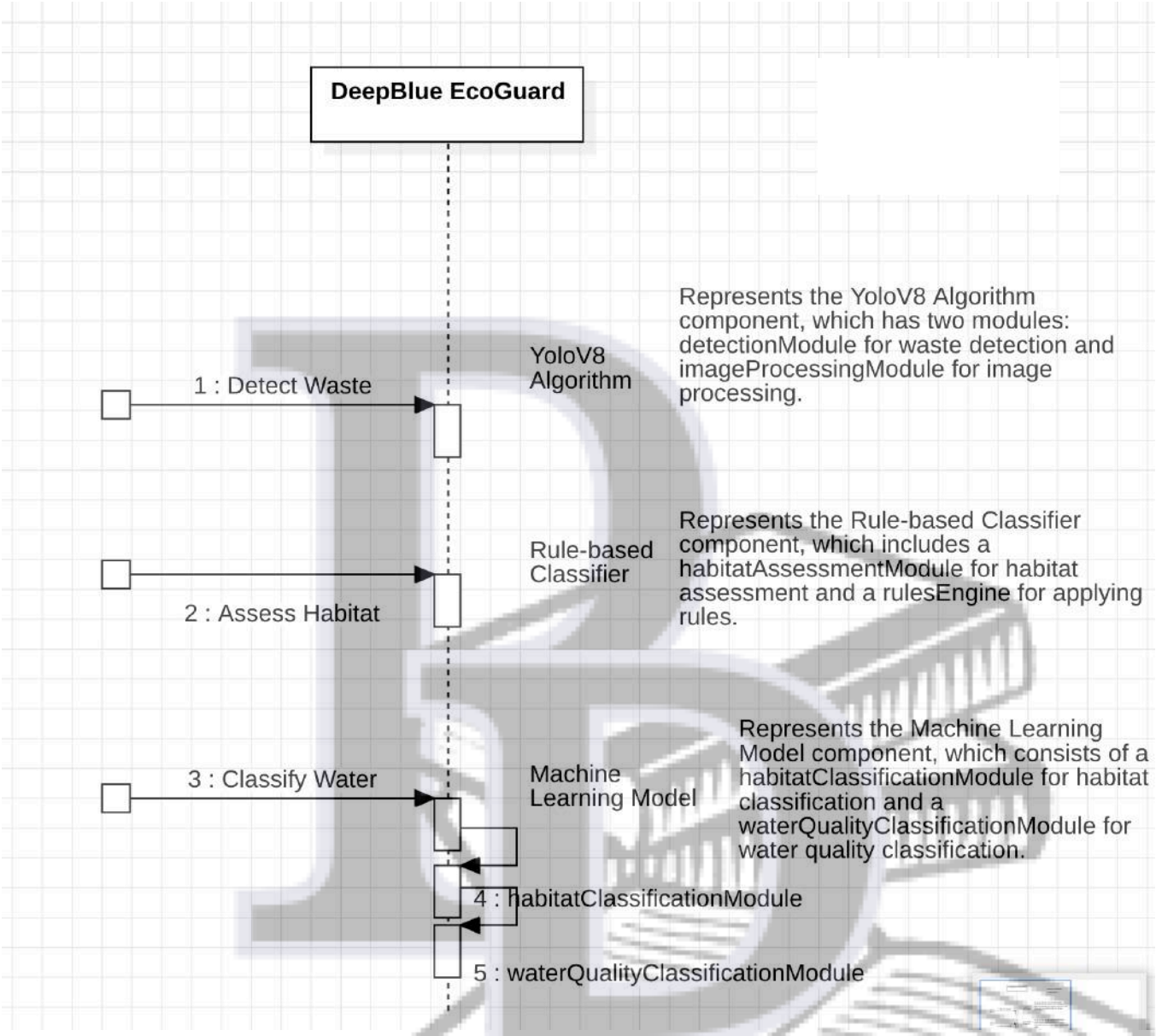
You can change the operator by setting `interactionOperator` property in **Property Editor**:

- `alt` : alternatives
- `opt` : option
- `par` : parallel
- `loop` : iteration
- `critical` : critical region
- `neg` : negative
- `assert` : assertion
- `strict` : strict sequencing
- `seq` : weak sequencing
- `ignore` : ignore
- `consider` : consider
- `break` : break

2. Collaboration Diagram



PAJAMA PADHAI



Explanation:

YoloV8 Algorithm: Represents the YoloV8 Algorithm component, which has two modules: detectionModule for waste detection and imageProcessingModule for image processing.

Rule-based Classifier: Represents the Rule-based Classifier component, which includes a habitatAssessmentModule for habitat assessment and a rulesEngine for applying rules.

Machine Learning Model: Represents the Machine Learning Model component, which consists of a habitatClassificationModule for habitat classification and a waterQualityClassificationModule for water quality classification.

This collaboration diagram illustrates the collaboration between different components and their respective modules within the DeepBlue EcoGuard project. Each component has specific functionalities represented by its associated modules.

Creating Collaboration Diagram using StarUML:

The following elements are available in a collaboration diagram.

Object
Link
SelfLink
Stimulus
Frame
Object
Procedure for creating object
In order to create Object,

Click [Toolbox] -> [Collaboration] -> [Object] button.

And click at the position where Object will be placed in the [main window].

Then quick dialog is shown. At the quick dialog, enter the object name.

And press [Enter] key.

Procedure for creating outgoing from object stimulus by using shortcut creation syntax
In order to create outgoing stimulus from selected object to another object,

Double-click from-object, or select from-object and press [Enter] key to pop up quick dialog.
At the quick dialog, enter stimulus name after "->" string ("<-" string for incoming and "<->" for outgoing with return).

Press [Enter] key and outgoing stimulus from selected object to target object is created and placed at the last order.

Procedure for setting active object
In order to set class to active object,

Set assigned class's [IsActive] property to true.

For MyObject, change MyClass's [IsActive] property.

If class property is not assigned, you can't change object to active object.
Procedure for setting to multi object
In order to set object to multi object,

Set object's IsMultiInstance property to true.

Then the object is assigned as multi object.

PAJAMA PADHAI

Procedure for creating object from class
In order to create object from class,

Select class in the [model explorer].

Drag it into collaboration diagram.

Then the object(instance of the class) is created.

Procedure for creating class from object

If class is not assigned to object,

Double-click object to pop up quick dialog. Then quick dialog is opened.
At the quick dialog, click add class button.

At the [Enter element name] dialog, enter new class name.

Then new class is created and assigned to object.

If you want existing class to be assigned to object, enter the existing class name at the [Select a model element] dialog.

Procedure for adding AttributeLink to object
There are two way to add attribute link to Object.

using object model in the main diagram or the [model explorer]
using [collection editor]
In the case of using object model,

Select object in the [main window] or in the [model explorer].
Right-click the selected object, select [Add] -> [Attribute Link] popup menu, and you can add Attribute Link.

Then new attribute link is created.

In the other case,

Select [Collection Editor...] popup menu of object or click button in [Slots] property on properties window.

At slots tab of the [collection editor], you can add attribute link by using button.

Link
Procedure for creating link
In order to create Link,

Click [Toolbox] -> [Collaboration] -> [Link] button.

Drag from one Object and drop to the other Object in the [main window].

Between two objects, the link is created.

SelfLink
Procedure for creating self-link
In order to create self-link,

Click [Toolbox] -> [Collaboration] -> [SelfLink] button.

And click the object that self-link will connect to in the [main window].

Procedure for creating self-stimulus
In order to create self-stimulus,

Click [Toolbox] -> [Collaboration] -> [ForwardStimulus/ReverseStimulus] button.

PAJAMA PADHAI

And click the self-link that the stimulus will be placed in the [main window].

And double-click the stimulus, enter the stimulus name at the quick dialog.

Stimulus
Procedure for creating stimulus
In order to create stimulus,

Click [Toolbox] -> [Collaboration] -> [ForwardStimulus/ReverseStimulus] button.

Click the link that the stimulus will be placed in the [main window].

And double-click the stimulus, enter the stimulus name at the quick dialog.

The result is as follows.

Procedure for changing ActionKind of stimulus
The [ActionKind] property of stimulus should be assigned to one of five sort as following. To change [ActionKind] property, select stimulus and select the [ActionKind] property on the properties window.

ActionKind Shape
CALL
SEND
RETURN
CREATE
DESTROY
Frame
Procedure for creating frame
생성 방법:
In order to create Frame,

Click [Toolbox] -> [Collaboration] -> [Frame] button.

And click at the position where Frame will be placed in the [main window].

The result is as follows.

Diagram
Procedure for showing sequence numbers in the diagram
In order to show or hide stimulus sequence number, select the diagram in the [model explorer] or in the [main window], and configure [ShowSequence] property of diagram to true or false.

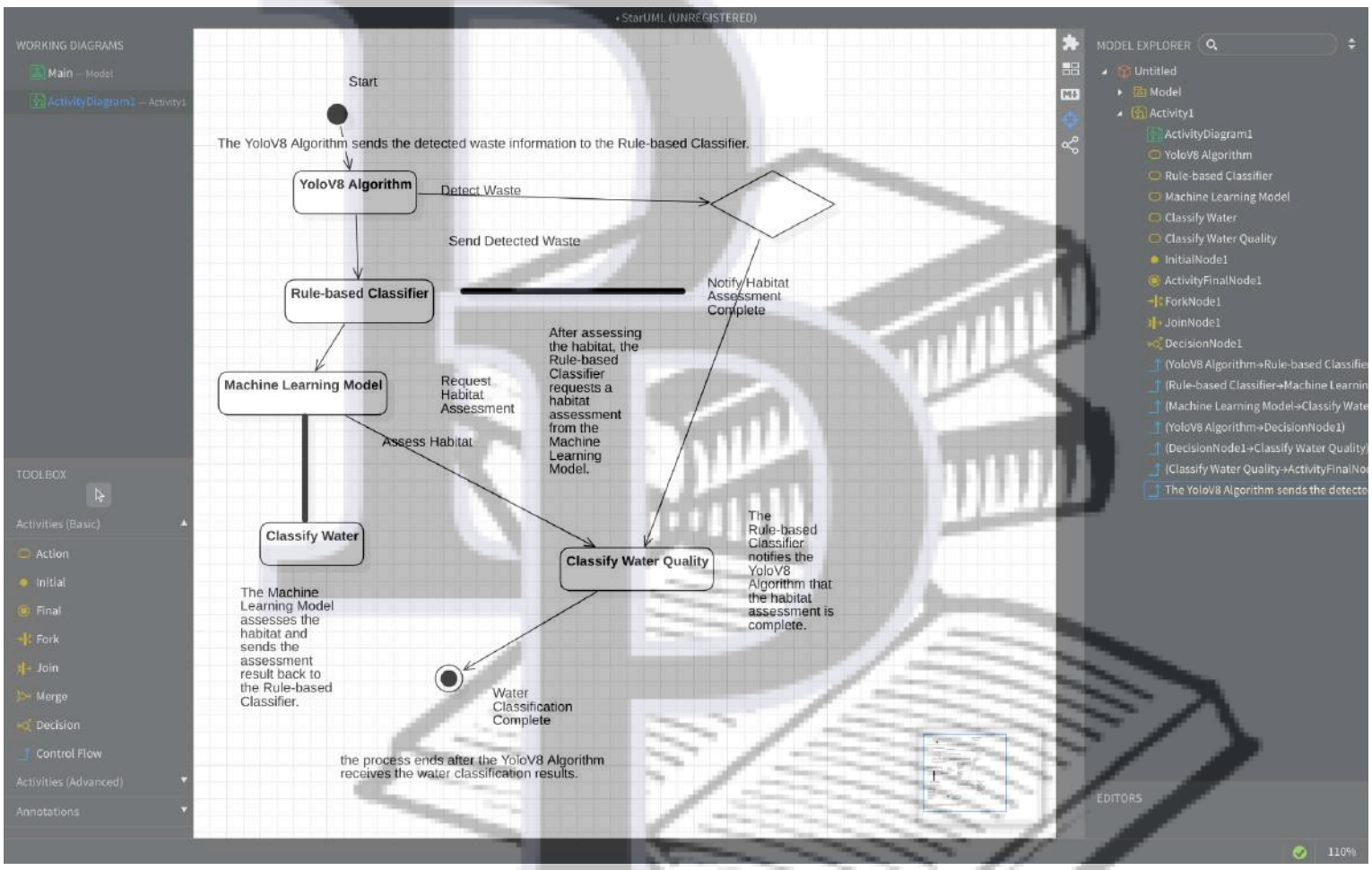
Procedure for changing signature style of message in the diagram
There are four message style. To change stimulus signature,

Select the diagram in the [model explorer] or in the [main window].

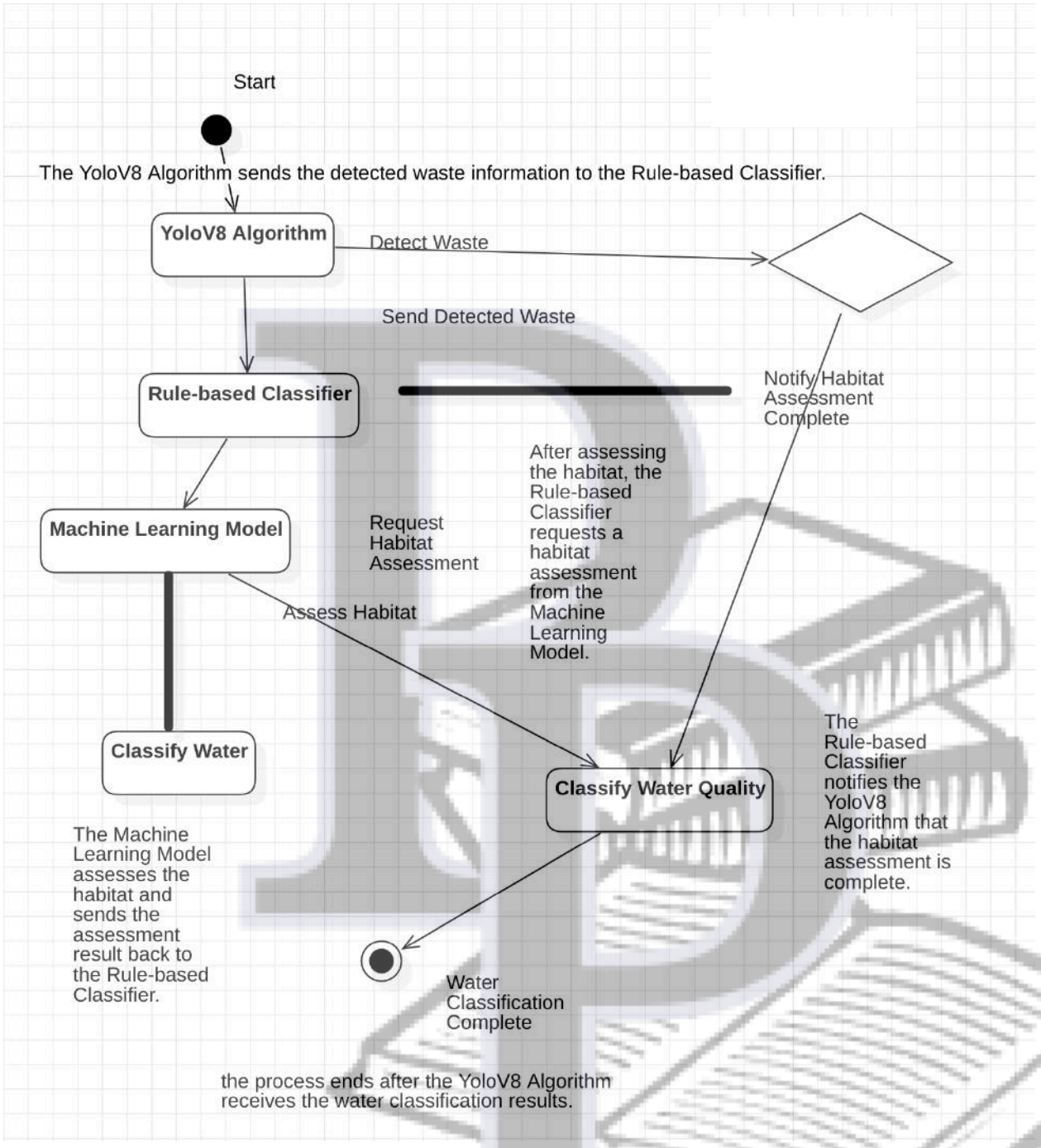
And configure [MessageSignature] property of diagram to one of the followings.

Style	Description
NONE	shows only message name
NAMEONLY	shows message name and arguement name
TYPEONLY	shows message name, arguement type, and return type
NAMEANDTYPE	shows message name, arguement name, arguement type, and return type

3. Activity Diagram



PAJAMA PADHAI



Creating Activity Diagram using StarUML:

To create an Action:

1. Select **Action** in **Toolbox**.
2. Drag on the diagram as the size of Action.

You can use **QuickEdit** for Action by double-click or press `Enter` on a selected Action.

- **Name Expression** : Edit name expression.

  Syntax of Name Expression

```
expression ::= [ '<<' stereotype `>>` ] [ visibility ] name
stereotype ::= (identifier)
visibility ::= '+' | '#' | '-' | '~'
name ::= (identifier)
```

PAJAMA PADHAI

- **Add Input Pin** : Add an input pin.
- **Add Output Pin** : Add an output pin.
- **Add Note** : Add a linked note.
- **Add Constraint** : Add a constraint.
- **Add Trigger Event** : Add a trigger event.
- **Add Outgoing Control Flow** : Add an outgoing control flow with an action.
- **Add Incoming Control Flow** : Add an incoming control flow with an action.
- **Add Outgoing Object Flow** : Add an outgoing object flow with an object node.
- **Add Incoming Object Flow** : Add an incoming object flow with an object node.
- **Add Decision** : Add a decision with two additional actions.
- **Add Merge** : Add a merge with two additional actions.
- **Add Fork** : Add a fork with two additional actions.
- **Add Join** : Add a join with two additional actions.
- **Add Initial Node** : Add an initial node with a connected control flow.
- **Add Final Node** : Add an final node with a connected control flow.

Trigger

To add a Trigger:

1. Select an Action.
2. Select **Model | Add | Trigger** in Menu Bar or **Add | Trigger** in Context Menu.

Initial Node

To create an Initial Node:

1. Select **Initial** in **Toolbox**.
2. Click at the position on the diagram.

Activity Final Node

To create an Activity Final Node:

1. Select **Activity Final** in **Toolbox**.
2. Click at the position on the diagram.

PAJAMA PADHAI

Fork Node

To create a Fork Node:

1. Select **Fork** in **Toolbox**.
2. Drag on the diagram as the size of Fork.

Join Node

To create a Join Node:

1. Select **Join** in **Toolbox**.
2. Drag on the diagram as the size of Join.

Merge Node

To create a Merge Node:

1. Select **Merge** in **Toolbox**.
2. Click at the position on the diagram.

Decision Node

To create a Decision Node:

1. Select **Decision** in **Toolbox**.
2. Click at the position on the diagram.

Swimlane (Partition)

To create a Swimlane (Vertical or Horizontal):

1. Select **Swimlane (Vertical)** or **Swimlane (Horizontal)** in **Toolbox**.
2. Drag on the diagram as the size of Swimlane.

You can use **QuickEdit** for Model Element (See Model Element).

PAJAMA PADHAI

Interruptible Activity Region

To create an Interruptible Activity Region:

1. Select **Interruptible Activity Region** in **Toolbox**.
2. Drag on the diagram as the size of Interruptible Activity Region.

Structured Activity Node

To create a Structured Activity Node:

1. Select **Structured Activity** in **Toolbox**.
2. Drag on the diagram as the size of Structured Activity Node.

You can use **QuickEdit** for Model Element (See Model Element).

Input Pin

To create an Input Pin:

1. Select **Input Pin** in **Toolbox**.
2. Click on an Action where Input Pin to be attached.

Output Pin

To create an Output Pin:

1. Select **Output Pin** in **Toolbox**.
2. Click on an Action where Output Pin to be attached.

Send Signal

To create a Send Signal:

1. Select **Send Signal** in **Toolbox**.
2. Drag on the diagram as the size of Send Signal.

Send Signal is actually an Action whose kind is `sendSignal`.

Accept Signal

PAJAMA PADHAI

To create an Accept Signal:

1. Select **Accept Signal** in **Toolbox**.
2. Drag on the diagram as the size of Accept Signal.

Accept Signal is actually an Action whose kind is `acceptSignal`.

Accept Time Event

To create an Accept Time Event:

1. Select **Accept Time Event** in **Toolbox**.
2. Drag on the diagram as the size of Accept Time Event.

Accept Time Event is actually an Action whose kind is `timeEvent`.

Flow Final Node

To create a Flow Final Node:

1. Select **Flow Final** in **Toolbox**.
2. Click at the position on the diagram.

Object Node

To create a Object Node:

1. Select **Object Node** in **Toolbox**.
2. Drag on the diagram as the size of Object Node.

Central Buffer

To create a Central Buffer:

1. Select **Central Buffer** in **Toolbox**.
2. Drag on the diagram as the size of Central Buffer.

You can use **QuickEdit** for Object Node (See Object Node).

Datastore

To create a Datastore:

1. Select **Datastore** in **Toolbox**.
2. Drag on the diagram as the size of Datastore.

You can use **QuickEdit** for Object Node (See Object Node).

PAJAMA PADHAI

Activity Parameter Node

To create an Activity Parameter Node:

1. Select **Activity Parameter Node** in **Toolbox**.
2. Drag on the diagram as the size of Activity Parameter Node.

If you set the name as blank, the associated parameter's textual notation will be shown as the name.

You can use **QuickEdit** for Object Node (See Object Node).

Expansion Region

To create a Expansion Region:

1. Select **Expansion Region** in **Toolbox**.
2. Drag on the diagram as the size of Expansion Region.

You can use **QuickEdit** for Model Element (See Model Element).

Object Flow

To create an Object Flow:

1. Select **Object Flow** in **Toolbox**.
2. Drag from a node and drop on another node.

Exception Handler

To create an Extension Handler:

1. Select **Extension Handler** in **Toolbox**.
2. Drag from a node and drop on another node.

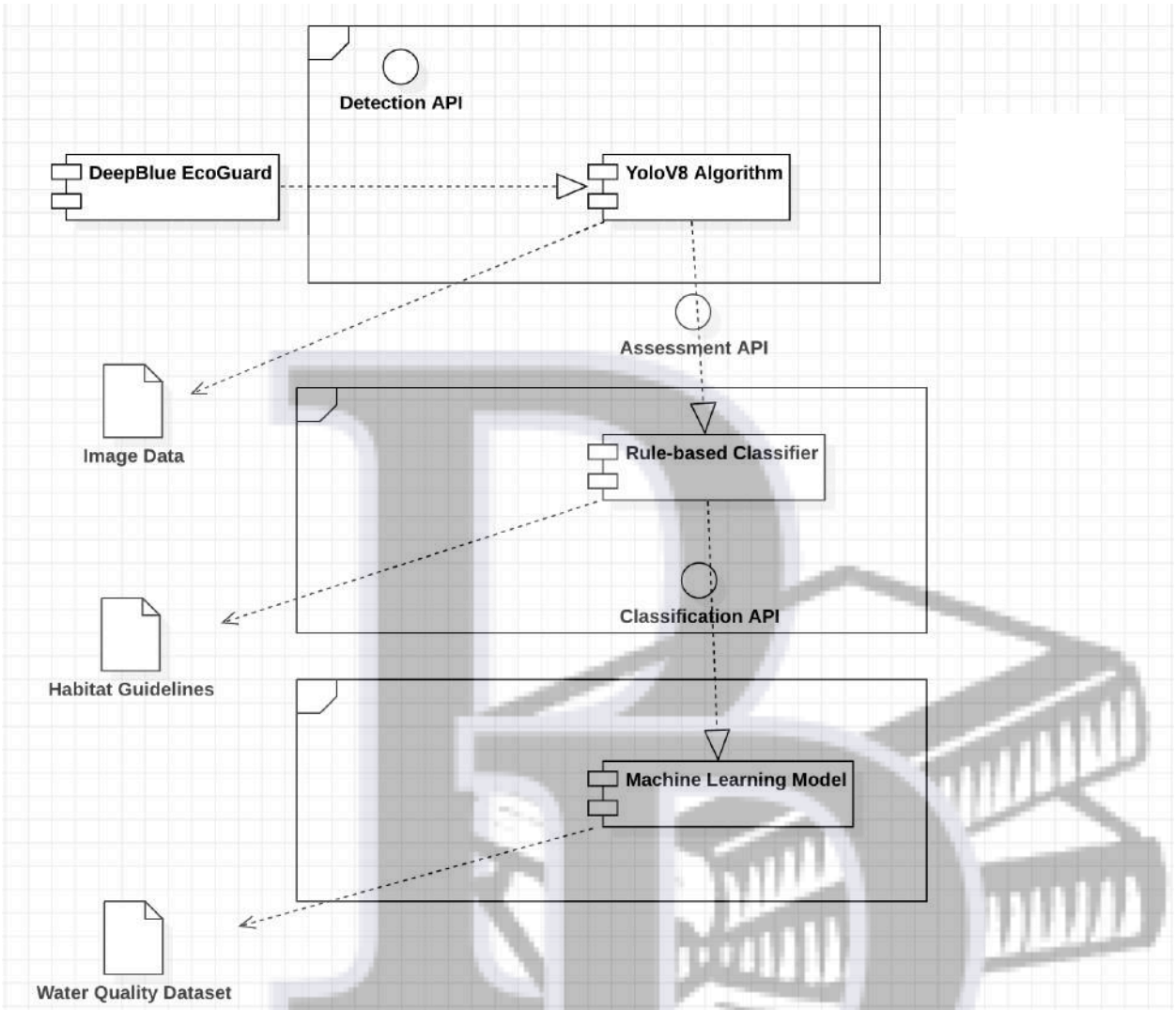
Activity Interrupt

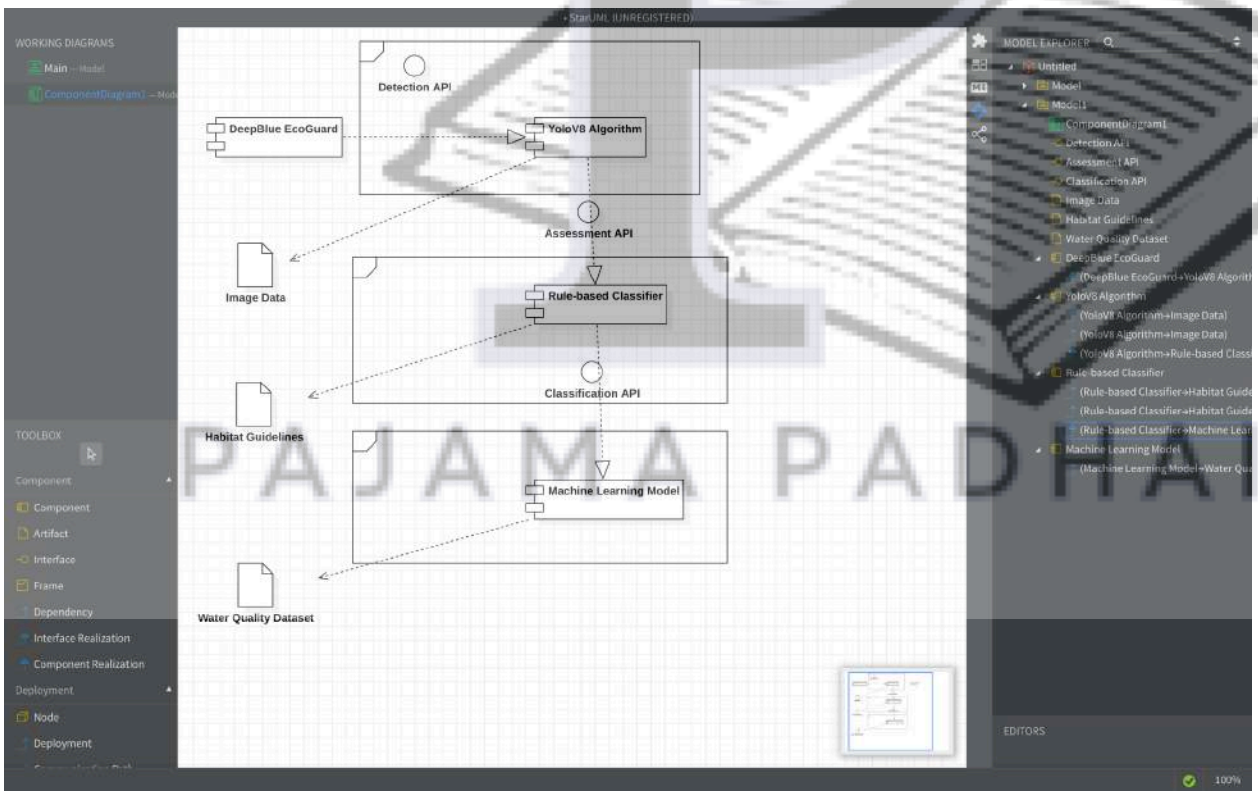
To create an Activity Interrupt:

1. Select **Activity Interrupt** in **Toolbox**.
2. Drag from a node and drop on another node.

PAJAMA PADHAI

4. Component Diagram


Detection API
DeepBlue EcoGuard
YoloV8 Algorithm
Assessment API
Image Data
Rule-based Classifier
Classification API
Habitat Guidelines
Machine Learning Model
Water Quality Dataset


WORKING DIAGRAMS
Main
ComponentDiagram1
Detection API
DeepBlue EcoGuard
YoloV8 Algorithm
Assessment API
Image Data
Rule-based Classifier
Classification API
Habitat Guidelines
Machine Learning Model
Water Quality Dataset
TOOLBOX
Component
Artifact
Interface
Frame
Dependency
Interface Realization
Component Realization
Deployment
Node
Deployment
MODEL EXPLORER
EDITORS

Explanation:

Components: Represented as boxes, these are the main functional units of the DeepBlue EcoGuard system. They include the YoloV8 Algorithm, Rule-based Classifier, and Machine Learning Model.

Artifacts: Represented as dashed boxes, these are the data or information used by the components. Examples include Image Data, Habitat Guidelines, and Water Quality Dataset.

Interfaces: Represented as labeled lines, these define the interactions between components. Examples include Detection API, Assessment API, and Classification API.

Frames: Represented as labeled boxes, these group related components and artifacts together. Examples include Waste Detection Frame, Habitat Assessment Frame, and Water Classification Frame.

Dependencies: Represented as arrows, these indicate that one component relies on another. For example, the YoloV8 Algorithm depends on Image Data.

Interface Realization: Represented as arrows with labels, these indicate how interfaces are implemented by components. For example, the Detection API is realized by the Detection Module in the YoloV8 Algorithm.

Component Realization: Represented as arrows with labels, these indicate how components implement interfaces. For example, the Detection Module realizes the Detection API.

This component diagram provides an overview of the system's structure, showing the relationships between components, artifacts, interfaces, frames, dependencies, interface realizations, and component realizations in the DeepBlue EcoGuard project.

Creating Component Diagram using StarUML:

To create a Component Diagram:

1. Select first an element where a new Component Diagram to be contained as a child.
2. Select **Model | Add Diagram | Component Diagram** in Menu Bar or select **Add Diagram | Component Diagram** in Context Menu.

PAJAMA PADHAI

To create a Component:

1. Select **Component** in **Toolbox**.
2. Drag on the diagram as the size of Component.

To create a Component (model element only) by Menu:

1. Select an Element where a new Component to be contained.
2. Select **Model | Add | Component** in Menu Bar or **Add | Component** in Context Menu.

You can use **QuickEdit** for Component by double-click or press `Enter` on a selected Component.

- **Name Expression** : Edit name expression.

  Syntax of Name Expression

  ```
  expression ::= [ '<<' stereotype `>>` ] [ visibility ] name
  stereotype ::= (identifier)
  visibility ::= '+' | '#' | '-' | '~'
  name ::= (identifier)
  ```

- **Visibility** : Change visibility property.
- **Add Note** : Add a linked note.
- **Add Constraint** : Add a constraint.
- **Add Attribute** (`Ctrl+Enter`) : Add an attribute.
- **Add Operation** (`Ctrl+Shift+Enter`) : Add an operation.
- **Add Reception** : Add a reception.
- **Add Provided Interface** : Add a provided interface.
- **Add Required Interface** : Add a required interface.
- **Add Port** : Add a port.
- **Add Part** : Add a part.

To create a Artifact:

1. Select **Artifact** in **Toolbox**.
2. Drag on the diagram as the size of Artifact.

To create a Artifact (model element only) by Menu:

1. Select an Element where a new Artifact to be contained.
2. Select **Model | Add | Artifact** in Menu Bar or **Add | Artifact** in Context Menu.

To create an Component Realization:

1. Select **Component Realization** in **Toolbox**.
2. Drag from an element (realizing) and drop on a Component (to be realized).

PAJAMA PADHAI