TB141Ic – ICT System Engineering and Rapid Prototyping

MVC in Mendix: Controller - Microflows and Data Validation

Jacopo De Stefani, Ph.D

Delft University of Technology, The Netherlands

27/03/2023

Licensing information

Learning objectives and related literature

Learning objectives

- Create a basic Mendix app
- Identify the components of a Mendix microflow
- Implement data validation via model constraints
- Implement data validation via a microflow

Related literature

- Mendix - Build an App in Mendix Studio Learning Path
- Mendix - Become a Rapid Developer Learning Path
- Mendix - Crash Course

Mendix

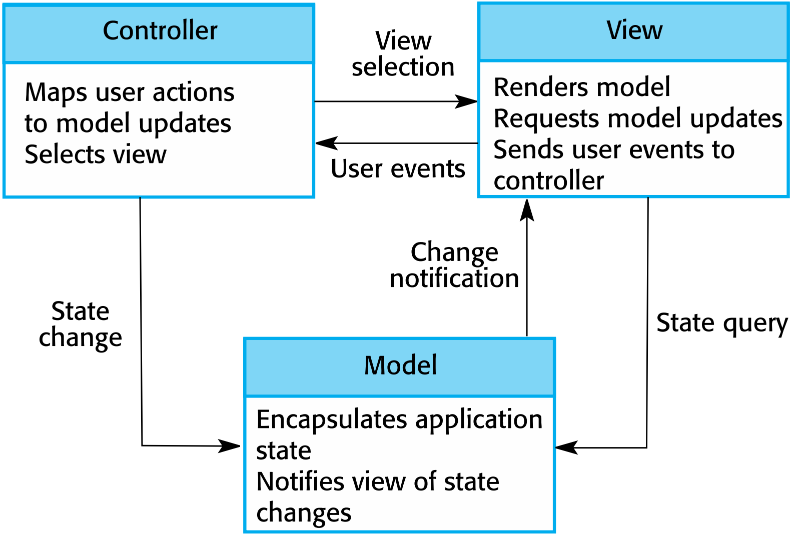
Mendix

- Low-code development platform
- UML-like Diagrams → Code/Data structures
- Visual Sketches → User Interface

Mendix

- **Software Development Methodology:** Agile (with SCRUM project management)
- **SW Architecture:** MVC
- **HW Architecture:** Client-Server
- **Programming Language:** Graphical
  - Domain model ≡ Data structures → UML Class-like diagrams
  - Microflows ≡ Code → UML Activity-like diagrams
  - Pages ≡ User Interface → Graphical sketches

Mendix - MVC



Controller

Maps user actions to model updates
Selects view

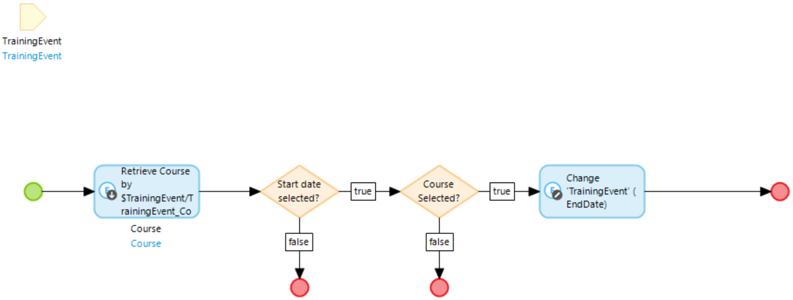
View

Renders model
Requests model updates
Sends user events to controller

Model

Encapsulates application state
Notifies view of state changes

Mendix - Microflow example

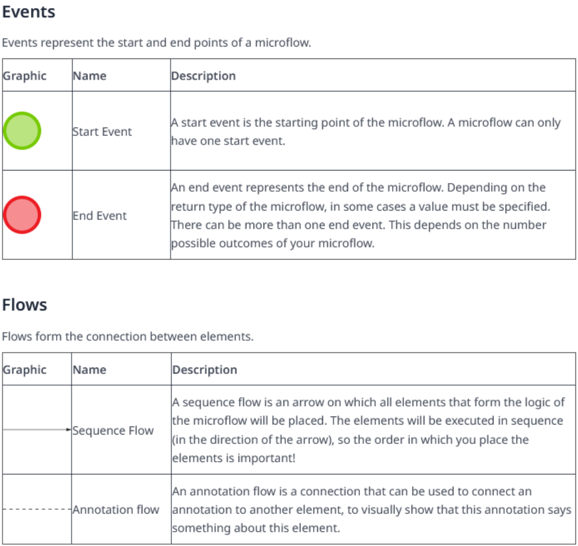


Source: Mendix Academy - Rapid Developer Learning Path

Mendix - Microflows

- Graphical representation of the logic of the applications
  - Close resemblance to UML Activity Diagrams
  - Easily understandable by technical and non-technical people
- Serve different purposes in the application
  - Validate data
  - Execute conditional operations based on input data
  - Create new entities
  - Include advanced built-in actions (integrations with external platforms)

Mendix - Microflows - Elements



Events

Events represent the start and end points of a microflow.

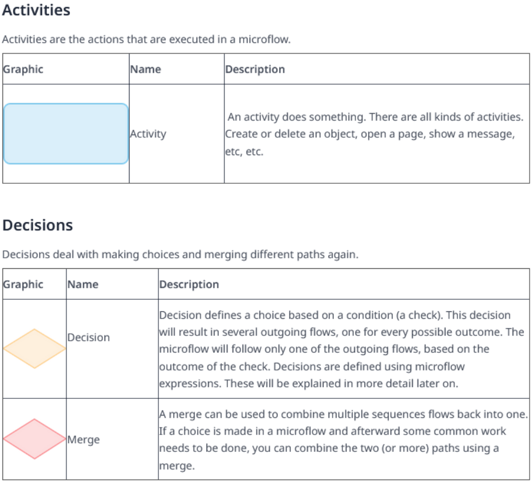
Graphic	Name	Description
🟢	Start Event	A start event is the starting point of the microflow. A microflow can only have one start event.
🔴	End Event	An end event represents the end of the microflow. Depending on the return type of the microflow, in some cases a value must be specified. There can be more than one end event. This depends on the number possible outcomes of your microflow.

Flows

Flows form the connection between elements.

Graphic	Name	Description
——→	Sequence Flow	A sequence flow is an arrow on which all elements that form the logic of the microflow will be placed. The elements will be executed in sequence (in the direction of the arrow), so the order in which you place the elements is important!
- - - - -	Annotation flow	An annotation flow is a connection that can be used to connect an annotation to another element, to visually show that this annotation says something about this element.

Mendix - Microflows - Elements



Activities

Activities are the actions that are executed in a microflow.

Graphic	Name	Description
▭	Activity	An activity does something. There are all kinds of activities. Create or delete an object, open a page, show a message, etc, etc.

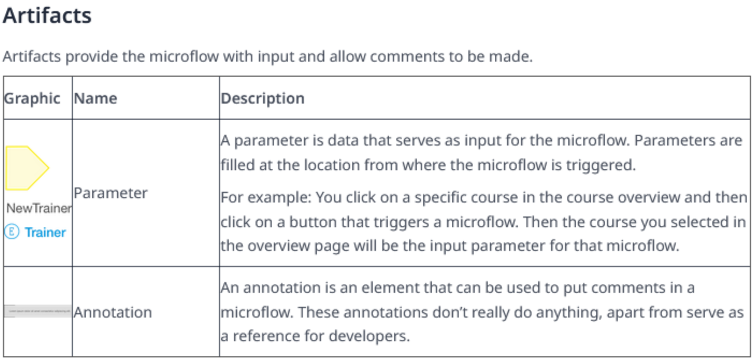
Decisions

Decisions deal with making choices and merging different paths again.

Graphic	Name	Description
◇	Decision	Decision defines a choice based on a condition (a check). This decision will result in several outgoing flows, one for every possible outcome. The microflow will follow only one of the outgoing flows, based on the outcome of the check. Decisions are defined using microflow expressions. These will be explained in more detail later on.
◇	Merge	A merge can be used to combine multiple sequences flows back into one. If a choice is made in a microflow and afterward some common work needs to be done, you can combine the two (or more) paths using a merge.

Source: Mendix Academy - Rapid Developer Learning Path

Mendix - Microflows - Elements



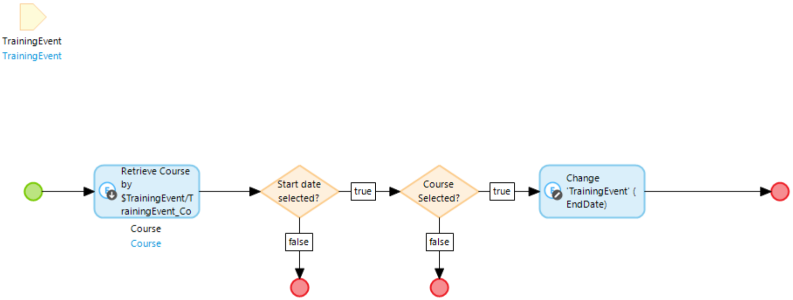
Artifacts

Artifacts provide the microflow with input and allow comments to be made.

Graphic	Name	Description
NewTrainer Ⓔ Trainer	Parameter	A parameter is data that serves as input for the microflow. Parameters are filled at the location from where the microflow is triggered. For example: You click on a specific course in the course overview and then click on a button that triggers a microflow. Then the course you selected in the overview page will be the input parameter for that microflow.
	Annotation	An annotation is an element that can be used to put comments in a microflow. These annotations don't really do anything, apart from serve as a reference for developers.

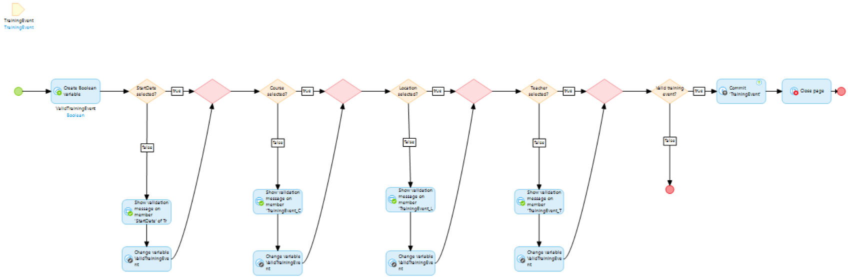
Source: Mendix Academy - Rapid Developer Learning Path

Mendix - Simple Microflow



Source: Mendix Academy - Rapid Developer Learning Path

Mendix - More complex microflow


TrainingEvent
TrainingEvent
Create Boolean variable
ValidTrainingEvent
Boolean
StartDate selected?
true
false
Show validation message on member 'StartDate' of Tr
Change variable ValidTrainingEvent
Course selected?
true
false
Show validation message on member 'TrainingEvent_C
Change variable ValidTrainingEvent
Location selected?
true
false
Show validation message on member 'TrainingEvent_L
Change variable ValidTrainingEvent
Teacher selected?
true
false
Show validation message on member 'TrainingEvent_T
Change variable ValidTrainingEvent
Valid training event?
true
false
Commit 'TrainingEvent'
Close page

Source: Mendix Academy - Rapid Developer Learning Path

Mendix - Validation Rules

Definition

Validation rules are conditions that should be met before an object is stored in the database, placed on the attributes of the entities.

- Validation can occur on **Pages, Domain Model** or **via Microflows**.
- A custom error message can be shown when the validation rule is not matched.

Source: Mendix Academy - Rapid Developer Learning Path

Mendix - Validation Rules

Definition

Validation rules are conditions that should be met before an object is stored in the database, placed on the attributes of the entities.

- Validation can occur on **Pages, Domain Model** or **via Microflows**.
- **On the Pages:**
  - **Local rules:** Validation to pages only applies on that page, not in other places.
  - Defined on a single field level
  - Each field can only have a single validation rule.
- A custom error message can be shown when the validation rule is not matched.

Source: Mendix Academy - Rapid Developer Learning Path

Mendix - Validation Rules

Definition

Validation rules are conditions that should be met before an object is stored in the database, placed on the attributes of the entities.

- Validation can occur on **Pages, Domain Model** or **via Microflows**.
- **On the Domain Model**
    - **Global rules:** Every component, page, or microflow in your project using that object must pass the validation check.
    - Defined on a single attribute level
- A custom error message can be shown when the validation rule is not matched.

Source: Mendix Academy - Rapid Developer Learning Path

Mendix - Validation Rules

Definition

Validation rules are conditions that should be met before an object is stored in the database, placed on the attributes of the entities.

- Validation can occur on **Pages, Domain Model** or **via Microflows**.
- **Via microflows**
  - Custom defined validation rules
  - The level of the rules (global or local) depends on the definition of the microflow
  - Can validate one or more attributes at the same time
- A custom error message can be shown when the validation rule is not matched.

Source: Mendix Academy - Rapid Developer Learning Path

Mendix - Validation Rules

Type	Description
Required	The attribute needs to have a value. It cannot be empty. For example: The description of a course can't be empty.
Unique	The attribute should have a value that is unique compared to the values of this attribute in all other objects of the same entity. For example: A course cannot have the same title as another course which is in the database already.
Equals	The attribute value needs to be equal to a specified value or equal to the value of another attribute of the same object. For example: When you have a form to change your password and you need to enter your new password twice.

Source: Mendix Academy - Rapid Developer Learning Path

Mendix - Validation Rules

Type	Description
Range	The attribute value needs to be in a range between specified values or between the values of other attributes of the same object. For example: The duration of a course needs to be a minimum of 1 day, but 10 days at the most.
Regular expression	The attribute needs to match a regular expression. A regular expression uses pattern recognition to check values. For example: The email address needs to be a valid email address (something@something.com).
Maximum length	The attribute may have no more than the specified number of characters. For example: An American zip code always has five characters, so when someone enters a zip code, the maximum length of that value is five.

Source: Mendix Academy - Rapid Developer Learning Path

Mendix - Validation Rules - What happens in case of failure?

- **Page, with an input widget for the attribute:** The end-user tried to add an object to the database, using a page that has an input widget for the attribute, but did not fill it out correctly. In that case an error message appears underneath the input widget.
- **Page, without an input widget for the attribute:** The end-user tried to add an object to the database, but the attribute that didn't pass the check doesn't have an input widget on the page the end-user is currently looking at. A pop-up message appears, with the error message in it.
- **Microflow:** A microflow attempts to commit the object to the database. In that case an error message appears in the log.

In each of the above cases the object is not committed to the database, preventing incomplete or wrong information from ending up in the database.

Source: Mendix Academy - Rapid Developer Learning Path

Mendix - Validation Rules - What happens in case of failure?

- **Page, with an input widget for the attribute:** The end-user tried to add an object to the database, using a page that has an input widget for the attribute, but did not fill it out correctly. In that case **an error message appears underneath the input widget.**
- **Page, without an input widget for the attribute:** The end-user tried to add an object to the database, but the attribute that didn't pass the check doesn't have an input widget on the page the end-user is currently looking at. A pop-up message appears, with the error message in it.
- **Microflow:** A microflow attempts to commit the object to the database. In that case an error message appears in the log.

In each of the above cases the object is not committed to the database, preventing incomplete or wrong information from ending up in the database.

Source: Mendix Academy - Rapid Developer Learning Path

Mendix - Validation Rules - What happens in case of failure?

- **Page, with an input widget for the attribute:** The end-user tried to add an object to the database, using a page that has an input widget for the attribute, but did not fill it out correctly. In that case an error message appears underneath the input widget.
- **Page, without an input widget for the attribute:** The end-user tried to add an object to the database, but the attribute that didn't pass the check doesn't have an input widget on the page the end-user is currently looking at. **A pop-up message appears, with the error message in it.**
- **Microflow:** A microflow attempts to commit the object to the database. In that case an error message appears in the log.

In each of the above cases the object is not committed to the database, preventing incomplete or wrong information from ending up in the database.

Source: Mendix Academy - Rapid Developer Learning Path

Mendix - Validation Rules - What happens in case of failure?

- **Page, with an input widget for the attribute:** The end-user tried to add an object to the database, using a page that has an input widget for the attribute, but did not fill it out correctly. In that case an error message appears underneath the input widget.
- **Page, without an input widget for the attribute:** The end-user tried to add an object to the database, but the attribute that didn't pass the check doesn't have an input widget on the page the end-user is currently looking at. A pop-up message appears, with the error message in it.
- **Microflow:** A microflow attempts to commit the object to the database. In that case **an error message appears in the log**.

In each of the above cases the object is not committed to the database, preventing incomplete or wrong information from ending up in the database.

Source: Mendix Academy - Rapid Developer Learning Path

Mendix - Development

Now it is your turn to develop...

Mendix - Development - Exercise 7

In the app defined in **Exercise 1**, create two new entities: `Product`, having as attributes: a name, a description and a price and `ProductStock` having as attributes: a `Business`, a `Product` and the quantity of products left in stock.

After that, introduce the following constraints on the **Domain Model** of the application.

Entity	**Attribute**	**Validation Rule**
Business	Name	Required, Unique
	Address	Required
	VatNumber	Required, $<=$ 50 characters
	PhoneNumber	Required, $<=$ 50 characters
	EmailAddress	Required, Email format
Product	Name	Required, Unique
	Price	Required, $>=$ 0
	Description	Required
ProductStock	StartDate	Required
	Business selection	Required
	Product selection	Required
	Quantity	Required,$>=$ 0

The constraints need to be introduced via **Mendix Studio Pro**.

Mendix - Development - Exercise 8

Based on the Domain Model defined in **Exercise 7**, employ a regular expression to enforce a validation rule on the EmailAddress fields, via the **Domain Model** of the application.
The constraints need to be introduced via **Mendix Studio Pro**.

Tip
You don't have to know how to write Regular Expressions. Just use your favorite search engine and look for Regex for the element that you want to validate.

Mendix - Development - Exercise 9

Based on the app defined in **Exercise 7**, create a form to be able to edit a `ProductStock`.

Then, create a microflow connected to the save Button named **ACT_ProductStock_Save**. The microflow will need to:

1. Commit the object to the database
2. Close the page

Mendix - Development - Exercise 10

Update on the microflow defined in **Exercise 9**, to perform a validation on a `ProductStock` object.

The microflow will need to:

1. Check that the `Quantity` is not `empty`

Mendix - Development - Exercise 11

Update on the microflow defined in **Exercise 10**, to perform additional validations on a `ProductStock` object.

The microflow will need to:

1. Check that the `Business` attribute is not empty
2. Check that the `Product` attribute is not empty

Mendix - Development - Exercise 12

Create pages to add new `Products`, `Businesses` and `ProductStocks` in order to test the functioning of the validation processes.