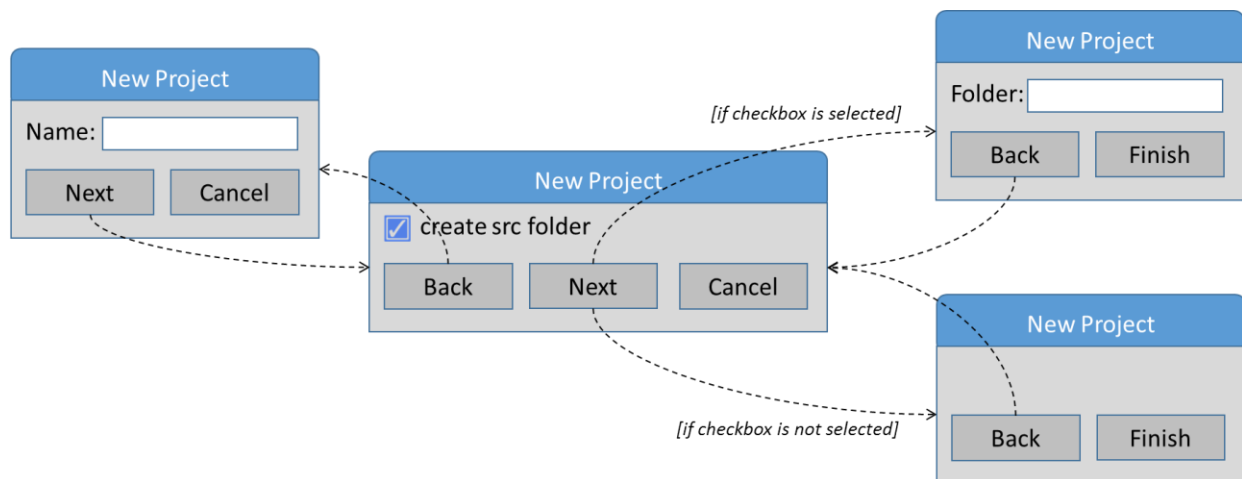


Create a textual domain-specific language that allows defining wizards with the following characteristics.

- A wizard has a title and consists of one or more pages, one of which must be the initial page. Each page of the wizard can include any number of text fields, at most one checkbox, and one or more buttons.
- Each text field has an associated label. It should be possible to specify the minimum and maximum length of the text field, or leave it open.
- Each checkbox has an associated label, and can be initially selected or not.
- Each button has an associated label with at most 10 characters, and there cannot be two buttons with the same label on the same page. Each button can perform one of the following actions: *navigate* to another page of the wizard, *return* to the previous page of the wizard, *show* a given message, or *close* the wizard. The initial page of the wizard cannot contain return-buttons, and the rest of pages can contain at most one return-button. No page can contain only buttons that show messages. A navigation-button cannot define as target the page that contains the button (i.e., self-loops are disallowed). All pages (but the initial one) must be reachable from the initial page by clicking on navigation-buttons. If a page does not include navigation-buttons or return-buttons, then it must contain at least one close-button. The target of a navigation-button may depend on the selection of a checkbox defined in the same page, e.g., going to page “p1” by default, or to page “p2” if the checkbox is selected.

Next, you can find an example of valid wizard according to the previous requirements. In this example, the buttons “Finish” and “Cancel” close the wizard. Other wizards may include close-buttons with a different label.



To build the domain-specific language, you will have to perform the following steps:

- **Step 1:** Definition of an ecore meta-model with the abstract syntax of the language. The constraints of the language can be encoded either using OCL, Xtext validators, or a mix of both.
- **Step 2:** Definition of a textual editor for the language, using Xtext. You should start from the default grammar generated from the ecore meta-model, and then modify the grammar to make the language more usable.
  - **Step 2.1:** Definition of a scope provider that filters non-admissible target pages for navigation buttons.
  - **Step 2.2:** Definition of a quickfix for some of the errors reported by the editor.

Create a zip/rar file containing the following artefacts, and submit it via Moodle by **October, 26th**:

- EMF project
- Projects of tree-like editor, and model created with the editor
- Projects of Xtext editor, and model created with the editor