

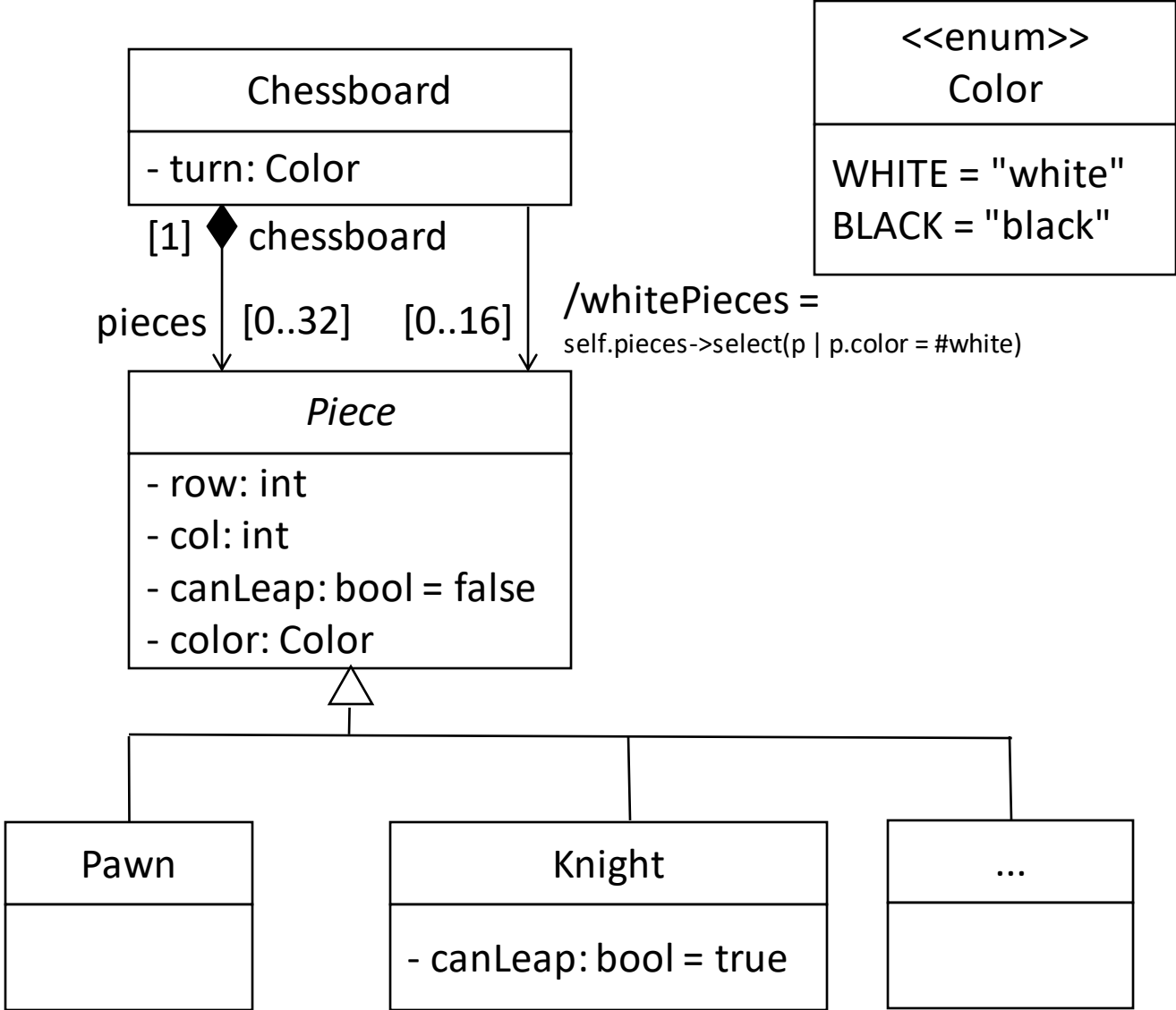
An Integrated CI/CD Workflow for eXecutable Domain-Specific Modelling Languages DevOps Oct. 2021

Nicolas Hili <nicolas.hili@univ-grenoble-alpes.fr>

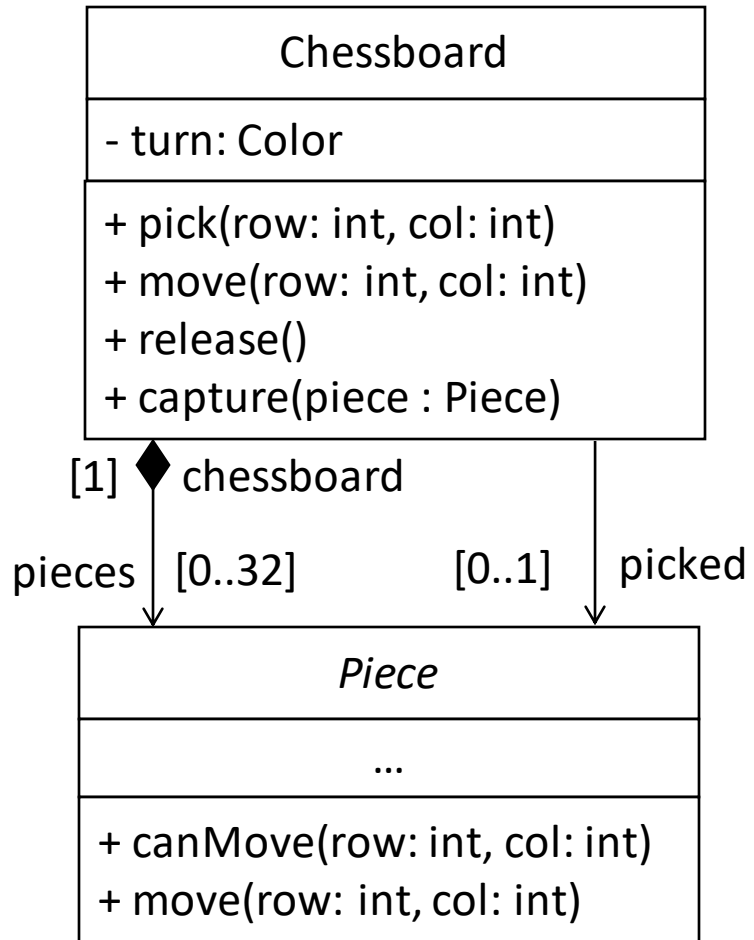


IUT2 
Université Grenoble Alpes

Example: Chess



Example: Chess



Return **TRUE** if:

The piece is moved according to its movement rules

and

The destination is on the board and unoccupied

or

The destination is occupied by an opponent's piece

and

Every cell b/w the current location and the destination is empty

or

The piece can leap

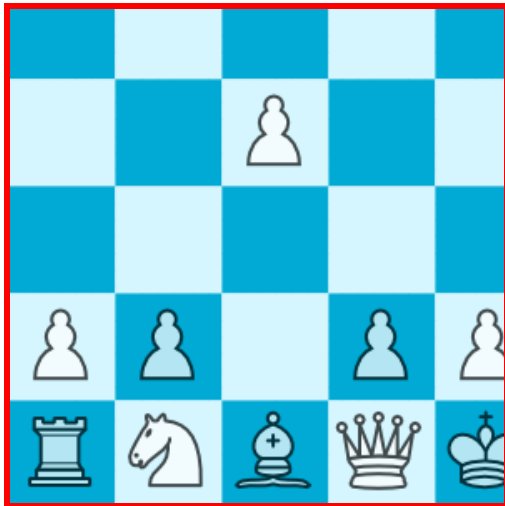
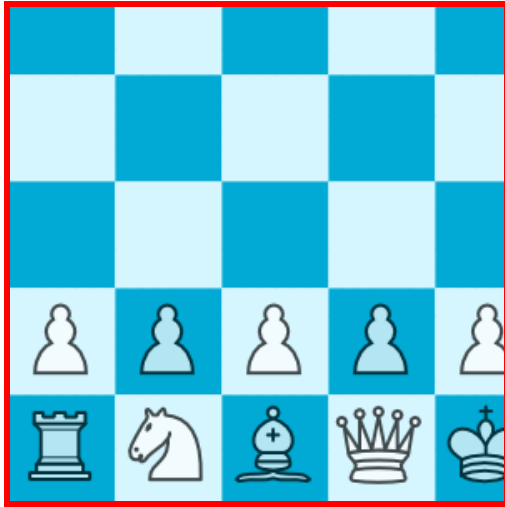
and

The player's king will not be in check at the end of the turn

Return **FALSE** otherwise

What would be the specification of the function
`canMove(row: int, col: int)`?

Test-Driven Development



Test suite #1: pawn moving

Test case #1: pawn can only move forward or diagonally

Test case #2: pawn can move two squares on its first move

Description: on its first move a pawn can advance two squares along the same file, provided both squares are unoccupied.

Initial model state: 

Operation context: 'White Pawn 3'

Operation name: 'move'

Operation parameters: [0, 0]

Expected model state: 

Expected operation result: true

Test case #3: pawn can move one square on its first move

Test case #4: pawn can move one square on subsequent moves

Test case #5: pawn can move diagonally to capture

...

Test suite #2: ...

FlexiMeta

The screenshot displays the FlexiMeta DevOps Interface, which is used for simulating and testing a chessboard model. The interface is divided into several key sections:

- Model Explorer:** Located on the left, it shows a hierarchical tree of the chessboard model. It includes categories for 'chessboard', 'pieces (32)', 'whitePieces (16)', and 'blackPieces (16)'. Under 'whitePieces', individual pieces like 'White pawn 1' through 'White pawn 8', 'White rook 1', 'White knight 1', 'White bishop 1', and 'White queen' are listed, each with associated functions such as `fx canMove(row, col)`, `fx move(row, col)`, `fx canPass(row, col)`, `fx canLand(row, col)`, and `fx canMove(row, col)`. Similar structures exist for black pieces.
- Chessboard:** The central area features a 6x6 grid representing the chessboard. It contains various chess pieces: a white king, white queen, white rook, white knight, white bishop, and white pawn. Black pieces are also present, including a black king, black queen, black rook, black knight, black bishop, and black pawn. The board is set on a blue and white checkered grid.
- Execution Engine:** Located at the bottom left, it allows for running simulations. It includes input fields for 'row' (set to 3) and 'col' (set to 5), and an 'EXECUTE' button.
- Property Panel:** At the bottom center, it displays the properties of the selected element. The ID is `a117e7bf-ace6-415a-a0af-7575b3d0a176`. Other visible properties include `color` (set to 'white'), `row` (set to 4), `col` (set to 6), and `canLeap` (checked).
- Test Suites:** On the right side, a list of test cases is shown, each with a status indicator (checkmark or arrow). The tests include: 'Move bishop' (passed), 'Move pawn' (passed), 'Black cannot move backward' (passed), 'Cannot move pawn forward 2 squares after first move' (passed), 'Move black pawn forward 1 square' (passed), 'Move pawn forward 1 square' (passed), 'Move pawn forward 2 squares from initial row' (passed), 'White cannot move backward' (passed), and 'Pick' (pending).

GitLab workflow

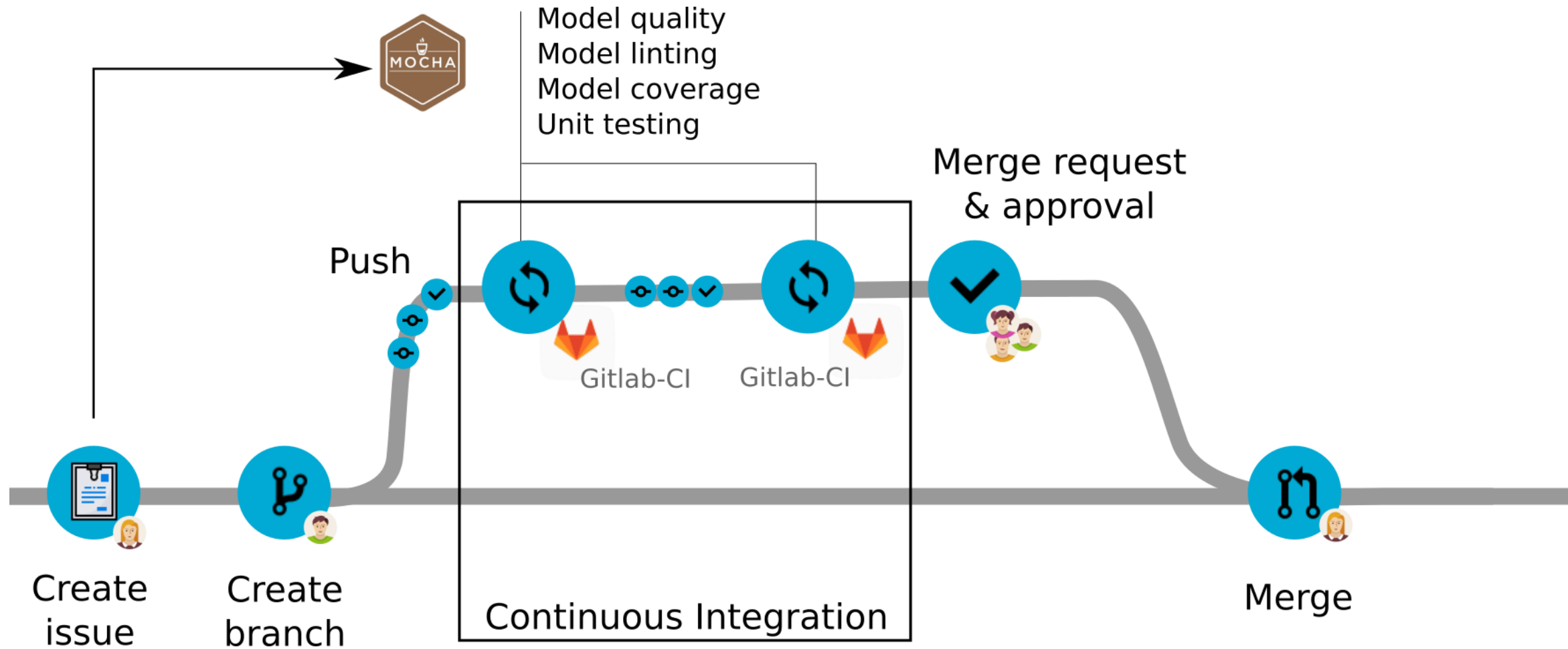


Figure adapted from <https://docs.gitlab.com/ee/ci/introduction/>