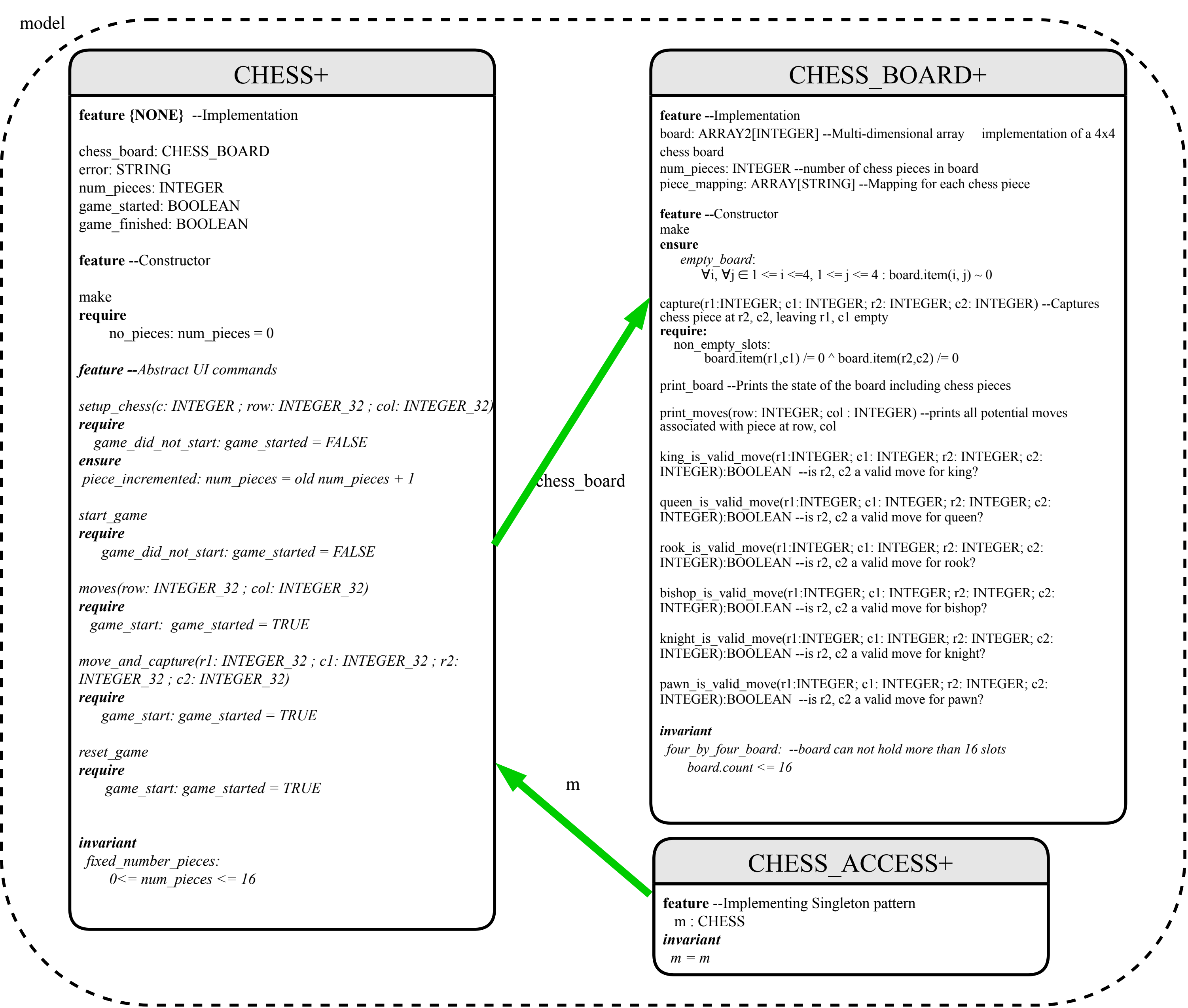

CHESS+

feature {NONE} --Implementation

chess_board: CHESS_BOARD
error: STRING
num_pieces: INTEGER
game_started: BOOLEAN
game_finished: BOOLEAN

feature --Constructor

make
require
no_pieces: num_pieces = 0

feature --*Abstract UI commands*

setup_chess(c: INTEGER ; row: INTEGER_32 ; col: INTEGER_32)
require
game_did_not_start: game_started = FALSE
ensure
piece_incremented: num_pieces = old num_pieces + 1

start_game
require
game_did_not_start: game_started = FALSE

moves(row: INTEGER_32 ; col: INTEGER_32)
require
game_start: game_started = TRUE

move_and_capture(r1: INTEGER_32 ; c1: INTEGER_32 ; r2: INTEGER_32 ; c2: INTEGER_32)
require
game_start: game_started = TRUE

reset_game
require
game_start: game_started = TRUE

invariant
fixed_number_pieces:
0<= num_pieces <= 16

CHESS_BOARD+

feature --Implementation
board: ARRAY2[INTEGER] --Multi-dimensional array implementation of a 4x4 chess board
num_pieces: INTEGER --number of chess pieces in board
piece_mapping: ARRAY[STRING] --Mapping for each chess piece

feature --Constructor
make
ensure
empty_board:
∀i, ∀j ∈ 1 <= i <=4, 1 <= j <= 4 : board.item(i, j) ~ 0

capture(r1:INTEGER; c1: INTEGER; r2: INTEGER; c2: INTEGER) --Captures chess piece at r2, c2, leaving r1, c1 empty
require:
non_empty_slots:
board.item(r1,c1) /= 0 ^ board.item(r2,c2) /= 0

print_board --Prints the state of the board including chess pieces

print_moves(row: INTEGER; col : INTEGER) --prints all potential moves associated with piece at row, col

king_is_valid_move(r1:INTEGER; c1: INTEGER; r2: INTEGER; c2: INTEGER):BOOLEAN --is r2, c2 a valid move for king?

queen_is_valid_move(r1:INTEGER; c1: INTEGER; r2: INTEGER; c2: INTEGER):BOOLEAN --is r2, c2 a valid move for queen?

rook_is_valid_move(r1:INTEGER; c1: INTEGER; r2: INTEGER; c2: INTEGER):BOOLEAN --is r2, c2 a valid move for rook?

bishop_is_valid_move(r1:INTEGER; c1: INTEGER; r2: INTEGER; c2: INTEGER):BOOLEAN --is r2, c2 a valid move for bishop?

knight_is_valid_move(r1:INTEGER; c1: INTEGER; r2: INTEGER; c2: INTEGER):BOOLEAN --is r2, c2 a valid move for knight?

pawn_is_valid_move(r1:INTEGER; c1: INTEGER; r2: INTEGER; c2: INTEGER):BOOLEAN --is r2, c2 a valid move for pawn?

invariant
four_by_four_board: --board can not hold more than 16 slots
board.count <= 16

CHESS_ACCESS+

feature --Implementing Singleton pattern
m : CHESS
invariant
m = m