**AspectOCL Constraints- Case Study 2**

**AspectOCL Constraint 1:**

**mapping** mapSudokuUnique

{

**let** T -> A : { RegisteredUser -> RegisteredUser :: username

Sudoku -> Sudoku :: id\_sudoku

}

}

**aspect** UniquenessSudoku

{

**import\_mapping** mapSudokuUnique

**pointcut** uniqueSudoku

**context** T

**intro:**

**inv** uniqueSudoku

T -> allInstances() -> **isUnique**(A)

}

**AspectOCL Constraint 2:**

**mapping** mapSudokuSize

{

**let** T -> A : { Sudoku -> Row :: row\_num

Sudoku -> Column :: col\_num

RowCell -> Cell:: correct\_value

ColumnCell -> Cell:: correct\_value

Region -> Cell:: correct\_value

}

}

**aspect** SizeOfSudoku

{

**import\_mapping** mapSudokuSize

**pointcut** SizeSudoku

**context** T

**intro:**

**inv** SizeSudoku

T.allInstance() -> **select**(s| s. A <> self. A) -> **size()**=1

}

**AspectOCL Constraint 3:**

**mapping** mapIndividualEffect

{

**let** T -> {S,A} : { NewPlayer :: effect() ->

Player :: play\_att , Player }

NewAdministrator :: effect() ->

Administrator :: admin , Administrator }

}

}

**aspect** EffectOfIndividual

{

**import\_mapping** mapIndividualEffect

**pointcut** effect

**context** T

**intro:**

**post** effect

S.oclIsNew() **and** oclIsTypeOf(A) and UserHasAttributes(S)

}

**AspectOCL Constraint 4:**

**mapping** mapfinishedStatus

{

**let** T : { SudokuChoice:: UnfinishedSudoku() : Boolean,

GameMove:: UnfinishedSudoku() : Boolean,

IncorrectCellsCheck:: UnfinishedSudoku() : Boolean,

CompoundGameMove:: UnfinishedSudoku() : Boolean

}

}

**aspect** finishedSudoku

{

**import\_mapping** mapfinishedStatus

**pointcut** selectsudokoFinish

**context** T

**intro:**

**pre** selectsudokoFinish

sudoku.finished = false

}

**AspectOCL Constraint 5:**

**mapping** mapMoveEffect

{

**let** T -> {A, B, S}: { Undo:: effect() ->

{Undo, UndoMove, Sudoku :: lastDisposableMove }

Redo:: effect() ->

{Redo, RedoMove, Sudoku :: lastUndoneMove }

}

}

**aspect** SudokuEffect

{

**import\_mapping** mapMoveEffect

**pointcut** selectEffectTarget

**context** T

**intro:**

**post** selectEffectTarget

A. oclIsNew() and A.oslIsTypeOf(B) and A. nonPredefinedCell = sudoku. S

Undo. oclIsNew() and undo.

}

**AspectOCL Constraint 6:**

**mapping** mapMailStatus

{

**let** T : { NewRegisteredUser:: CorrectMail() : Boolean

MailUpdate:: CorrectMail() : Boolean

}

}

**aspect** MailCorrection

{

**import\_mapping** mapMailStatus

**pointcut** Mail

**context** T

**intro:**

**body** Mail

mail. CorrectMail()

}

**AspectOCL Constraint 7:**

**mapping** mapCurrentPlayer

{

**let** T : { SudokuChoice:: SudokuIsTheCurrentOfPlayer() : Boolean

IncorrectCellsCheck:: SudokuIsTheCurrentOfPlayer() : Boolean

CompoundGameMove:: SudokuIsTheCurrentOfPlayer() : Boolean

}

}

**aspect** CurrentGamePlayer

{

**import\_mapping** mapCurrentPlayer

**pointcut** GamePlayer

**context** T

**intro:**

**body** GamePlayer

player.currentSudoku = Sudoku **and** sudoku.finished = false

}

**AspectOCL Constraint 8:**

**mapping** mapCellValue

{

**let** T : { PutValueInACell:: CellsPartOfCurrentSudoku() : Boolean

CellCheck:: CellsFromCurrentSudoku() : Boolean

}

}

**aspect** PredefinedCellValue

{

**import\_mapping** mapCellValue

**pointcut** CellValue

**context** T

**intro:**

**body** CellValue

player.currentSudoku.nonPredefinedCell -> **includes** (nonPredefinedCell)

}