Component Specification
FeedbackHandler

FeedbackHandler provides functions for UserInterface to display feedbacks and highlight error location.

Following functions are to be realized:

1. Locate an error in a term and highlight it
   Name and signature:
   locate_error_rec(+ProgramTerm, +Error, -HighlightedProgram)

2. Locate a syntax error and highlight it.
   Name and signature: locate_syntax_error_rec(+Syntax_Error -HighlightedProgram)

3. Return a feedback for an given error:

   Name and signature: return_feedback(+Error, -Feedback)

A feedback contains information about location, explanation and solution of this error. Explanation for an error should only tell half of the source of the error. The other half will be the solution of the error. For example:

error: missing_arg

explanation: `The argument number  for this predicate is not correct.`
help: `please check the argument number for this predicate and add an corresponding argument.`

Especially, error no_pattern_found should be solved by linking to the help page, where there is description of possible patterns which can be used to solve the corresponding exercise.

Error: no_pattern_found

explanation: we cannot understand your solution.
Help: Use can use pattern A or B to solve this exercise. Please read description of these pattern in the help page.

Errors of type missing/superfluous clauses/subgoals should be solved by giving a link to the corresponding pattern.

Error: missing/superfluous clause/subgoal

explanation: you have more/less clause/subgoal than necessary.
Help: please add/remove the clause/subgoal required. More information for this pattern can be found here. (a link to a concerning page)

We have following potential errors:

Error	**Note**
matching_error(Index, missing_basecase (Clause), 20)	Index: [C], C is clause number
matching_error(Index, missing_recursivecase (Clause),20)	
matching_error(Index, superluous_basecase (Clause), 20)	
matching_error(Index, superfluous_recursivecase(Clause), 20)	
matching_error(Index, missing_arg(Arg), 10)	Index: [T,..,C], T is position of a term, C is position of a clause. Between clause and term there may be nested terms. Normally the index in this case is [S,C], S is subgoal position. Because the argument is missing we cannot locate the argument, but we can locate the subgoal. Just say "this subgoal need more arguments."
matching_error(Index, superfluous_arg(Arg), 10)	Index: [A,T,..,C] as above, A is position of superfluous argument. Between T and C and nested terms
matching_error(Index, missing_subgoal(X), 15)	Index: [C] C is clause position. We cannot highlight the position of missing subgoal, just the position of the clause.
matching_error(Index, superfluous_subgoal (X), 15)	Index: [S,C], S is subgoal, C is clause position
matching_error(Index, arg_unmatchable (ArgStudent,ArgPattern), 10)	Index: [A,T,..,C] as above. Explain: this argument does not fit in this predicate. Help: see pattern description. More information please read StructureMatcher_diagnoseSpec.doc
matching_error(Index, misspelled_predicate (Wrong,Right), 15)	Index [S,C], Wrong is the wrong predicate name of the student. Right is the expected predicate name.
matching_error(Index, unknown_predicate (Pred), 15)	Index [S,C]

Error	Note

matching_error(Index, conflicting_type(Var, Type), 10)	Index[A,T,..,C]
	This code will get a list of conflict types for this argument Var.
	`session_getValue(vartypes, VarTypes),`
	`findall(Varname=ConflictType, member (Varname=ConflictType,VarTypes), ConflictTypeList).`

matching_error(Index, must_be_number(Var), 10)	Index [A,T,..,C]
matching_error(Index, no_arithmetic_function (X), 15)	Index [S,C]. Term X is not an arithmetic function.
matching_error (Index,misplaced_argument_type (Pos1,Var1,Pos2,TypeVar2)	Index [A,T,..,C]. This is a type error. It is assumed that argument Var1 of position Pos1 should have the type TypeVar2 of argument which has position Pos2, and Var2 should be on position Pos1. Please read StructureMatcher_Diagnosespec for more example. For this error: Explain: Type of this predicate seems to be incorrect. Help: On position Pos2 argument should have type TypeVar2. Please note: TypeVar2 can have following values: number, atom, list(number), list(atom). **More ???**
matching_error(Index, superfluous_argument_type(Pos,Arg)	Index [Pos,T,..,C]. This is a type error. Arg is superfluous in an predicate. Explain: type of this predicate seems to be incorrect. Help: Arg seems to be superfluous.
idiom_error([[Index1], [Index2]], Term,Explain, Penalty)	This case happens when we use constraint checking arg_value, and_value, or_value, before. Index1 shows the position of the first concerning element, Index2 the second clause/subgoal/argument. Term=[ErrorType,StudentVar1,StudentVar2] ErrorType is the name of wrong technique which is defined in XML DB. This error concerns two arguments of student solution: StudentVar1, StudentVar2 which have positions Index1, Index2.

Error	Note
idiom_error([Index1], Term,Explain, Penalty)	This case happens when we use op_value(Op,value). Term=[ErrorType,StudentVar1]. Wrong technique produces ErrorType and concerns StudentVar1.

We have following error types:

- error(syntax_error(Type), string(Clause,Position))

For example:
error(syntax_error(operator_expected), string("deleteall([H|T],A,Result):-H=>A,deleteall(T,A,Result) . ", 28))

- matching_error(Index, ErrorType, Penalty)

- idiom_error(Index, Term ,Explain, Penalty)

Reference document:

- StructureMatcher_DiagnoseSpec.doc
- ConstraintAnalyser.doc