MODULE *WorkflowRepairPretty*

EXTENDS *WorkflowRepair*

\$*W*\$ ... the sequence of workflow tasks to repair

$W \triangleq Workflow$ *e.g.* \$*W* = ⟨"*EVI*", "*IVI*"⟩\$

\$*ValidWorkflows*\$ ... the set of all valid workflows that act as candidate recommendations

$ValidWorkflows$ *e.g.* \$*ValidWorkflows* = \{⟨⟩, ⟨"*EVI*"⟩, ... \}\$

\$*W*_best\$ ... the best alternative sequence of workflow tasks

W_best *e.g.* \$*W*_best = ⟨"*EVI*", "*XRAY*", "*IVI*"⟩\$

\$*W*_worse\$ ... any other worse alternative sequence of workflow tasks

W_worse *e.g.* \$*W*_worse = ⟨"*EVI*"⟩\$

$$
\begin{array}{l}
\forall\, W\_worse \in ValidWorkflows : W\_best \neq W\_worse \Rightarrow \\
\quad \text{1. minimize deletion of tasks while going from old to new wf} \\
\quad \lor\ missingTaskTypes(W\_worse,\ W) > missingTaskTypes(W\_best,\ W) \\
\quad \lor \\
\qquad \land\ missingTaskTypes(W\_worse,\ W) = missingTaskTypes(W\_best,\ W) \\
\qquad \land \\
\qquad\quad \text{2. minimize addition of tasks while going from old to new wf} \\
\qquad\quad \lor\ additionalTaskTypes(W\_worse,\ W) > additionalTaskTypes(W\_best,\ W) \\
\qquad\quad \lor \\
\qquad\qquad \land\ additionalTaskTypes(W\_worse,\ W) = additionalTaskTypes(W\_best,\ W) \\
\qquad\qquad \land \\
\qquad\qquad\quad \text{3. minimize reduction of task repetitions while going from old to new wf} \\
\qquad\qquad\quad \lor\ missingTaskAmount(W\_worse,\ W) > missingTaskAmount(W\_best,\ W) \\
\qquad\qquad\quad \lor \\
\qquad\qquad\qquad \land\ missingTaskAmount(W\_worse,\ W) = missingTaskAmount(W\_best,\ W) \\
\qquad\qquad\qquad \land \\
\qquad\qquad\qquad\quad \text{4. minimize increase of task repetitions while going from old to new wf} \\
\qquad\qquad\qquad\quad \lor\ additionalTaskAmount(W\_worse,\ W) > additionalTaskAmount(W\_best,\ W) \\
\qquad\qquad\qquad\quad \lor \\
\qquad\qquad\qquad\qquad \land\ additionalTaskAmount(W\_worse,\ W) = additionalTaskAmount(W\_best,\ W) \\
\qquad\qquad\qquad\qquad \land \\
\qquad\qquad\qquad\qquad\quad \text{5. minimize ordering difference of matching tasks in old and new wf} \\
\qquad\qquad\qquad\qquad\quad \lor\ diffTaskOrder(W\_worse,\ W) > diffTaskOrder(W\_best,\ W) \\
\qquad\qquad\qquad\qquad\quad \lor \\
\qquad\qquad\qquad\qquad\qquad \land\ diffTaskOrder(W\_worse,\ W) = diffTaskOrder(W\_best,\ W) \\
\qquad\qquad\qquad\qquad\qquad \land \\
\qquad\qquad\qquad\qquad\qquad\quad \text{6. minimize length difference between old and new wf} \\
\qquad\qquad\qquad\qquad\qquad\quad \lor\ diffWorkflowLength(W\_worse,\ W) > diffWorkflowLength(W\_best,\ W) \\
\qquad\qquad\qquad\qquad\qquad\quad \lor\ diffWorkflowLength(W\_worse,\ W) = diffWorkflowLength(W\_best,\ W)
\end{array}
$$