MODULE *WorkflowValidation*

```
EXTENDS $WorkflowDefinition$
LOCAL INSTANCE $Utilities$
```

THEN an error is returned for unknown tasks

```
$ErrorUnknownTasks \triangleq$
    LET
        $KnownTaskConstraint(t) \triangleq$
            $\exists\, task \in Tasks : task.name = t$
    IN
        $\{t \in RAN(Workflow) : \neg KnownTaskConstraint(t)\}$
```

such that it adheres to the expected structure

```
ASSUME $IsFiniteSet(ErrorUnknownTasks)$
ASSUME $\forall\, t \in ErrorUnknownTasks : t \in$ STRING
```

THEN an error is returned for non-repeatable tasks being repeated

```
$ErrorNonRepeatableTasks \triangleq$
    LET
        $RepeatabilityConstraint(t) \triangleq$
            $\land\ Contains(Workflow,\ t)$
            $\land\ \neg Task(t).repeatable$
            $\Rightarrow Count(Workflow,\ t) \leq 1$
    IN
        $\{t \in TaskNames : \neg RepeatabilityConstraint(t)\}$
```

such that it adheres to the expected structure

```
ASSUME $IsFiniteSet(ErrorNonRepeatableTasks)$
ASSUME $\forall\, t \in ErrorNonRepeatableTasks : t \in$ STRING
```

THEN an error is returned for destructive tasks coming before non-destructive ones

```
$ErrorDestructiveBeforeNonDestructive \triangleq$
    LET
        $DestructiveOrderConstraint(d) \triangleq$
            $\forall\, n \in TaskNames :$
                $\land\ d \neq n$
                $\land\ Contains(Workflow,\ d)$
                $\land\ Contains(Workflow,\ n)$
                $\land\ Task(d).group =$ "destructive"
                $\land\ Task(n).group =$ "non-destructive"
                $\Rightarrow FirstIndex(Workflow,\ d) > LastIndex(Workflow,\ n)$
    IN
        $\{t \in TaskNames : \neg DestructiveOrderConstraint(t)\}$
```

such that it adheres to the expected structure

```
ASSUME $\mathit{IsFiniteSet}(\mathit{ErrorDestructiveBeforeNonDestructive})$
ASSUME $\forall t \in \mathit{ErrorDestructiveBeforeNonDestructive} : t \in \text{STRING}$
```

THEN an error is returned for partial-order violations

```
$\mathit{ErrorPartialOrderViolations} \triangleq$
    LET
        $\mathit{PartialOrderConstraint}(s, d) \triangleq$
            $\land\ s \neq d \land \mathit{TransConRel}[s, d]$
            $\land\ \mathit{Contains}(\mathit{Workflow}, s) \land \mathit{Contains}(\mathit{Workflow}, d)$
            $\land\ \neg \mathit{Task}(s).\mathit{repeatable} \lor \neg \mathit{Task}(d).\mathit{repeatable}$
            $\Rightarrow \mathit{LastIndex}(\mathit{Workflow}, s) < \mathit{FirstIndex}(\mathit{Workflow}, d)$
    IN
        UNION $\{\mathit{ErrorConn}(s, d, \mathit{PartialOrderConstraint}) : s, d \in \mathit{TaskNames}\}$
```

such that it adheres to the expected structure

```
ASSUME $\mathit{IsFiniteSet}(\mathit{ErrorPartialOrderViolations})$
ASSUME $\forall \mathit{conn} \in \mathit{ErrorPartialOrderViolations} :$
            $\forall \mathit{id} \in \mathit{DOM}(\mathit{conn}) : \mathit{id} \in \{$“name”, “srcName”, “dstName”$\}$
ASSUME $\forall \mathit{conn} \in \mathit{ErrorPartialOrderViolations} : \mathit{conn}.\mathit{name} \in$
    {   “has_successor”
    ,   “has_predecessor”
    ,   “has_mandatory_predecessor”
    ,   “has_mandatory_successor”
    }
ASSUME $\forall \mathit{conn} \in \mathit{ErrorPartialOrderViolations} : \mathit{conn}.\mathit{srcName} \in \text{STRING}$
ASSUME $\forall \mathit{conn} \in \mathit{ErrorPartialOrderViolations} : \mathit{conn}.\mathit{dstName} \in \text{STRING}$
```

THEN an error is returned for missing mandatory dependency tasks

```
$\mathit{ErrorMissingMandatoryDependencies} \triangleq$
    LET
        $\mathit{MandatoryDependencyConstraint}(s, d) \triangleq$
        $\land$
            $\land\ s \neq d \land \mathit{TransConRel}[s, d]$
            $\land\ \mathit{RequiresRel}[s, d] \land \mathit{Contains}(\mathit{Workflow}, s)$
            $\Rightarrow$   $\land\ \mathit{Contains}(\mathit{Workflow}, d)$
                 $\land\ \mathit{LastIndex}(\mathit{Workflow}, s) < \mathit{LastIndex}(\mathit{Workflow}, d)$
        $\land$
            $\land\ s \neq d \land \mathit{TransConRel}[s, d]$
            $\land\ \mathit{RequiresRel}[d, s] \land \mathit{Contains}(\mathit{Workflow}, d)$
            $\Rightarrow$   $\land\ \mathit{Contains}(\mathit{Workflow}, s)$
                 $\land\ \mathit{FirstIndex}(\mathit{Workflow}, s) < \mathit{FirstIndex}(\mathit{Workflow}, d)$
    IN
        UNION $\{\mathit{ErrorConns}(s, d, \mathit{MandatoryDependencyConstraint}) : s, d \in \mathit{TaskNames}\}$
```

such that it adheres to the expected structure

```
$\textsc{assume}\ \mathit{IsFiniteSet}(\mathit{ErrorMissingMandatoryDependencies})$
$\textsc{assume}\ \forall\, \mathit{conn} \in \mathit{ErrorMissingMandatoryDependencies} :$
        $\forall\, \mathit{id} \in \mathit{DOM}(\mathit{conn}) : \mathit{id} \in \{\text{“name”}, \text{“srcName”}, \text{“dstName”}\}$
$\textsc{assume}\ \forall\, \mathit{conn} \in \mathit{ErrorMissingMandatoryDependencies} : \mathit{conn.name} \in$
    $\{$  “has_successor”
    ,   “has_predecessor”
    ,   “has_mandatory_predecessor”
    ,   “has_mandatory_successor”
    $\}$
$\textsc{assume}\ \forall\, \mathit{conn} \in \mathit{ErrorMissingMandatoryDependencies} : \mathit{conn.srcName} \in \textsc{string}$
$\textsc{assume}\ \forall\, \mathit{conn} \in \mathit{ErrorMissingMandatoryDependencies} : \mathit{conn.dstName} \in \textsc{string}$
```

THEN an error is returned for missing mandatory dependency repetitions

```
$\mathit{ErrorMissingMandatoryDependencyRepetitions} \triangleq$
    $\textsc{let}$
        $\mathit{MandatoryRepetitionConstraint}(s, d) \triangleq$
        $\land$
            $\land\ s \neq d \land \mathit{TransConRel}[s, d] \land \mathit{RequiresRel}[s, d]$
            $\land\ \mathit{Contains}(\mathit{Workflow}, s) \land \mathit{Contains}(\mathit{Workflow}, d)$
            $\Rightarrow\ \forall\, i, j \in \mathit{Indexes}(\mathit{Workflow}, s) :$
                $i < j \Rightarrow \exists\, k \in \mathit{Indexes}(\mathit{Workflow}, d) : i < k \land k < j$
        $\land$
            $\land\ s \neq d \land \mathit{TransConRel}[s, d] \land \mathit{RequiresRel}[d, s]$
            $\land\ \mathit{Contains}(\mathit{Workflow}, s) \land \mathit{Contains}(\mathit{Workflow}, d)$
            $\Rightarrow\ \forall\, i, j \in \mathit{Indexes}(\mathit{Workflow}, d) :$
                $i < j \Rightarrow \exists\, k \in \mathit{Indexes}(\mathit{Workflow}, s) : i < k \land k < j$
    $\textsc{in}$
        $\textsc{union}\ \{\mathit{ErrorConns}(s, d, \mathit{MandatoryRepetitionConstraint}) : s, d \in \mathit{TaskNames}\}$
```

such that it adheres to the expected structure

```
$\textsc{assume}\ \mathit{IsFiniteSet}(\mathit{ErrorMissingMandatoryDependencyRepetitions})$
$\textsc{assume}\ \forall\, \mathit{conn} \in \mathit{ErrorMissingMandatoryDependencyRepetitions} :$
        $\forall\, \mathit{id} \in \mathit{DOM}(\mathit{conn}) : \mathit{id} \in \{\text{“name”}, \text{“srcName”}, \text{“dstName”}\}$
$\textsc{assume}\ \forall\, \mathit{conn} \in \mathit{ErrorMissingMandatoryDependencyRepetitions} : \mathit{conn.name} \in$
    $\{$  “has_successor”
    ,   “has_predecessor”
    ,   “has_mandatory_predecessor”
    ,   “has_mandatory_successor”
    $\}$
$\textsc{assume}\ \forall\, \mathit{conn} \in \mathit{ErrorMissingMandatoryDependencyRepetitions} : \mathit{conn.srcName} \in \textsc{string}$
$\textsc{assume}\ \forall\, \mathit{conn} \in \mathit{ErrorMissingMandatoryDependencyRepetitions} : \mathit{conn.dstName} \in \textsc{string}$
```

FINALLY a structure of all errors is returned

```
$\mathit{Errors} \triangleq [$
    $\mathit{ErrorUnknownTasks} \mapsto \mathit{ErrorUnknownTasks},$
```

$$
\begin{array}{l}
\quad \mathit{ErrorNonRepeatableTasks} \mapsto \mathit{ErrorNonRepeatableTasks}, \\
\quad \mathit{ErrorDestructiveBeforeNonDestructive} \mapsto \mathit{ErrorDestructiveBeforeNonDestructive}, \\
\quad \mathit{ErrorPartialOrderViolations} \mapsto \mathit{ErrorPartialOrderViolations}, \\
\quad \mathit{ErrorMissingMandatoryDependencies} \mapsto \mathit{ErrorMissingMandatoryDependencies}, \\
\quad \mathit{ErrorMissingMandatoryDependencyRepetitions} \mapsto \mathit{ErrorMissingMandatoryDependencyRepetitions} \\
]
\end{array}
$$

```
e.g. Errors ==
[ ErrorUnknownTasks |-> {"IVI"}
, ErrorNonRepeatableTasks |-> {"EVI"}
, ErrorDestructiveBeforeNonDestructive |-> {"IVI"}
, ErrorPartialOrderViolations |->
  { [ name |-> "has_successor", srcName |-> "EVI", dstName |-> "IVI" ] }
, ErrorMissingMandatoryDependencies |->
{ [ name |-> "has_mandatory_predecessor", srcName |-> "IVI", dstName |->
"EVI" ] }
, ErrorMissingMandatoryDependencyRepetitions |->
{ [ name |-> "has_mandatory_predecessor", srcName |-> "IVI", dstName |->
"EVI" ] }
]
```

WHILE the structure containing no errors matches

$$
\begin{array}{l}
\mathit{NoErrors} \triangleq [\\
\quad \mathit{ErrorUnknownTasks} \mapsto \{\}, \\
\quad \mathit{ErrorNonRepeatableTasks} \mapsto \{\}, \\
\quad \mathit{ErrorDestructiveBeforeNonDestructive} \mapsto \{\}, \\
\quad \mathit{ErrorPartialOrderViolations} \mapsto \{\}, \\
\quad \mathit{ErrorMissingMandatoryDependencies} \mapsto \{\}, \\
\quad \mathit{ErrorMissingMandatoryDependencyRepetitions} \mapsto \{\} \\
]
\end{array}
$$