

```

MODULE WorkflowValidationTest
LOCAL INSTANCE TLC

ValidationTests  $\triangleq$ 

should report no errors for trivial valid workflow

 $\langle$ LET Validation  $\triangleq$  INSTANCE WorkflowValidation WITH
  Connections  $\leftarrow$ 
  {
  },
  Tasks  $\leftarrow$ 
  { [name  $\mapsto$  "EVI", repeatable  $\mapsto$  FALSE, group  $\mapsto$  "both"]
  },
  Workflow  $\leftarrow$ 
   $\langle$ "EVI" $\rangle$ 

  ACTUAL  $\triangleq$  Validation!Errors

  EXPECTED  $\triangleq$  Validation!NoErrors
IN
  Assert(
    Print(ACTUAL, ACTUAL = EXPECTED),
    Print(EXPECTED, "should report no errors for trivial valid workflow")
  )

should not report errors for complex, valid workflow

, LET Validation  $\triangleq$  INSTANCE WorkflowValidation WITH
  Connections  $\leftarrow$ 
  { [name  $\mapsto$  "has_successor", srcName  $\mapsto$  "EVI", dstName  $\mapsto$  "IVI"]
  , [name  $\mapsto$  "has_mandatory_successor", srcName  $\mapsto$  "EVI", dstName  $\mapsto$  "IVI"]
  , [name  $\mapsto$  "has_successor", srcName  $\mapsto$  "IVI", dstName  $\mapsto$  "DES"]
  },
  Tasks  $\leftarrow$ 
  { [name  $\mapsto$  "EVI", repeatable  $\mapsto$  FALSE, group  $\mapsto$  "non-destructive"]
  , [name  $\mapsto$  "IVI", repeatable  $\mapsto$  TRUE, group  $\mapsto$  "destructive"]
  , [name  $\mapsto$  "DES", repeatable  $\mapsto$  TRUE, group  $\mapsto$  "destructive"]
  },
  Workflow  $\leftarrow$ 
   $\langle$ "EVI", "IVI", "DES", "IVI", "DES" $\rangle$ 

  ACTUAL  $\triangleq$  Validation!Errors

  EXPECTED  $\triangleq$  Validation!NoErrors
IN
  Assert(

```

```

    Print(ACTUAL, ACTUAL = EXPECTED),
    Print(EXPECTED, "should not report errors for complex, valid workflow")
)

```

should report errors for unknown task

```

, LET Validation  $\triangleq$  INSTANCE WorkflowValidation WITH
    Connections  $\leftarrow$ 
    {
    },
    Tasks  $\leftarrow$ 
    { [name  $\mapsto$  "EVI", repeatable  $\mapsto$  FALSE, group  $\mapsto$  "both"]
    },
    Workflow  $\leftarrow$ 
    <"IVI">

    ACTUAL  $\triangleq$  Validation!Errors

    EXPECTED  $\triangleq$  [Validation!NoErrors EXCEPT
        !.ErrorUnknownTasks = {"IVI"}
    ]
IN
    Assert(
        Print(ACTUAL, ACTUAL = EXPECTED),
        Print(EXPECTED, "should report errors for unknown task")
    )

```

should report errors for repeating unrepeatable task

```

, LET Validation  $\triangleq$  INSTANCE WorkflowValidation WITH
    Connections  $\leftarrow$ 
    {
    },
    Tasks  $\leftarrow$ 
    { [name  $\mapsto$  "EVI", repeatable  $\mapsto$  FALSE, group  $\mapsto$  "both"]
    },
    Workflow  $\leftarrow$ 
    <"EVI", "EVI">

    ACTUAL  $\triangleq$  Validation!Errors

    EXPECTED  $\triangleq$  [Validation!NoErrors EXCEPT
        !.ErrorNonRepeatableTasks = {"EVI"}
    ]
IN
    Assert(
        Print(ACTUAL, ACTUAL = EXPECTED),
    )

```

```

    Print(EXPECTED, "should report errors for repeating unrepeatable task")
)

```

should report no errors for mixing tasks that can be both destructive and non-destructive

```

, LET Validation  $\triangleq$  INSTANCE WorkflowValidation WITH
    Connections  $\leftarrow$ 
    {
    },
    Tasks  $\leftarrow$ 
    { [name  $\mapsto$  "EVI", repeatable  $\mapsto$  FALSE, group  $\mapsto$  "both"]
    , [name  $\mapsto$  "IVI", repeatable  $\mapsto$  FALSE, group  $\mapsto$  "both"]
    },
    Workflow  $\leftarrow$ 
    <"EVI", "IVI">

    ACTUAL  $\triangleq$  Validation!Errors

    EXPECTED  $\triangleq$  Validation!NoErrors

IN
    Assert(
        Print(ACTUAL, ACTUAL = EXPECTED),
        Print(EXPECTED, "should report no errors for mixing tasks that can be both destructive and non-destructive")
    )

```

should report no errors for doing non-destructive task before destructive one

```

, LET Validation  $\triangleq$  INSTANCE WorkflowValidation WITH
    Connections  $\leftarrow$ 
    {
    },
    Tasks  $\leftarrow$ 
    { [name  $\mapsto$  "EVI", repeatable  $\mapsto$  FALSE, group  $\mapsto$  "non-destructive"]
    , [name  $\mapsto$  "IVI", repeatable  $\mapsto$  FALSE, group  $\mapsto$  "destructive"]
    },
    Workflow  $\leftarrow$ 
    <"EVI", "IVI">

    ACTUAL  $\triangleq$  Validation!Errors

    EXPECTED  $\triangleq$  Validation!NoErrors

IN
    Assert(
        Print(ACTUAL, ACTUAL = EXPECTED),
        Print(EXPECTED, "should report no errors for doing non-destructive task before destructive one")
    )

```

should report errors for doing destructive task before non-destructive one

```
, LET Validation  $\triangleq$  INSTANCE WorkflowValidation WITH
  Connections  $\leftarrow$ 
  {
  },
  Tasks  $\leftarrow$ 
  { [name  $\mapsto$  "EVI", repeatable  $\mapsto$  FALSE, group  $\mapsto$  "non-destructive"]
  , [name  $\mapsto$  "IVI", repeatable  $\mapsto$  FALSE, group  $\mapsto$  "destructive"]
  },
  Workflow  $\leftarrow$ 
  <"IVI", "EVI">

  ACTUAL  $\triangleq$  Validation!Errors

  EXPECTED  $\triangleq$  [Validation!NoErrors EXCEPT
    !.ErrorDestructiveBeforeNonDestructive = { "IVI" }
  ]

IN
  Assert(
    Print(ACTUAL, ACTUAL = EXPECTED),
    Print(EXPECTED, "should report errors for doing destructive task before non-destructive one")
  )
```

should report no errors for containing a mandatory successor task

```
, LET Validation  $\triangleq$  INSTANCE WorkflowValidation WITH
  Connections  $\leftarrow$ 
  { [name  $\mapsto$  "has_mandatory_successor", srcName  $\mapsto$  "EVI", dstName  $\mapsto$  "IVI"]
  },
  Tasks  $\leftarrow$ 
  { [name  $\mapsto$  "EVI", repeatable  $\mapsto$  FALSE, group  $\mapsto$  "both"]
  , [name  $\mapsto$  "IVI", repeatable  $\mapsto$  FALSE, group  $\mapsto$  "both"]
  },
  Workflow  $\leftarrow$ 
  <"EVI", "IVI">

  ACTUAL  $\triangleq$  Validation!Errors

  EXPECTED  $\triangleq$  Validation!NoErrors

IN
  Assert(
    Print(ACTUAL, ACTUAL = EXPECTED),
    Print(EXPECTED, "should report no errors for containing a mandatory successor task")
  )
```

should report errors for missing mandatory successor task

```

, LET Validation  $\triangleq$  INSTANCE WorkflowValidation WITH
  Connections  $\leftarrow$ 
  { [name  $\mapsto$  "has_mandatory_successor", srcName  $\mapsto$  "EVI", dstName  $\mapsto$  "IVI"]
  },
  Tasks  $\leftarrow$ 
  { [name  $\mapsto$  "EVI", repeatable  $\mapsto$  FALSE, group  $\mapsto$  "both"]
  , [name  $\mapsto$  "IVI", repeatable  $\mapsto$  FALSE, group  $\mapsto$  "both"]
  },
  Workflow  $\leftarrow$ 
  <"EVI">

ACTUAL  $\triangleq$  Validation!Errors

EXPECTED  $\triangleq$  [Validation!NoErrors EXCEPT
  !..ErrorMissingMandatoryDependencies = {
    [name  $\mapsto$  "has_mandatory_successor", srcName  $\mapsto$  "EVI", dstName  $\mapsto$  "IVI"]
  }
]
IN
  Assert(
    Print(ACTUAL, ACTUAL = EXPECTED),
    Print(EXPECTED, "should report errors for missing mandatory successor task")
  )

```

should report no errors for containing a mandatory predecessor task

```

, LET Validation  $\triangleq$  INSTANCE WorkflowValidation WITH
  Connections  $\leftarrow$ 
  { [name  $\mapsto$  "has_mandatory_predecessor", srcName  $\mapsto$  "IVI", dstName  $\mapsto$  "EVI"]
  },
  Tasks  $\leftarrow$ 
  { [name  $\mapsto$  "EVI", repeatable  $\mapsto$  FALSE, group  $\mapsto$  "both"]
  , [name  $\mapsto$  "IVI", repeatable  $\mapsto$  FALSE, group  $\mapsto$  "both"]
  },
  Workflow  $\leftarrow$ 
  <"EVI", "IVI">

ACTUAL  $\triangleq$  Validation!Errors

EXPECTED  $\triangleq$  Validation!NoErrors
IN
  Assert(
    Print(ACTUAL, ACTUAL = EXPECTED),
    Print(EXPECTED, "should report no errors for containing a mandatory predecessor task")
  )

```

should report errors for missing mandatory predecessor task

```

, LET Validation  $\triangleq$  INSTANCE WorkflowValidation WITH
  Connections  $\leftarrow$ 
  { [name  $\mapsto$  "has_mandatory_predecessor", srcName  $\mapsto$  "IVI", dstName  $\mapsto$  "EVI"]
  },
  Tasks  $\leftarrow$ 
  { [name  $\mapsto$  "EVI", repeatable  $\mapsto$  FALSE, group  $\mapsto$  "both"]
  , [name  $\mapsto$  "IVI", repeatable  $\mapsto$  FALSE, group  $\mapsto$  "both"]
  },
  Workflow  $\leftarrow$ 
  <"IVI">

ACTUAL  $\triangleq$  Validation!Errors

EXPECTED  $\triangleq$  [Validation!NoErrors EXCEPT
  !.ErrorMissingMandatoryDependencies = {
    [name  $\mapsto$  "has_mandatory_predecessor", srcName  $\mapsto$  "IVI", dstName  $\mapsto$  "EVI"]
  }
]
IN
  Assert(
    Print(ACTUAL, ACTUAL = EXPECTED),
    Print(EXPECTED, "should report errors for missing mandatory predecessor task")
  )

```

should report no errors for respecting partial order between non-repeatable tasks

```

, LET Validation  $\triangleq$  INSTANCE WorkflowValidation WITH
  Connections  $\leftarrow$ 
  { [name  $\mapsto$  "has_successor", srcName  $\mapsto$  "EVI", dstName  $\mapsto$  "IVI"]
  },
  Tasks  $\leftarrow$ 
  { [name  $\mapsto$  "EVI", repeatable  $\mapsto$  FALSE, group  $\mapsto$  "both"]
  , [name  $\mapsto$  "IVI", repeatable  $\mapsto$  FALSE, group  $\mapsto$  "both"]
  },
  Workflow  $\leftarrow$ 
  <"EVI", "IVI">

ACTUAL  $\triangleq$  Validation!Errors

EXPECTED  $\triangleq$  Validation!NoErrors
IN
  Assert(
    Print(ACTUAL, ACTUAL = EXPECTED),
    Print(EXPECTED, "should report no errors for respecting partial order between non-repeatable tasks")
  )

```

should report errors for violating partial order between non-repeatable tasks

```

, LET Validation  $\triangleq$  INSTANCE WorkflowValidation WITH
  Connections  $\leftarrow$ 
  { [name  $\mapsto$  "has_successor", srcName  $\mapsto$  "EVI", dstName  $\mapsto$  "IVI"]
  },
  Tasks  $\leftarrow$ 
  { [name  $\mapsto$  "EVI", repeatable  $\mapsto$  FALSE, group  $\mapsto$  "both"]
  , [name  $\mapsto$  "IVI", repeatable  $\mapsto$  FALSE, group  $\mapsto$  "both"]
  },
  Workflow  $\leftarrow$ 
  <"IVI", "EVI">

  ACTUAL  $\triangleq$  Validation!Errors

  EXPECTED  $\triangleq$  [Validation!NoErrors EXCEPT
    !.ErrorPartialOrderViolations = {
      [name  $\mapsto$  "has_successor", srcName  $\mapsto$  "EVI", dstName  $\mapsto$  "IVI"]
    }
  ]
IN
  Assert(
    Print(ACTUAL, ACTUAL = EXPECTED),
    Print(EXPECTED, "should report errors for violating partial order between tasks")
  )

```

should automatically infer partial order from other connection types - has\\_predecessor

```

, LET Validation  $\triangleq$  INSTANCE WorkflowValidation WITH
  Connections  $\leftarrow$ 
  { [name  $\mapsto$  "has_predecessor", srcName  $\mapsto$  "IVI", dstName  $\mapsto$  "EVI"]
  },
  Tasks  $\leftarrow$ 
  { [name  $\mapsto$  "EVI", repeatable  $\mapsto$  FALSE, group  $\mapsto$  "both"]
  , [name  $\mapsto$  "IVI", repeatable  $\mapsto$  FALSE, group  $\mapsto$  "both"]
  },
  Workflow  $\leftarrow$ 
  <"IVI", "EVI">

  ACTUAL  $\triangleq$  Validation!Errors

  EXPECTED  $\triangleq$  [Validation!NoErrors EXCEPT
    !.ErrorPartialOrderViolations = {
      [name  $\mapsto$  "has_successor", srcName  $\mapsto$  "EVI", dstName  $\mapsto$  "IVI"]
    }
  ]
IN
  Assert(

```

```

    Print(ACTUAL, ACTUAL = EXPECTED),
    Print(EXPECTED, "should automatically infer partial order from other connection types - has_predecessor")
)

should automatically infer partial order from other connection types - has\_mandatory\_predecessor

, LET Validation  $\triangleq$  INSTANCE WorkflowValidation WITH
    Connections  $\leftarrow$ 
    { [name  $\mapsto$  "has_mandatory_predecessor", srcName  $\mapsto$  "IVI", dstName  $\mapsto$  "EVI"]
    },
    Tasks  $\leftarrow$ 
    { [name  $\mapsto$  "EVI", repeatable  $\mapsto$  FALSE, group  $\mapsto$  "both"]
    , [name  $\mapsto$  "IVI", repeatable  $\mapsto$  FALSE, group  $\mapsto$  "both"]
    },
    Workflow  $\leftarrow$ 
    <"IVI", "EVI">

    ACTUAL  $\triangleq$  Validation!Errors

    EXPECTED  $\triangleq$  [Validation!NoErrors EXCEPT
        !.ErrorPartialOrderViolations = {
            [name  $\mapsto$  "has_successor", srcName  $\mapsto$  "EVI", dstName  $\mapsto$  "IVI"]
        }
    ]
IN
    Assert(
        Print(ACTUAL, ACTUAL.ErrorPartialOrderViolations = EXPECTED.ErrorPartialOrderViolations)
        Print(EXPECTED, "should automatically infer partial order from other connection types - has_mandatory_predecessor")
    )

```

```

should automatically infer partial order from other connection types - has\_mandatory\_successor

, LET Validation  $\triangleq$  INSTANCE WorkflowValidation WITH
    Connections  $\leftarrow$ 
    { [name  $\mapsto$  "has_mandatory_successor", srcName  $\mapsto$  "EVI", dstName  $\mapsto$  "IVI"]
    },
    Tasks  $\leftarrow$ 
    { [name  $\mapsto$  "EVI", repeatable  $\mapsto$  FALSE, group  $\mapsto$  "both"]
    , [name  $\mapsto$  "IVI", repeatable  $\mapsto$  FALSE, group  $\mapsto$  "both"]
    },
    Workflow  $\leftarrow$ 
    <"IVI", "EVI">

    ACTUAL  $\triangleq$  Validation!Errors

    EXPECTED  $\triangleq$  [Validation!NoErrors EXCEPT

```



```

        !.ErrorPartialOrderViolations = {
            [name ↦ "has_successor", srcName ↦ "EVI", dstName ↦ "IVI"]
        }
    ]
IN
    Assert(
        Print(ACTUAL, ACTUAL.ErrorPartialOrderViolations = EXPECTED.ErrorPartialOrderViolations),
        Print(EXPECTED, "should automatically infer partial order from other connection types - has_mandator")
    )

```

should report no errors for respecting partial order between non-repeatable and repeatable task

```

, LET Validation  $\triangleq$  INSTANCE WorkflowValidation WITH
    Connections  $\leftarrow$ 
    { [name ↦ "has_successor", srcName ↦ "EVI", dstName ↦ "IVI"]
    },
    Tasks  $\leftarrow$ 
    { [name ↦ "EVI", repeatable ↦ FALSE, group ↦ "both"]
    , [name ↦ "IVI", repeatable ↦ TRUE, group ↦ "both"]
    },
    Workflow  $\leftarrow$ 
    <"EVI", "IVI">

    ACTUAL  $\triangleq$  Validation!Errors

    EXPECTED  $\triangleq$  Validation!NoErrors
IN
    Assert(
        Print(ACTUAL, ACTUAL = EXPECTED),
        Print(EXPECTED, "should report no errors for respecting partial order between non-repeatable and repeatable task")
    )

```

should report errors for violating partial order between non-repeatable and repeatable task

```

, LET Validation  $\triangleq$  INSTANCE WorkflowValidation WITH
    Connections  $\leftarrow$ 
    { [name ↦ "has_successor", srcName ↦ "EVI", dstName ↦ "IVI"]
    },
    Tasks  $\leftarrow$ 
    { [name ↦ "EVI", repeatable ↦ FALSE, group ↦ "both"]
    , [name ↦ "IVI", repeatable ↦ TRUE, group ↦ "both"]
    },
    Workflow  $\leftarrow$ 
    <"IVI", "EVI">

    ACTUAL  $\triangleq$  Validation!Errors

```

```

    EXPECTED  $\triangleq$  [Validation!NoErrors EXCEPT
      !.ErrorPartialOrderViolations = {
        [name  $\mapsto$  "has_successor", srcName  $\mapsto$  "EVI", dstName  $\mapsto$  "IVI"]
      }
    ]
  IN
    Assert(
      Print(ACTUAL, ACTUAL = EXPECTED),
      Print(EXPECTED, "should report errors for violating partial order between non-repeatable and repeatable")
    )

```

should report no errors for respecting partial order between repeatable and non-repeatable task

```

, LET Validation  $\triangleq$  INSTANCE WorkflowValidation WITH
  Connections  $\leftarrow$ 
  { [name  $\mapsto$  "has_successor", srcName  $\mapsto$  "EVI", dstName  $\mapsto$  "IVI"]
  },
  Tasks  $\leftarrow$ 
  { [name  $\mapsto$  "EVI", repeatable  $\mapsto$  TRUE, group  $\mapsto$  "both"]
  , [name  $\mapsto$  "IVI", repeatable  $\mapsto$  FALSE, group  $\mapsto$  "both"]
  },
  Workflow  $\leftarrow$ 
  <"EVI", "IVI">

  ACTUAL  $\triangleq$  Validation!Errors

  EXPECTED  $\triangleq$  Validation!NoErrors
IN
  Assert(
    Print(ACTUAL, ACTUAL = EXPECTED),
    Print(EXPECTED, "should report no errors for respecting partial order between repeatable and non-repeatable")
  )

```

should report errors for violating partial order between repeatable and non-repeatable task

```

, LET Validation  $\triangleq$  INSTANCE WorkflowValidation WITH
  Connections  $\leftarrow$ 
  { [name  $\mapsto$  "has_successor", srcName  $\mapsto$  "EVI", dstName  $\mapsto$  "IVI"]
  },
  Tasks  $\leftarrow$ 
  { [name  $\mapsto$  "EVI", repeatable  $\mapsto$  TRUE, group  $\mapsto$  "both"]
  , [name  $\mapsto$  "IVI", repeatable  $\mapsto$  FALSE, group  $\mapsto$  "both"]
  },
  Workflow  $\leftarrow$ 
  <"IVI", "EVI">

```

```

ACTUAL  $\triangleq$  Validation!Errors

EXPECTED  $\triangleq$  [Validation!NoErrors EXCEPT
  !.ErrorPartialOrderViolations = {
    [name  $\mapsto$  "has_successor", srcName  $\mapsto$  "EVI", dstName  $\mapsto$  "IVI"]
  }
]
IN
  Assert(
    Print(ACTUAL, ACTUAL = EXPECTED),
    Print(EXPECTED, "should report errors for violating partial order between non-repeatable and repeatable")
  )

```

should report no errors for repeating non-mandatory tasks in any order

```

, LET Validation  $\triangleq$  INSTANCE WorkflowValidation WITH
  Connections  $\leftarrow$ 
  { [name  $\mapsto$  "has_successor", srcName  $\mapsto$  "EVI", dstName  $\mapsto$  "IVI"]
  },
  Tasks  $\leftarrow$ 
  { [name  $\mapsto$  "EVI", repeatable  $\mapsto$  TRUE, group  $\mapsto$  "both"]
  , [name  $\mapsto$  "IVI", repeatable  $\mapsto$  TRUE, group  $\mapsto$  "both"]
  },
  Workflow  $\leftarrow$ 
  <"EVI", "IVI", "EVI", "EVI", "IVI", "IVI", "IVI">

  ACTUAL  $\triangleq$  Validation!Errors

  EXPECTED  $\triangleq$  Validation!NoErrors

IN
  Assert(
    Print(ACTUAL, ACTUAL = EXPECTED),
    Print(EXPECTED, "should report no errors for repeating non-mandatory tasks in any order")
  )

```

should report no errors for repeating a task and its mandatory successor properly

```

, LET Validation  $\triangleq$  INSTANCE WorkflowValidation WITH
  Connections  $\leftarrow$ 
  { [name  $\mapsto$  "has_mandatory_successor", srcName  $\mapsto$  "EVI", dstName  $\mapsto$  "IVI"]
  },
  Tasks  $\leftarrow$ 
  { [name  $\mapsto$  "EVI", repeatable  $\mapsto$  TRUE, group  $\mapsto$  "both"]
  , [name  $\mapsto$  "IVI", repeatable  $\mapsto$  TRUE, group  $\mapsto$  "both"]
  },
  Workflow  $\leftarrow$ 

```

```

    <"EVI", "IVI", "EVI", "IVI">
    ACTUAL  $\triangleq$  Validation!Errors
    EXPECTED  $\triangleq$  Validation!NoErrors
IN
    Assert(
        Print(ACTUAL, ACTUAL = EXPECTED),
        Print(EXPECTED, "should report no errors for repeating a task and its mandatory successor properly")
    )

```

should report errors for repeating a task more often than its mandatory successor at the end

```

, LET Validation  $\triangleq$  INSTANCE WorkflowValidation WITH
    Connections  $\leftarrow$ 
    { [name  $\mapsto$  "has_mandatory_successor", srcName  $\mapsto$  "EVI", dstName  $\mapsto$  "IVI"]
    },
    Tasks  $\leftarrow$ 
    { [name  $\mapsto$  "EVI", repeatable  $\mapsto$  TRUE, group  $\mapsto$  "both"]
    , [name  $\mapsto$  "IVI", repeatable  $\mapsto$  TRUE, group  $\mapsto$  "both"]
    },
    Workflow  $\leftarrow$ 
    <"EVI", "IVI", "EVI">
    ACTUAL  $\triangleq$  Validation!Errors
    EXPECTED  $\triangleq$  [Validation!NoErrors EXCEPT
        !.ErrorMissingMandatoryDependencies = {
            [name  $\mapsto$  "has_mandatory_successor", srcName  $\mapsto$  "EVI", dstName  $\mapsto$  "IVI"]
        }
    ]
IN
    Assert(
        Print(ACTUAL, ACTUAL = EXPECTED),
        Print(EXPECTED, "should report errors for repeating a task more often than its mandatory successor at
    )

```

should report errors for repeating a task more often than its mandatory successor in between

```

, LET Validation  $\triangleq$  INSTANCE WorkflowValidation WITH
    Connections  $\leftarrow$ 
    { [name  $\mapsto$  "has_mandatory_successor", srcName  $\mapsto$  "EVI", dstName  $\mapsto$  "IVI"]
    },
    Tasks  $\leftarrow$ 
    { [name  $\mapsto$  "EVI", repeatable  $\mapsto$  TRUE, group  $\mapsto$  "both"]
    , [name  $\mapsto$  "IVI", repeatable  $\mapsto$  TRUE, group  $\mapsto$  "both"]
    },

```

```

Workflow ←
⟨“EVI”, “EVI”, “IVI”⟩

ACTUAL ≜ Validation!Errors

EXPECTED ≜ [Validation!NoErrors EXCEPT
  !.ErrorMissingMandatoryDependencyRepetitions = {
    [name ↦ “has_mandatory_successor”, srcName ↦ “EVI”, dstName ↦ “IVI”]
  }
]
IN
  Assert(
    Print(ACTUAL, ACTUAL = EXPECTED),
    Print(EXPECTED, “should report errors for repeating a task more often than its mandatory successor in
  )

```

should report errors for repeating a task before its mandatory successor

```

, LET Validation ≜ INSTANCE WorkflowValidation WITH
  Connections ←
  { [name ↦ “has_mandatory_successor”, srcName ↦ “EVI”, dstName ↦ “IVI”]
  },
  Tasks ←
  { [name ↦ “EVI”, repeatable ↦ TRUE, group ↦ “both”]
  , [name ↦ “IVI”, repeatable ↦ TRUE, group ↦ “both”]
  },
  Workflow ←
  ⟨“EVI”, “EVI”, “IVI”, “IVI”⟩

ACTUAL ≜ Validation!Errors

EXPECTED ≜ [Validation!NoErrors EXCEPT
  !.ErrorMissingMandatoryDependencyRepetitions = {
    [name ↦ “has_mandatory_successor”, srcName ↦ “EVI”, dstName ↦ “IVI”]
  }
]
IN
  Assert(
    Print(ACTUAL, ACTUAL = EXPECTED),
    Print(EXPECTED, “should report errors for repeating a task before its mandatory successor”)
  )

```

should report no errors for repeating a task and its mandatory predecessor properly

```

, LET Validation ≜ INSTANCE WorkflowValidation WITH
  Connections ←
  { [name ↦ “has_mandatory_predecessor”, srcName ↦ “IVI”, dstName ↦ “EVI”]
  }

```

```

    },
    Tasks ←
    { [name ↦ "EVI", repeatable ↦ TRUE, group ↦ "both"]
    , [name ↦ "IVI", repeatable ↦ TRUE, group ↦ "both"]
    },
    Workflow ←
    ⟨"EVI", "IVI", "EVI", "IVI"⟩

    ACTUAL ≜ Validation!Errors

    EXPECTED ≜ Validation!NoErrors
IN
    Assert(
        Print(ACTUAL, ACTUAL = EXPECTED),
        Print(EXPECTED, "should report no errors for repeating a task and its mandatory predecessor properly")
    )

```

should report errors for repeating a task more often than its mandatory predecessor at the beginning

```

, LET Validation ≜ INSTANCE WorkflowValidation WITH
    Connections ←
    { [name ↦ "has_mandatory_predecessor", srcName ↦ "IVI", dstName ↦ "EVI"]
    },
    Tasks ←
    { [name ↦ "EVI", repeatable ↦ TRUE, group ↦ "both"]
    , [name ↦ "IVI", repeatable ↦ TRUE, group ↦ "both"]
    },
    Workflow ←
    ⟨"IVI", "EVI", "IVI"⟩

    ACTUAL ≜ Validation!Errors

    EXPECTED ≜ [Validation!NoErrors EXCEPT
        !.ErrorMissingMandatoryDependencies = {
            [name ↦ "has_mandatory_predecessor", srcName ↦ "IVI", dstName ↦ "EVI"]
        }
    ]
IN
    Assert(
        Print(ACTUAL, ACTUAL = EXPECTED),
        Print(EXPECTED, "should report errors for repeating a task more often than its mandatory predecessor")
    )

```

should report errors for repeating a task more often than its mandatory predecessor in between

```

, LET Validation ≜ INSTANCE WorkflowValidation WITH

```

```

Connections ←
{ [name ↦ "has_mandatory_predecessor", srcName ↦ "IVI", dstName ↦ "EVI"]
},
Tasks ←
{ [name ↦ "EVI", repeatable ↦ TRUE, group ↦ "both"]
, [name ↦ "IVI", repeatable ↦ TRUE, group ↦ "both"]
},
Workflow ←
⟨"EVI", "IVI", "IVI"⟩

ACTUAL ≜ Validation!Errors

EXPECTED ≜ [Validation!NoErrors EXCEPT
!.ErrorMissingMandatoryDependencyRepetitions = {
[name ↦ "has_mandatory_predecessor", srcName ↦ "IVI", dstName ↦ "EVI"]
}
]
IN
Assert(
Print(ACTUAL, ACTUAL = EXPECTED),
Print(EXPECTED, "should report errors for repeating a task more often than its mandatory predecessor")
)

```

should report errors for repeating a task before its mandatory predecessor

```

, LET Validation ≜ INSTANCE WorkflowValidation WITH
Connections ←
{ [name ↦ "has_mandatory_predecessor", srcName ↦ "IVI", dstName ↦ "EVI"]
},
Tasks ←
{ [name ↦ "EVI", repeatable ↦ TRUE, group ↦ "both"]
, [name ↦ "IVI", repeatable ↦ TRUE, group ↦ "both"]
},
Workflow ←
⟨"EVI", "EVI", "IVI", "IVI"⟩

ACTUAL ≜ Validation!Errors

EXPECTED ≜ [Validation!NoErrors EXCEPT
!.ErrorMissingMandatoryDependencyRepetitions = {
[name ↦ "has_mandatory_predecessor", srcName ↦ "IVI", dstName ↦ "EVI"]
}
]
IN
Assert(
Print(ACTUAL, ACTUAL = EXPECTED),

```

```

    Print(EXPECTED, "should report errors for repeating a task before its mandatory predecessor")
)

```

should report no errors for repeating co-dependent tasks properly

```

, LET Validation  $\triangleq$  INSTANCE WorkflowValidation WITH
    Connections  $\leftarrow$ 
    { [name  $\mapsto$  "has_mandatory_successor", srcName  $\mapsto$  "EVI", dstName  $\mapsto$  "IVI"]
    , [name  $\mapsto$  "has_mandatory_predecessor", srcName  $\mapsto$  "IVI", dstName  $\mapsto$  "EVI"]
    },
    Tasks  $\leftarrow$ 
    { [name  $\mapsto$  "EVI", repeatable  $\mapsto$  TRUE, group  $\mapsto$  "both"]
    , [name  $\mapsto$  "IVI", repeatable  $\mapsto$  TRUE, group  $\mapsto$  "both"]
    },
    Workflow  $\leftarrow$ 
    <"EVI", "IVI", "EVI", "IVI">

    ACTUAL  $\triangleq$  Validation!Errors

    EXPECTED  $\triangleq$  Validation!NoErrors

IN
    Assert(
        Print(ACTUAL, ACTUAL = EXPECTED),
        Print(EXPECTED, "should report no errors for repeating co-dependent tasks properly")
    )

```

should report errors for repeating one of the co-dependent tasks more often at the end

```

, LET Validation  $\triangleq$  INSTANCE WorkflowValidation WITH
    Connections  $\leftarrow$ 
    { [name  $\mapsto$  "has_mandatory_successor", srcName  $\mapsto$  "EVI", dstName  $\mapsto$  "IVI"]
    , [name  $\mapsto$  "has_mandatory_predecessor", srcName  $\mapsto$  "IVI", dstName  $\mapsto$  "EVI"]
    },
    Tasks  $\leftarrow$ 
    { [name  $\mapsto$  "EVI", repeatable  $\mapsto$  TRUE, group  $\mapsto$  "both"]
    , [name  $\mapsto$  "IVI", repeatable  $\mapsto$  TRUE, group  $\mapsto$  "both"]
    },
    Workflow  $\leftarrow$ 
    <"EVI", "IVI", "EVI", "IVI", "EVI">

    ACTUAL  $\triangleq$  Validation!Errors

    EXPECTED  $\triangleq$  [Validation!NoErrors EXCEPT !.ErrorMissingMandatoryDependencies =
        { [name  $\mapsto$  "has_mandatory_successor", srcName  $\mapsto$  "EVI", dstName  $\mapsto$  "IVI"]
        , [name  $\mapsto$  "has_mandatory_predecessor", srcName  $\mapsto$  "IVI", dstName  $\mapsto$  "EVI"]
        }
    ]
]

```



```

IN
  Assert(
    Print(ACTUAL, ACTUAL = EXPECTED),
    Print(EXPECTED, "should report errors for repeating one of the co-dependent tasks more often at the e
  )

```

should report errors for repeating one of the co-dependent tasks more often at the beginning

```

, LET Validation  $\triangleq$  INSTANCE WorkflowValidation WITH
  Connections  $\leftarrow$ 
  { [name  $\mapsto$  "has_mandatory_successor", srcName  $\mapsto$  "EVI", dstName  $\mapsto$  "IVI"]
    , [name  $\mapsto$  "has_mandatory_predecessor", srcName  $\mapsto$  "IVI", dstName  $\mapsto$  "EVI"]
  },
  Tasks  $\leftarrow$ 
  { [name  $\mapsto$  "EVI", repeatable  $\mapsto$  TRUE, group  $\mapsto$  "both"]
    , [name  $\mapsto$  "IVI", repeatable  $\mapsto$  TRUE, group  $\mapsto$  "both"]
  },
  Workflow  $\leftarrow$ 
  <"IVI", "EVI", "IVI", "EVI", "IVI">

  ACTUAL  $\triangleq$  Validation!Errors

  EXPECTED  $\triangleq$  [Validation!NoErrors EXCEPT !.ErrorMissingMandatoryDependencies =
    { [name  $\mapsto$  "has_mandatory_successor", srcName  $\mapsto$  "EVI", dstName  $\mapsto$  "IVI"]
      , [name  $\mapsto$  "has_mandatory_predecessor", srcName  $\mapsto$  "IVI", dstName  $\mapsto$  "EVI"]
    }
  ]

```

```

IN
  Assert(
    Print(ACTUAL, ACTUAL = EXPECTED),
    Print(EXPECTED, "should report errors for repeating one of the co-dependent tasks more often at the b
  )

```

should report errors for incorrectly repeating co-dependent tasks

```

, LET Validation  $\triangleq$  INSTANCE WorkflowValidation WITH
  Connections  $\leftarrow$ 
  { [name  $\mapsto$  "has_mandatory_successor", srcName  $\mapsto$  "EVI", dstName  $\mapsto$  "IVI"]
    , [name  $\mapsto$  "has_mandatory_predecessor", srcName  $\mapsto$  "IVI", dstName  $\mapsto$  "EVI"]
  },
  Tasks  $\leftarrow$ 
  { [name  $\mapsto$  "EVI", repeatable  $\mapsto$  TRUE, group  $\mapsto$  "both"]
    , [name  $\mapsto$  "IVI", repeatable  $\mapsto$  TRUE, group  $\mapsto$  "both"]
  },
  Workflow  $\leftarrow$ 
  <"EVI", "IVI", "EVI", "EVI", "IVI", "IVI">

```

```

ACTUAL  $\triangleq$  Validation!Errors

EXPECTED  $\triangleq$  [Validation!NoErrors EXCEPT !.ErrorMissingMandatoryDependencyRepetitions =
  { [name  $\mapsto$  "has_mandatory_successor", srcName  $\mapsto$  "EVI", dstName  $\mapsto$  "IVI"]
    , [name  $\mapsto$  "has_mandatory_predecessor", srcName  $\mapsto$  "IVI", dstName  $\mapsto$  "EVI"]
  }
]
IN
  Assert(
    Print(ACTUAL, ACTUAL = EXPECTED),
    Print(EXPECTED, "should report errors for incorrectly repeating co-dependent tasks")
  )

```

should report no errors for correctly repeating dependent tasks in a more complex example

```

, LET Validation  $\triangleq$  INSTANCE WorkflowValidation WITH
  Connections  $\leftarrow$ 
  { [name  $\mapsto$  "has_mandatory_successor", srcName  $\mapsto$  "EVI", dstName  $\mapsto$  "XRAY"]
    , [name  $\mapsto$  "has_mandatory_predecessor", srcName  $\mapsto$  "IVI", dstName  $\mapsto$  "XRAY"]
  },
  Tasks  $\leftarrow$ 
  { [name  $\mapsto$  "EVI", repeatable  $\mapsto$  TRUE, group  $\mapsto$  "both"]
    , [name  $\mapsto$  "XRAY", repeatable  $\mapsto$  TRUE, group  $\mapsto$  "both"]
    , [name  $\mapsto$  "IVI", repeatable  $\mapsto$  TRUE, group  $\mapsto$  "both"]
  },
  Workflow  $\leftarrow$ 
  < "EVI", "XRAY"
    , "EVI", "XRAY", "IVI"
    , "XRAY"
    , "EVI", "XRAY", "IVI"
    , "XRAY", "IVI"
  >

  ACTUAL  $\triangleq$  Validation!Errors

  EXPECTED  $\triangleq$  Validation!NoErrors
IN
  Assert(
    Print(ACTUAL, ACTUAL = EXPECTED),
    Print(EXPECTED, "should report no errors for correctly repeating dependent tasks in a more complex ex
  )

```

should report errors for incorrectly repeating dependent tasks in a more complex example I

```

, LET Validation  $\triangleq$  INSTANCE WorkflowValidation WITH
  Connections  $\leftarrow$ 

```

```

{ [name ↦ "has_mandatory_successor", srcName ↦ "EVI", dstName ↦ "XRAY"]
, [name ↦ "has_mandatory_predecessor", srcName ↦ "IVI", dstName ↦ "XRAY"]
},
Tasks ←
{ [name ↦ "EVI", repeatable ↦ TRUE, group ↦ "both"]
, [name ↦ "XRAY", repeatable ↦ TRUE, group ↦ "both"]
, [name ↦ "IVI", repeatable ↦ TRUE, group ↦ "both"]
},
Workflow ←
⟨ "EVI"
, "EVI", "XRAY", "IVI"
, "IVI"
⟩
ACTUAL ≜ Validation!Errors

EXPECTED ≜ [Validation!NoErrors EXCEPT !.ErrorMissingMandatoryDependencyRepetitions =
{ [name ↦ "has_mandatory_successor", srcName ↦ "EVI", dstName ↦ "XRAY"]
, [name ↦ "has_mandatory_predecessor", srcName ↦ "IVI", dstName ↦ "XRAY"]
}
]
IN
Assert(
Print(ACTUAL, ACTUAL = EXPECTED),
Print(EXPECTED, "should report errors for incorrectly repeating dependent tasks in a more complex example I")
)

```

should report errors for incorrectly repeating dependent tasks in a more complex example II

```

, LET Validation ≜ INSTANCE WorkflowValidation WITH
Connections ←
{ [name ↦ "has_mandatory_successor", srcName ↦ "EVI", dstName ↦ "XRAY"]
, [name ↦ "has_mandatory_predecessor", srcName ↦ "IVI", dstName ↦ "XRAY"]
},
Tasks ←
{ [name ↦ "EVI", repeatable ↦ TRUE, group ↦ "both"]
, [name ↦ "XRAY", repeatable ↦ TRUE, group ↦ "both"]
, [name ↦ "IVI", repeatable ↦ TRUE, group ↦ "both"]
},
Workflow ←
⟨ "EVI"
, "EVI", "XRAY", "IVI"
, "XRAY", "IVI"
⟩
ACTUAL ≜ Validation!Errors

EXPECTED ≜ [Validation!NoErrors EXCEPT !.ErrorMissingMandatoryDependencyRepetitions =

```

```

    { [name ↦ "has_mandatory_successor", srcName ↦ "EVI", dstName ↦ "XRAY"]
    }
  ]
IN
  Assert(
    Print(ACTUAL, ACTUAL = EXPECTED),
    Print(EXPECTED, "should report errors for incorrectly repeating dependent tasks in a more complex example")
  )

```

should report no errors for correctly repeating co-dependent tasks in a more complex example

```

, LET Validation ≜ INSTANCE WorkflowValidation WITH
  Connections ←
  { [name ↦ "has_mandatory_successor", srcName ↦ "EVI", dstName ↦ "XRAY"]
  , [name ↦ "has_mandatory_predecessor", srcName ↦ "XRAY", dstName ↦ "EVI"]
  , [name ↦ "has_mandatory_successor", srcName ↦ "XRAY", dstName ↦ "IVI"]
  , [name ↦ "has_mandatory_predecessor", srcName ↦ "IVI", dstName ↦ "XRAY"]
  },
  Tasks ←
  { [name ↦ "EVI", repeatable ↦ TRUE, group ↦ "both"]
  , [name ↦ "XRAY", repeatable ↦ TRUE, group ↦ "both"]
  , [name ↦ "IVI", repeatable ↦ TRUE, group ↦ "both"]
  },
  Workflow ←
  { "EVI", "XRAY", "IVI"
  , "EVI", "XRAY", "IVI"
  , "EVI", "XRAY", "IVI"
  }

  ACTUAL ≜ Validation!Errors
  EXPECTED ≜ Validation!NoErrors
IN
  Assert(
    Print(ACTUAL, ACTUAL = EXPECTED),
    Print(EXPECTED, "should report no errors for correctly repeating co-dependent tasks in a more complex example")
  )

```

should report errors for incorrectly repeating co-dependent tasks in a more complex example

```

, LET Validation ≜ INSTANCE WorkflowValidation WITH
  Connections ←
  { [name ↦ "has_mandatory_successor", srcName ↦ "EVI", dstName ↦ "XRAY"]
  , [name ↦ "has_mandatory_predecessor", srcName ↦ "XRAY", dstName ↦ "EVI"]
  , [name ↦ "has_mandatory_successor", srcName ↦ "XRAY", dstName ↦ "IVI"]
  , [name ↦ "has_mandatory_predecessor", srcName ↦ "IVI", dstName ↦ "XRAY"]
  }

```

```

    },
    Tasks ←
    { [name ↦ "EVI", repeatable ↦ TRUE, group ↦ "both"]
    , [name ↦ "XRAY", repeatable ↦ TRUE, group ↦ "both"]
    , [name ↦ "IVI", repeatable ↦ TRUE, group ↦ "both"]
    },
    Workflow ←
    ⟨ "EVI", "XRAY", "IVI"
    , "EVI", "XRAY", "IVI"
    , "XRAY", "IVI"
    ⟩
    ACTUAL ≜ Validation!Errors

    EXPECTED ≜ [Validation!NoErrors EXCEPT !.ErrorMissingMandatoryDependencyRepetitions =
    { [name ↦ "has_mandatory_successor", srcName ↦ "EVI", dstName ↦ "XRAY"]
    , [name ↦ "has_mandatory_predecessor", srcName ↦ "XRAY", dstName ↦ "EVI"]
    }
    ]
IN
    Assert(
        Print(ACTUAL, ACTUAL = EXPECTED),
        Print(EXPECTED, "should report errors for incorrectly repeating co-dependent tasks in a more complex")
    )

```

should report no errors for respecting transitive partial order between tasks

```

, LET Validation ≜ INSTANCE WorkflowValidation WITH
    Connections ←
    { [name ↦ "has_successor", srcName ↦ "EVI", dstName ↦ "XRAY"]
    , [name ↦ "has_successor", srcName ↦ "XRAY", dstName ↦ "IVI"]
    },
    Tasks ←
    { [name ↦ "EVI", repeatable ↦ FALSE, group ↦ "both"]
    , [name ↦ "XRAY", repeatable ↦ FALSE, group ↦ "both"]
    , [name ↦ "IVI", repeatable ↦ FALSE, group ↦ "both"]
    },
    Workflow ←
    ⟨ "EVI", "IVI" ⟩

    ACTUAL ≜ Validation!Errors

    EXPECTED ≜ Validation!NoErrors
IN
    Assert(
        Print(ACTUAL, ACTUAL = EXPECTED),
        Print(EXPECTED, "should report errors for violating transitive partial order between tasks")
    )

```

```

)

should report errors for violating transitive partial order between tasks

, LET Validation  $\triangleq$  INSTANCE WorkflowValidation WITH
  Connections  $\leftarrow$ 
  { [name  $\mapsto$  "has_successor", srcName  $\mapsto$  "EVI", dstName  $\mapsto$  "XRAY"]
  , [name  $\mapsto$  "has_successor", srcName  $\mapsto$  "XRAY", dstName  $\mapsto$  "IVI"]
  },
  Tasks  $\leftarrow$ 
  { [name  $\mapsto$  "EVI", repeatable  $\mapsto$  FALSE, group  $\mapsto$  "both"]
  , [name  $\mapsto$  "XRAY", repeatable  $\mapsto$  FALSE, group  $\mapsto$  "both"]
  , [name  $\mapsto$  "IVI", repeatable  $\mapsto$  FALSE, group  $\mapsto$  "both"]
  },
  Workflow  $\leftarrow$ 
  <"IVI", "EVI">

  ACTUAL  $\triangleq$  Validation!Errors

  EXPECTED  $\triangleq$  [Validation!NoErrors EXCEPT
    !.ErrorPartialOrderViolations = {
      [name  $\mapsto$  "has_successor", srcName  $\mapsto$  "EVI", dstName  $\mapsto$  "IVI"]
    }
  ]
IN
  Assert(
    Print(ACTUAL, ACTUAL = EXPECTED),
    Print(EXPECTED, "should report errors for violating transitive partial order between tasks")
  )
}


```

---