-- This is the USE model for only part of agentTool. Specifically for
Goal Diagram
-- Hence the model assumes that the diagram contains only Goals
-- Hence only the association between Goal and Diagram is included here.

model agentTool

--

-- CLASSES
--

class Diagram
dend

class ModelElement
attributes
  name : String
  stereotype : String
dend

class Organization < ModelElement
dend

class Goal < ModelElement
attributes
  currentChildNumber : String
operations
  addGoal(o : Diagram) : Set(Goal) = o.hasGoals->union(Set{self})
  deleteGoal(o : Diagram) : Set(Goal) = o.hasGoals - (Set{self})
  addSubGoal(o : Diagram, g: Goal, c: Relationship)
dend

class Role < ModelElement
dend

class Capability < ModelElement
dend

class Agent < ModelElement
dend

class Service < ModelElement
dend

class Relationship
attributes
  name : String
  stereotype : String
  srcElement : ModelElement
  tgtElement : ModelElement
dend

class AndORRelationship < Relationship
attributes
  type : String
  parentGoal : Goal
subGoal : Goal
end

--
-- ASSOCIATIONS
--

association GoalBelongsTo between
  Goal[*] role hasGoals
  Diagram[1] role belongsTo
end

association RelBelongsTo between
  Relationship[*] role hasRelationships
  Diagram[1] role belongsTo
end

association ANDORRel between
  Goal[2] role connects
  AndORRelationship[*] role isConnectedTo
end

--
-- CONSTRAINTS
--

constraints

--
-- Unique names. Goal Names should be unique
--
context o:Diagram
inv UniqueGoals:
  o.hasGoals->forAll(g1, g2 | g1 <> g2 implies g1.name <> g2.name)

--
-- OPERATIONS
--

--
-- Adding a goal
-- Added object must be new
-- Object should be added to the list of goals in the diagram
--
context Goal::addGoal(o : Diagram): Set(Goal)
  pre cond1 : o.hasGoals->excludes(self)
  post cond2 : o.hasGoals = o.hasGoals@pre->union(Set{self})
  post cond3 : result = o.hasGoals
  post cond4 : (o.hasGoals - o.hasGoals@pre)->size() = 1

--
-- Deleting a goal
-- Object to be deleted must exist in the organization
-- Object should be deleted from the list of goals in the organization
--
context Goal::deleteGoal(o : Diagram): Set(Goal)
  pre cond1 : o.hasGoals->includes(self)
  post cond2 : o.hasGoals = o.hasGoals@pre->excluding(self)
  post cond3 : result = o.hasGoals
  post cond4 : (o.hasGoals@pre - o.hasGoals)->size() = 1

--
-- Adding a sub goal
-- The goals to be connected via the parent-child relationship should both exist in the organization
-- The name of the sub-goal should become one more than the child number of the parent
-- The relationship type of all sub goals of the parent goal should be same.
-- AND / OR connections not allowed together.

context Goal::addSubGoal(o : Diagram, g: Goal, c: AndORRelationship)
  pre condpre1 : o.hasGoals->includes(self)
  pre condpre2 : o.hasGoals->includes(g)
  pre condpre3 : c.parentGoal = self

  post condpost1 : g.name = self.name->concat('.')->concat(self.currentChildNumber)
  post condpost2 : c.subGoal = g
  post condpost3 : AndORRelationship.allInstances->select(c1 |
  c1.parentGoal = self)->forAll(c2 | c2.type = c.type)