Introduction

This model represents a cake bakery. It contains chefs who are able to bake cakes using utensils and also buy ingredients. All ingredients are required to bake a cake. The optimal solution to a problem file would be one in which the goal state is achieved in the shortest time. A typical goal state would require each cake to be decorated or just baked.

Domain

Objects

The following objects are in the domain:

- Ingredient
  The amount of any ingredient can vary during the run.

- Individuals
  Chef, oven, cake, and utensil are of type individual. These are individual whole items for which there can be a fixed number for any run.

- Utensil
  Spoon, bowl and tray are types of utensils. They are required to bake the cake.

Predicates

- Baked
  This indicates whether or not a cake has been baked.

- Decorated
  This indicates whether or not a cake has been decorated. A cake must be decorated after it has been baked.

- Available
  This indicates whether an individual is available or currently involved in an action.

- Clean
  This indicates whether or not a utensil is clean.

- Contains
  This indicates whether or not a particular cake currently contains a particular ingredient.

- In-bowl
  This indicates whether or not a particular bowl is being used for a particular cake.

Functions

There are two ways in which functions are used in the domain. The first is to mark the time taken by a chef to perform a task. The other is for maximum,

required and current levels of a chef's energy or the amount held of an ingredient.

Actions

There is a single action cake-in-bowl which involves a chef assigning a bowl to cake. It requires all involved individuals to be available and that the bowl is washed.

Durative Actions

All actions are taken by a chef and, aside from rest require the chef to have enough energy, decreasing the energy by this required amount. The chef must also be available in order to perform the action. Throughout all durative actions all involved individuals are unavailable with the exception of the chef in bake-cake as the cake is in the oven.

- Buy-ingredient
  This restocks an ingredient to maximum amount. A chef goes on a journey to buy the ingredient which takes an amount of time that is dependent on the chef.

- Rest
  This refreshes a chef to the maximum energy and takes a certain amount of time for all chefs.

- Bake-cake
  This bakes a cake. All ingredients must be present in the cake which, along with both a clean tray and an oven, must be available. The bowl previously used to mix the ingredients is now free and needs washing.

- Decorate-cake
  This decorates a cake. The cake must already have been baked and be available. It is assumed that there are always sufficient decorations.

- Add-ingredient
  This adds an ingredient to a cake. It requires; a sufficient amount of the ingredient, that the cake is available and assigned to a bowl, and a clean spoon. It decreases the current amount of the ingredient by the amount used.

- Wash-utensil
  This cleans a utensil. The utensil must be available.

Appendix

```
(define (domain bake-a-cake)

   (:requirements :strips :typing :durative-actions :numeric-fluents)

   (:types

      ingredient - object
      individual - object
      chef - individual
      cake - individual
      oven - individual
      utensil - individual
      spoon, bowl, tray - utensil
     )

(:predicates

      (baked ?cake - cake)
      (decorated ?cake - cake)
      (available ?individual - individual)
      (clean ?utensil - utensil)
      (contains ?cake - cake ?ingredient - ingredient)
      (in-bowl ?cake - cake ?bowl - bowl)
)

  (:functions

      (bake-time ?oven - oven) - number
      (max-amount) - number
      (required-amount) - number
      (amount ?ingredient - ingredient) - number
      (max-energy) - number
      (required-energy) - number
      (energy ?chef - chef) - number
      (travel-time ?chef - chef) - number
      (decorate-time ?chef - chef) - number
      (mix-time ?chef - chef) - number
      (wash-time ?chef - chef) - number
      (rest-time) - number
)

(:durative-action buy-ingredient

   :parameters (?ingredient - ingredient ?chef - chef)
   :duration (= ?duration (travel-time ?chef))
   :condition (and
      (at start (available ?chef))
      (at start (< (amount ?ingredient) max-amount))
      (at start (>= (energy ?chef) required-energy))
      )
```

```
   :effect (and
      (at end (assign (amount ?ingredient) max-amount ))

      (at start (not (available ?chef)))
      (at end (available ?chef))
      (at end (decrease (energy ?chef) required-energy))
      )
)

(:durative-action rest

   :parameters (?chef - chef)
   :duration (= ?duration rest-time)
   :condition (and
      (at start (available ?chef))
      (at start (< (energy ?chef) max-energy))
      )
   :effect (and
      (at end (assign (energy ?chef) max-energy ))
      (at start (not (available ?chef)))
      (at end (available ?chef))
      )
)

(:durative-action bake-cake

  :parameters (?cake - cake ?tray - tray ?oven - oven ?chef - chef ?bowl -
bowl)
  :duration (= ?duration (bake-time ?oven))
   :condition (and
      (at start (in-bowl ?cake ?bowl))
      (at start (available ?chef))
      (at start (available ?cake))
      (at start (available ?oven))
      (at start (available ?tray))
      (at start (>= (energy ?chef) required-energy))
      (at start (clean ?tray))
      (at start (forall (?ingredient - ingredient)
                        (contains ?cake ?ingredient)))
      )
   :effect (and
      (at end (baked ?cake))
      (at start (not (in-bowl ?cake ?bowl)))
      (at start (available ?bowl))
      (at start (not (clean ?bowl)))
      (at start (not (available ?tray)))
      (at end (available ?tray))
      (at start (not (available ?oven)))
      (at end (available ?oven))
      (at start (not (available ?cake)))
      (at end (available ?cake))
```

```
      (at start (not (clean ?tray)))
      (at end (decrease (energy ?chef) required-energy))
      )

)

(:durative-action decorate-cake

   :parameters (?cake - cake ?chef - chef)
   :duration (= ?duration (decorate-time ?chef))
   :condition (and
      (at start (available ?chef))
      (at start (available ?cake))
      (at start (baked ?cake))
      (at start (>= (energy ?chef) required-energy))
      )
   :effect (and
      (at end (decorated ?cake ))
      (at start (not (available ?cake)))
      (at end (available ?cake))
      (at start (not (available ?chef)))
      (at end (available ?chef))
      (at end (decrease (energy ?chef) required-energy))
      )
)

(:action cake-in-bowl
   :parameters (?cake - cake ?bowl - bowl ?chef - chef)
   :precondition (and
      (available ?bowl)
      (clean ?bowl)
      (available ?cake)
      (available ?chef)
      )
   :effect(and
      (in-bowl ?cake ?bowl)
      (not (available ?bowl))
      )
)

(:durative-action add-ingredient

   :parameters (?ingredient - ingredient ?cake - cake ?spoon - spoon ?bowl -
bowl ?chef - chef)
   :duration (= ?duration (mix-time ?chef))
   :condition (and
      (at start (in-bowl ?cake ?bowl))
      (at start (available ?chef))
      (at start (available ?spoon))
      (at start (>= (energy ?chef) required-energy))
      (at start (clean ?spoon))
```

```
      (at start (>= (amount ?ingredient) required-amount))
      )
   :effect (and
      (at start (not (clean ?spoon)))
      (at start (not (clean ?bowl)))
      (at start (not (available ?spoon)))
      (at end (available ?spoon))

      (at end (contains ?cake ?ingredient ))
      (at start (not (available ?chef)))
      (at end (available ?chef))
      (at start (decrease (amount ?ingredient) required-amount))
      (at end (decrease (energy ?chef) required-energy))
      )
)

(:durative-action wash-utensil

   :parameters (?utensil - utensil ?chef - chef)
   :duration (= ?duration (wash-time ?chef))
   :condition (and
      (at start (available ?utensil))
      (at start (available ?chef))
      (at start (>= (energy ?chef) required-energy))
      )
   :effect(and
      (at start (not (available ?utensil)))
      (at end (available ?utensil))
      (at end (clean ?utensil))
      (at start (not (available ?chef)))
      (at end (available ?chef))
      (at end (decrease (energy ?chef) required-energy))
      )
)
)
```