Scenario Example: microwave usage

Here is an example of the description of a scenario model with the **sam-l** language. The purpose is to monitor and recognize activities of old adult in an instrumented home, using sensing technology. The example concerns the recognition of the *using microwave* activity. This activity is depicted in the main scenario *usingMicrowave*. This scenario takes as argument a person and an equipment, thus in the data part we define two external types *Person* and *Equipment*. Functions related to camera (*close_to*, *far_from*) and to contact sensor on microwave (*is_used*, *is_not_used*) are also declared. These latter are Boolean functions giving information on the context at each instant. They will be implemented in the external code of the recognition engine provided to analyze this activity. Moreover an external constant is also declared in this data part. To define the main scenario, we rely on four primitive scenarios (*staysAt*, *microwaveUsed*, *microwaveNotUsed* and *farFrom*) corresponding to primitive events of the environment. Finally, the main scenario has these four primitive scenarios as components. Its relation part describes how these sub scenarios are temporally related. *usingMicrowave* scenario has also two constraints : a temporal constraint concerning the duration of *m_used* component and a constraint to check its eq argument type.

```
;; Using microwaveScenario
;;

Data:
Type: Person, Equipment;
Function: far_from (Person, Equipment) : boolean;
          close_to (Person, Equipment) : boolean;
          is_used(Equipment) : boolean;
          is_not_used(Equipment) : boolean;
          is_microwave(Equipment) : boolean;
Constant: threshold : integer;
end Data


primitive scenario staysAt(person:Person;  eq : Equipment)
{
     close_to(person, eq)
     Constraint:  is_microwave(eq);
}
primitive scenario microwaveUsed(eq : Equipment)
{
     is_used(eq)
     Constraint:
       is_microwave(eq);
}
```

```
primitive scenario microwaveNotUsed(eq : Equipment)
{
     is_not_used(eq)
     Constraint:  is_microwave(eq);
}

primitive scenario farFrom(person:Person; eq : Equipment)
{
    far_from(person, eq)
     Constraint:  is_microwave(eq);
}


main scenario usingMicrowave (person : Person; eq: Equipment)
{
   Component:
     p_stay : staysAt(person, eq);
     m_used : microwaveUsed(eq);
     m_not_used : microwaveNotUsed(eq);
     p_far:  farFrom(person, eq);

   Relation:
     m_used during p_stays;
     m_used before m_not_used;
     m_not_used before p_far;

   Constraint:
     duration(m_used) >= threshold;
     is_microwave(eq);
}
```