A Model for Energy-Saving in an IoT Smarthome accounting for End-User Convenience

Alistair Francis Bowman Grevis-James

November 2019

Abstract

The preface pretty much says it all.
Second paragraph of abstract starts here.

Contents

0.0.0.1 Algorithm 1

Text

Algorithm 1: Extraction of data from S1 Activities dataset

Result: Intermediate dataframe with 'activity', 'date', 'startTime', 'endTime' as attributes
Input : S1 Activities
Output: S1 Activities Intermediate

```
 1  begin
 2  |    array1-21 = [ ]
 3  |    while i < length(dataframe) do
 4  |    |    array1.append(dataframe[i][0])
 5  |    |    array2.append([x.strip() for x in array1[i].split(',')])
 6  |    |    array3.append(array2[i][0])
 7  |    |    array4.append(array2[i][1])
 8  |    |    array5.append(array2[i][2])
 9  |    |    array6.append(array2[i][3])
10  |    |    i = i + 1
11  |    end
12  |    while i < length(dataframe) do
13  |    |    array7.append(dataframe[i][1])
14  |    |    array8.append([x.strip() for x in array7[i].split(',')])
15  |    |    array9.append(dataframe[i][2])
16  |    |    array10.append([x.strip() for x in array9[i].split(',')])
17  |    |    array11.append(dataframe[i][3])
18  |    |    array12.append([x.strip() for x in array11[i].split(',')])
19  |    |    array13.append(dataframe[i][4])
20  |    |    array14.append([x.strip() for x in array13[i].split(',')])
21  |    |    i = i + 1
22  |    end
23  |    while i < length(dataframe) do
24  |    |    for x in range(len(array8[i])) : array15.append(array4[i])
25  |    |    i = i + 1
26  |    end
27  |    for sublist in array8: for item in sublist: array16.append(item)
28  |    for sublist in array10: for item in sublist: array17.append(item)
29  |    for sublist in array12: for item in sublist: array18.append(item)
30  |    for sublist in array13: for item in sublist: array19.append(item)
31  |    dfIntermediate = pandas.DataFrame(list(zip(array16, array17, array15, array18, array19)))
32  |    start = (dfIntermediate.date + " " + dfIntermediate.startTime)
33  |    end = (dfIntermediate.date + " " + dfIntermediate.endTime)
34  |    while i < length(start) do
35  |    |    array20.append(datetime.strptime(start[i], mm/dd/yyyy HH:MM:SS))
36  |    |    array21.append(datetime.strptime(end[i], mm/dd/yyyy HH:MM:SS))
37  |    |    i = i + 1
38  |    end
39  |    dfFinal = pandas.DataFrame(list(zip(array16, array17, array20, array21)))
40  |    return dfIntermediate
41  |    return dfFinal
42  end
```

%algorithm usage

Algorithm 2: Step planning

Input: entities; actors; player; customRules;
Output: new world state

```
1  let entities be all entities, excluding actors and the player;
2  let actors be all actors, including the player;
3  foreach actor in actors do
4  |    actor.rules.Invoke();

   /* It's possible that one of the rules gave the player a constraint, so we'll check                    */
5  if player.constraint == null then
6  |    GenerateConstraint(player);

7  foreach rule in customRules do
8  |    rule.Invoke();
```

%procedure usage

Procedure GenerateConstraint(Player)

Input: player
Output: new constraint for the player
1 if *rules.count ≥ 0* **then**
2   **foreach** *item in rules* **do**
3   rules.invoke(player);

4 **else**
5   **select random** *e* **from** *entities*;
6   player.constraint = *e*;

Algorithm 3: Algorithm with procedure

---

1 **Algorithm** `algo()`
3   xxx
5   xxx
7   proc()
9   **return**
1 **Procedure** `proc()`
3   xxx
5   **return**