```
/*********************************************
 * OPL 12.4 Model
 * Author: alexander
 * Creation Date: Apr 16, 2014 at 1:02:08 PM
 *********************************************/
using CP;

{string} userGroups  = {"u1", "u2"};
int cardUserGroups = card(userGroups);
{string} topics = {"robots", "microbiology"};
{string} frames = {"a1", "a2", "a3", "a4", "b1", "b2",
  "b3"};

float probabilityTopics[userGroups][topics] = [
    [0.9, 0.2],
    [0.7, 0.3]
];

// this should all be generated by a program
int minFrameId = 1;
int maxFrameId = 7;

range frameIdRange = minFrameId..maxFrameId;
int frameId[frames] = [ 1, 2, 3, 4, 5, 6, 7];

// before constraints
tuple Pair {
       int pred;
       int succ;
    };

// this set has to be provided by the users
// as of now, specify transitive closure manually
{Pair} beforeConstraints = {<frameId["a1"], frameId
  ["a2"]>};
```

```
34
35{int} seenBy[userGroups] = [
36    {frameId["a1"], frameId["a3"], frameId["a2"]},
37    {frameId["a1"], frameId["a3"]}
38];
39
40{int} belongsTo[topics] = [
41    {frameId["a1"], frameId["a2"], frameId["a3"],
  frameId["a4"]},
42    {frameId["b1"], frameId["b2"], frameId["b3"]}
43];
44
45
46
47dvar int chosenFrame in frameIdRange;
48
49dexpr int belongsToTopic[f in frameIdRange][t in topics]
  = (f in belongsTo[t]);
50dexpr int selectedTopic[t in topics] = belongsToTopic
  [chosenFrame][t];
51
52dexpr float preference = (1/cardUserGroups) * sum(u in
  userGroups, t in topics) selectedTopic[t] *
  probabilityTopics[u][t];
53
54// assuming a common prior over user groups
55dexpr float satisfaction = chosenFrame;
56
57maximize preference;
58
59subject to {
60    // should not have been seen
61    forall(u in userGroups) {
62      !(chosenFrame in seenBy[u]);
63    }
```

```
64
65    // now the before constraints
66    forall(p in beforeConstraints) {
67        // if we chose a frame that has to be preceded,
  we expect that each group has seen that
68        forall(u in userGroups) {
69            // either the before constraint does not
  refer to the chosen frame or
70            // its predecessor has been seen by each
  user
71            (!(p.succ == chosenFrame) || (p.pred in
  seenBy[u]));
72        }
73    }
74}
75
76execute {
77 writeln(chosenFrame);
78
79}
```