Generating program code for psychological experiments from high-level descriptions

Igor Dejanović[1], Mirjana Dejanović[2]

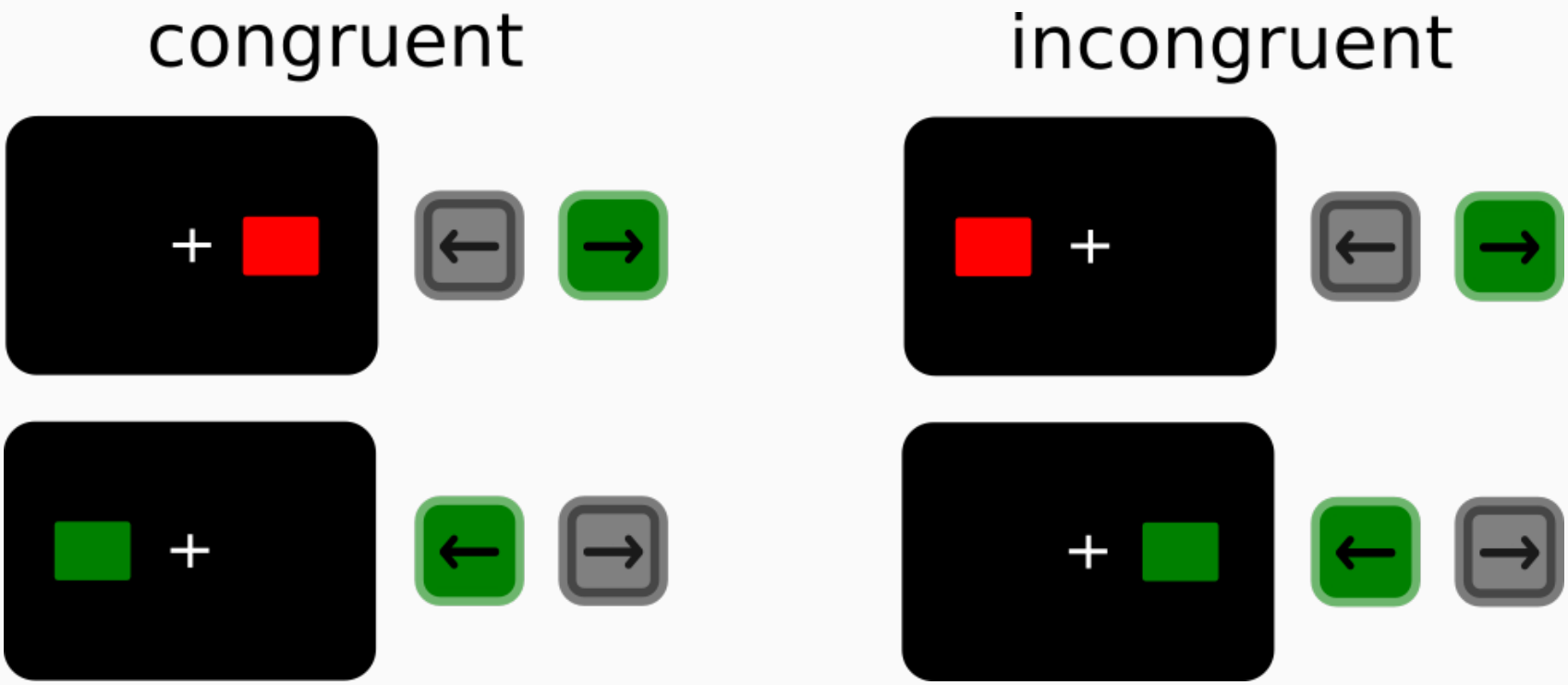
University of Novi Sad[1], University of Priština[2]

September, 2019 @ ERK Portorož Slovenia

Introduction

Experiments in Psychology

An example – the Simon effect test



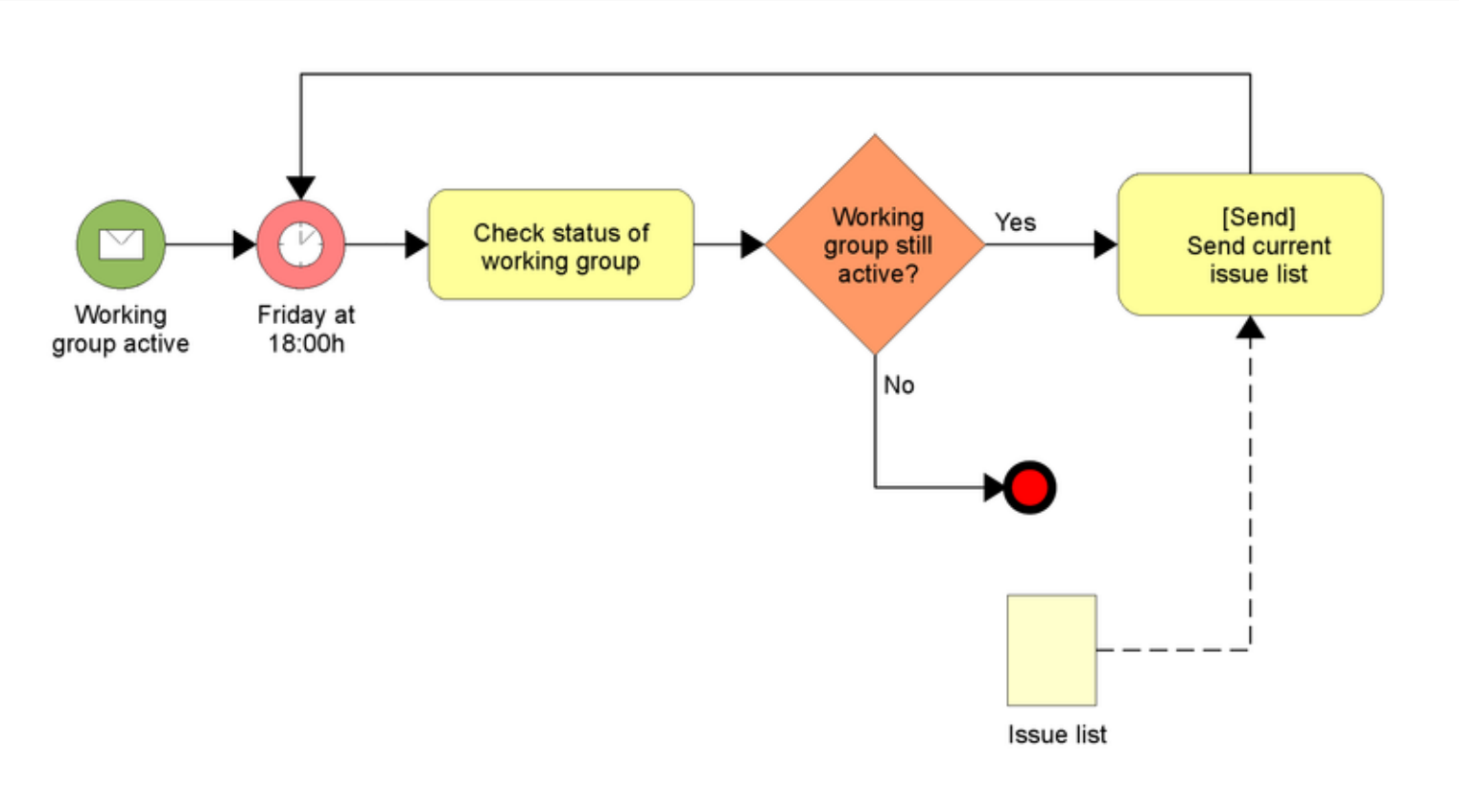
Motivation

Domain-Specific Languages

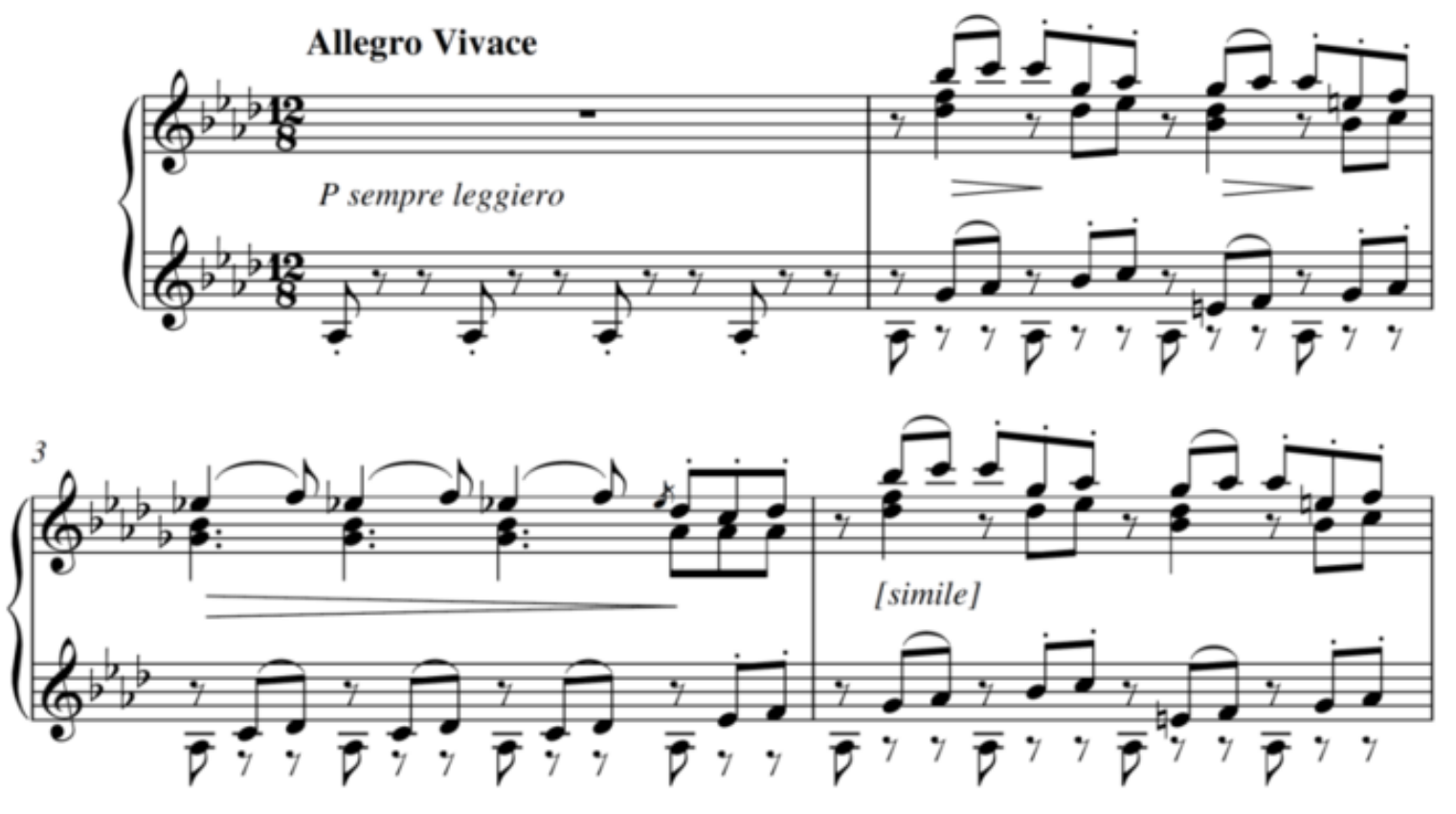
SQL

```
SELECT player, stadium
    FROM game JOIN goal ON (id=matchid)
```

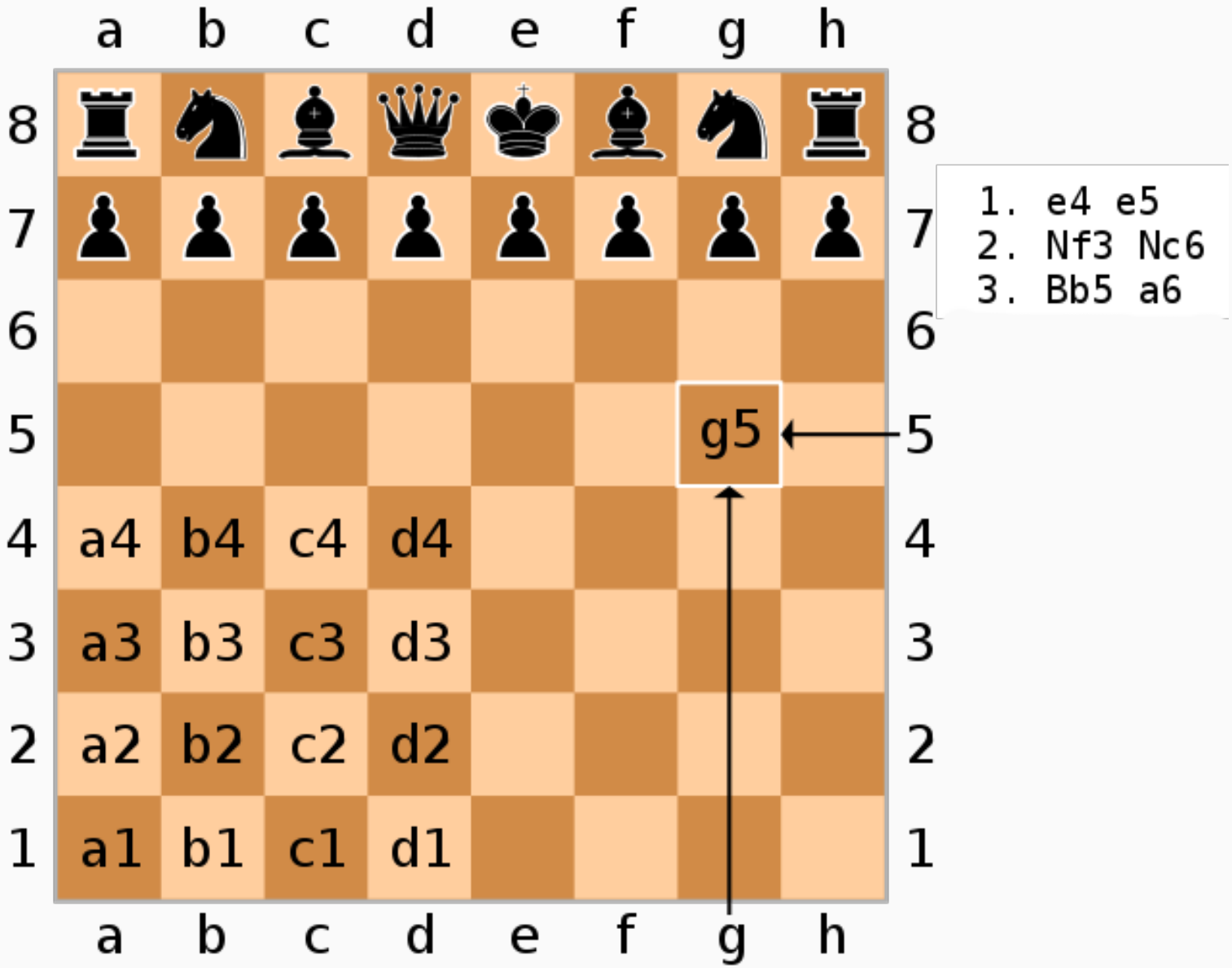
Business processes - BPMN

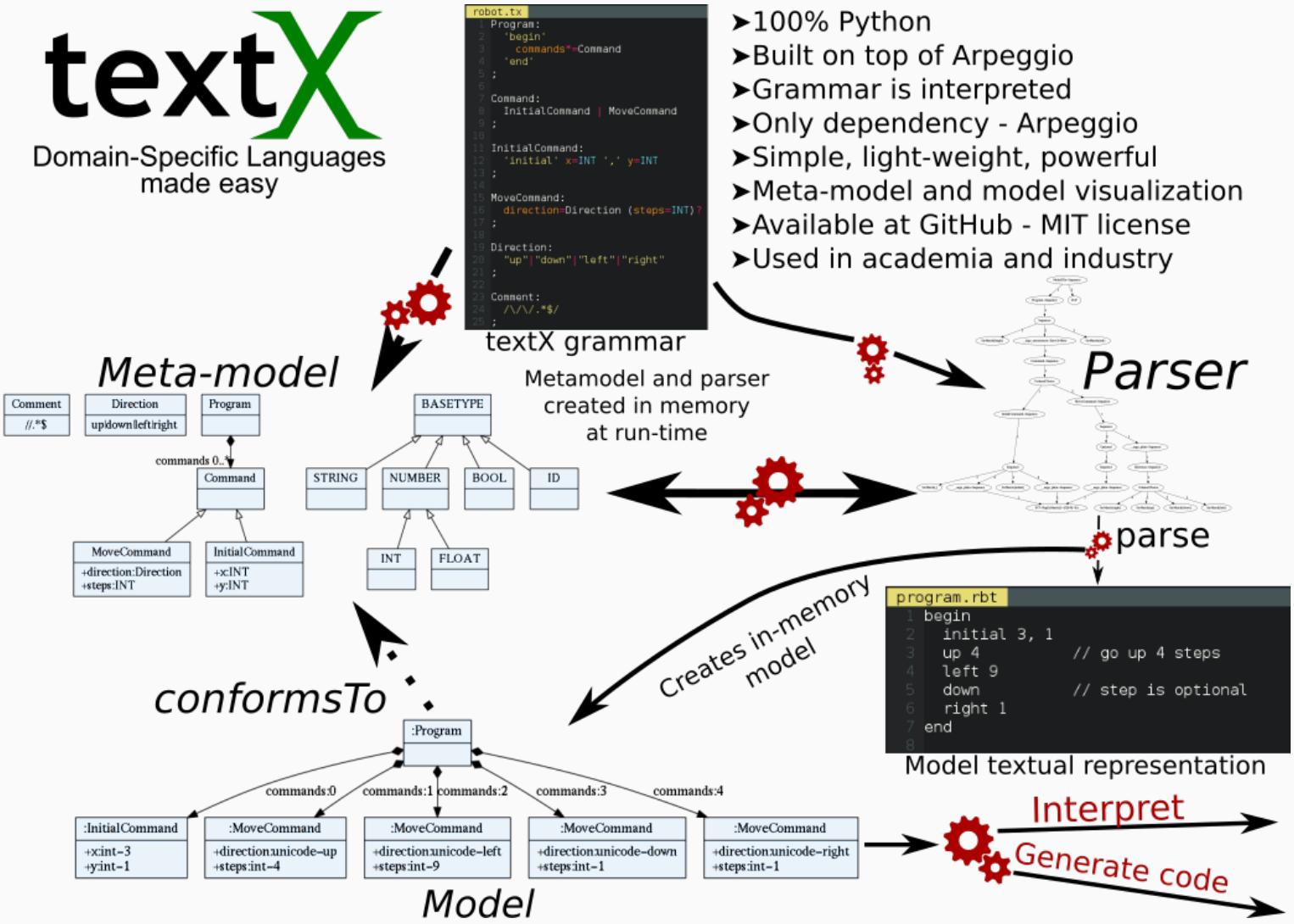


But also. . .



Or. . .


a b c d e f g h
8 7 6 5 4 3 2 1
g5
a4 b4 c4 d4
a3 b3 c3 d3
a2 b2 c2 d2
a1 b1 c1 d1
1. e4 e5
2. Nf3 Nc6
3. Bb5 a6



https://github.com/textX/textX

I. Dejanović, R. Vaderna, G. Milosavljević, Ž. Vuković, TextX: A Python tool for Domain-Specific Languages implementation, Knowledge-Based Systems 115, 1-4, 2017.

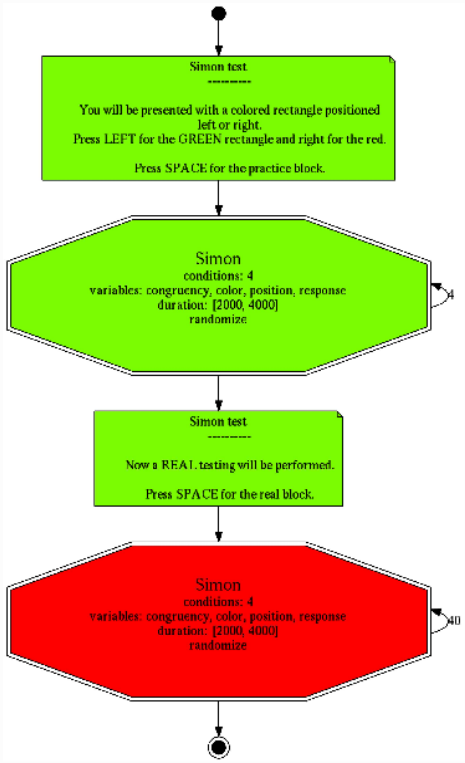
pyFlies

pyFlies - DSL for psychological RT experiments

pyFlies

Simon.pf | PosnerCueing.pf | Parity.pf

```
experiment "Simon"
"
A simple behavioural task to assess a Simon effect.

See also:
http://en.wikipedia.org/wiki/Simon_effect
"

test Simon {
    conditions {
        position     color        congruency        response

        left         green        congruent         left
        left         red          incongruent       right
        right        green        incongruent       left
        right        red          congruent         right
    }

    stimuli{
        all: shape(rectangle, position position, color color, fillColor color)
        /* Or in extended form
        1: shape(rectangle, position left, color green, fillColor green)
        2: shape(rectangle, position left, color red, fillColor red)
        3: shape(rectangle, position right, color green, fillColor green)
        4: shape(rectangle, position right, color red, fillColor red)
        */
        error: sound(1000, duration 300)
        correct: sound(500, duration 300)
        fixation: shape(cross)
    }
}

screen Practice {
    Simon test
    ----------

    You will be presented with a colored rectangle positioned
    left or right.
    Press LEFT for the GREEN rectangle and right for the red.
```

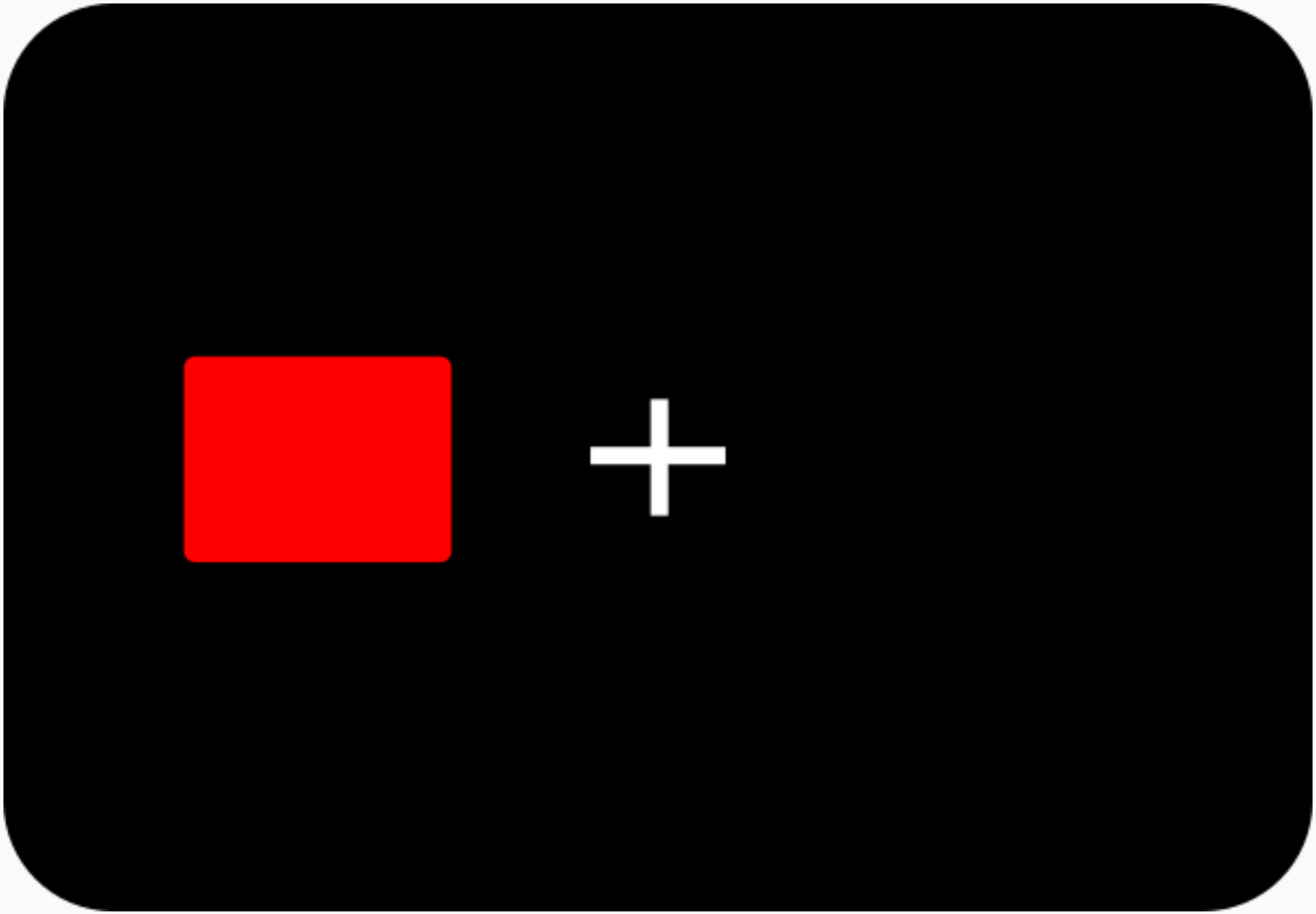


pyFlies code for the Simon effect test

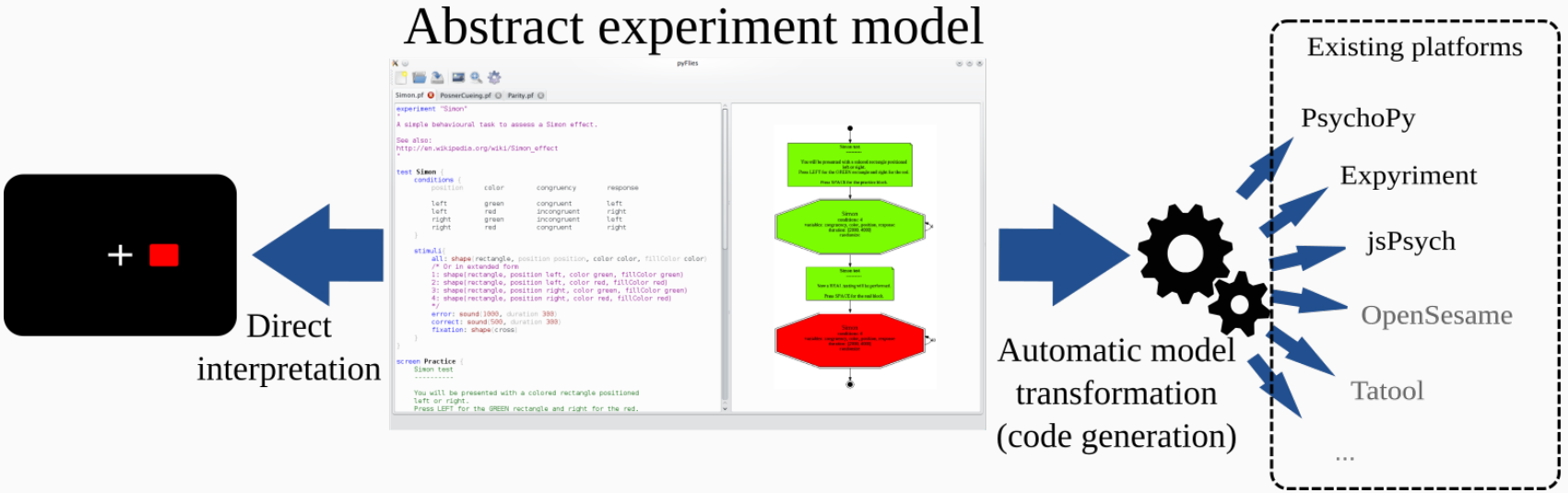
```
test Simon {
  conditions {
    position  color  congruency   response

    left      green  congruent    left
    left      red    incongruent  right
    right     green  incongruent  left
    right     red    congruent    right
  }

  stimuli{
    all: shape(rectangle, position position,
               color color)
    error: sound(1000)
    fixation: shape(cross)
  }
}
```

Connecting stimuli and conditions

```
parity=odd and position=left: image('red_square.png', position position)

position=left and color=red
```



Target code generators



Template engines

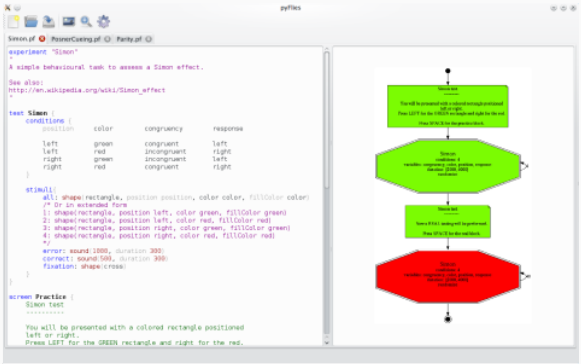
```
# Run experiment
# Create folder for experiment results.
timestr = time.strftime("%Y%m%d-%H%M%S")
exp_folder = '{{m.name}}-%s' % timestr
os.mkdir(exp_folder)

{%- for e in m.structure.elements %}
    {%- if e.__class__.__name__ == "ScreenInstance" %}

present_stimuli(screen_{{e.type.name}})
    {%- elif e.__class__.__name__ == "TestInstance" %}

run_block(exp_folder, "{{e.type.name}}", {{e.type.name}}_varNames,
          {{e.type.name}}_conditions, {{e.type.name}}_condition_stimuli,
          {{e.trials}}, {{e.practice}}, {{e.randomize}},
          {{e.type.name}}_fixation, {{e.type.name}}_error,
          {{e.type.name}}_correct)
    {%- endif %}
{%- endfor %}
```

Template

Template engine

```
# Run experiment
# Create folder for experiment results.
timestr = time.strftime("%Y%m%d-%H%M%S")
exp_folder = 'Simon-%s' % timestr
os.mkdir(exp_folder)

present_stimuli(screen_Practice)

run_block(exp_folder, "Simon", Simon_varNames,
          Simon_conditions, Simon_condition_stimuli,
          1, True, True,
          Simon_fixation, Simon_error, Simon_correct)
```

Generated source code



pyFlies model

Conclusion

Thanks! Q&A?