Template Design and
Automatic Generation of Controllers
for Industrial Robots

Lenko Grigorov[1], José E.R. Cury[2],
Karen Rudie[3], Steffi Klinge[4]

[1]School of Computing, Queen's University, Kingston, Canada, lenko.grigorov@banica.org
[2]Department of Automation and Systems, Federal University of Santa Catarina, Brazil
[3]Dept. of Electrical and Computer Engineering, Queen's University, Kingston, Canada
[4]Institute of Automation Technology, University of Magdeburg, Germany
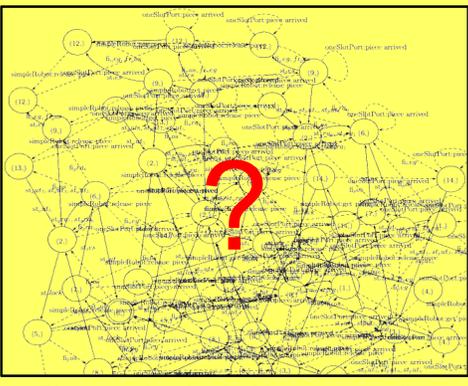
Problem

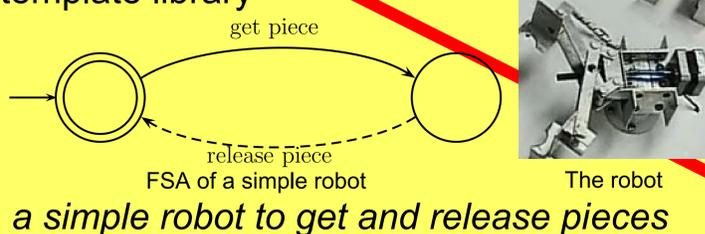Diagram of a simple 64-state controller

- Did I misspell a synchronization event?

- Will a small change in the robot model make any difference?

- What will the PLC-code look like?

- Will my solution work?!

Template Design is a method which is modular, robust, easy to use and easy to debug. Using templates, Discrete-Event Systems can be modeled and suitable supervisors built using the following steps:
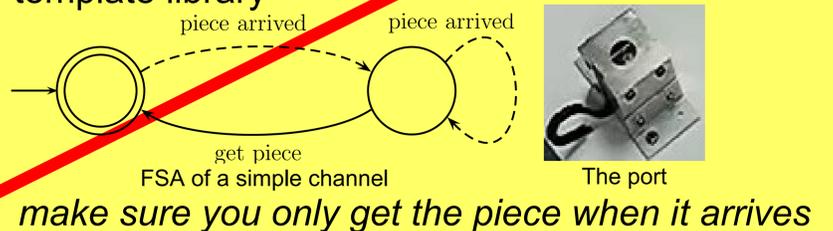
1. Modeling the system

1.1 Modules

Build all model components as finite state automata (FSA) or use a module from the template library

get piece

release piece

FSA of a simple robot   The robot

a simple robot to get and release pieces

1.2 Channels

Build all control requirements as finite state automata (FSA) or use a channel from the template library

piece arrived   piece arrived

get piece

FSA of a simple channel   The port

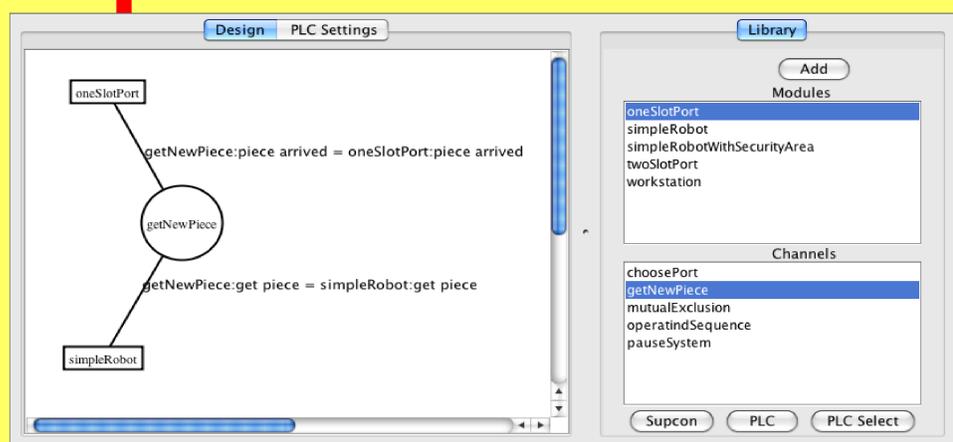*make sure you only get the piece when it arrives*

1.3 Link modules and channels

Connect the corresponding events of modules and channels

1.4 Input PLC-code for subroutines

For each event, enter the custom Instruction List commands necessary to control the hardware of the plant

Template Design in IDES

2. The software creates the optimal supervisor and generates the PLC-code

Creates the appropriate, optimal, nonblocking supervisor and the corresponding PLC-code in Instruction List automatically

```
NETWORK 0 // initialization
LDN ilc_inited
S      ilc_inited, 1
S      SUP0S0, 1
S      PS0S1, 1
S      PS1S0, 1

NETWORK 1 // Supervisor State Machine 0
LD      SUP0S0
A       oneSlotPortpiecearrived
R       SUP0S0, 1
S       SUP0S1, 1
...      ...
```

Part of the corresponding, generated PLC-Code

oneSlotPort:piece arrived

oneSlotPort:piece arrived

simpleRobot:release piece

simpleRobot:release piece

simpleRobot:get piece

oneSlotPort:piece arrived

oneSlotPort:piece arrived

FSA of the corresponding optimal, nonblocking supervisor