Computer Controlled Systems II.
Tutorial: Introduction to Stateflow

Anna I. Pózna

University of Pannonia
Faculty of Information Technology
Department of Electrical Engineering and Information Systems

pozna.anna@virt.uni-pannon.hu

January 27, 2020

Overview

Discrete event systems (DES)

Discrete state space

- $x(t) \in \mathbb{X} = \{x_0, x_1, \ldots, x_n\}$

Discrete time

- $T = \{t_0, t_1, \ldots, t_n\}$

Event

- change in the value of the discrete value of a variable

The state transition is **event driven**

Only the order of events is considered

- parallel or serial events

Stateflow

- Part of Simulink
- Graphical programming environment based on finite state machines
- Application areas
  - reactive control systems
  - control system logic
  - finite state machine
  - scheduling
  - fault detection
  - event driven systems
- Video tutorials
  - Part 1: `https://www.youtube.com/watch?v=thBxzulFuyg`
  - Part 2: `https://www.youtube.com/watch?v=jvSjBDnvbxE`
  - Part 3: `https://www.youtube.com/watch?v=64iuG25g-0g`
- Documentation
  - `https://www.mathworks.com/help/stateflow/index.html`

Basic concepts

- State
  - state name
  - entry action: executed when entering the state
  - during action: executed while the state is active
  - exit action: executed when a state is active and a transition out of the state occurs
- Transition
  - arcs between states
  - event_or_message trigger[condition]{condition_action}/{transition_action}
  - condition: boolean expression, in(state_name), temporal expression...
  - action: executing a function, setting a variable...
- Flowchart
  - creating functions
  - programming logic patterns
  - graphical form

Basic concepts cont.

- Hierarchy
  - compact models
  - superstates and subcharts
- Temporal logic
  - implement time delay between state transitions
- Variables
  - local data
  - input/output data from Simulink
  - constant
  - parameter
- Truth tables
  - condition table: description and conditional expression, decision: T, F, - (don't care)
  - action table: description and action

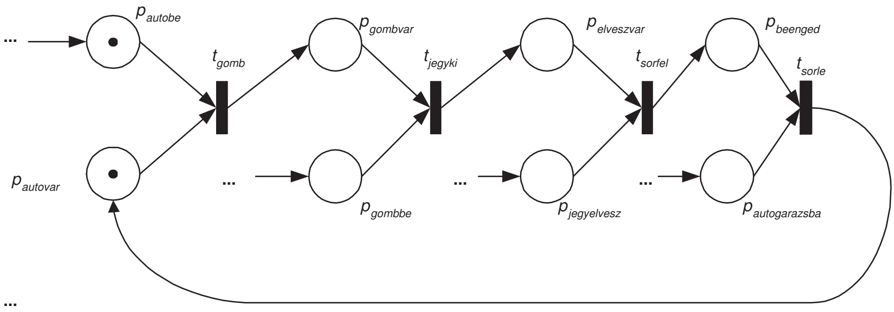
Simple example

Redundant sensor pair

- Open the model and examine it!
- openExample('stateflow/ModelingARedundantSensorPair UsingAtomicSubchartExample')

Example 2

Operation of a garage gate

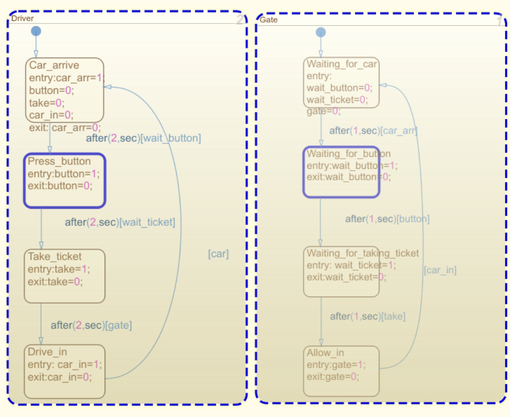


- State machine for the gate
- State machine for the driver

States of the two state machines

States

- Gate
  - Waiting_for_car
  - Waiting_for_button
  - Waiting_for_take_ticket
  - Allow_in
- Driver
  - Car_arrive
  - Press_button
  - Take_ticket
  - Drive_in



Variables

Local data variables

- Gate
  - gate: 0 - closed, 1 - opened
  - wait_button: 0 - inactive, 1 - active
  - wait_ticket: 0 - inactive, 1 - active
- Driver
  - car_arr: 0 - no car, 1 - car arrived
  - button: 0 - not pressed, 1 - pressed
  - take: 0 - ticket is not taken, 1 -ticket is taken
  - car_in: 0 - car not in the garage, 1 - car in the garage

Input data from Simulink

- car

Model creation

- Create a state machine for the driver/gate
  - states
  - entry actions: setting the variables
  - transitions, conditions
- Make it subcharted
- Parallel decomposition
  - the two state machines are evaluated in parallel
  - set the priority order (1-gate, 2-driver)
- Simulate the model
- Add time delay to the transitions
  - e.g. after(10,sec) $\rightarrow$ the transition is evaluated after 10 seconds
  - take care of the evaluation precedence!