# Discrete Event Systems

The DEVS generator has the following specification:

## Abbreviations

GEN : the generator

In1 : one inpt port of the generator

In2 : a second input por of the generator

σ : the time currently scheduled for the next scheduled internal event

ψ : the inter-arrival-time, the specified time between generated output events

φ : the phase, the discrete state of the generator

e : internal time elapsed since last event

## Assumptions:

Multiple input events with the same time-stamp are allowed.

When σ is zero the internal event must be generated immediately.

Given a fixed set of input the internal state and output must be determinate.

Neither the input value nor ψ can be zero or less, that would cause thrashing.

## Functional Specification:

GEN has In1 which, upon receiving a value sets ψ to that value and adjusts σ. There is a bit of ambiguity here in the word "immediately", there are several choices for σ. Candidate values for σ are:

- 0 : this would cause the immediate production of an internal event and the scheduling at the new time. (rejected)
- ψ : this is similar but would not create an immediate internal event. (rejected)
- ψ – e : a pretty good compromise, it credits the elapsed time. It has the problem that it could be 0, in that case 0 should be used. (accepted)

GEN has In2 which, upon receiving a value updates ψ to that value.

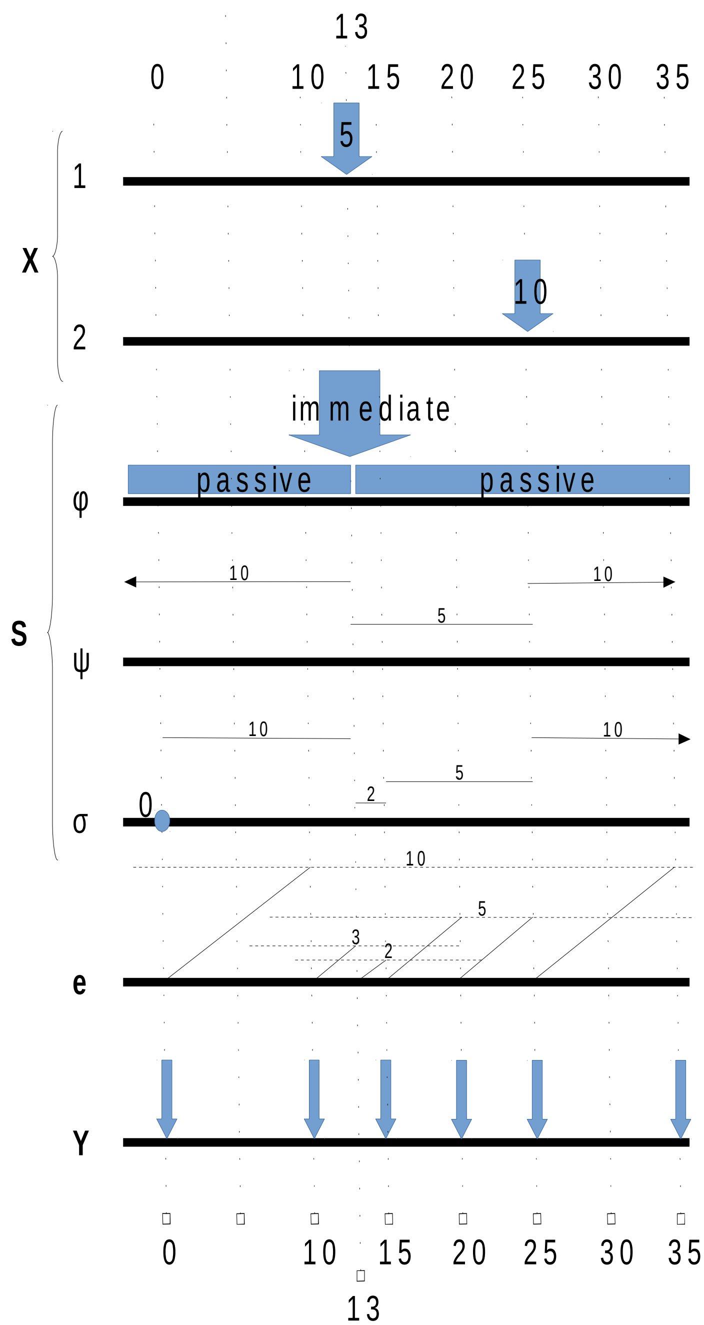
The initial state has: σ = 0; ψ = 10: and φ = "passive".

The formal DEVS model:

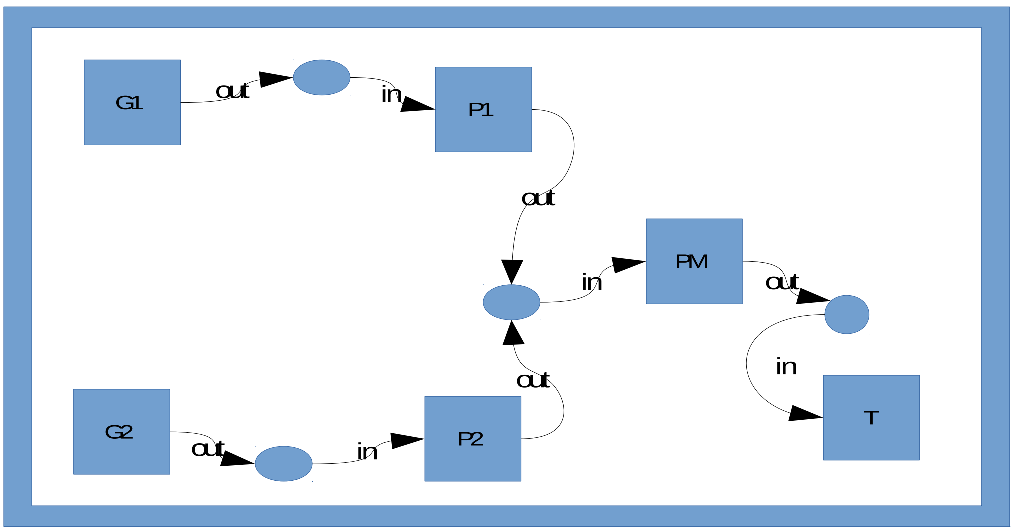
$$DEVS_{\mathrm{gen}} = [X_M, Y_M, S, \delta_{\mathrm{ext}}, \delta_{\mathrm{int}}, \delta_{\mathrm{con}}, \lambda, ta].$$

$$
\begin{aligned}
& X_M = \{[t, v]; t \in \{\mathrm{In1}, \mathrm{In2}\}; v \in \mathbb{R}_{pos}^{inf}\} \\
& Y_M = \{1\} \\
& S = \{[\varphi, \psi, \sigma]; \varphi \in \{\mathrm{passive}, \mathrm{immediate}\}; \psi \in \mathbb{R}_{\mathrm{pos}}^{inf}; \sigma \in \mathbb{R}_0^{inf}\}. \\
& \delta_{\mathrm{ext}}([\varphi, \psi, \sigma], e, [t, v]) \\
& \quad = [\mathrm{immediate}, v, v - e]; \quad \textit{if } t = \mathrm{In1} \wedge v > e; \\
& \quad = [\mathrm{immediate}, v, 0]; \quad \textit{if } t = \mathrm{In1} \wedge v \le e; \\
& \quad = [\mathrm{passive}, v, \sigma - e]; \quad \textit{otherwise}. \\
& \delta_{\mathrm{int}}([\varphi, \psi, \sigma]) = [\mathrm{passive}, \psi, \psi]. \\
& \delta_{\mathrm{con}}([\varphi, \psi, \sigma], e, [t, v]) = \delta_{\mathrm{ext}}(\delta_{\mathrm{int}}(\varphi), 0, \varphi). \\
& \lambda([\varphi, \psi, \sigma]) = 1. \\
& ta([\varphi, \psi, \sigma]) = \sigma.
\end{aligned}
$$

There is an ambiguity due to events which may occur at the same time, in the case of an internal and an external event the internal event is handled first. In the case of two external events the *confluent transition* function selects the “in” event over the “in1” event.


13
0 10 15 20 25 30 35
X
1
5
2
10
S
immediate
passive
passive
φ
10
10
5
ψ
10
10
5
2
0
σ
10
5
3
2
e
Y
0 10 15 20 25 30 35
13

**Problem 2:** A three-processor system with two classes of job arrivals. The size of the buffers at all processors are infinite. The jobs are periodic and the event times are as indicated in the model.




The formal devs model has a mechanism in the “Select” function which will use a reasonable mechanism when the “PM” processor has a job of each class. A round-robin approach would be reasonable.

$N_{\text{mpipe}} = \{X, Y, D, \{M_d | d \in D\}, EIC, EOC, IC, Select\}$.

$V =$ the set of allowed jobs and their states
$J = \{1, 2\}$ the types of jobs, the job classes
$InPorts = \{\text{in}\}; OutPorts = \{\text{out}\};$ possible port names

$X = \{(p, c, v) | \ p \in InPorts \wedge c \in J \wedge v \in V\};$ the set of input events representing the jobs
$Y = \{(p, c, v) | \ p \in OutPorts \wedge c \in J \wedge v \in V\};$ the set of output events representing the jobs
$D = \{G1, G2, P1, P2, M0, PM, T\};$ the names of the subcomponents

$M_{G1} = GEN(20); \quad M_{G2} = GEN(21);$ the job generators
$M_{P1} = BPROC(5); \quad M_{P2} = BPROC(7);$ the buffered job processors
$M_{PM} = BPROC(10); \quad T = SINK();$ the final job processor and sink
$EIC = \emptyset; \quad EOC = \emptyset;$ the generators and transducer are part of the model

$IC =$
$\{[(G1, \text{out}), (P1, \text{in})]$
$[(P1, \text{out}), (PM, \text{in1})]$
$[(G2, \text{out}), (P2, \text{in})]$
$[(P2, \text{out}), (PM, \text{in})]$
$[(PM, \text{out}), (T, \text{in})]\}$
$Select(D') =$ fair selection of a component ready to produce an event

This model makes use of several DEV models, some of which take parameters. Such as the following generator model.

Has a control variable "PERIOD".
$DEVS_{\text{gen}} = [X_M, Y_M, S, \delta_{\text{ext}}, \delta_{\text{int}}, \delta_{\text{con}}, \lambda, ta]$.

$X = \{\}$. takes no input
$Y = \{1\}$
$S = \{[\varphi, \sigma] \mid \varphi \in \{\text{passive}, \text{active}\}; \sigma \in \mathbb{R}_0^{inf}\}$.
$\delta_{\text{ext}}([\varphi, \sigma], e, x) = \emptyset$. no input so no input events
$\delta_{\text{int}}([\varphi, \sigma]) = [\text{active}, PERIOD]$.
$\lambda([\varphi, \sigma]) = 1$.
$ta([\varphi, \sigma]) = \sigma$.

The processor model also takes a parameter but as it has an infinite buffer no buffer length is required.

Has a control variable "PROC_TIME" which indicates the time to com
This processer can receive jobs while it is processing. It does this by in

$DEVS_{\text{buff\_proc}} = [X, Y, S, \delta_{\text{ext}}, \delta_{\text{int}}, \lambda, ta].$

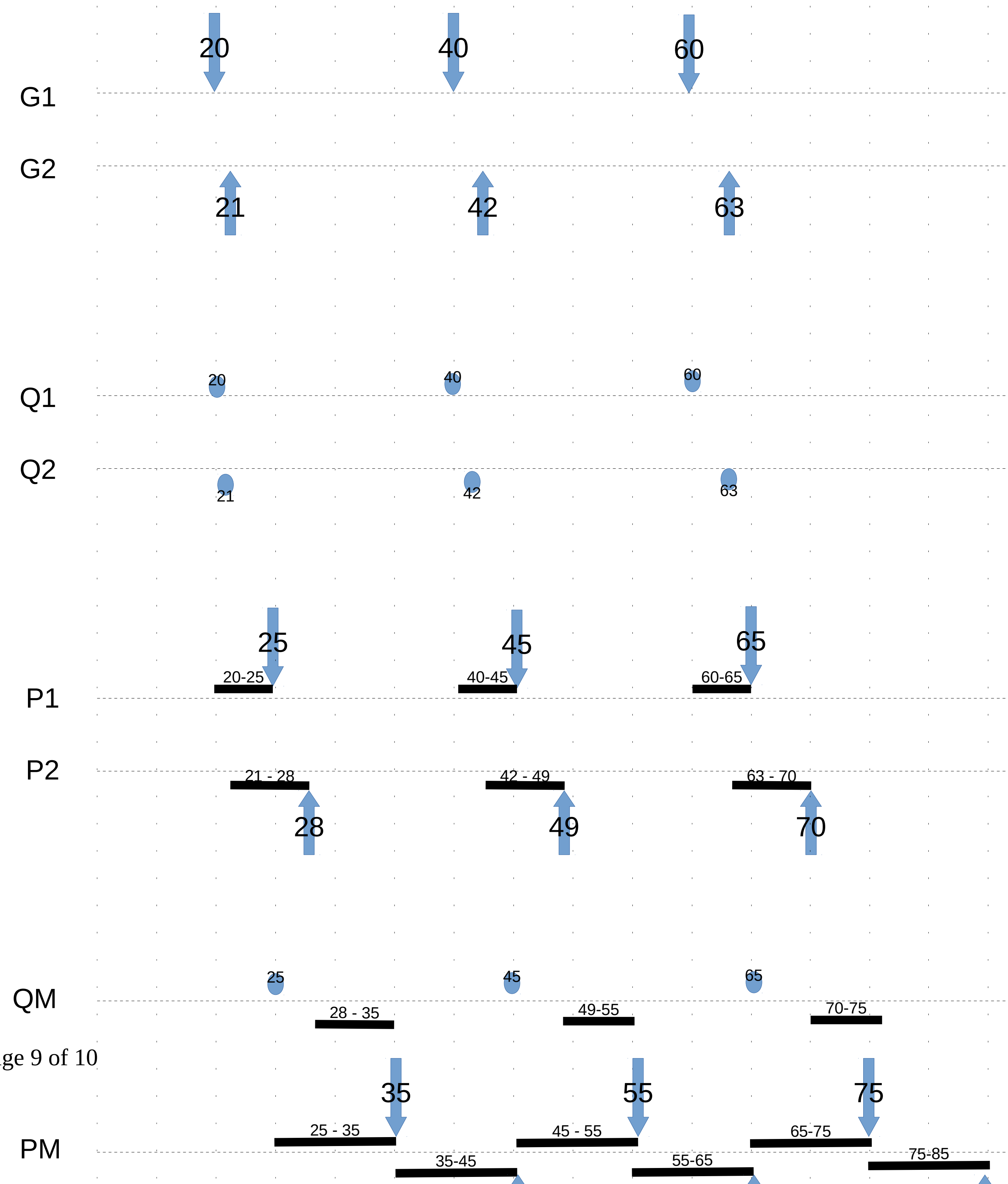
$X = \mathbb{R}.$ could be anything really
$Y = \mathbb{R}.$ could be anything really (yes that is a pun)
$S = \{[\varphi, \sigma, \psi,]; \varphi \in \{\text{passive}, \text{immediate}\}; \psi \in \mathbb{R}_{pos}^{inf}; \sigma \in \mathbb{R}_0^{inf}\}.$
$\delta_{\text{ext}}([\varphi, \psi, \sigma], e, [t, v])$
$= [\text{immediate}, v, v - e];\ \ if\ t = \text{In}1 \wedge v > e;$
$= [\text{immediate}, v, 0];\ \ if\ t = \text{In}1 \wedge v \le e;$
$= [\text{passive}, v, \sigma - e];\ \ otherwise$
$\delta_{\text{int}}([\varphi, \psi, \sigma]) = [\text{passive}, \psi, \psi].$
$\delta_{\text{con}}([\varphi, \psi, \sigma], e, [t, v]) = \delta_{\text{ext}}(\delta_{\text{int}}(\varphi), 0, \varphi).$
$\lambda([\varphi, \psi, \sigma]) = 1.$
$ta([\varphi, \psi, \sigma]) = \sigma.$

The transducer (sink) takes no parameters.

$$DEVS_{\text{transducer}} = [X_M, Y_M, S, \delta_{\text{ext}}, \delta_{\text{int}}, \lambda, ta].$$

$X = \{\text{anything}\}$
$Y = \emptyset$
$S = \{[\varphi] \mid \varphi \in \{\text{passive}\}.$
$\delta_{\text{ext}}([\varphi], e, x) = [\text{passive}]\};$
$\delta_{\text{int}}([\varphi]) = [\text{passive}].$
$\lambda([\varphi]) = 1.$
$ta([\varphi]) = \infty.$

b) The following timing diagram shows the first 6 events (3 of each class). Note that there is no initial event the first event comes after the generator period has expired. The figure shows Class-1 events above the line and Class-2 below the line.


20
40
60
G1
G2
21
42
63
20
40
60
Q1
Q2
21
42
63
25
45
65
20-25
40-45
60-65
P1
P2
21 - 28
42 - 49
63 - 70
28
49
70
25
45
65
QM
28 - 35
49-55
70-75
35
55
75
25 - 35
45 - 55
65-75
PM
35-45
55-65
75-85

c) This model has been implemented with Matlab/SimEvents using server blocks as the processor. The following figures are from that model. The job classes have been implemented the attribute JobClass, setting it to 1 for Class-1 and 2 for Class-2.