Proces-Interaction approach

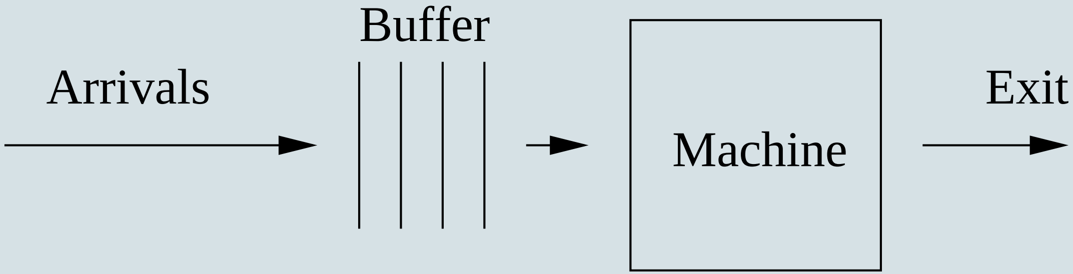
This approach focusses on describing *processes*;
In the event-scheduling approach one regards a simulation as executing a sequence of events ordered in time; but *no time elapses* within an event.

The process-interaction approach provides a process for *each entity* in the system; and *time elapses* during a process.

In production systems we have processes for:

- Arrivals
- Buffers
- Machines
- Exit

Example: Single-stage production system

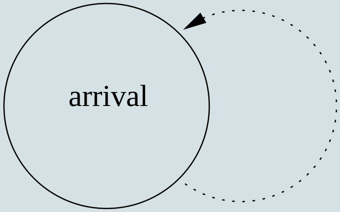


A single machine processes jobs in order of arrival. The interarrival times and processing times are exponential with parameters λ and μ (with $\lambda < \mu$).

- What is the mean waiting time?
- What is the mean queue length?
- What is the mean length of a busy period?
- How does the performance change if we speed up the machine?

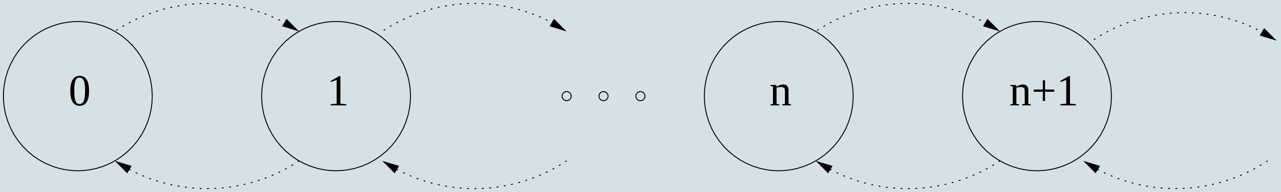
Arrival process

Generate arrival after random (exponential) time units



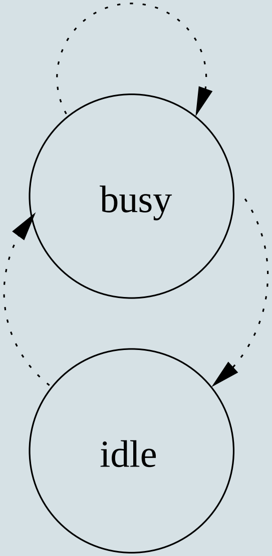
Buffer process

Add job to buffer and remove job from buffer (if there is any)



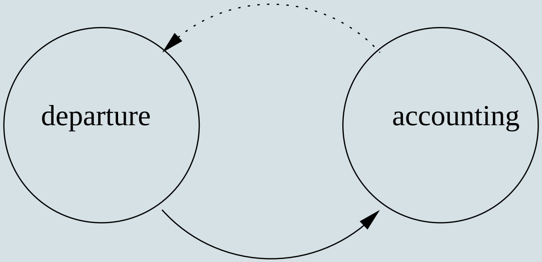
Machine process

Process job (if there is any)



Exit process

Accept completed job and do accounting



The specification language χ:

Modelling and simulation tool for the design of manufacturing systems

The language χ has been developed by the Systems Engineering group

For documentation, see http://se.wtb.tue.nl/documentation

Arrival process

```
type job=real

proc G(a: !job, ta: real) =
|[ u: -> real
 | u:=negexp(ta)
 ; *[ true -> a!time; delta sample u ]
]|
```

Buffer process

```
proc B(a: ?job, b:!job) =
|[ xs: job*, x: job
 | xs:=[]
 ; *[ true;      a?x       -> xs:= xs ++ [x]
    | len(xs)>0; b!hd(xs) -> xs:= tl(xs)
    ]
]|
```

Machine process

```
proc M(a: ?job, b: !job, te: real) =
|[ u: -> real, x: job
 | u:=negexp(te)
 ; *[ true -> a?x; delta sample u; b!x ]
]|
```

Exit process

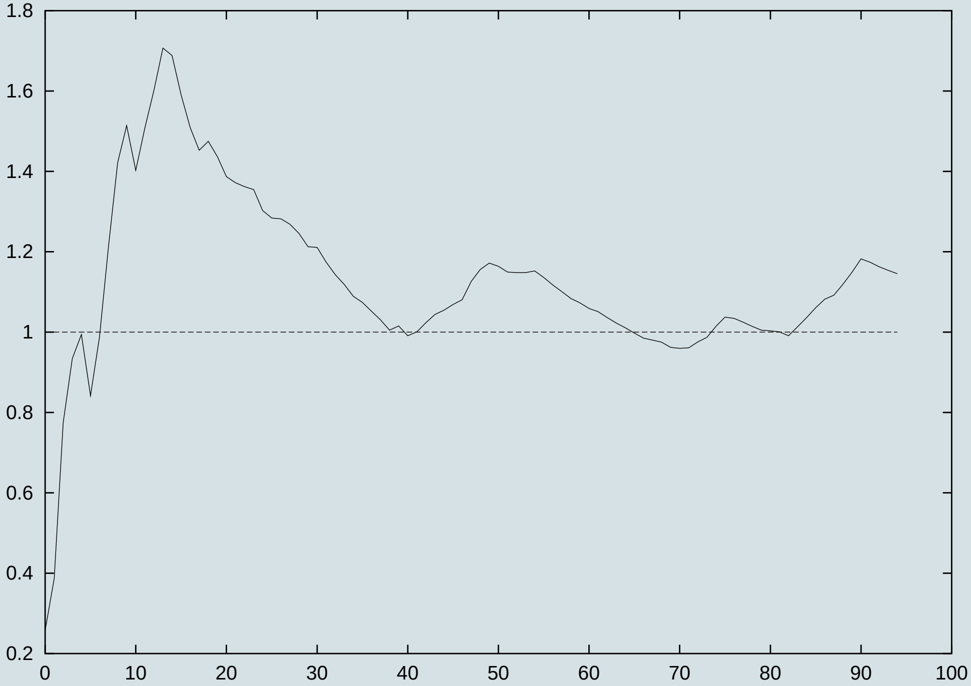
```
proc E(a: ?job) =
|[ ct,mct: real, n: nat, x: job
 | ct:= 0.0
 ; mct:= 0.0
 ; n:= 0
 ; *[ true -> a?x
            ; ct:= time - x
            ; n:= n + 1
            ; mct:= (n-1)/n*mct + ct/n
            ; !"Mean throughput time ", mct, nl()
    ]
]|
```

System and simulation experiment

```
syst S() =
|[ a,b,c: -job
 | G(a,1.0) || B(a,b) || M(b,c,0.5) || E(c)
]|

xper = |[ S() ]|
```

Complete χ code

```
type job=real

proc G(a: !job, ta: real) =
|[ u: -> real
 | u:=negexp(ta)
 ; *[ true -> a!time; delta sample u ]
]|

proc B(a: ?job, b: !job) =
|[ xs: job*, x: job
 | xs:=[]
 ; *[ true;      a?x      -> xs:= xs ++ [x]
    | len(xs)>0; b!hd(xs) -> xs:= tl(xs)
    ]
]|

proc M(a: ?job, b: !job, te: real) =
|[ u: -> real, x: job
 | u:=negexp(te)
 ; *[ true -> a?x; delta sample u; b!x ]
]|
proc E(a: ?job) =
|[ ct,mct: real, n: nat, x: job
 | ct:= 0.0
 ; mct:= 0.0
 ; n:= 0
 ; *[ true -> a?x
            ; ct:= time - x
            ; n:= n + 1
            ; mct:= (n-1)/n*mct + ct/n
            ; !"Mean throughput time ", mct, nl()
    ]
]|

syst S() =
|[ a,b,c: -job
 | G(a,1.0) || B(a,b) || M(b,c,0.5) || E(c)
]|

xper = |[ S() ]|
```



Mean throughput time as a function of the number of jobs processed for $\lambda = 1$ and $\mu = 2$

More examples...

Other interarrival and service time distributions

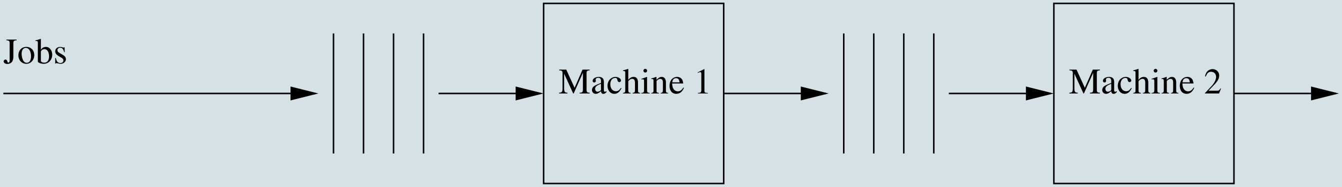
χ has a library available for sampling from distributions, e.g.,

- Bernouilli
- Binomial
- Poisson
- Beta
- Gamma
- Normal
- etc...

Example: Single-stage production system with three parallel machines
In the χ program we have to add channels to the buffer and exit process:

```
proc B(a: ?job, b,c,d: !job) =
|[ xs: job*, x: job
 | xs:=[]
 ; *[ true;      a?x      -> xs:= xs ++ [x]
    | len(xs)>0; b!hd(xs) -> xs:= tl(xs)
    | len(xs)>0; c!hd(xs) -> xs:= tl(xs)
    | len(xs)>0; d!hd(xs) -> xs:= tl(xs)
    ]
]|
proc E(a,b,c: ?job) =
|[ ct,mct: real, n: nat, x: job
 | ct:= 0.0
 ; mct:= 0.0
 ; n:= 0
 ; *[ true -> [ true; a?x -> skip
              | true; b?x -> skip
              | true; c?x -> skip
              ]
            ; ct:= time - x
            ; n:= n + 1
            ; mct:= (n-1)/n*mct + ct/n
            ; !"Mean throughput time ", mct, nl()
    ]
]|
syst S() =
|[ a,b,c,d,e,f,g: -job
 |                            M(b,e,0.5)
 || G(a,1.0) || B(a,b,c,d) || M(c,f,0.5) || E(e,f,g)
 ||                           M(d,g,0.5)
]|
```

Example: Two-stage production system



Jobs are processed by two machines in series. Each machine has its own local buffer and processes jobs in order of arrival. The interarrival and processing times of jobs are exponential with parameters λ, μ_1 and μ_2.

What is the mean (overall) throughput time?

In the χ program we only have to change the system:

```
syst S() =
|[ a,b,c,d,e: -job
 | G(a,1.0) || B(a,b) || M(b,c,0.5) || B(c,d) || M(d,e,0.5) || E(e)
]|
```

The simulation system Arena

In Arena you can construct simulation models without programming, but simply with click, drag and drop...

Student version of Arena is available in the Public Folders in Outlook; look in Software/Overig

Book with CD-ROM:

W. David Kelton, Randall P. Sadowski, Deborah A. Sadowski:
Simulation with Arena. 2nd ed., London: McGraw-Hill, 2002