

# Lab #2

Software Engineering

소프트웨어공학 연구실 김경륜

# Discrete-Event System Modeler and Simulator

DEVSJAVA modeling package provides

- Discrete-Event System Modeling and Simulation
- Hierarchical and modular modeling
- Plug-in approach (using Model Base)

## 2. DEVS basic model – Atomic, Coupled

---

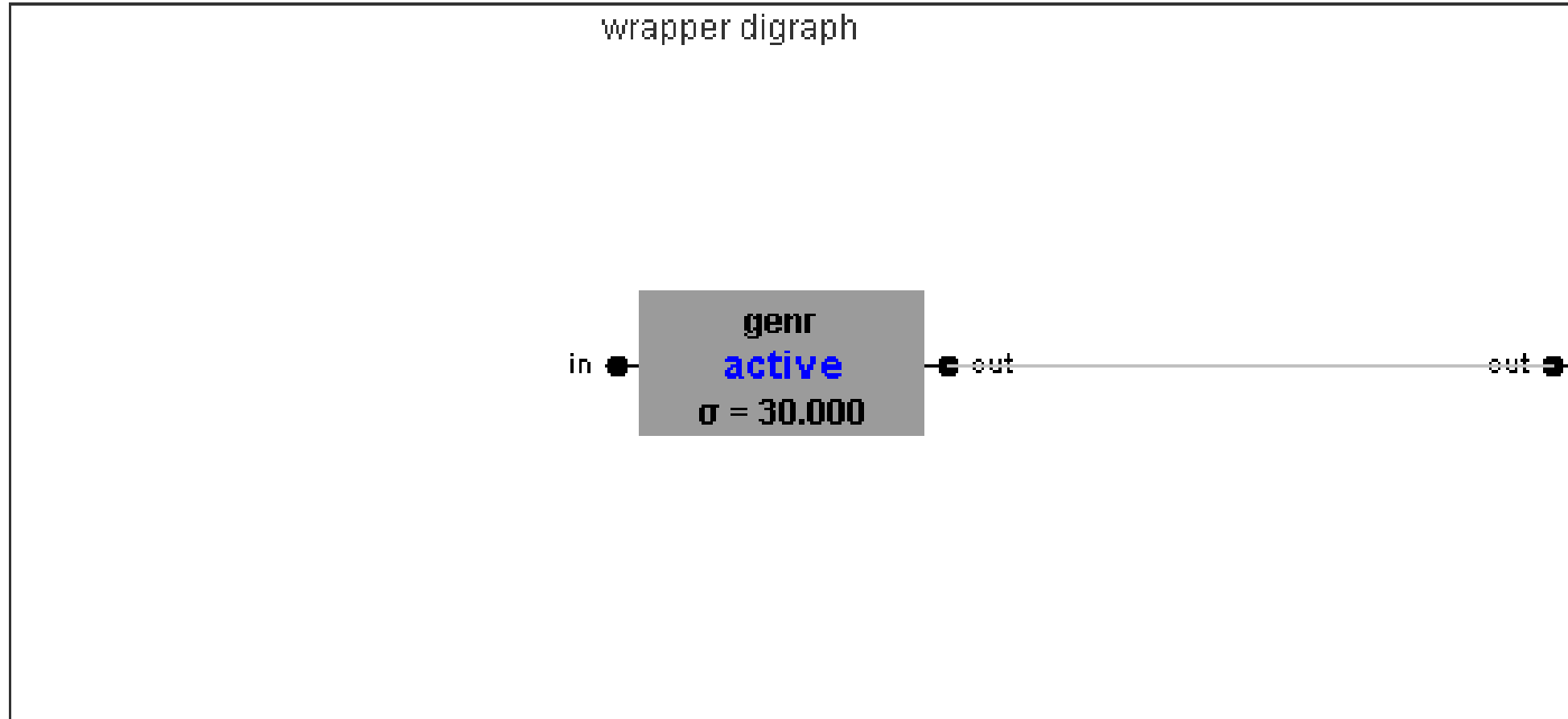
- **Atomic Model**

- Basic model for DEVS System
- Has input and output ports
- Has several states
- Consists of four major functions

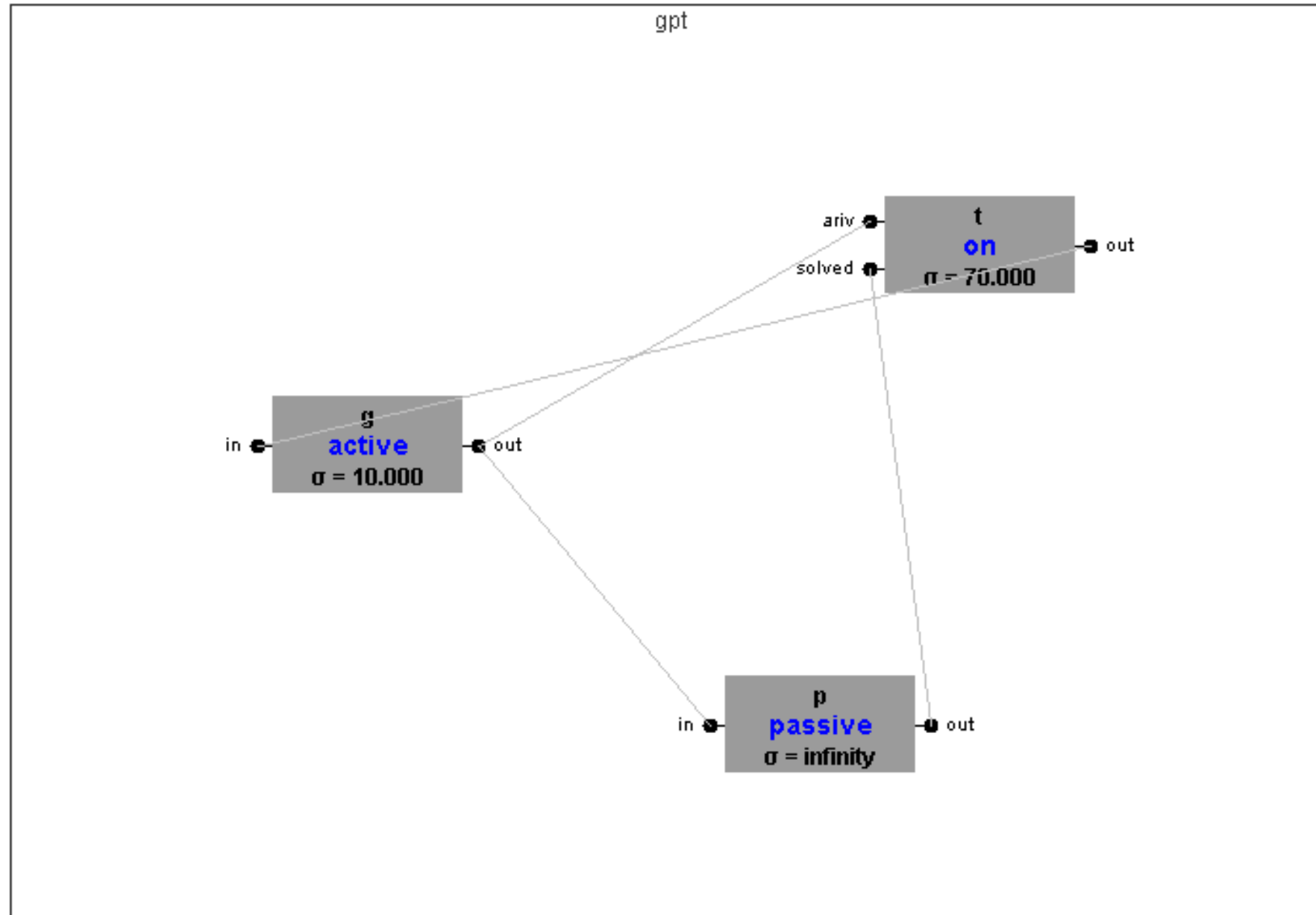
- **Coupled Model**

- Set of DEVS models
- Has coupling between DEVS model
- Can be a submodule for coupled model

## 2. DEVS basic model – Atomic model looks like...

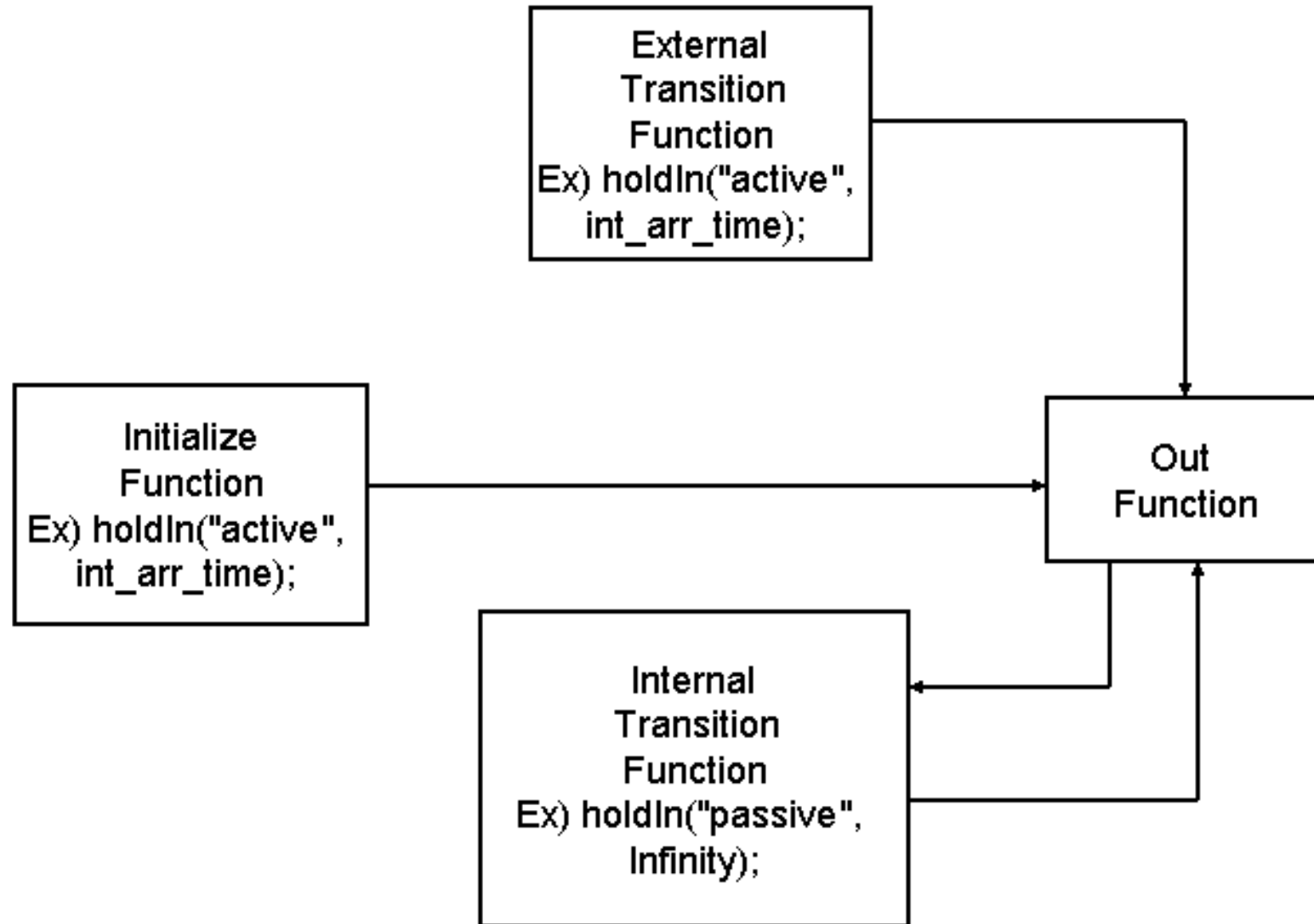


## 2. DEVS basic model – Coupled model looks like...



### 3. Functions – Transition in Atomic model

---



### 3. Functions – Functions in Atomic model

---

#### External Transition Function

- Handles external events (Receives messages from other DEVS atomic model)
- Does the scheduling with state and advanced time  
ex) `holdIn( " active " , int_arr_time)`

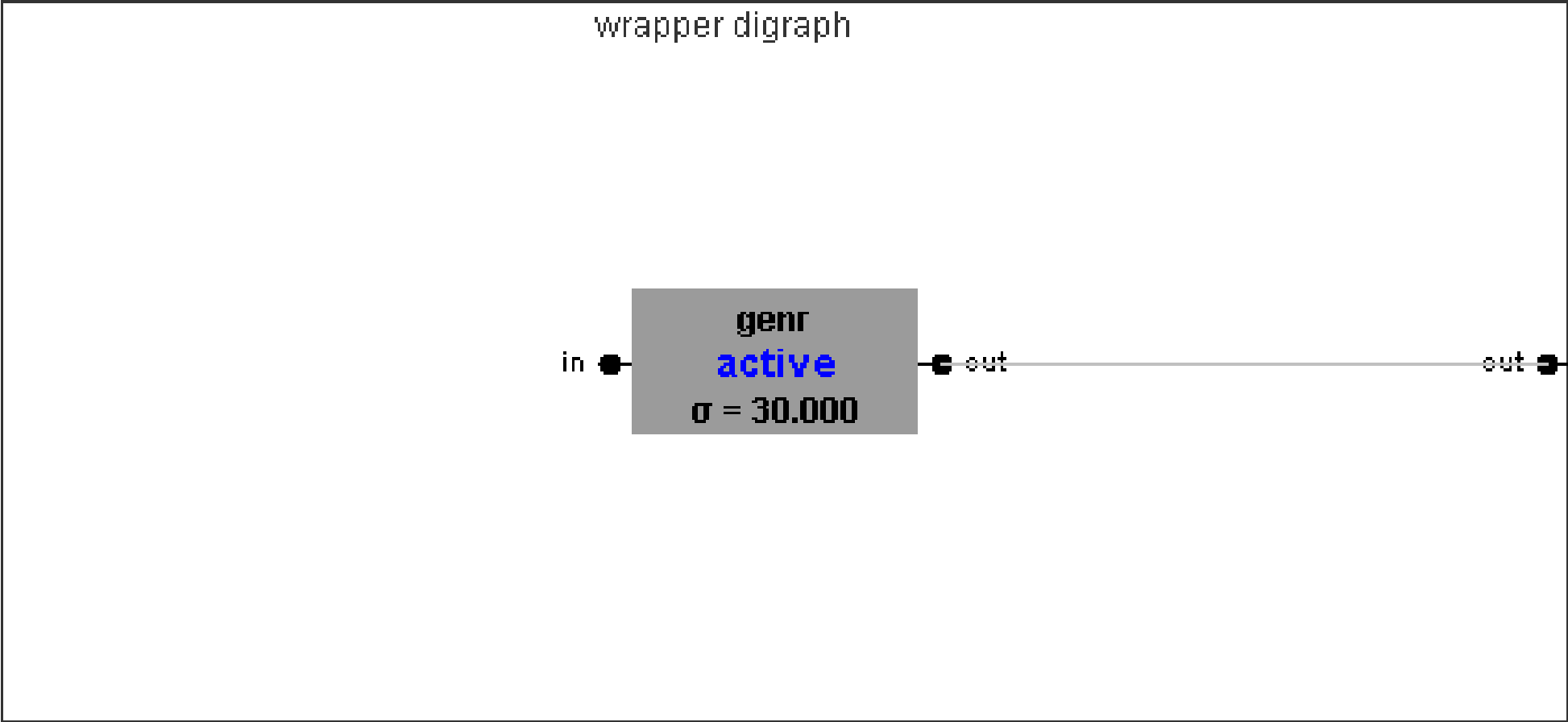
#### Internal Transition Function

- Handles internal events without external events
- Does the scheduling with state and advanced time  
ex) `holdIn( " active " , int_arr_time);`  
ex) `holdIn( " passive " , INFINITY);`

#### Output Function

- Handles output events (Sends messages to other DEVS atomic model)
- No scheduling, **Automatically call internal transition function**

# 4. Generator





## 4. Generator – Pseudo-code description

---

### Pseudo-code description for a generator

#### **Primary States:**

Phase: active

Sigma: any positive number

#### **Parameters:**

count: alpha-numeric (e.g. job-23) // Job-count

int\_arr\_time: any positive number // inter-arrival-time

**Input port:** stop, start

**Output port:** out

## 4. Generator – Pseudo-code description

---

### External Transition Function:

```
if phase is passive
    when receive input on input port "start"
        hold_in "active" for int_arr_time

if phase is active
    when receive input on input port "stop"
        set phase to "stop"
```

### Internal Transition Function:

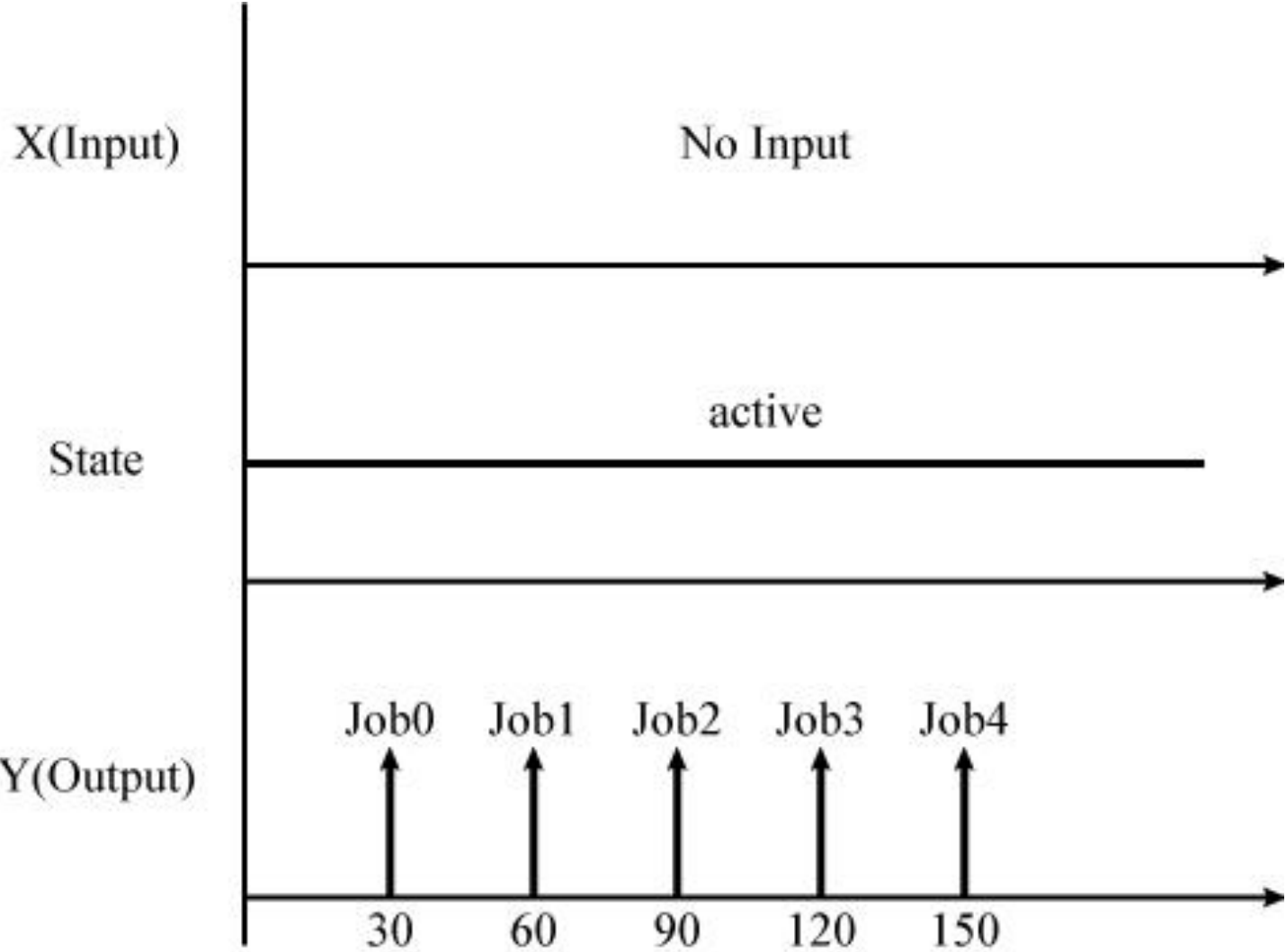
```
If phase is active
    count = count + 1
    set int_arr_time to 30
    hold_in "active" for int_arr_time
```

```
If phase is not active
    passivate ( set sigma to Infinity)
```

### Output Function:

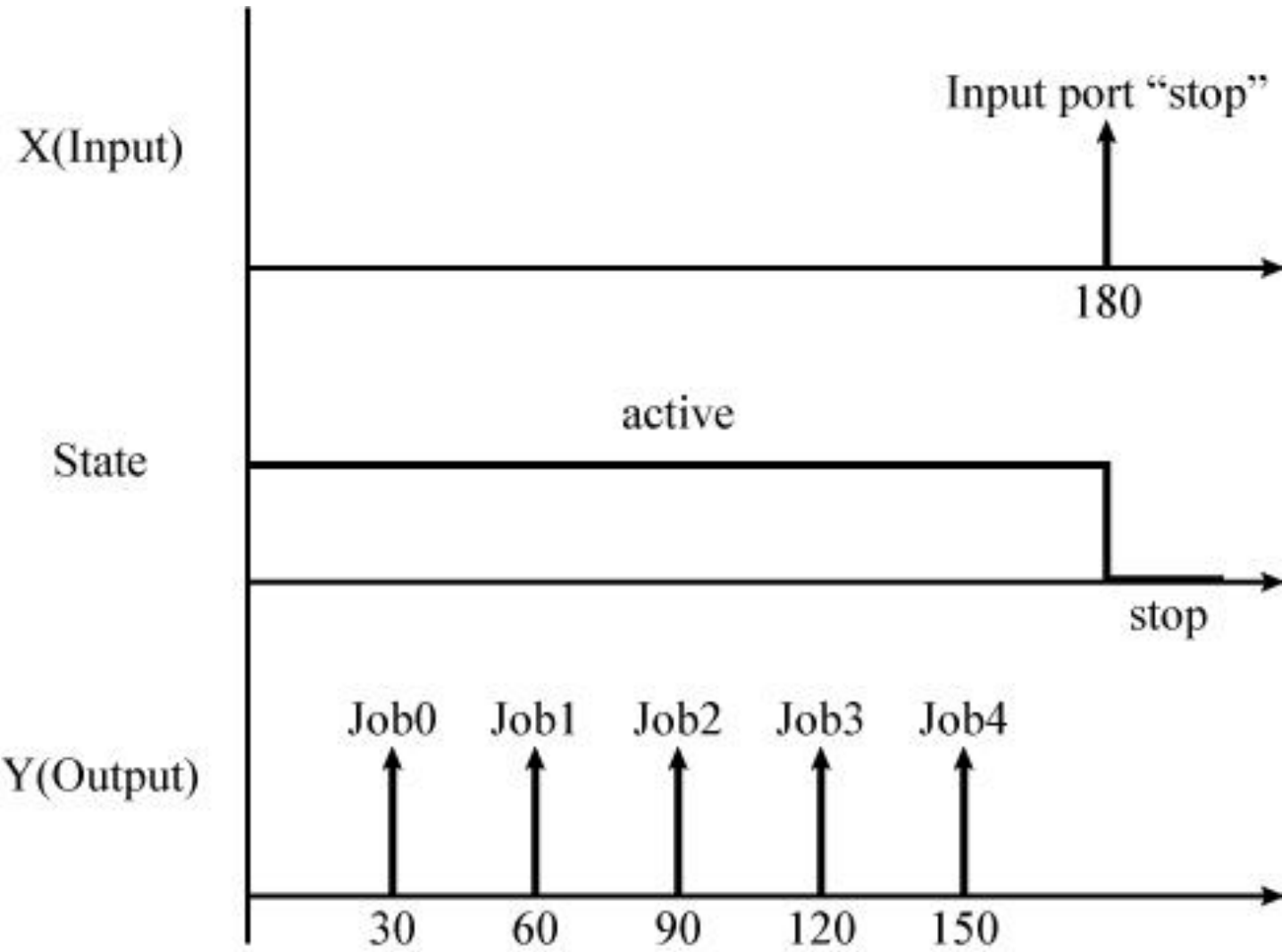
```
Send "job" + count to output port "out"
```

# 4. Generator – Trajectory



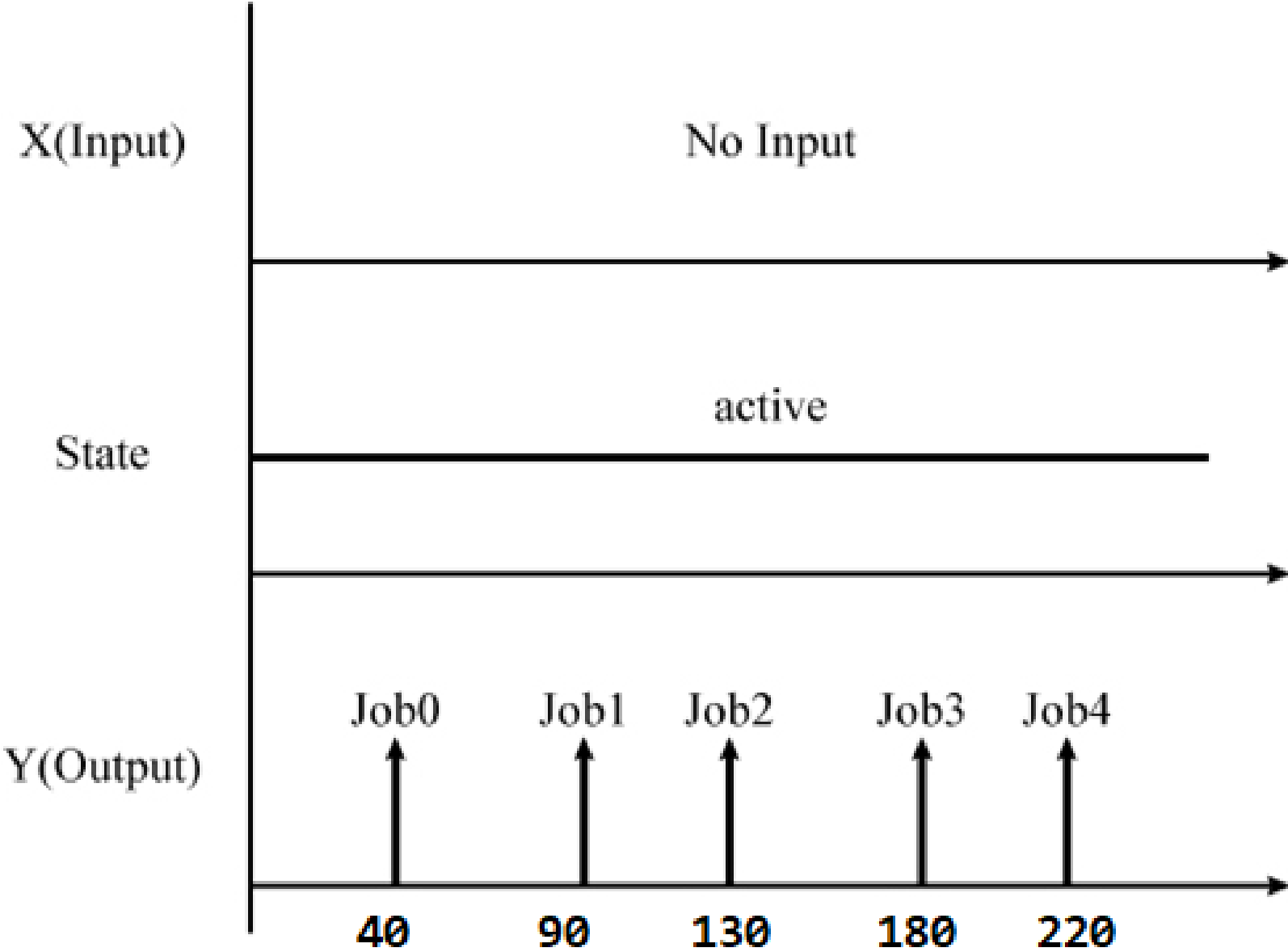
The trajectory of a generator  
without input

# 4. Generator – Trajectory



The trajectory of a generator  
with input

# 5. Assignment



## 5. Assignment (cont'd)

---

### 과제 – generator)

위 슬라이드의 수정된 generator의 trajectory를 보고 같은 trajectory를 그리는 generator를 작성하시오

### 과제 제출 유의사항

수정된 modified\_genr.java의 원본 코드

\* 코드에는 주석이 포함되어 있어야 함

수정한 부분의 코드와 설명(주석) 스크린샷

\* 코드를 수정한 부분의 주석은 필수

왜냐하면 정해진 답을 맞추는 것이 아닌, 본인이 구상한 같은 trajectory를 그리는 generator를 만드는 것이기 때문

위 2개를 압축하여 제출

제출 전 주석, 코드, 파일 이름의 인코딩이 올바른지 확인  
압축이 올바르게 잘 되었는지 확인

\* 위반 시 부분점수 없습니다  
\* 위반 시 부분점수 없습니다