Bee-up 강의

소프트웨어 순환공학 연구실
이성현

목차

I. Bee-up이란?

Bee-up

- Bee-up
  - ADOxx기반의 하이브리드 모델링 도구
  - 아래의 다섯 가지의 모델링 언어의 앞 글자를 따온 약어임
    - BPMN (Business Process Model and Notation)
    - EPC (Event-driven Process Chains)
    - ER (Entity Relationship Diagrams)
    - UML (Unified Modeling Language)
    - Petri Nets
  - 다섯 가지 모델 중 EPC와 ER을 제외한 BPMN, Petri Nets을 실습

II. 모델링 언어 소개

BPM이란?

- BPM (Business Process Modeling)
  - 업무 공정을 모델링 하기 위한 표기법
  - 현재 BPMN 2.0이 BPM 표기법의 표준임
- BPM의 특징
  - 공정의 흐름에 따른 모델링
  - 같은 공정에 속한 조직원들 간에 원활한 협업이 가능하도록 도와줌
  - 풍부하지만 직관적인 표기법으로 업무의 명확한 이해가 가능함

BPMN

- BPMN (Business Process Modeling Notation)
  - BPM의 표기법 정의
- 표기법의 종류
  - 활동 (Activity)
  - 이벤트 (Event)
  - 게이트웨이 (Gateway)
  - 업무 흐름 (Sequence flow)과 메시지 흐름 (Message flow)
  - 풀 (Pool)과 레인 (lane)

활동

- 업무 공정 안에서 실행되는 작업을 나타냄
- 활동 (Activity)의 종류
  - 작업 (Task)
    - 가장 최소한의 활동 단위를 의미함
    - ex) 홈페이지에 게시, 코드 컴파일, 항공편 예약, 코드 검토
  - 하위 공정 (Sub Process)
    - 복잡한 프로세스를 간략하게 표시하고자 할 때 사용함

이벤트

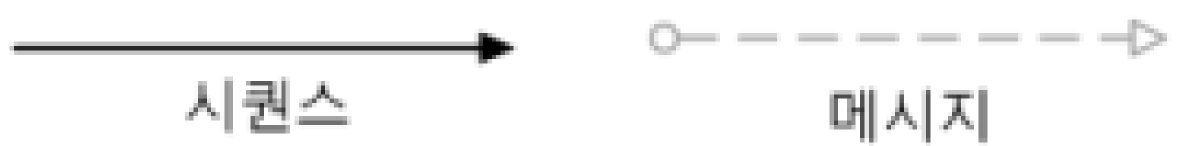
- 발생된 사건의 신호 (Signal)를 표현하기 위해 사용함
- 시작 (Start), 중간 (Immediate), 종료 (End) 이벤트가 있음
- 이벤트를 발생시키는 요소
  - 일반적인 시작, 종료
  - 메시지의 도착, 이메일이나 편지
  - 특정 시점의 도달(알람)
  - 특정 시간의 종료
  - 참으로 판명되는 조건
  - 오류 발생

게이트웨이

- 공정의 논리적인 흐름을 표현하고, 흐름을 분할 및 병합하는데 사용됨
- 게이트웨이의 종류
  - 배타적 (Exclusive) 게이트웨이
  - 포함 (Inclusive) 게이트웨이
  - 병렬 (Parallel) 게이트웨이
  - 양방향 (Complex) 게이트웨이

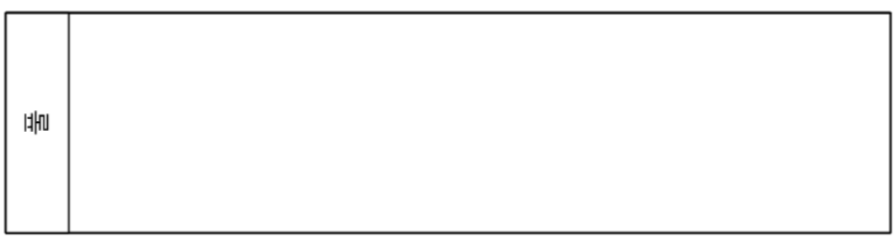
업무 흐름과 메시지 흐름

- 객체 사이를 연결하며, 객체의 흐름이나, 객체 간의 데이터 전달을 표시할 때 사용함
- 시퀀스 (흐름)
  - 주로 업무의 순서 관계를 나타낼 때 사용
- 메시지
  - 업무 간의 정보 전달에 사용



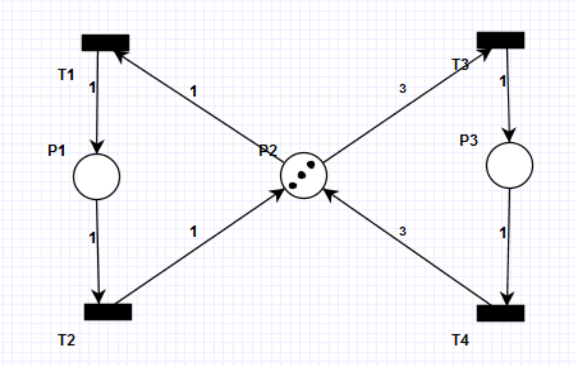
풀과 레인

- 프로세스의 흐름을 하나로 묶어, 묶인 프로세스의 유형을 나타냄
  - ex) 참가자 풀(role pool), 다이어그램 레인(diagram lane) 등
- 레인
  - 풀 안에 포함되며, 풀 안의 프로세스들의 역할 혹은 참가자에 따라서 풀을 나누는 역할을 함



Petri Nets

- 개요
  - Carl Adam Petrl의 박사 논문에서 처음 소개된 상태기계 이론
  - 병렬 시스템과 동적 이산 이벤트 시스템을 명세하고 분석하기 위해 고안됨
- 문법
  - 5개의 투플로 구성됨
    - $N = \{P, T, I, O, M_0\}$
    - 각 투플의 의미
      - P (Place): 조건, 자원의 사용가능성 또는 공정 상태 (circle)
        - 유한 집합 $P = \{p_1, p_2, \dots, p_n\}$
      - T (Transition): 사건 또는 공정의 시작과 끝 (bar)
        - 유한 집합 $T = \{t_1, t_2, \dots, t_n\}$
      - I (Input function): Place로부터 transition으로의 아크 (arrow)
      - O (Output function): Transition로부터 Place로의 아크 (arrow)
      - M (Marking): 각 place에 있는 토큰의 개수 (M_0: 초기 마킹)



Petri Nets 수학적 구조 표현

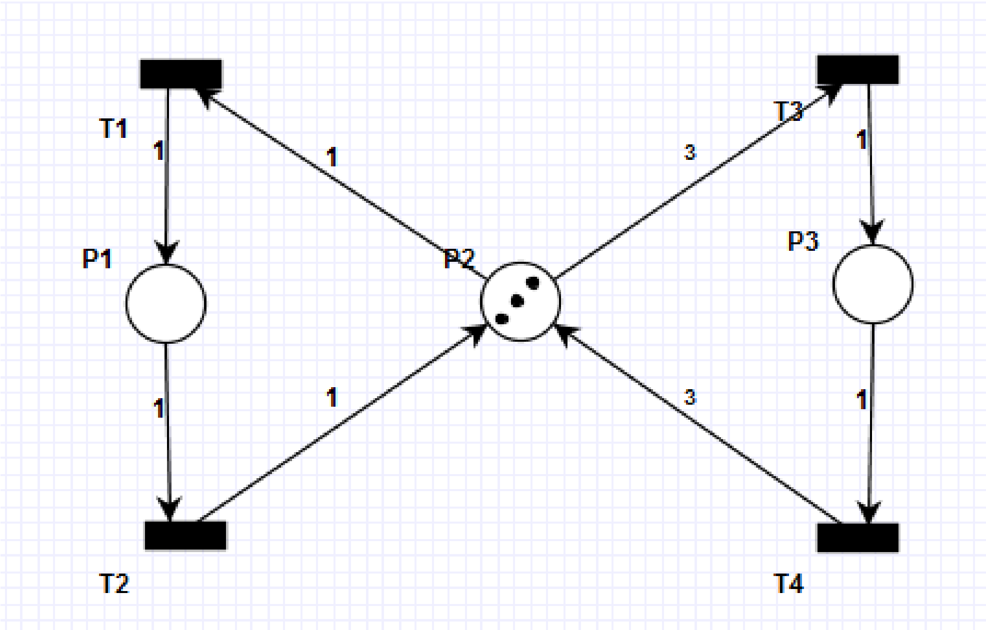
- Petri net은 matrix 구조로 나타낼 수 있음
  - $P = \{P1, P2, P3\}$
  - $T = \{T2, T2, T3, T4\}$
  - $I = \begin{matrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 3 & 0 \\ 0 & 0 & 0 & 1 \end{matrix}$
  - $O = \begin{matrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 3 \\ 0 & 0 & 1 & 0 \end{matrix}$
  - $M_0 = \begin{matrix} 0 \\ 3 \\ 0 \end{matrix}$

- Petri net은 matrix 구조로 나타낼 수 있음
  - $P = \{P1, P2, P3\}$
  - $T = \{T2, T2, T3, T4\}$
  - $I = \begin{matrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 3 & 0 \\ 0 & 0 & 0 & 1 \end{matrix}$
  - $O = \begin{matrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 3 \\ 0 & 0 & 1 & 0 \end{matrix}$
  - $M_0 = \begin{matrix} 0 \\ 3 \\ 0 \end{matrix}$

Input

Place에서 Transition으로 향하는 arc를 나타냄

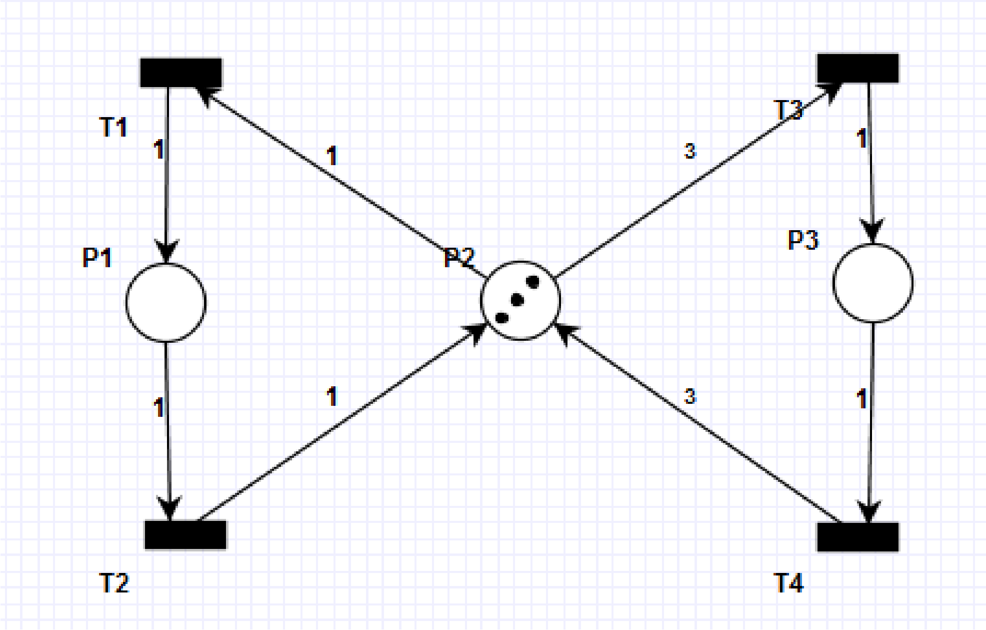
	T1	T2	T3	T4
P1	0	1	0	0
P2	1	0	3	0
P3	0	0	0	1



Output

Transition에서 Place로 향하는 arc를 나타냄

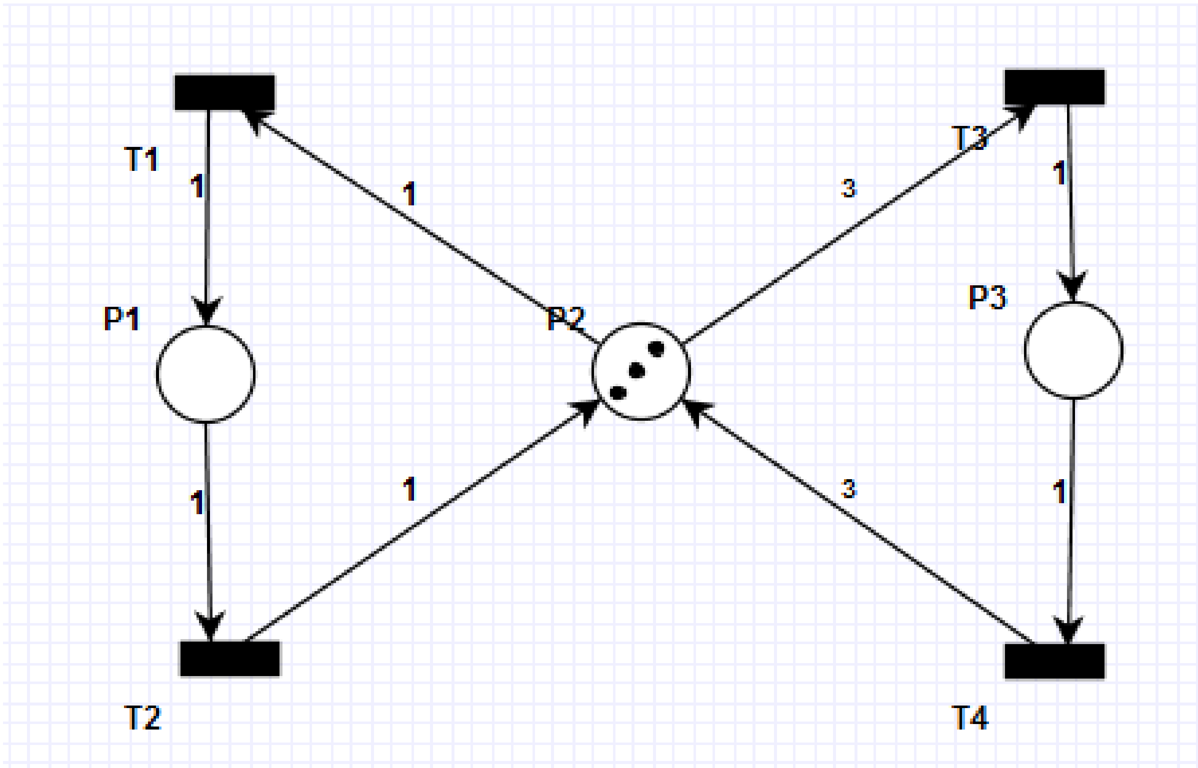
	T1	T2	T3	T4
P1	1	0	0	0
P2	0	1	0	3
P3	0	0	1	0



Marking

각 Place에 포함된 token의 수를 나타냄

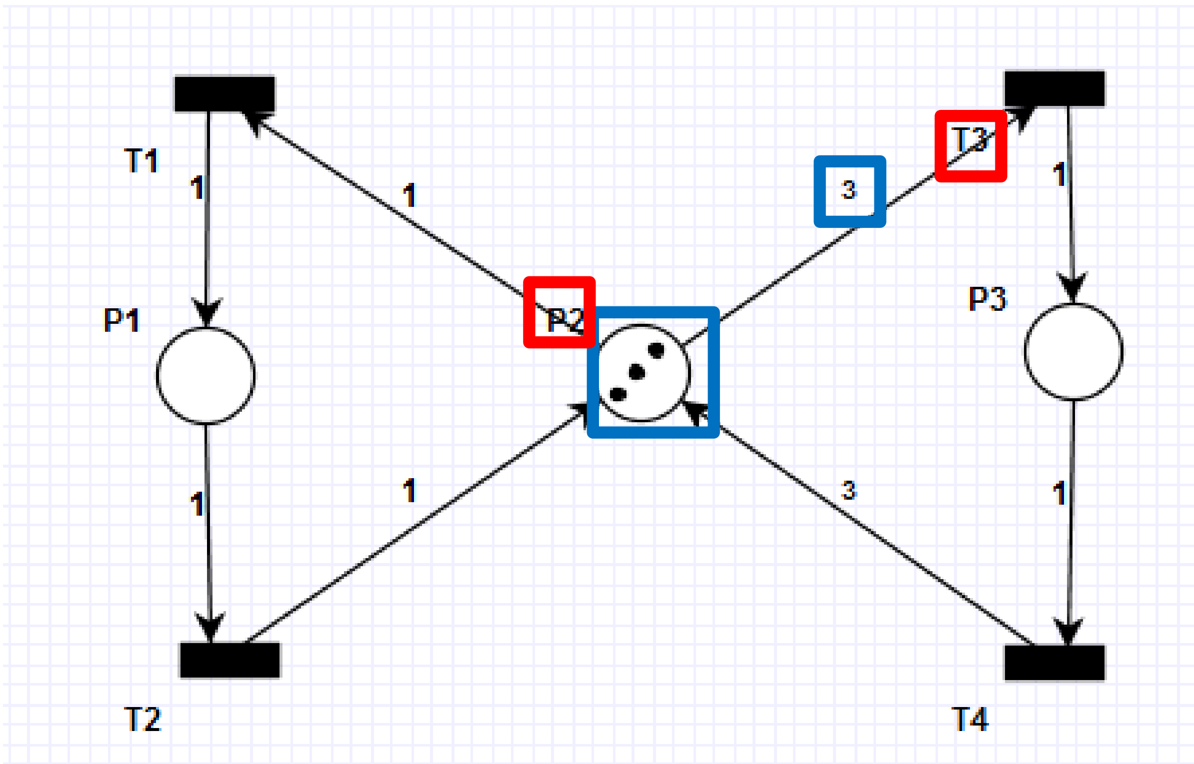
	M
P1	0
P2	3
P3	0



Marking

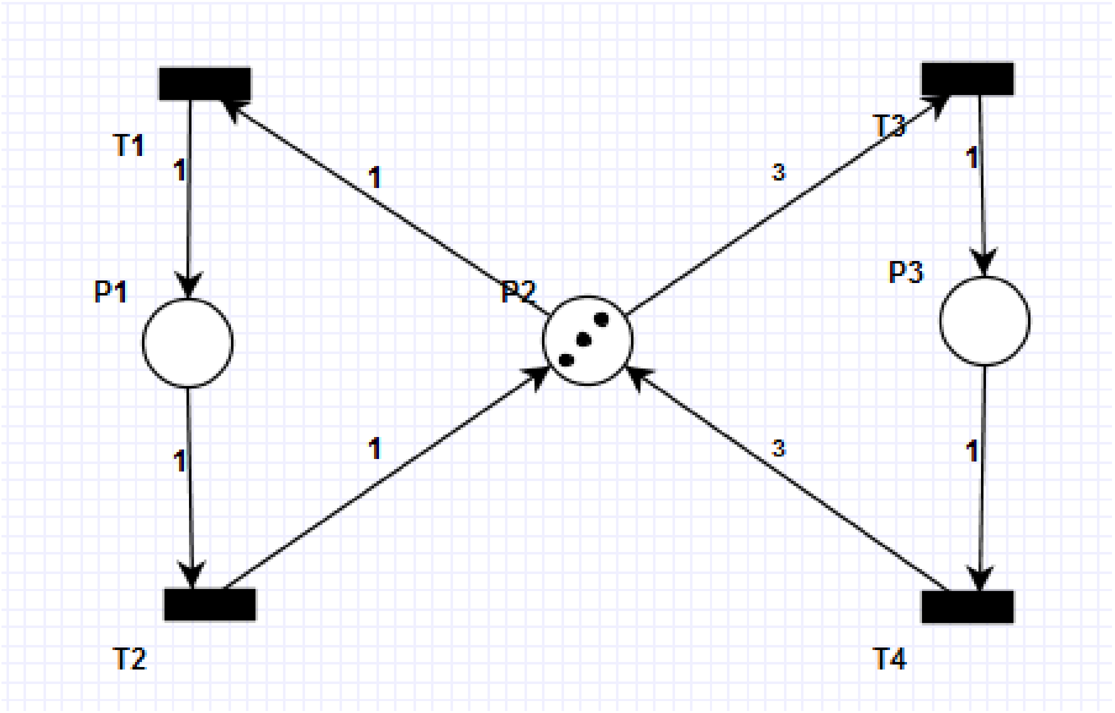
- T3가 발생한 경우 Token 변화 예시
  - $M = \begin{matrix}0\\3\\0\end{matrix} \xrightarrow{T2} \begin{matrix}0\\0\\1\end{matrix}$

	M
P1	0
P2	0
P3	1



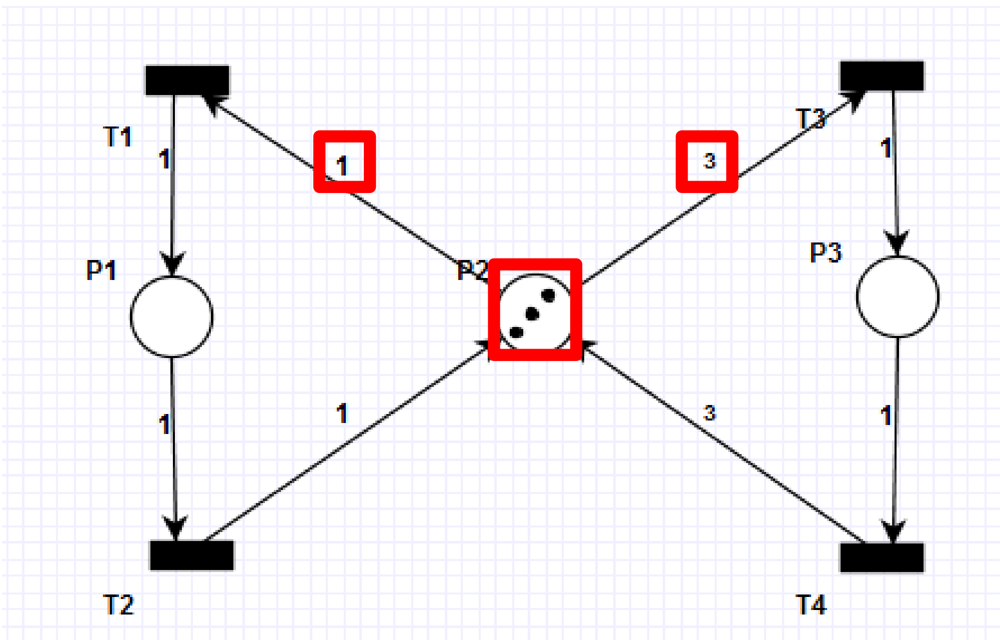
Incidence Matrix

- $C = Output - Input$
- $C = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 3 \\ 0 & 0 & 1 & 0 \end{bmatrix} - \begin{bmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 3 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & -1 & 0 & 0 \\ -1 & 1 & -3 & 3 \\ 0 & 0 & 1 & -1 \end{bmatrix}$



Semantics

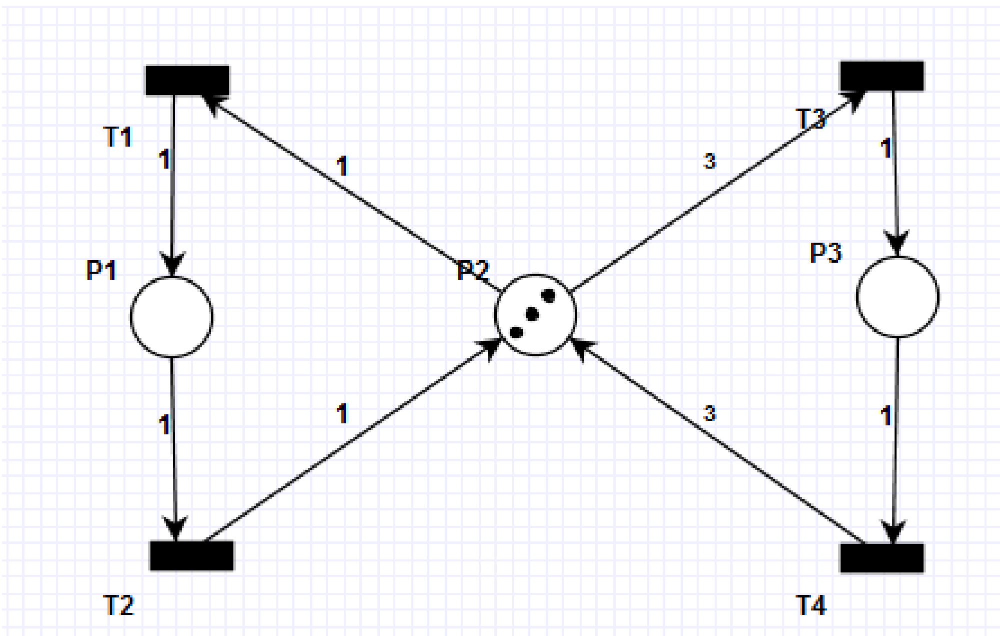
- Transition
  - [정의] Petri net $N = \{P, T, I, O, M_0\}$의 transition t는 현재 M에 대하여 $M \geq I(\cdot, t)$인 경우에만 활성화된다.
  - 예) $M_0 = \begin{matrix} 0 \\ 3 \\ 0 \end{matrix}$이고, $\mathrm{Input}(\cdot, T1) = \begin{matrix} 0 \\ 1, \\ 0 \end{matrix}$ $\mathrm{Input}(\cdot, T3) = \begin{matrix} 0 \\ 3, \\ 0 \end{matrix}$ 이므로 M_0인 상태에서는 T1과 T3만 활성화된다.



OMLAB KOREA

Semantics

- Transition
  - Transition이 $M \xrightarrow{t} M'$이면 M'은 다음과 같이 변화한다:
    - $M' = M + Output(\cdot, t) - Input(\cdot, t) = M' = M + C(\cdot, t)$
    - T1 발생 시: $\begin{bmatrix}1\\2\\0\end{bmatrix} = \begin{bmatrix}0\\3\\0\end{bmatrix} + \begin{bmatrix}1\\0\\0\end{bmatrix} - \begin{bmatrix}0\\1\\0\end{bmatrix}$



OMLAB KOREA

Semantics

- Reachability
  - [정의] Petri net N의 M과 $t \in T$에 대하여 $M \xrightarrow{t} M'$ 조건을 만족할 때에만 M'은 도달 가능하다.
    - 즉 M_0로부터 특정한 transition을 거쳐 M'이 된다면 M'은 도달 가능하다고 할 수 있다.
  - [정의] $s = t_1 t_2 \ldots t_k$를 Petri net N의 transition의 순서라 하면 s는 다음의 조건을 만족하는 경우에만 M_1을 발생한다:
    - Transition t_1에 대하여 $M_1 \xrightarrow{t_1} M_2$가 성립한다.
    - $s' = t_2 t_3 \ldots t_k$와 M_2에 대하여 재귀적으로 성립한다.

Semantics

- Reachability
  - 즉, M_1과 s에 대하여 $M_1 \xrightarrow{s} M_{k+1}$관계가 성립한다. s를 vector로 표현하기 위해 다음과 같이 정의한다.
  - [정의] Petri net N의 s를 vector로 표현하기 위한 함수 $\bar{s}$는 다음과 같다.
    - $\bar{s}: T \rightarrow N$
    - $\bar{s}(t)$는 transition t가 s에서 나타난 횟수를 의미한다.
    - s가 $T1T1T2$ 인 경우, $\overline{T1T1T2} = \begin{bmatrix} 2 \\ 1 \\ 0 \\ 0 \end{bmatrix}$
    - $M_{k+1} = M_1 + C \cdot \bar{s}$

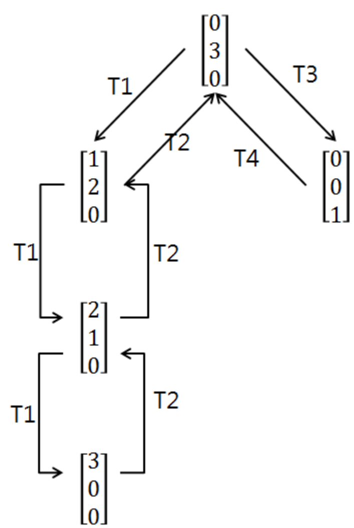
OMLAB KOREA

Semantics

- Reachability
  - [정의] Petri net N의 M과 transition의 순서 s 에 대하여 $M_1 \xrightarrow{s} M'$조건을 만족할 때에만 M'은 도달 가능하다
    - 예) $\mathrm{M} = \begin{bmatrix} 3 \\ 0 \\ 0 \end{bmatrix}$ 은 다음의 식에 따라 도달 가능함.
      - $\begin{bmatrix} 0 \\ 3 \\ 0 \end{bmatrix} \xrightarrow{T1T1T1} \begin{bmatrix} 3 \\ 0 \\ 0 \end{bmatrix}$

Semantics

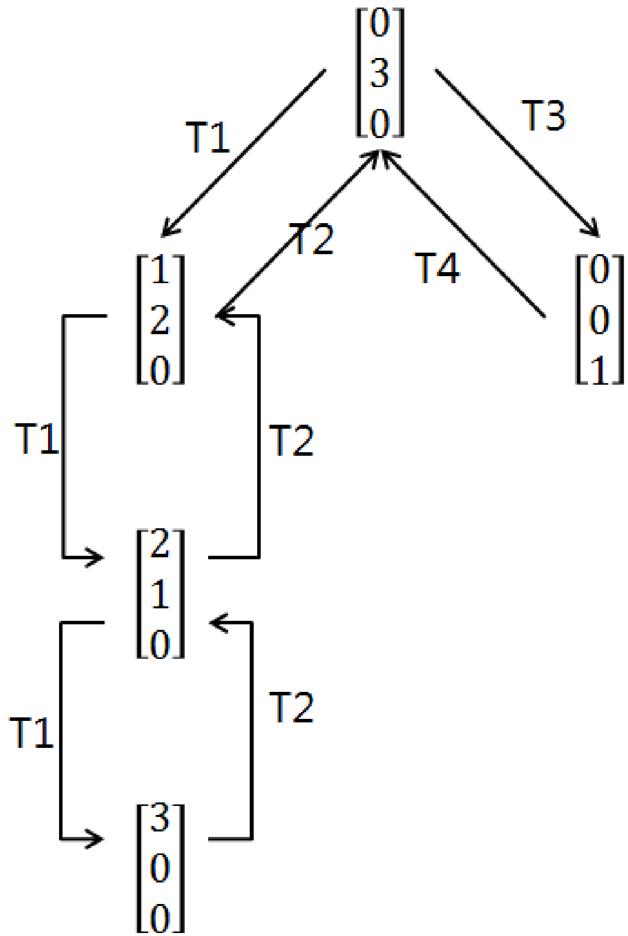
- Reachability
  - [정의] Petri net $N = \{P, T, I, O, M_0\}$의 reachability set은 $\Re(N, M_0)$이며 $\left(\mathrm{M} \in \Re(N, M_0)\right) \Leftrightarrow (\exists s M_0 \xrightarrow{s} M)$을 만족한다.
  - [정의] Petri net $N = \{P, T, I, O, M_0\}$의 reachability graph는 다음의 조건을 만족한다:
    - Node는 $\Re(N, M_0)$의 원소이다.
    - Transition $t \in T$에 대하여 $M \xrightarrow{t} M'$조건을 만족하는 경우에만 M에서 M'방향으로 연결된다.



OMiLAB KOREA

Semantics

- Reachability graph



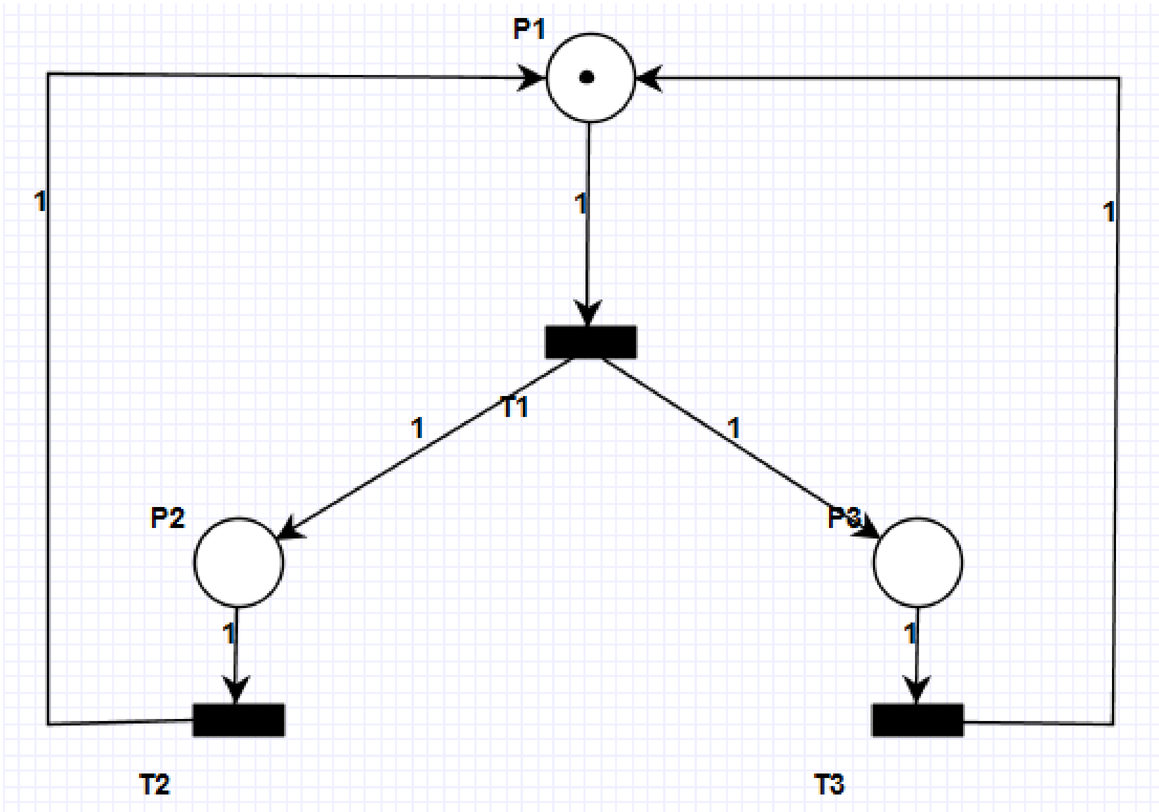
Properties

- Boundedness
  - [정의] Petri net $N = \{P, T, I, O, M_0\}$이 다음의 조건을 만족하는 경우에만 k-bounded라고 한다:
    - $\forall M \in \Re(N, M_0)\ and\ \forall p \in P\ then\ M(p) \leq k(k \in N\ and\ k \neq 0)$
  - k-bounded: 모든 place가 최대 k개까지의 token을 지닐 수 있음
    - 1-bounded인 경우 safe Petri net이라 함
  - Safe Petri net의 경우 token을 통해 참, 거짓 조건을 표현할 수 있음
    - 토큰이 1개: 참
    - 토큰이 0개: 거짓
  - Petri net이 un-bounded하다면 token이 무한하게 증가함. 이러한 시스템은 검증이 불가능함
- Liveness
- Reversibility

OMiLAB KOREA

Properties

- Un-Bounded한 경우
  - T1, T2가 반복된다면, T3에 token이 무한히 쌓임



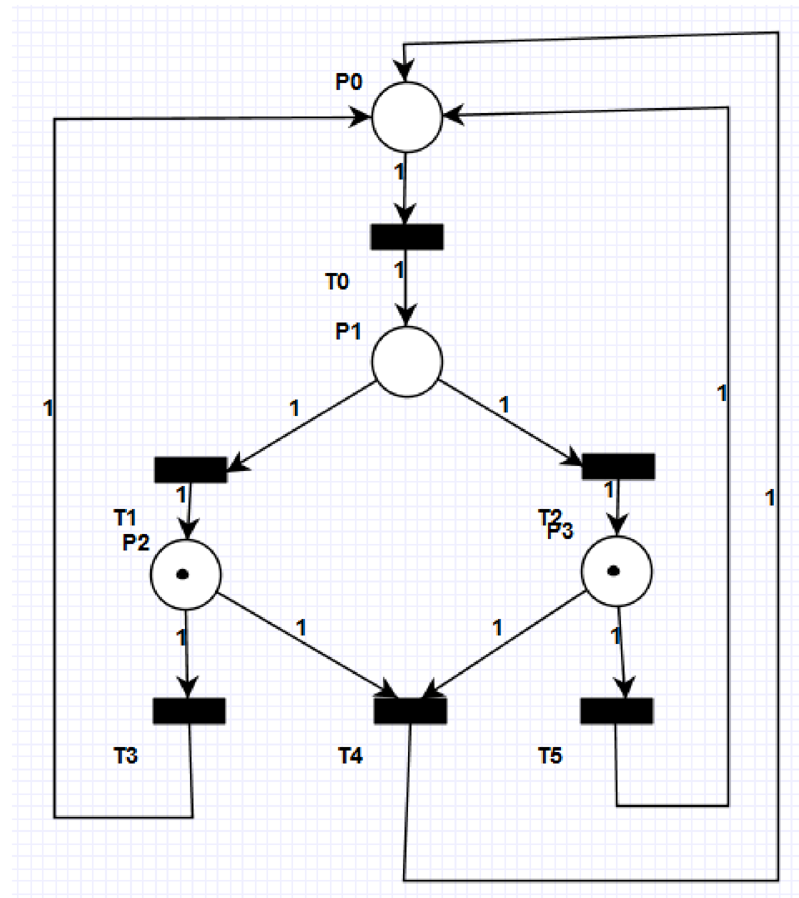
OMLAB KOREA

Properties

- Boundedness
- Liveness
  - [정의] Petri net $N = \{P, T, I, O, M_0\}$이 transition t에 대하여 다음의 조건을 만족하는 경우 transition t는 live하다고 함:
    - $\forall M \in \Re(N, M_0), \exists s \; such \; that \; M \xrightarrow{s} M' and \; M' \xrightarrow{t} M''$
  - Transition t가 live하다면 Petri net N이 어떠한 상태에 있더라도 transition t를 발생시킬 수 있는 상태로 전이될 수 있다는 것을 의미함
  - [정의] Petri net $N = \{P, T, I, O, M_0\}$의 모든 transition이 live한 경우 Petri net N은 live함
    - Deadlock이 발생하지 않음을 보장
    - 위의 두 정의를 만족하는 Petri net은 어느 transition에 있더라도 적어도 하나의 transition이 발생할 수 있음
- Reversibility

Properties

- Liveness 속성은 deadlock 방지보다 더욱 강력한 제약 조건
  - 특정 Petri net에서 deadlock이 전혀 발생하지 않더라도 liveness 속성을 만족하지 못할 수 있음
- 예)
  - 1) T4 T0로 전이
  - 2) P1에 토큰이 하나 존재
  - 3) 두 가지 경로의 전이만 가능함
    - 3-1) T1 T3 T0
    - 3-2) T2 T5 T0
  - 즉, T4가 한 번 발생한 후에는 더 이상 T4를 발생시킬 수 있는 상태로 전이될 수 없음
    - Live하지 않지만 deadlock은 없음



OMLAB KOREA

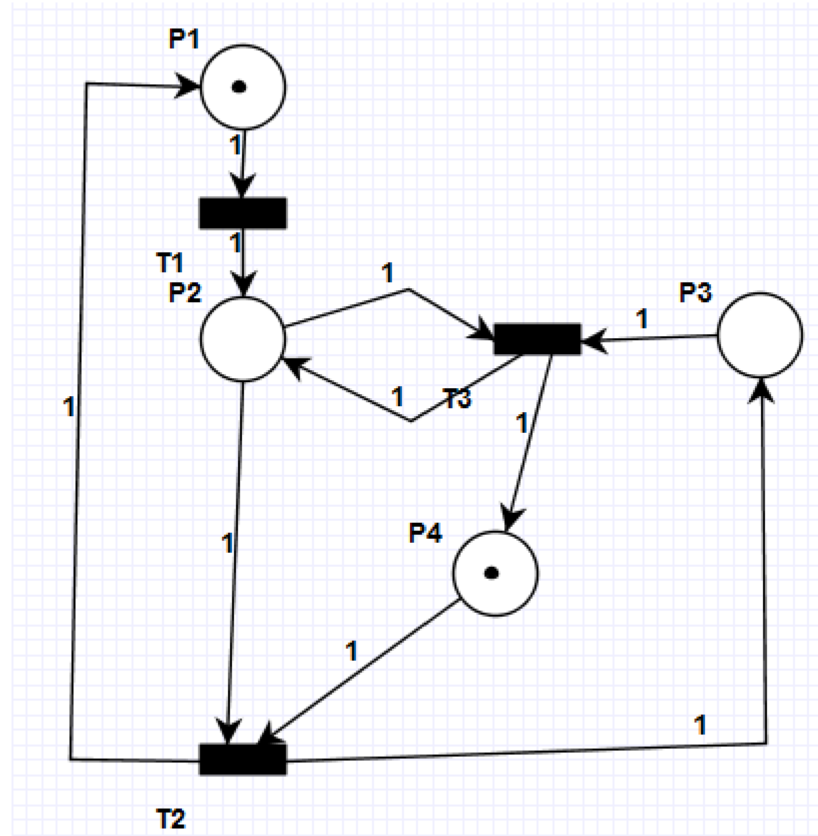
Properties

- Boundedness
- Liveness
- Reversibility
  - [정의] Petri net $N = \{P, T, I, O, M_0\}$이 다음 조건을 만족하는 경우에만 reversible함:
    - $\forall M \in \mathfrak{R}(N, M_0), \exists s \text{ such that } M \xrightarrow{s} M_0$
  - 대부분의 시스템이 주기적으로 작동하기 때문에 일반적으로 reversible 속성이 요구됨
  - Liveness와 reversibility 속성은 비슷해 보이지만 두 속성 간에는 연관관계가 없음

OMiLAB KOREA

Properties

- Liveness 속성은 만족하지만 reversibility 속성은 만족하지 못하는 예
  - 1) $M_0 = \begin{bmatrix}1\\0\\0\\1\end{bmatrix}$ 상태에서만 T1 발생
  - 2) $M_1 = \begin{bmatrix}0\\1\\0\\1\end{bmatrix}$ 상태에서 T2 발생
  - 3) $M_2 = \begin{bmatrix}1\\0\\1\\0\end{bmatrix}$ 상태에서 T1 발생
  - 4) $M_3 = \begin{bmatrix}0\\1\\1\\0\end{bmatrix}$ 상태에서 T3 발생하여 M_1발생
  - $M_0 \xrightarrow{T1} M_1 \xrightarrow{T2} M_2 \xrightarrow{T1} M_3 \xrightarrow{T3} M_1 \xrightarrow{T2} M_2 \xrightarrow{T1} M_3 \xrightarrow{T3} \ldots$
  - Liveness 조건을 만족하지만 M_0상태로 돌아올 수 없음



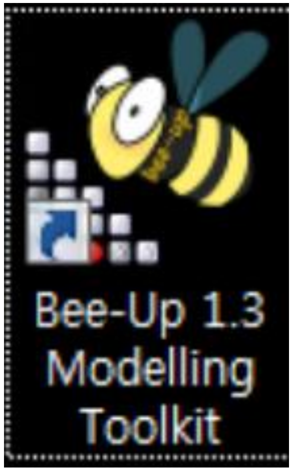
III. 실습

Bee-up 설치

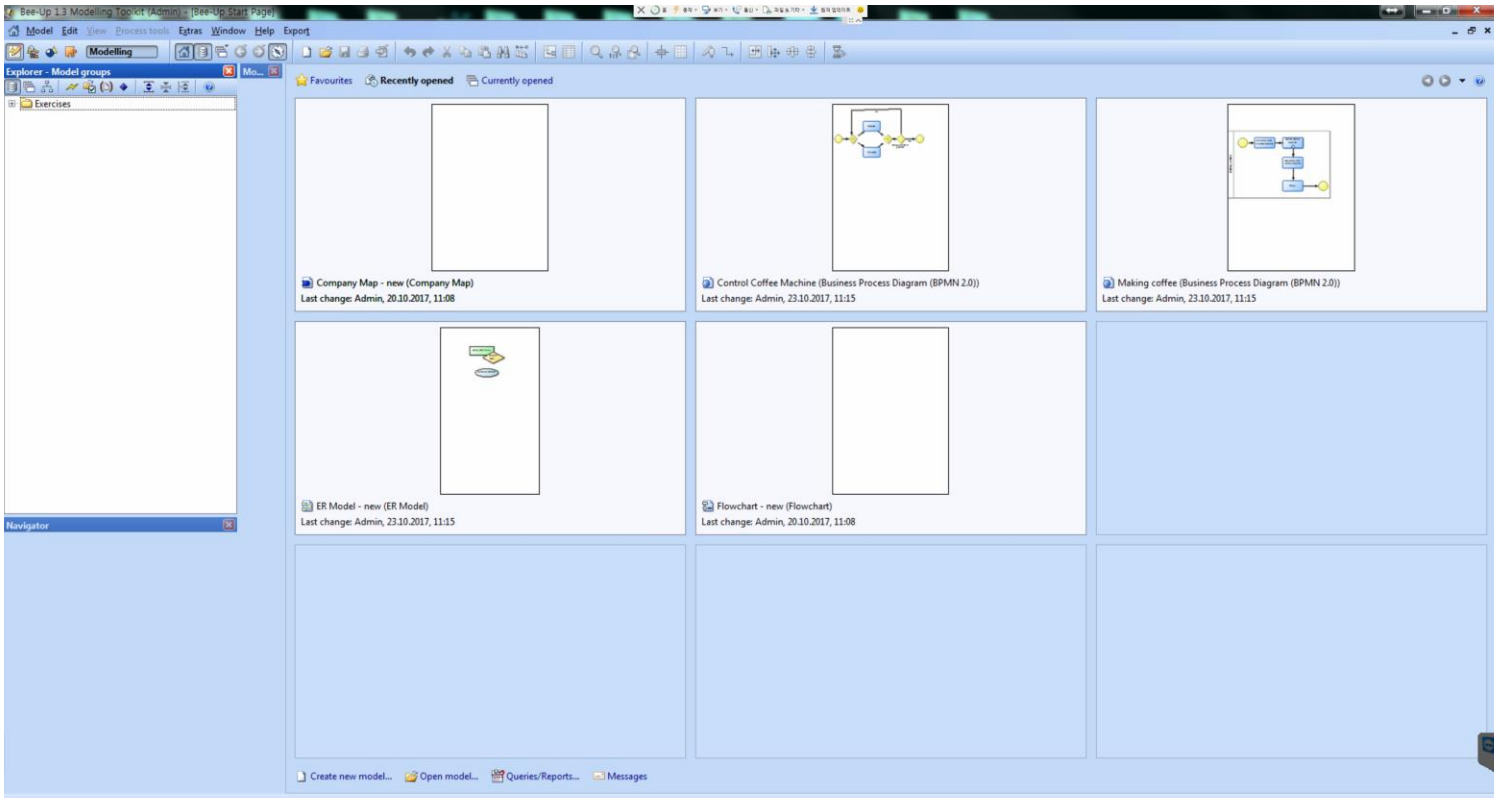
- 다운로드 홈페이지
  - http://Austria.omilab.org/psm/content/bee-up/info
  - 좌측 메뉴 중 Tool-Download에서 다운로드 가능함
- 설치 전 준비사항
  - 제어판-국가 및 언어에서 날짜 표기를 "영어(미국)" 으로 변경
  - 제어판-국가 및 언어에서 시스템 로캘을 "영어(미국)" 으로 변경
  - DB설치
    - 첨부자료 참고: ADOxx강의 1.pdf
- Bee-up 설치
  - 다운받은 인스톨러를 실행하여 설치

Bee-up 실행

- Bee-up 실행
  - 정상적으로 설치되었다면 Bee-up 1.3 Modelling Toolkit 아이콘이 바탕화면에 생성되어 있음
  - Bee-up 1.3 Modelling Toolkit을 실행

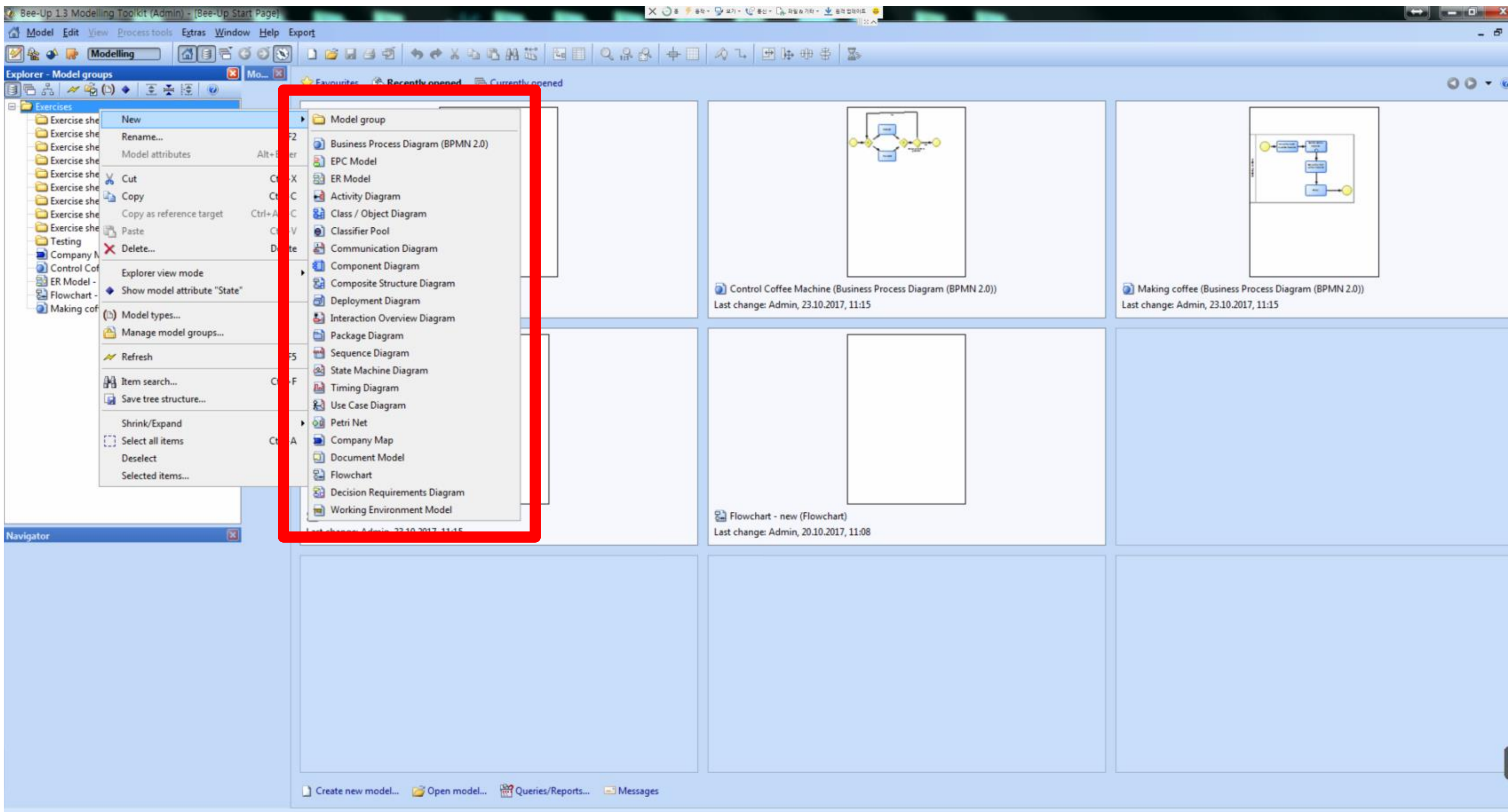


Bee-up 실행화면


Bee-Up 1.3 Modelling Toolkit (Admin) - [Bee-Up Start Page]
Model Edit View Process tools Extras Window Help Export
Modelling
Explorer - Model groups
Exercises
Navigator
Favourites Recently opened Currently opened
Company Map - new (Company Map)
Last change: Admin, 20.10.2017, 11:08
Control Coffee Machine (Business Process Diagram (BPMN 2.0))
Last change: Admin, 23.10.2017, 11:15
Making coffee (Business Process Diagram (BPMN 2.0))
Last change: Admin, 23.10.2017, 11:15
ER Model - new (ER Model)
Last change: Admin, 23.10.2017, 11:15
Flowchart - new (Flowchart)
Last change: Admin, 20.10.2017, 11:08
Create new model... Open model... Queries/Reports... Messages

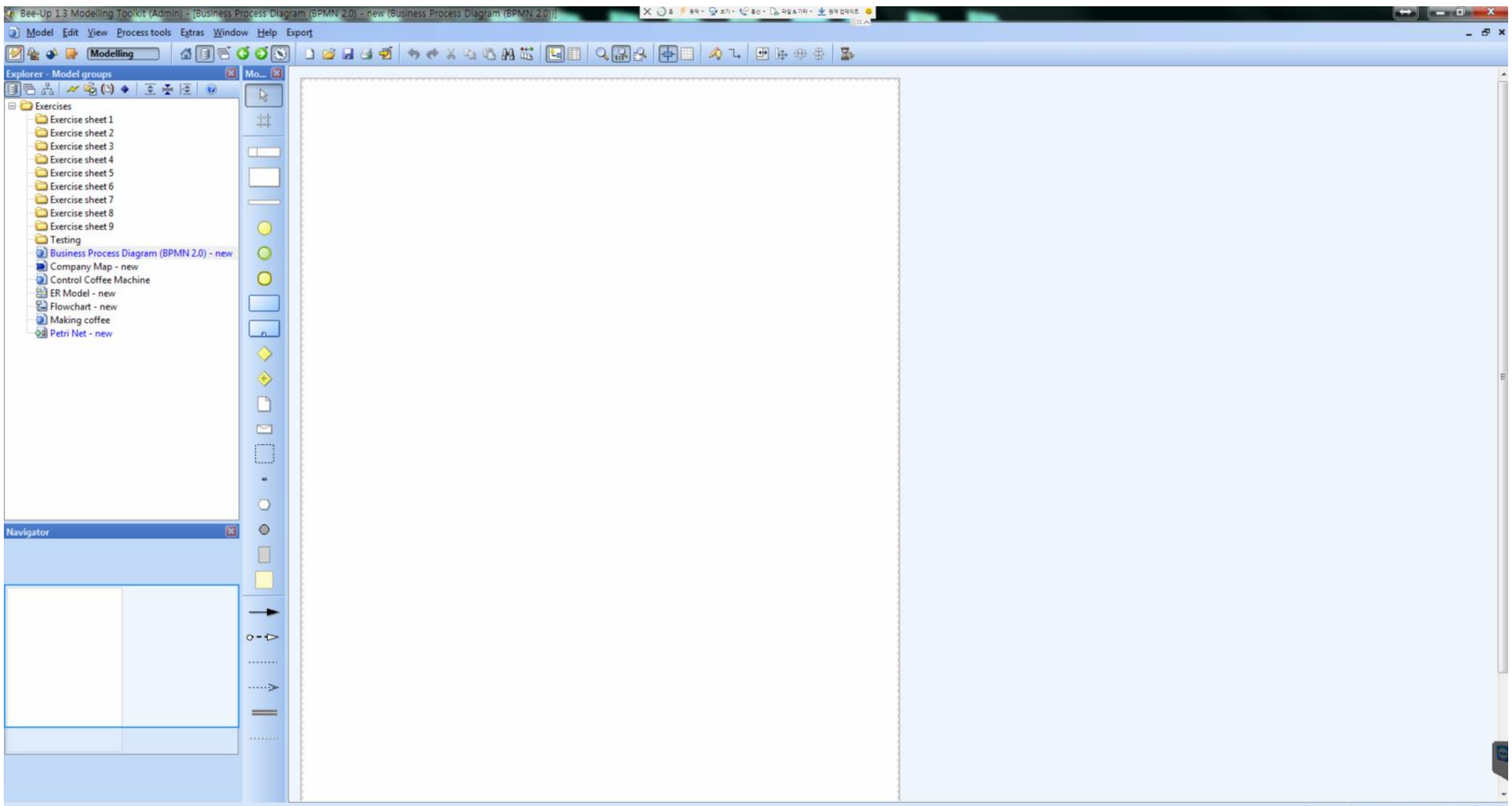
BEE-UP: BPM 모델 생성

- 우측의 익스플로러에서 폴더를 우클릭 -> new의 하위 항목에서 원하는 모델을 생성할 수 있음
  - 모델 중 BPD와 Petri Net 생성



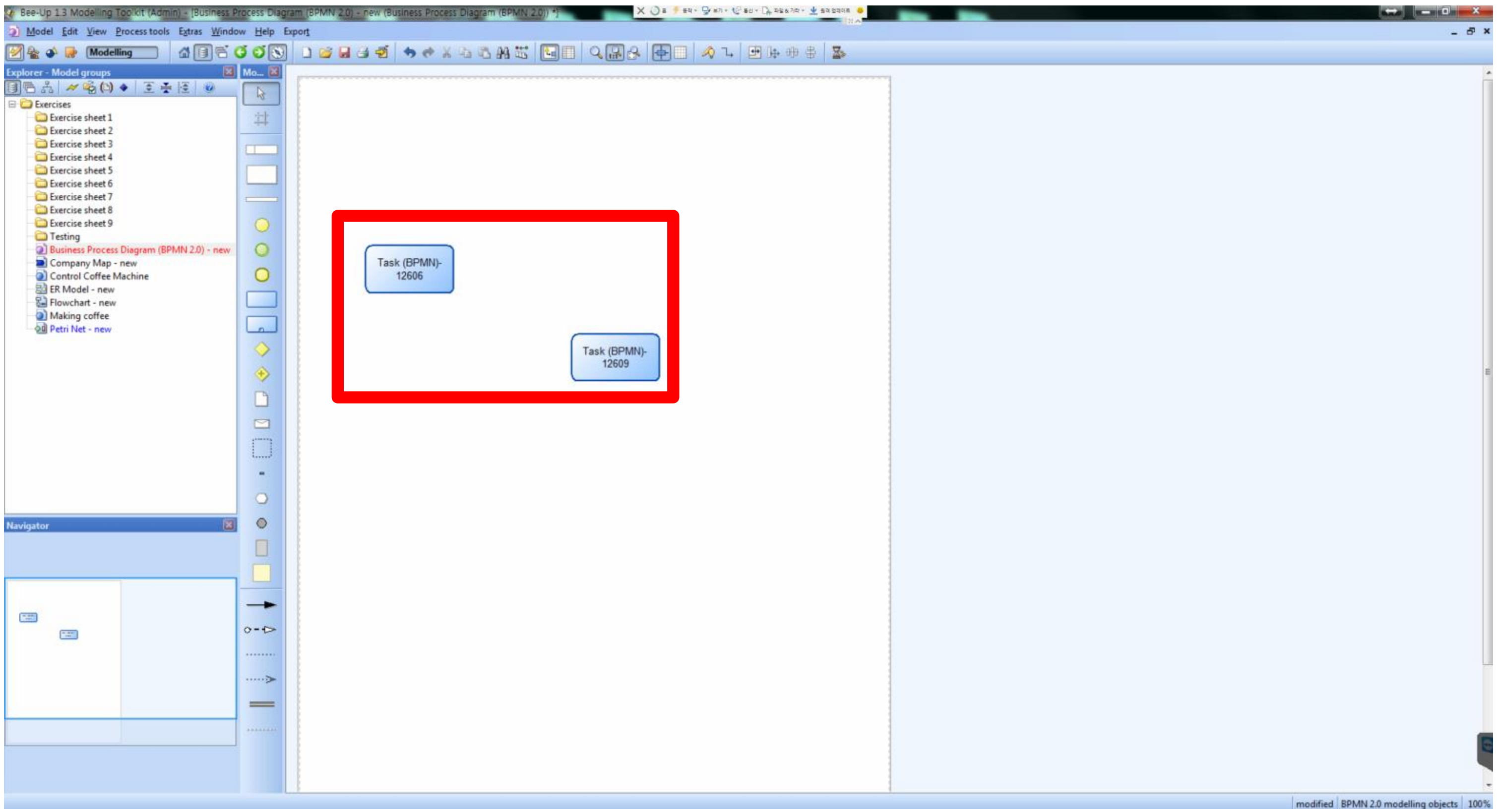
BEE-UP: BPM 모델 생성

- 생성된 모델 화면
  - 좌측의 익스플로러에서 생성된 모델의 아이콘을 확인할 수 있음



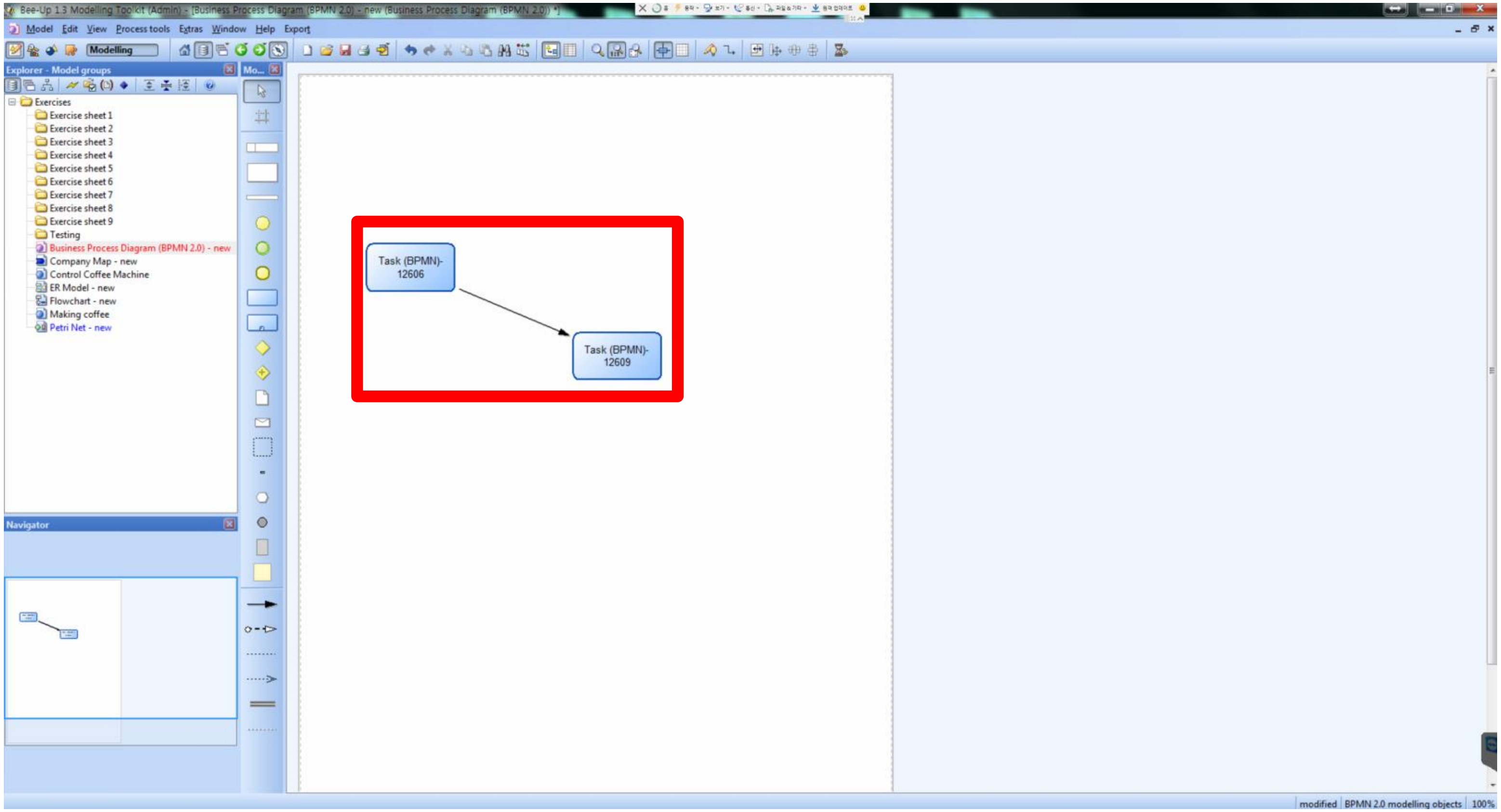
BEE-UP: BPM 클래스 생성

- 생성된 클래스 화면
  - 모델 익스플로러와 모델 화면 사이의 아이콘을 클릭하여 클래스 생성
  - 중앙의 창에서 생성된 클래스 확인 가능



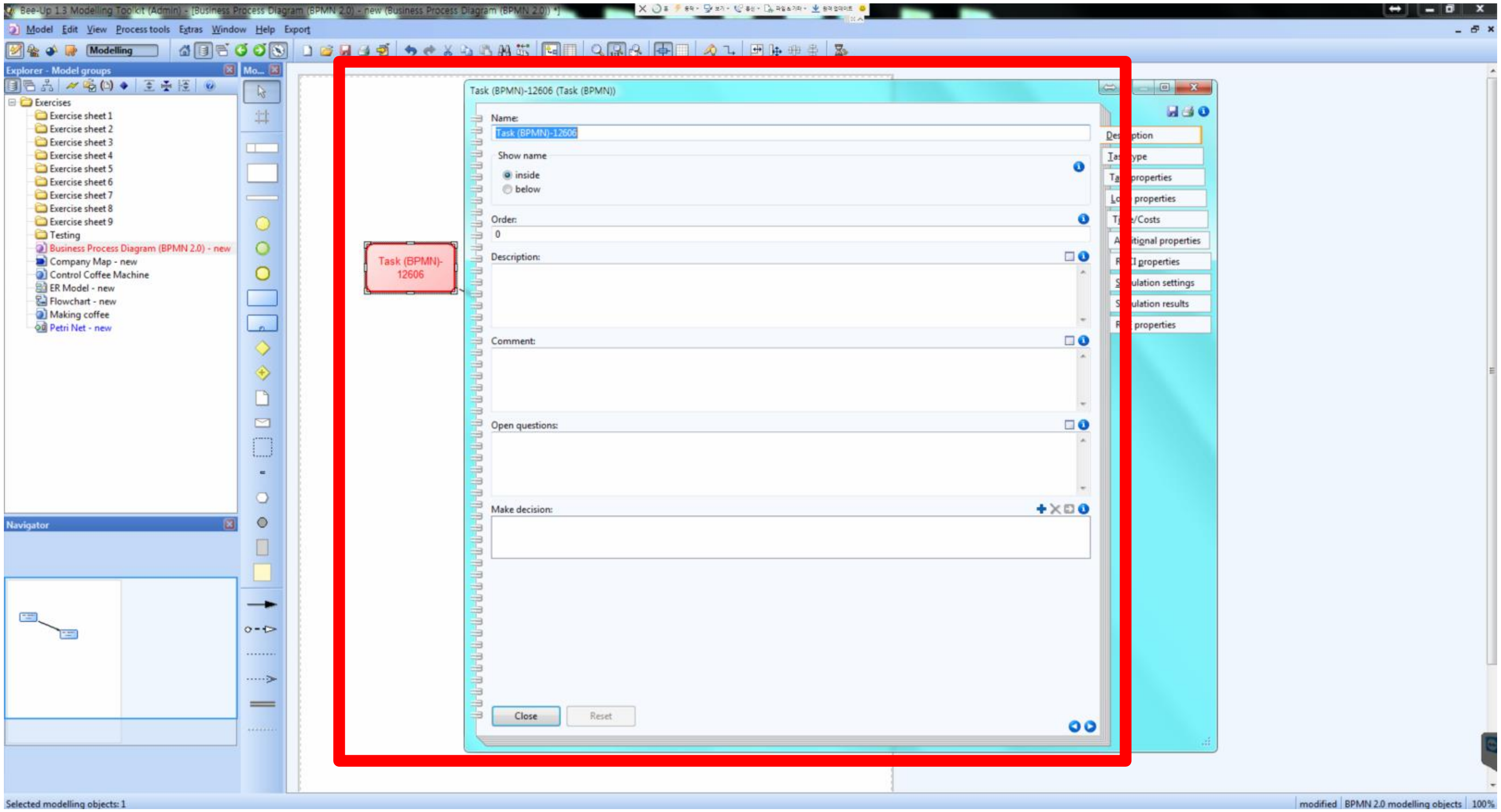
BEE-UP: BPM 노드 생성

- 생성된 노드 화면
  - 모델과 마찬가지로 노드 선택 후 두 클래스를 클릭하여 생성 가능



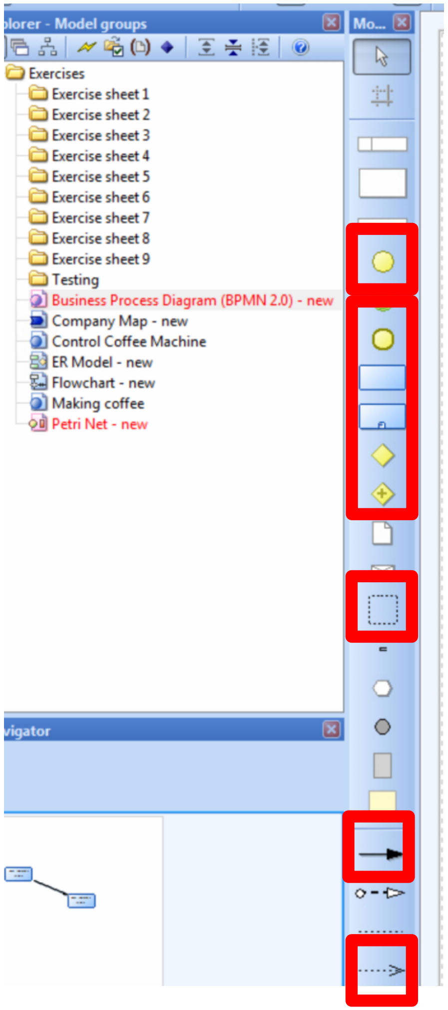
BEE-UP: BPM 노트북 활성화

- 생성된 모델 화면
  - 생성된 클래스를 더블클릭하여 노트북을 활성화 할 수 있음
  - 노트북에서 클래스의 이름이나 값 등을 수정 가능



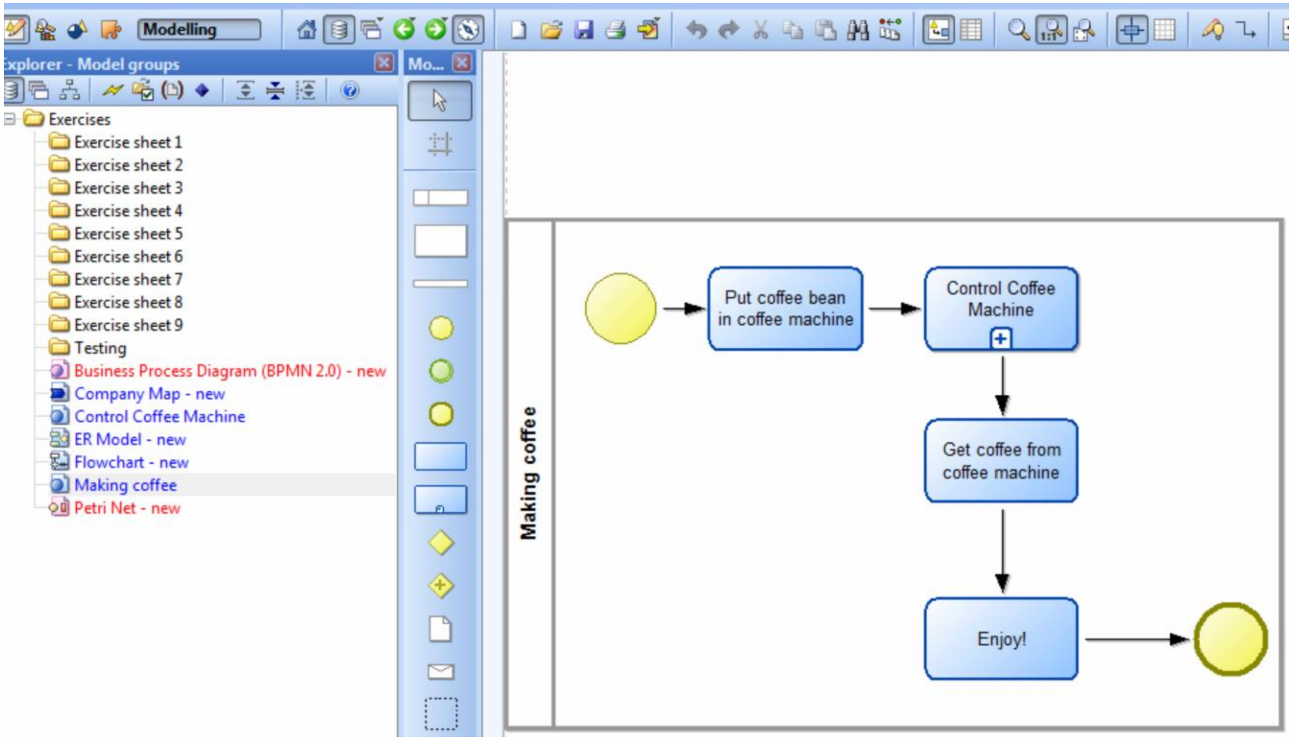
BEE-UP: BPM 클래스 목록

- 생성할 클래스
  - BPM에서 생성할 클래스는 다음과 같음
    - 시작이벤트
    - 중간이벤트
    - 종료이벤트
    - 게이트웨이
    - 액티비티
    - 풀과 레인
    - 시퀀스 플로우
    - 메시지 플로우



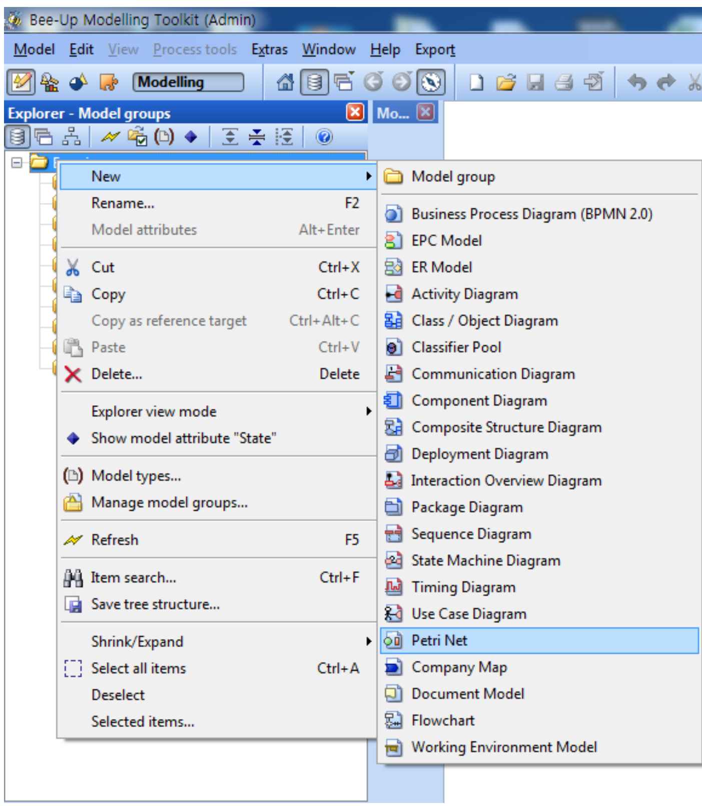
BEE-UP: BPM 생성 예제 화면

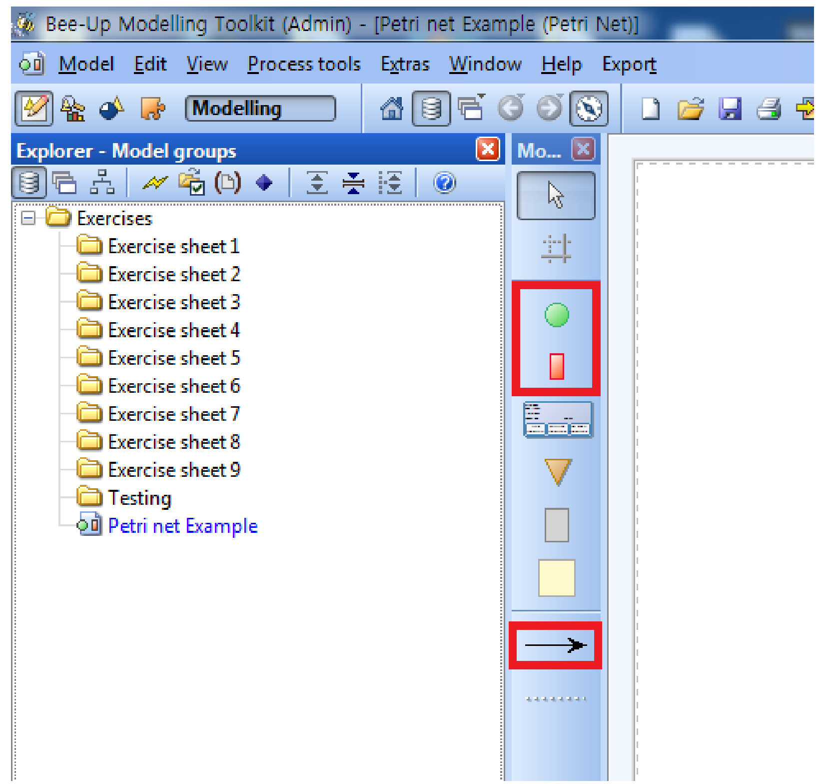
- BPM을 활용한 예제생성 화면



BEE-UP: Petri Net

- 모델 생성
  - Place, transition, arc와 추가 기능을 위한 여러 node를 생성 가능

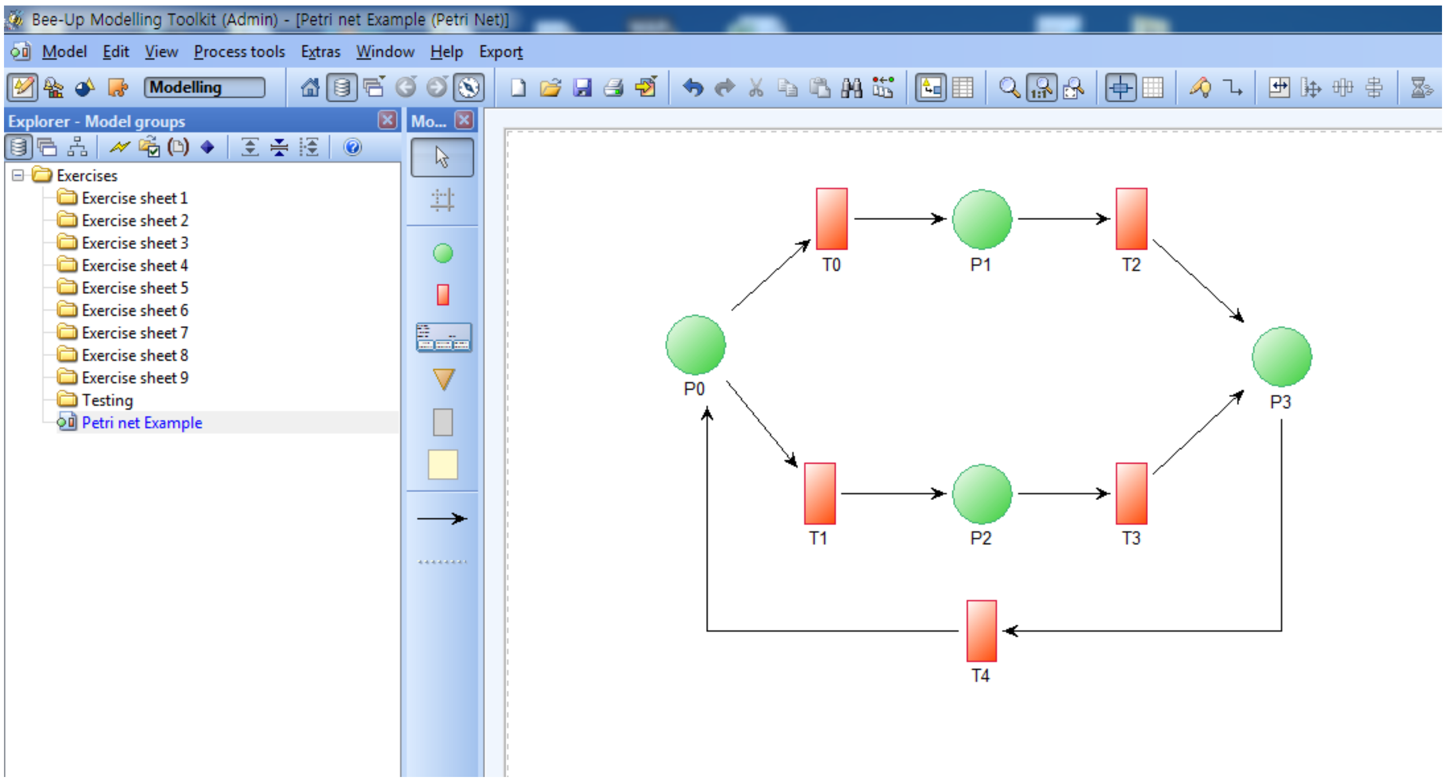




OMiLAB KOREA

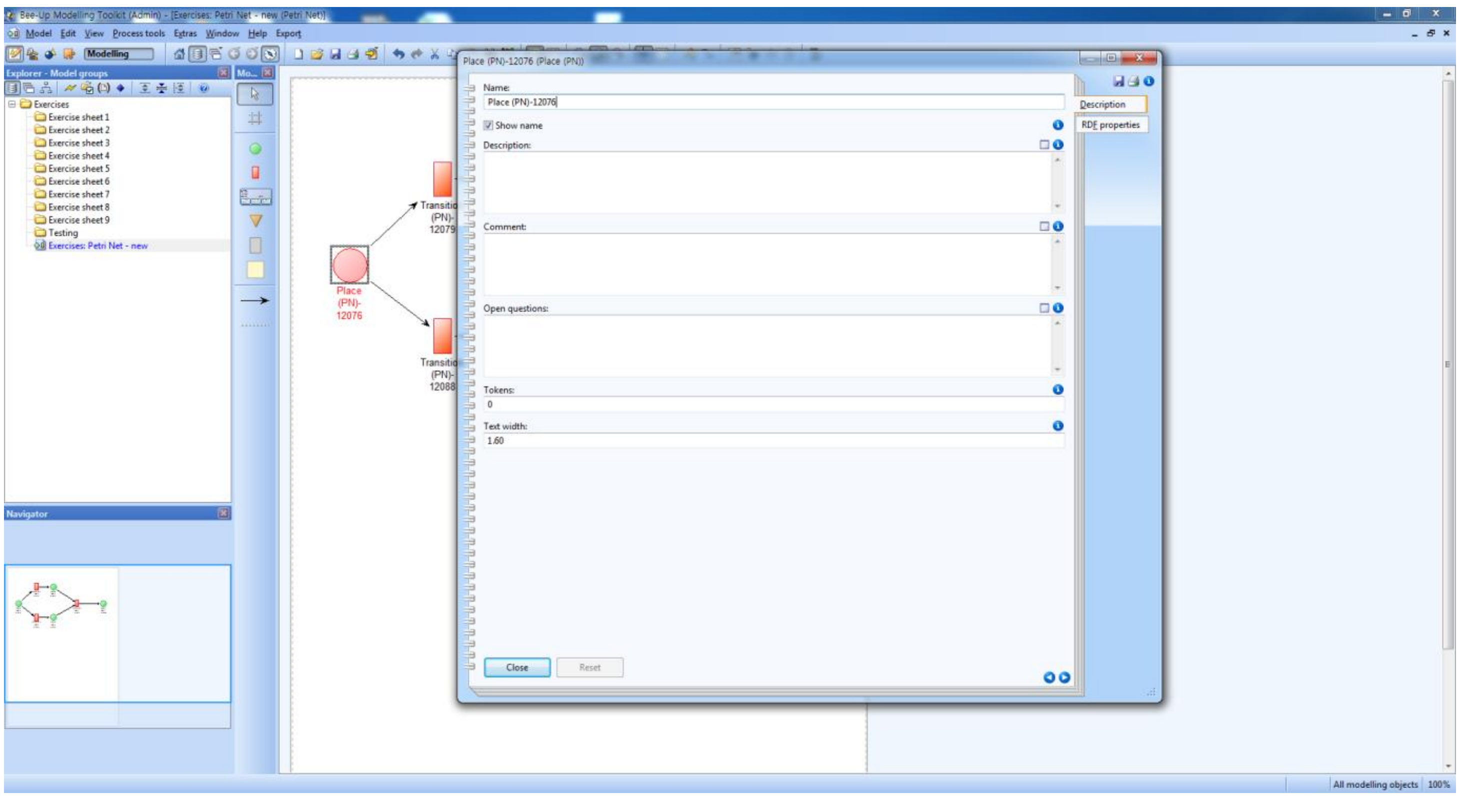
BEE-UP: Petri Net

- Petri net 모델 예시



BEE-UP: Petri Net

- 노트북 열기 및 node 정보 입력
  - Node를 더블 클릭하면 노트북이 열림.
  - 노트북을 활용하여 각 node에 대한 이름, 설명 등의 다양한 추가 정보 입력

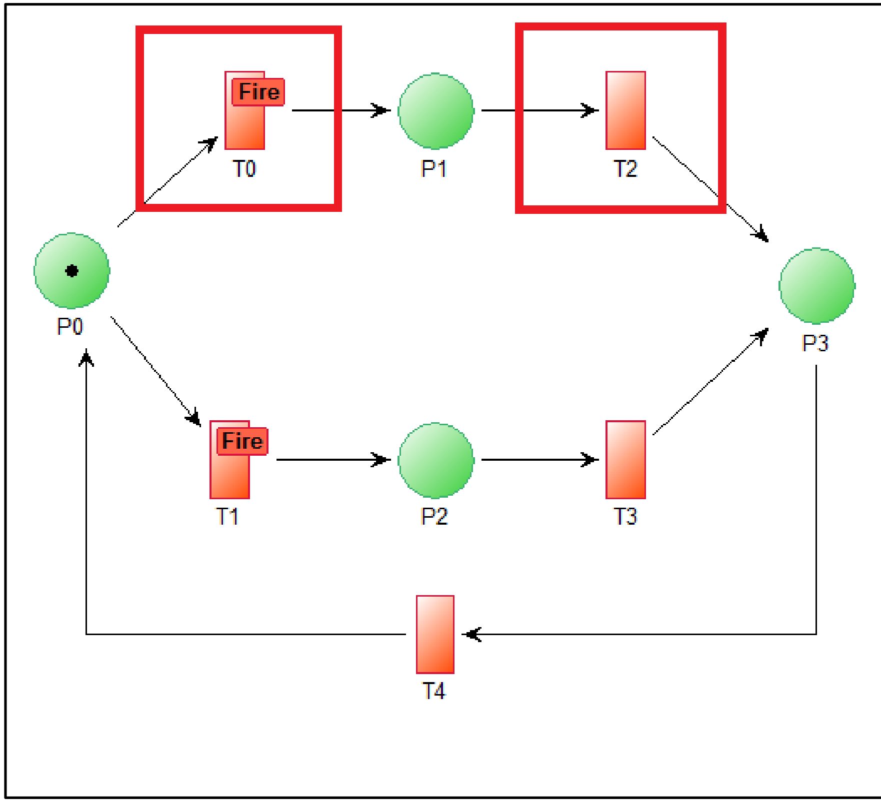


BEE-UP: Petri Net

- 시뮬레이션
  - 두 가지 방법의 시뮬레이션
    - 직접 transition을 하나씩 Fire 시키기
    - 시뮬레이션 구성자를 사용하여 자동으로 시뮬레이션을 수행

BEE-UP: Petri Net

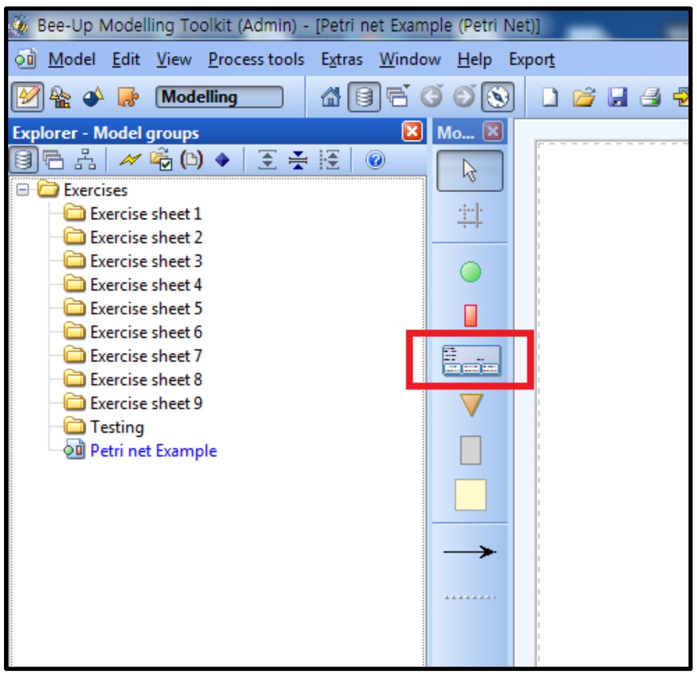
- 직접 transition을 하나씩 Fire 시키기
  - 절차
    - 모든 node와 token을 배치하면 여러 transition 중 fire 조건을 충족한 transition이 하나 이상 존재하게 됨
    - Fire 가능한 상태의 transition 아이콘은 변경됨
    - Fire 문구를 클릭하면 transition이 수행됨
  - 장단점
    - 전이 단계 하나하나를 천천히 살펴볼 수 있음
    - 선택에 따라 발생할 수 있는 deadlock을 발견하지 못할 가능성이 존재

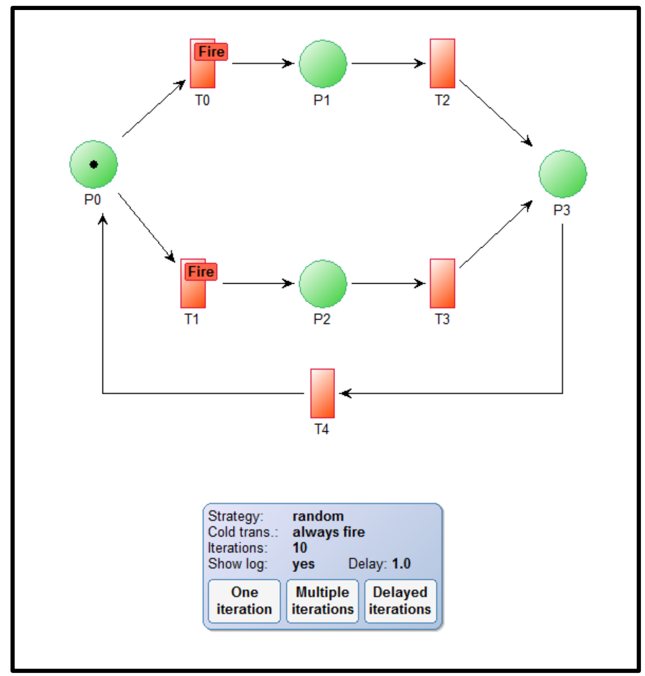


OMiLAB KOREA

BEE-UP: Petri Net

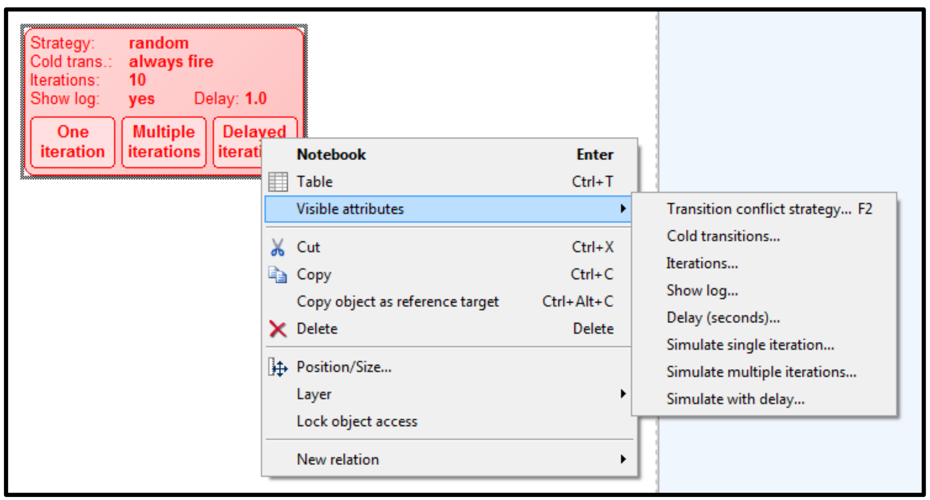
- 시뮬레이션 구성자 사용
  - One iteration: 한번 Fire
  - Multiple iteration: 설정된 수 만큼 Fire
  - Delayed iteration: 무한 반복 Fire, transition이 일어날 때마다 delay 값만큼 대기





BEE-UP: Petri Net

- 시뮬레이션 구성자 사용
  - 설정을 통해 iteration 회수, delay 값, 시뮬레이션 방식 등을 설정 가능
    - Transition conflict strategy: transition 선택하는 방법 설정
      - Default: 먼저 연결된 transition을 우선적으로 선택, 한 가지 방법으로만 시뮬레이션이 실행됨
      - Random: Fire 가능한 transition을 랜덤으로 하나 선택
      - 확률에 따른 선택 방법 등이 있음
    - Iteration: Multiple iteration에서 사용, transition을 fire할 회수를 정함
    - Delay (seconds): delayed iteration 방법에서 사용, transition이 실행되는 사이사이의 delay 값을 정함



BEE-UP: Petri Net

- 시뮬레이션 결과
  - Multiple, Delayed iteration 중 하나의 방법으로 시뮬레이션하면, 시뮬레이션 종료 후 시뮬레이션 결과를 표기하는 창이 나타남.
  - 각 상태 별 token의 개수를 확인할 수 있음