

# ペトリネットによる マルチエージェントシステムのモデル化

平石 邦彦

北陸先端科学技術大学院大学情報科学研究科

[hira@jaist.ac.jp](mailto:hira@jaist.ac.jp)

# 内容

- マルチエージェントシステム .
- ペトリネットによるマルチエージェントシステムの記述とその問題点 .
- オブジェクト指向ペトリネット:
  - Elementary Object Systems
  - Reference Nets
- エージェント指向ペトリネット:
  - $PN^2$  (Petri Nets in a Petri Net)
- 今後の研究課題 .

# マルチエージェントシステム

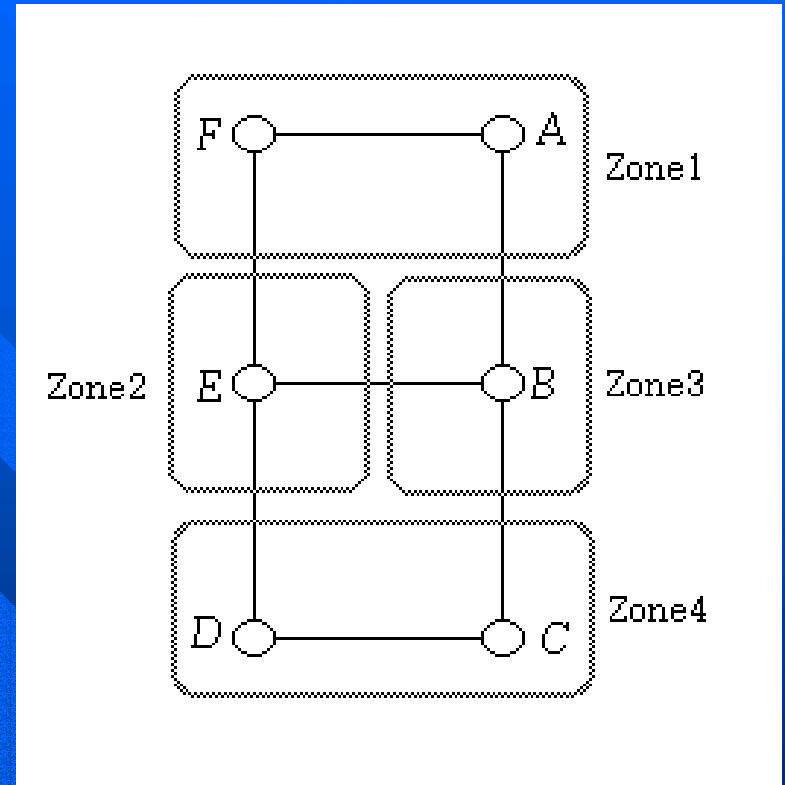
## ■ エージェント (Agent):

“An agent is a computer system that is situated in some *environment*, and that is capable of *autonomous* action in this environment in order to meet its *design objectives*.” (G. Weiss (ed.): Multiagent Systems – A Modern Approach to Distributed Artificial Intelligence, The MIT Press, 1999).

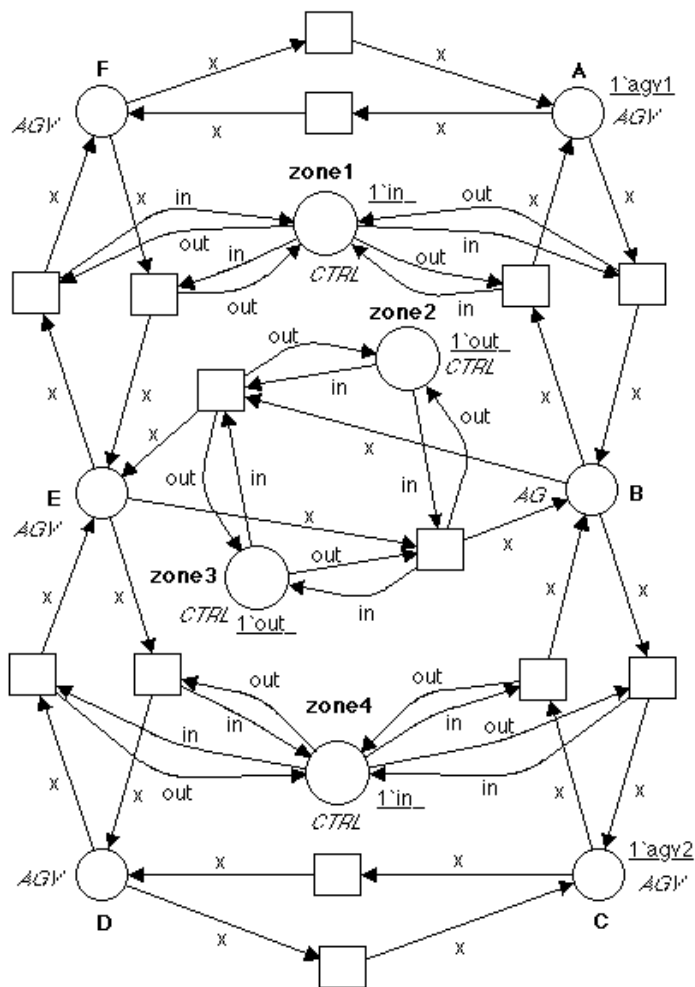
- マルチエージェントシステム = 環境 + 複数のエージェント。
  - 環境の状態遷移
  - 個々のエージェントの状態遷移
  - それらのインタラクション

# 例：AGVシステム

- 複数の自動搬送車 (AGV: Automated Guided Vehicle) が軌道上を移動して部品を運搬する。
- 搬送要求:  $AGV_1$  は  $A$  と  $D$  を交互に,  $AGV_2$  は  $C$  と  $F$  を交互に訪れる。
- 軌道は4つのゾーンに分割されており, 各ゾーン内にはたかだか1台の AGV しか存在してはならない。



# Coloured Petri Net によるモデル化



- 各 AGV は単にデータ型 AGV をもつトークンとして表現されているにすぎない。
- 各 AGV の詳細化において、以下が満たされることが望ましい。
  - 各 AGV の内部状態の変更はそれ自身が行う。
  - ある AGV の動作プログラムの変更が他のオブジェクト (軌道, ゾーン制御) の記述に影響を与えない。
- CPN で実現すると, モデルが複雑になる。

# オブジェクト指向ペトリネット (1)

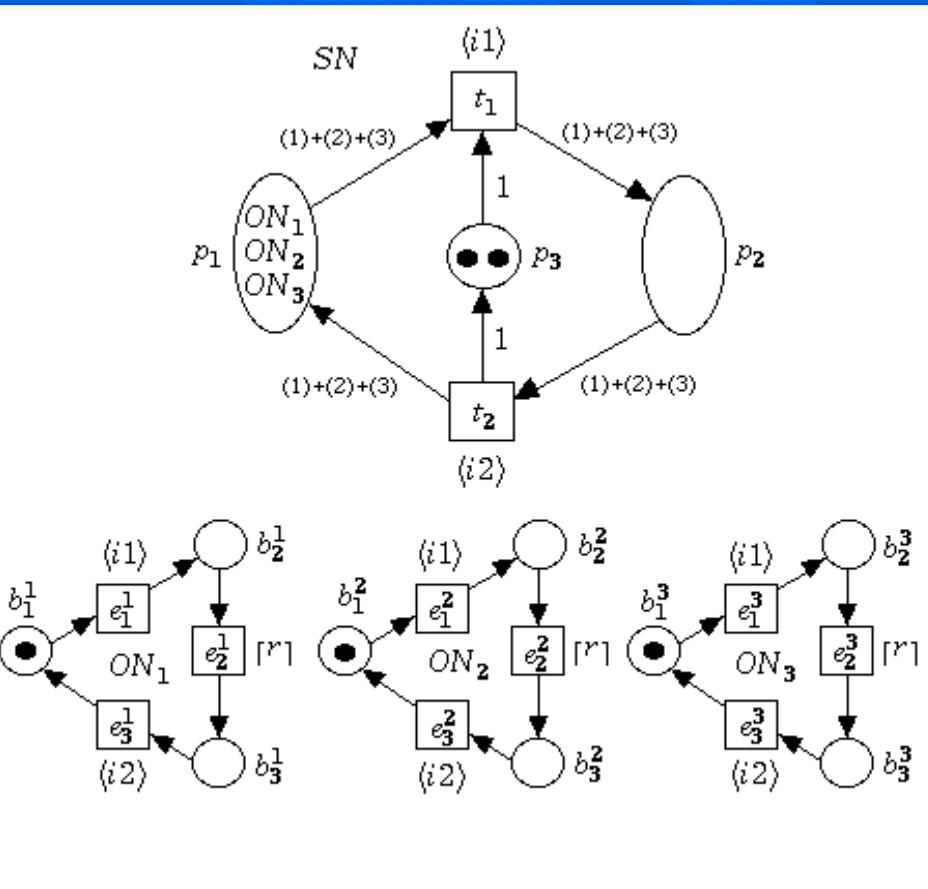
- オブジェクト指向の考え方を取り入れたペトリネットの拡張モデルが提案されている。
  - 多くのオブジェクト指向ペトリネットにおいて、トークンはそれに対応するオブジェクトへのリファレンス(ポインタ)に関する情報を持つことができる。
  - オブジェクトをトークンとするようなペトリネットは、データ構造をもつトークンを扱えるカラーペトリネットの拡張であると言える。
- 継承の考えを導入し、すでに定義されたオブジェクトからの差分の記述により新しいオブジェクトを定義する機能を取り入れたモデルもある。

# オブジェクト指向ペトリネット (2)

- OBJSA Nets [E. Battiston et al. 88],
- POTs, POPs [J. Engelfriet et al. 90],
- CO-OPN, CO-OPN2 [D. Buch et al. 93],
- PNTalk [M. Ceska et al. 94],
- Cooperative Nets [C. Sibertin-Blanc et al. 94],
- Object Petri Nets [C. Lakos 95],
- OB(PN)<sup>2</sup> [J. Lilius 96],
- Elementary Object Systems [R. Valk 88],
- Reference Nets [O. Kummer et al. 99], ...

# 初等的オブジェクトシステム (1)

初等的オブジェクトシステム (Elementary Object Systems) はオブジェクト・システムの理論的解析を目的としたモデルである。



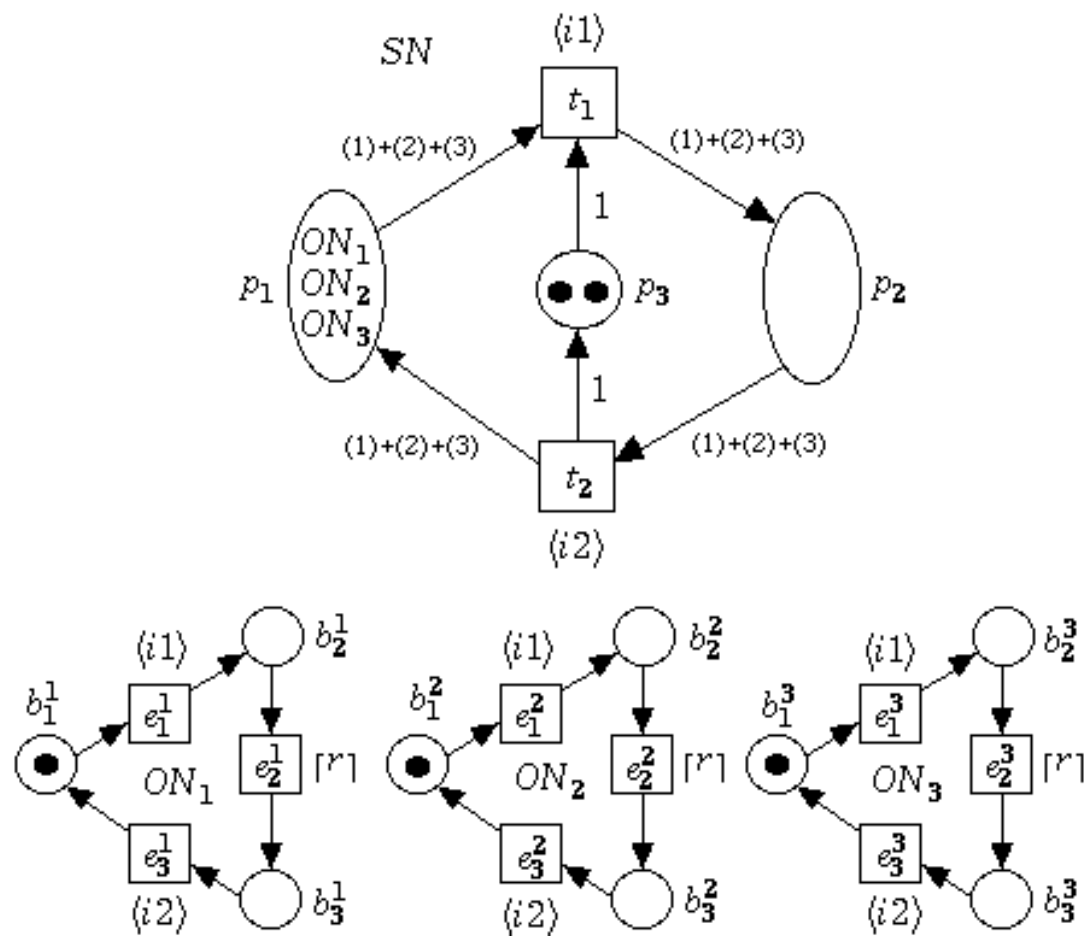
- $SN$  をシステムネットといい,  $ON_1, ON_2, ON_3$  をオブジェクトネットという.
- $SN$  のプレース  $p_1$  には3つのトークンがあるが, これらは対応するオブジェクトネットへのリファレンスである.



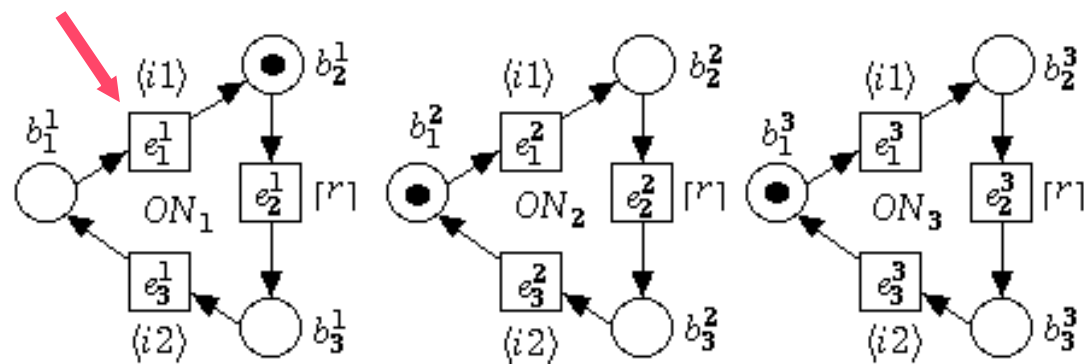
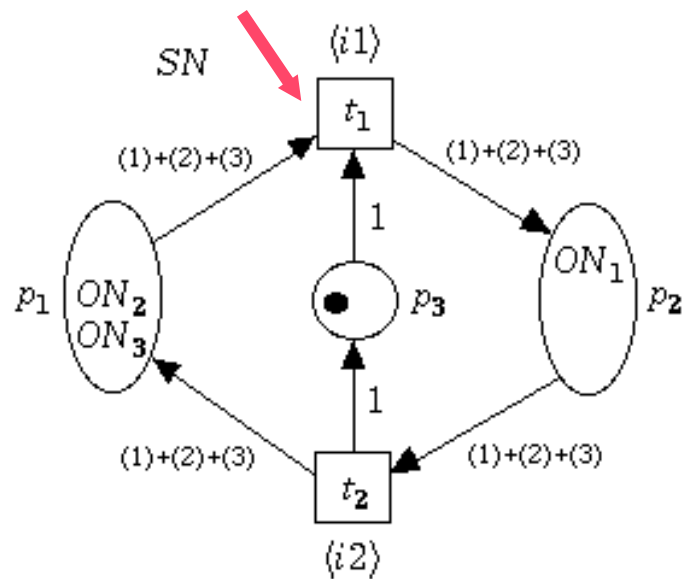
# 初等的オブジェクトシステム (2)

- トランジションの発火にはつぎの4種類がある：
  - $SN$  のトランジションが単独で発火する；
  - $SN$  のトランジションと  $ON_i$  のトランジションの組が同期して発火する；
  - 同じプレース内に存在する  $ON_i, ON_j$  のトランジションの組が同期して発火する；
  - $ON_i$  のトランジションが単独で発火する。

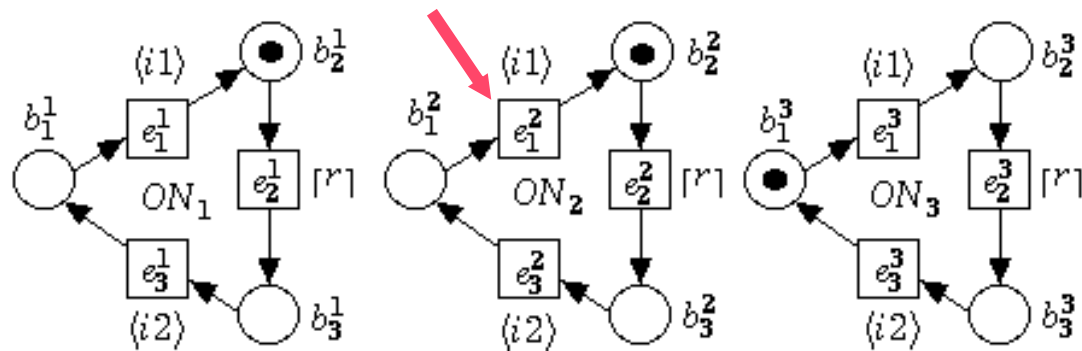
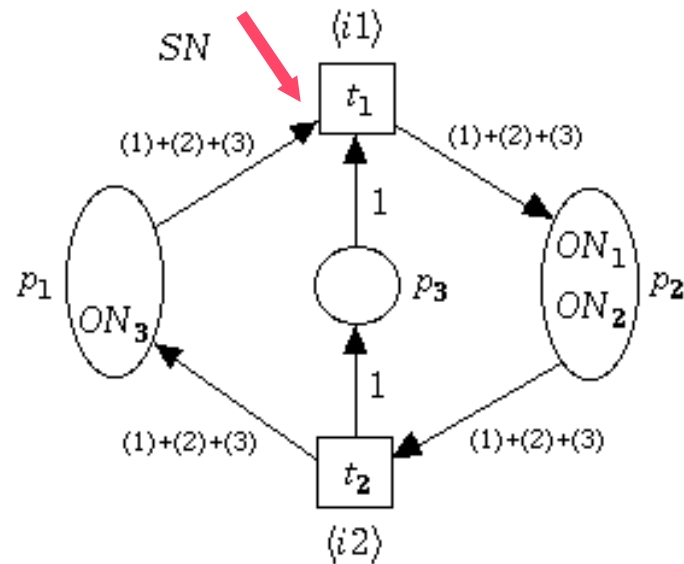
# 狀態遷移



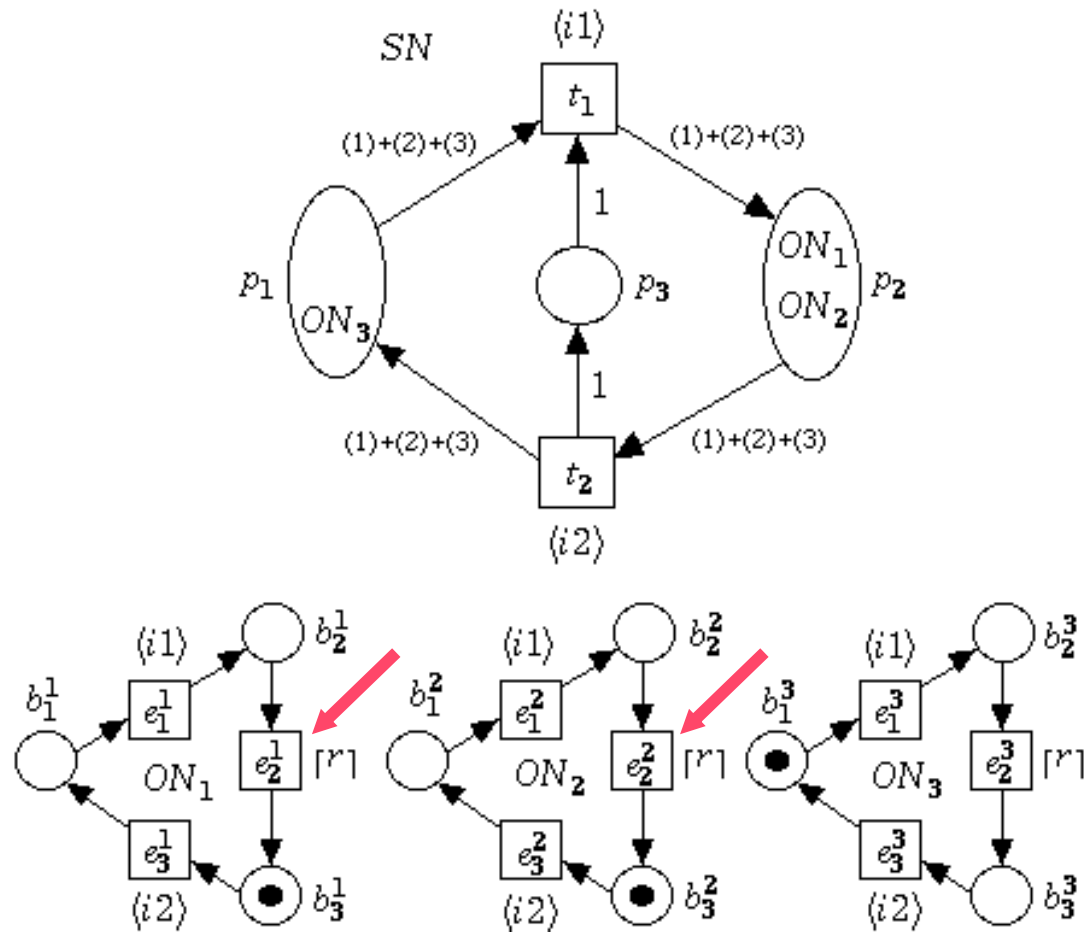
# $[t_1, e^1_1]$ が発火



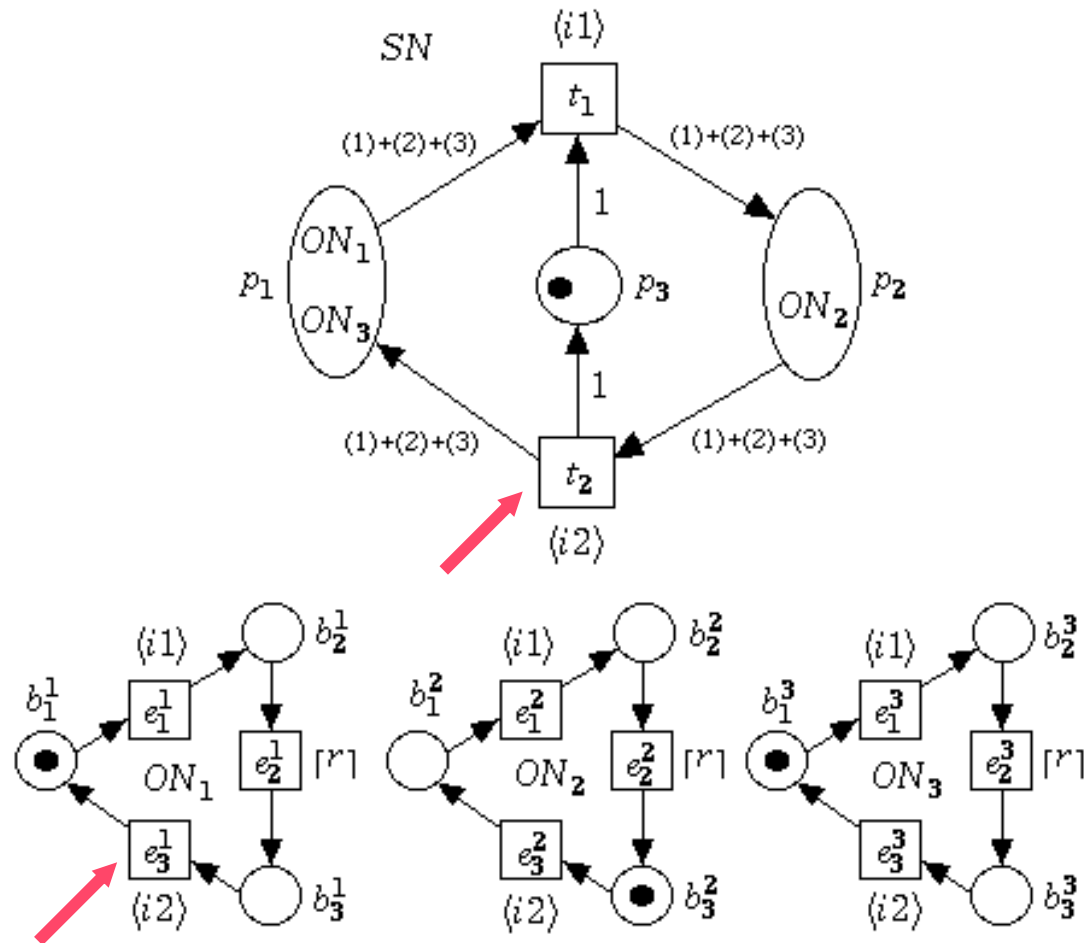
# $[t_1, e^2_1]$ が発火



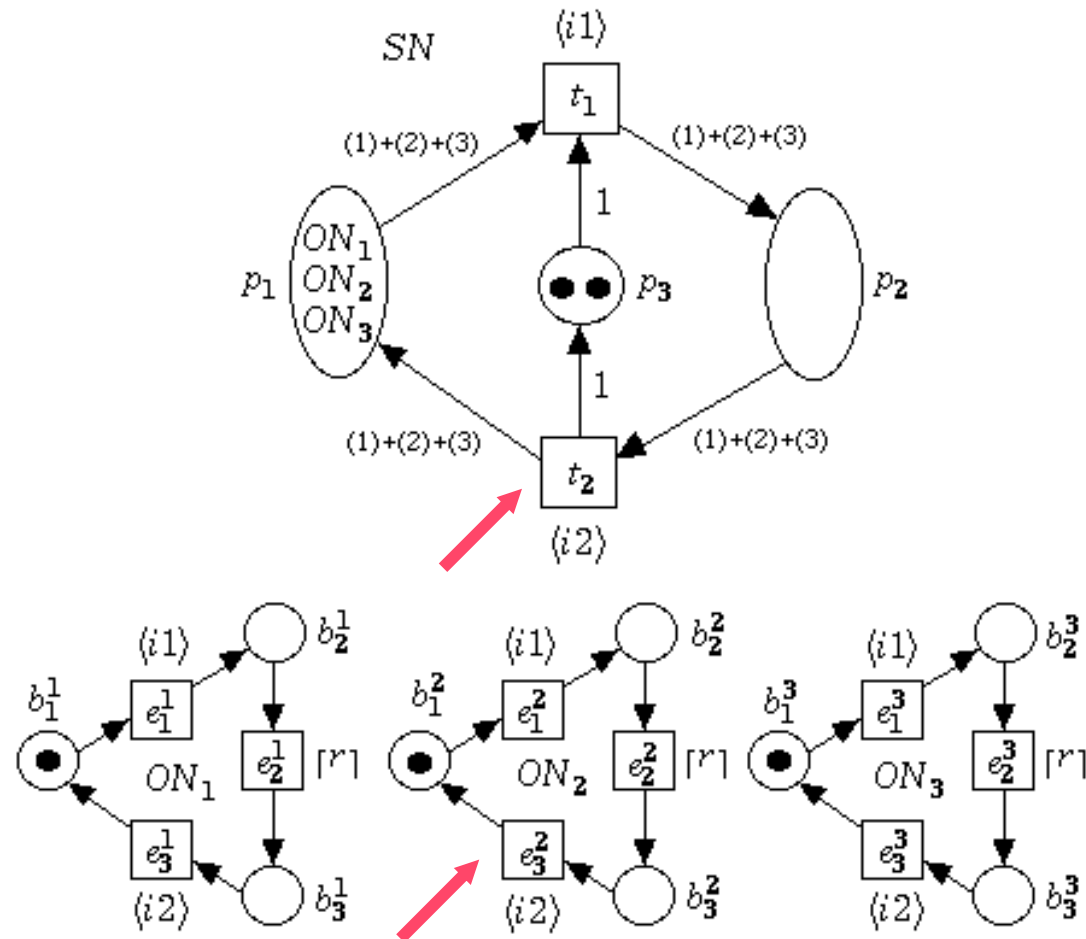
# $[e^1_2, e^2_2]$ が発火



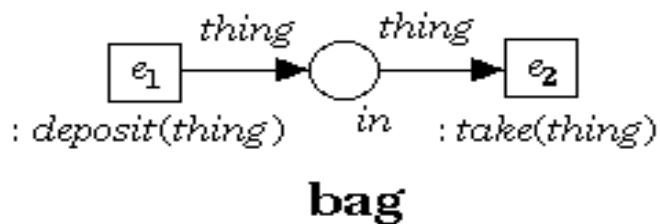
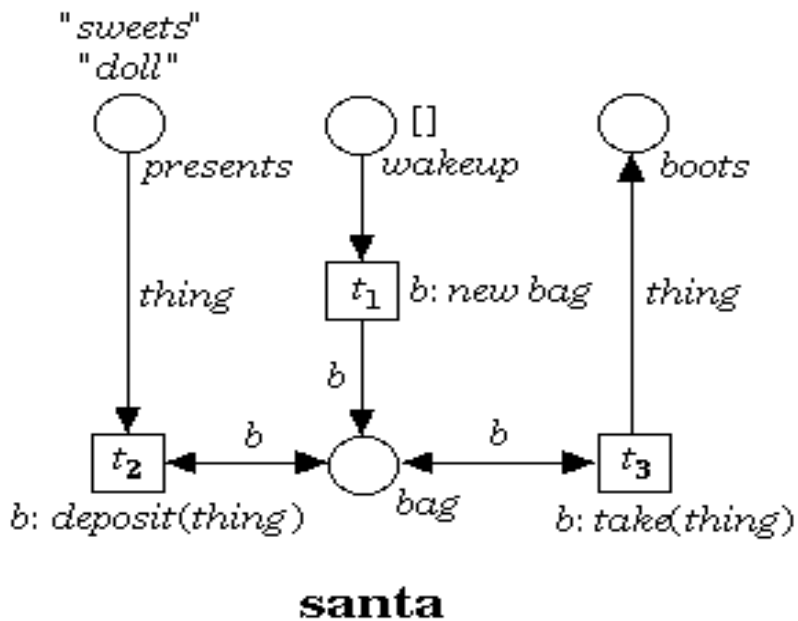
# $[t_2, e^1_3]$ が発火



## [ $t_2, e^2_3$ ] が発火



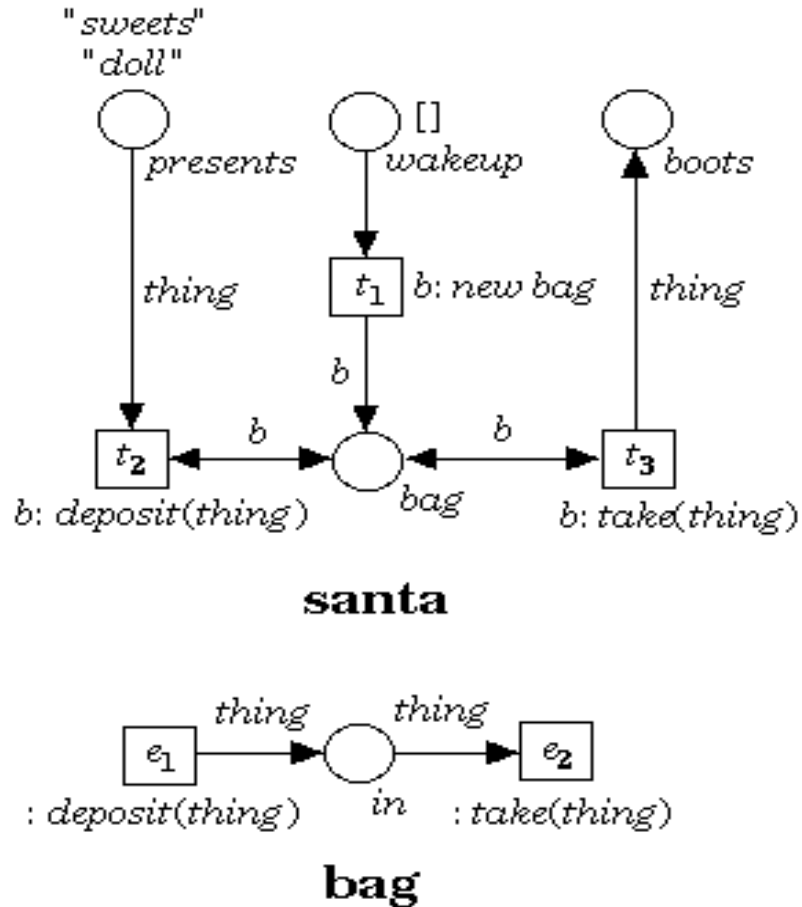
# リファレンスネット (1)



- リファレンスネット (Reference Nets) は、高水準ペトリネットのようなプログラミング言語として使用することを目的としたオブジェクト指向ペトリネットである。
- 機能的にはカラーペトリネットとオブジェクト指向プログラミングのスタイルを融合させたものである。

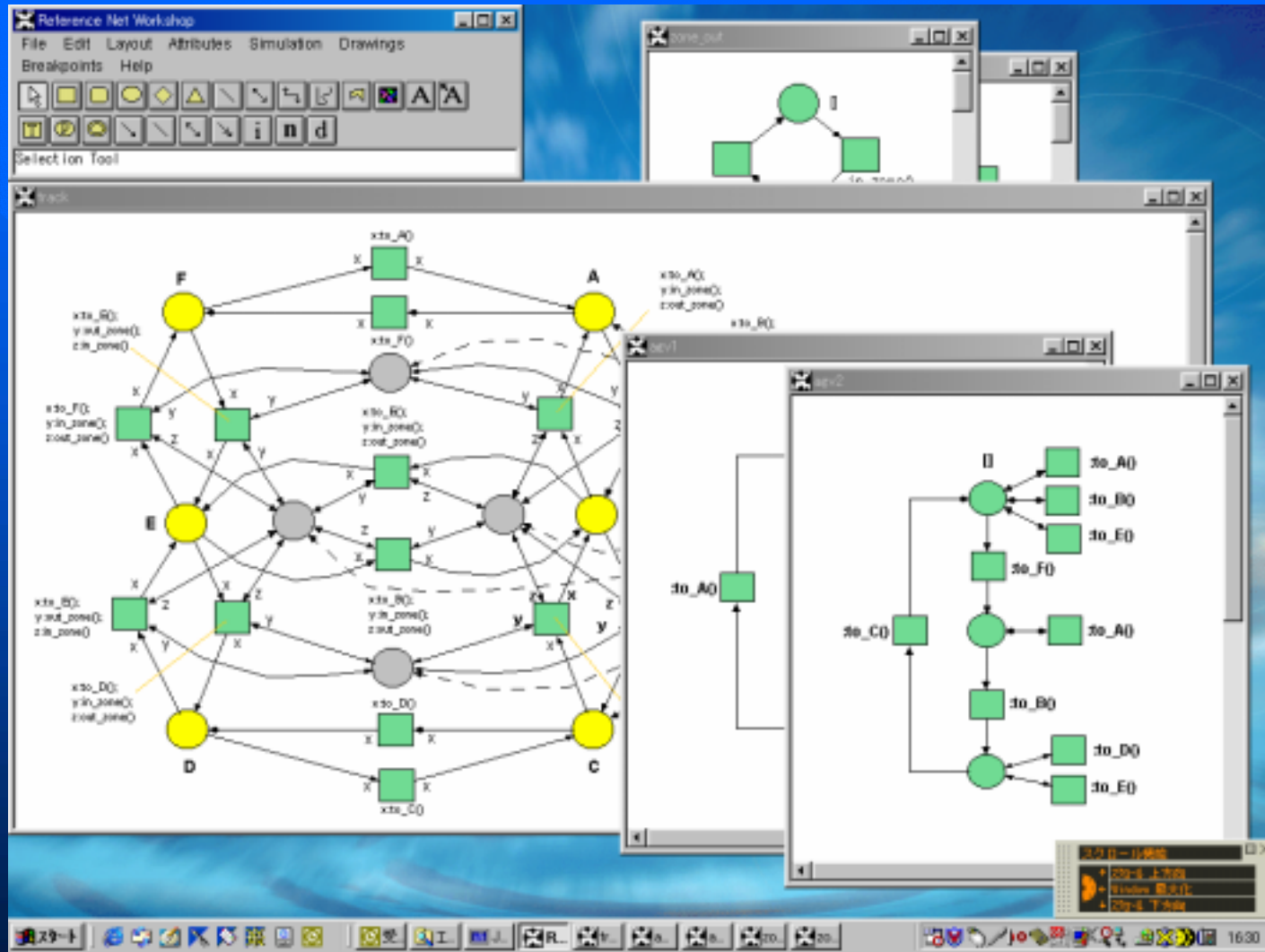


# リファレンスネット (2)



- トランジション  $t_1$  の発火により, メソッド  $b: new\ bag$  が実行され, ネット bag のインスタンスが作られる. 同時に, それへのリファレンスがプレース  $bag$  に入る.
- トランジション  $t_2$  の発火により, プレース  $presents$  からトークン  $sweets$  または  $doll$  が取り出され, ネット bag のインスタンスのプレース  $in$  に入る.
- ネット  $santa$  の  $b: deposit(thing)$  と ネット  $bag$  の  $:deposit(thing)$  の記述は通信チャネルの定義を表し, トランジション  $t_2$  と  $e_1$  が同期して発火し, その際, チャネル  $deposit$  を通じて変数  $thing$  の内容がネット  $bag$  に渡されることを表す.

# リファレンスネット (3)



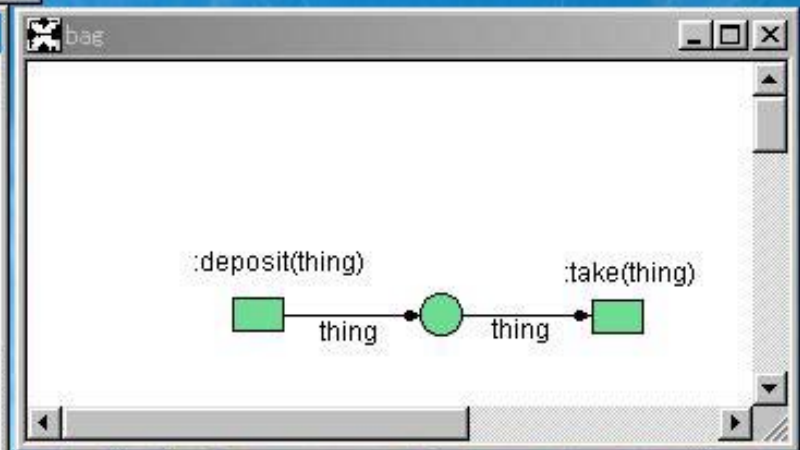
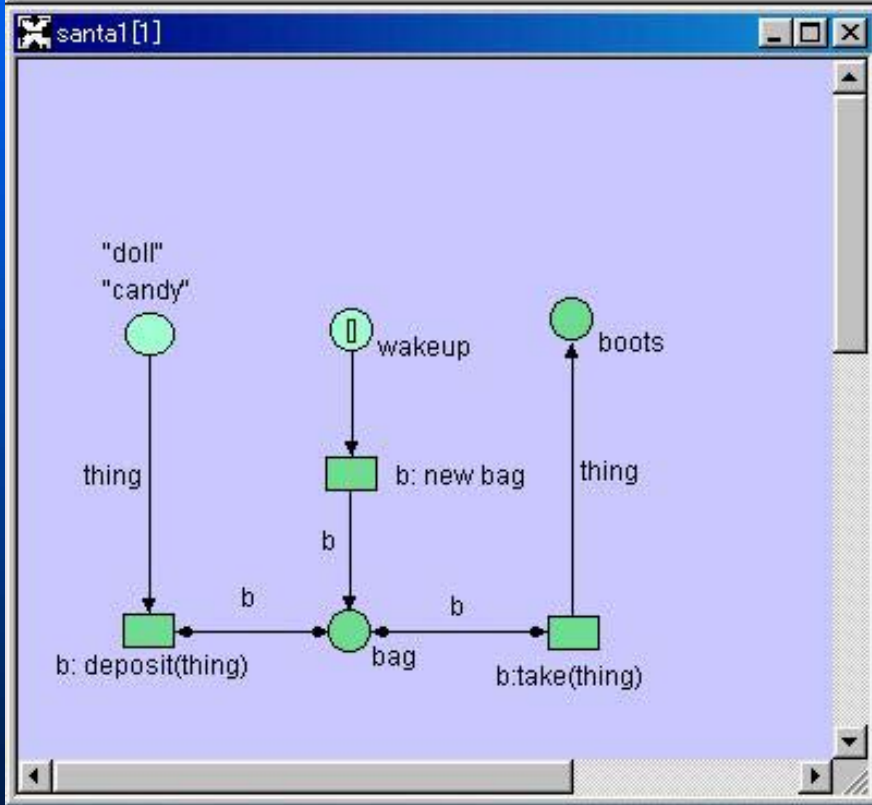
RENEW – the Reference Net Workshop

<http://www.renew.de/>

Reference Net Workshop

File Edit Layout Attributes Simulation Drawings  
Breakpoints Help

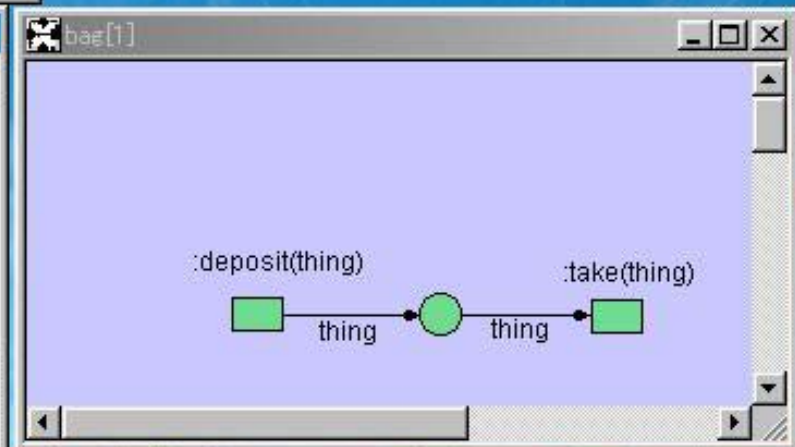
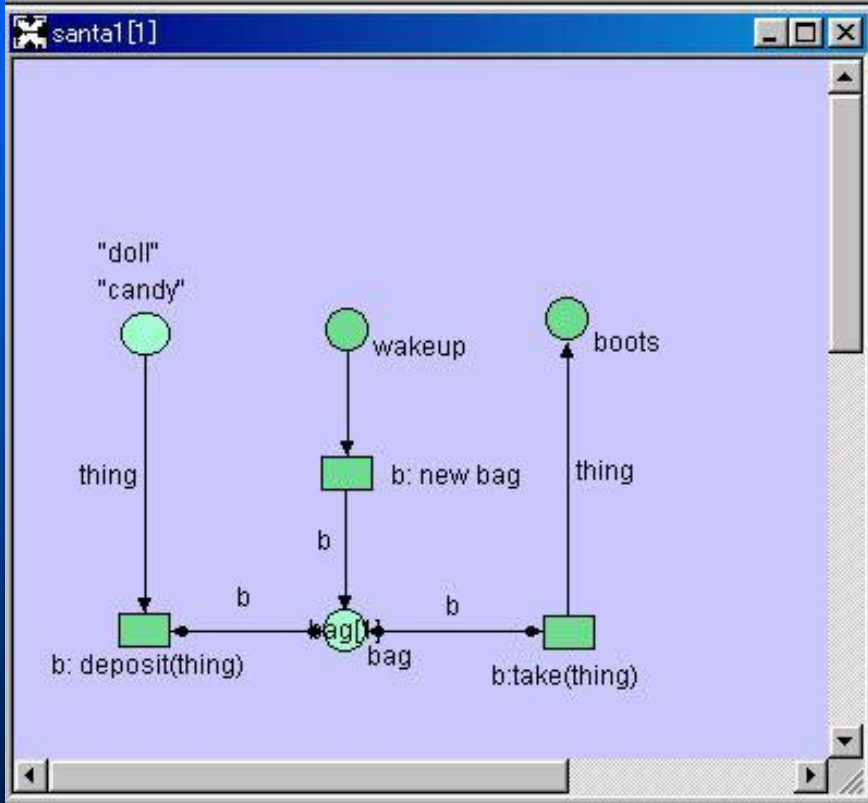
Simulation initialized.



Reference Net Workshop

File Edit Layout Attributes Simulation Drawings  
Breakpoints Help

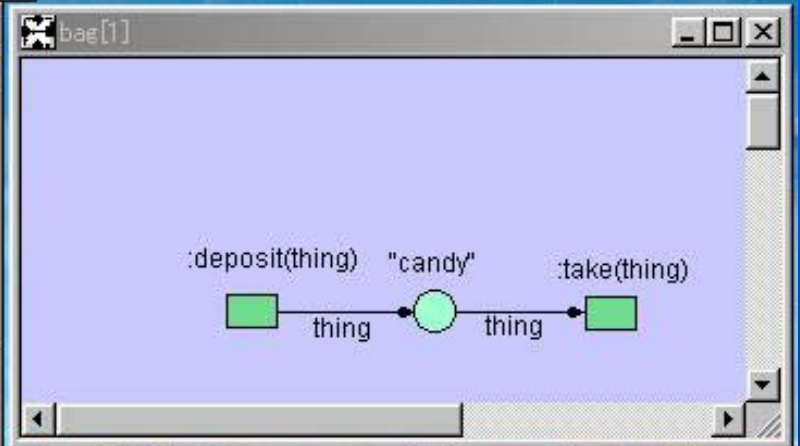
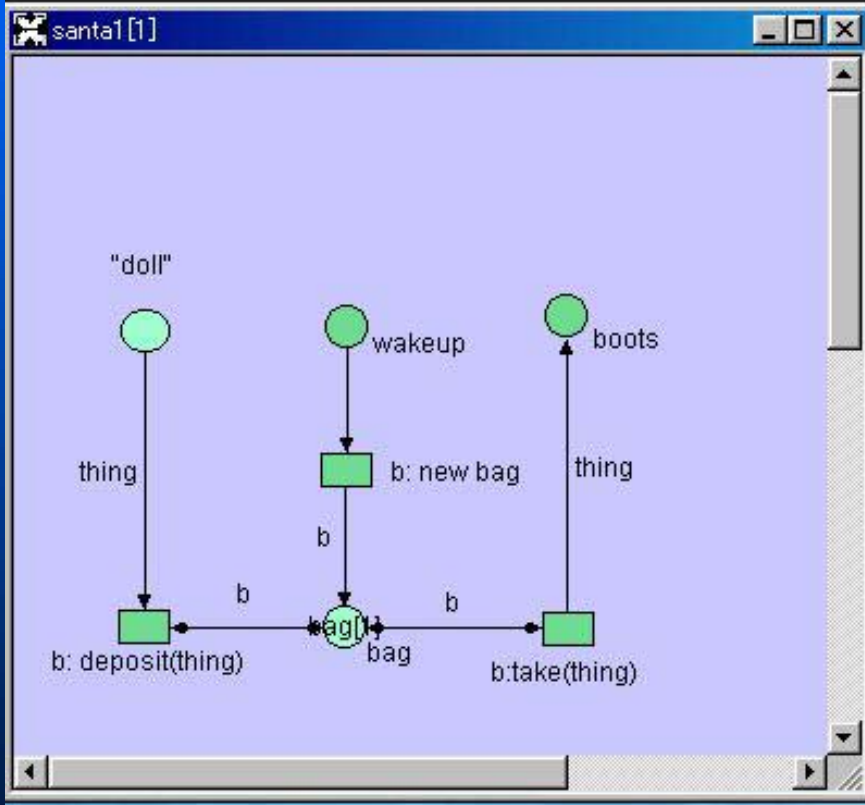
Simulation step completed.



Reference Net Workshop

File Edit Layout Attributes Simulation Drawings  
Breakpoints Help

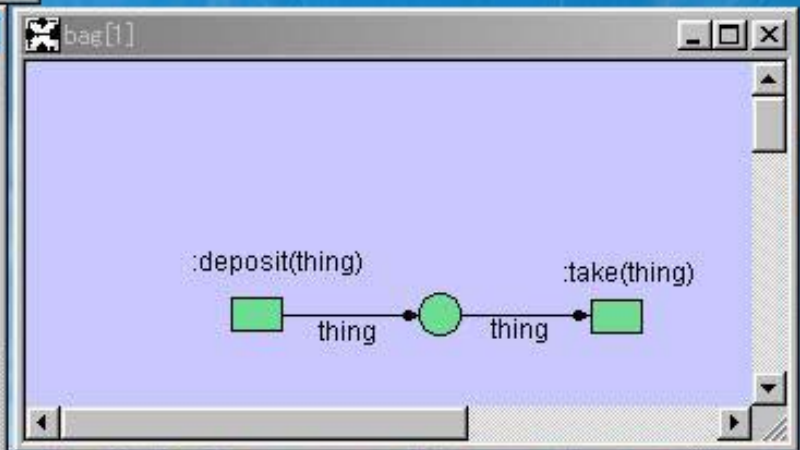
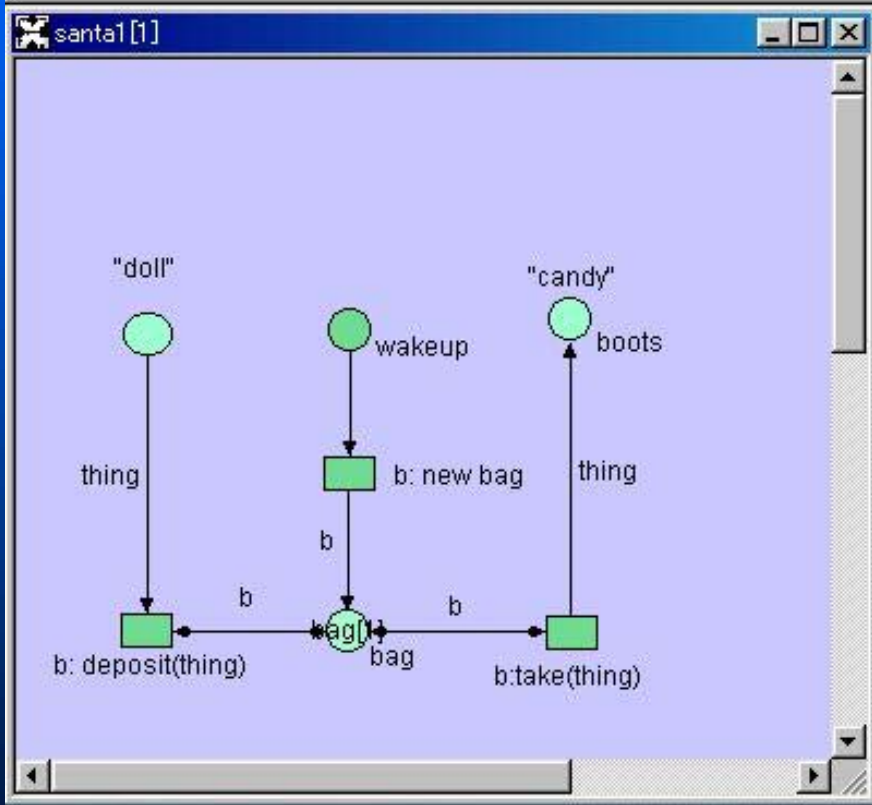
Selection Tool



Reference Net Workshop

File Edit Layout Attributes Simulation Drawings  
Breakpoints Help

Selection Tool



# エージェントとオブジェクトの違い (1)

## 1. 何がエージェント / オブジェクトの挙動に対する制御権を持つか?

- オブジェクトは自身の内部状態への制御権は持つが、その挙動に対する制御権はメソッドを呼び出した側がもつ。メソッドを呼び出したときに、指定したオブジェクトがそのメソッドを持たないならば、それはエラーとして扱われる。
- エージェントは自分自身の挙動に対する制御権をもつ。エージェントは他のエージェントからある動作の実行を要求されたとき、もし自己の目的にかなうならば実行するが、そうでないときは実行を拒否できる。

# エージェントとオブジェクトの違い (2)

## 2. オブジェクトに対応するトークンが何を表すか?

- リファレンス意味論 (reference semantics) と値意味論 (value semantics).
- リファレンス意味論では、トークンはオブジェクトそのものではなくそれへのリファレンスとして扱われる。したがって、トランジションの発火によってトークンが複製されても、それはリファレンスが複製されるだけで、実体であるオブジェクトが複製されるわけではない。
- 値意味論では、トークンはオブジェクトそのものを表し、トークンの複製はオブジェクトの複製を意味する。
- トークンをエージェントに対応させ、エージェントの動的な生成、消滅を表現したいときには、値意味論の方が自然である。



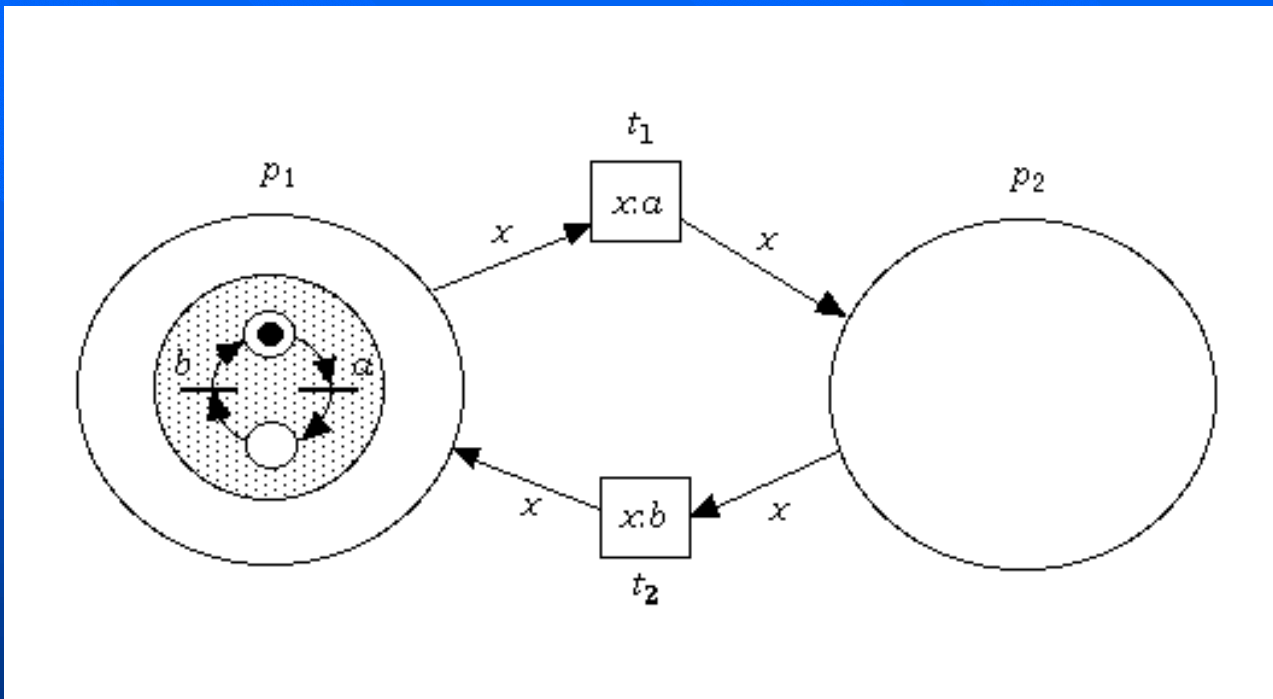
# エージェントとオブジェクトの違い (3)

3. エージェントは独立したスレッドで実行されるべきであるが、オブジェクトは必ずしもそうではない。

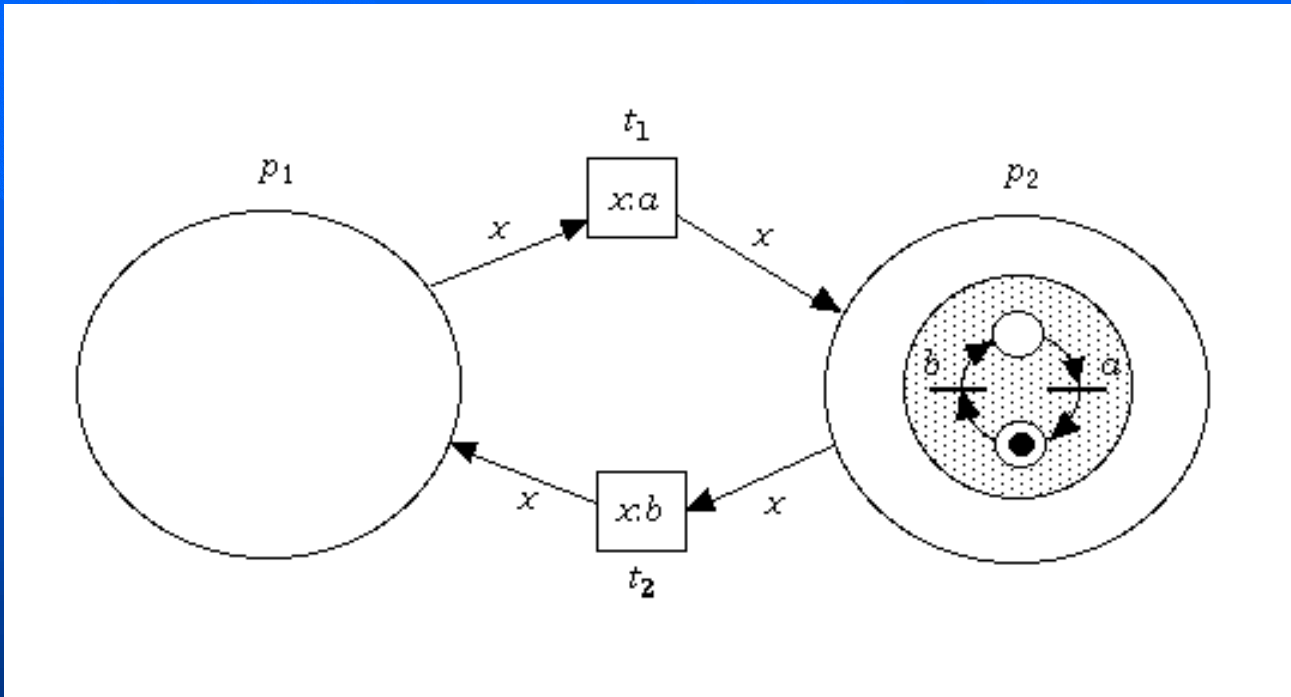
# $PN^2$ (Petri Nets in a Petri Net)

- マルチエージェント・システムの解析を目的としたペトリネット・ベースのモデルである。高水準ペトリネットのような複雑な変数の束縛や式の評価などは必要とせず、立場としてはEOSに近い。
  1. エージェントが動作する環境を表す1つの上位ネットとエージェントを表す複数の下位ネットから構成される。
  2. 動的なエージェントの生成、消滅を表現できる。
  3. 上位ネットのトランジションに対し、それと同期して発火する下位ネットのトランジションが動的に決定される。これで、エージェント間の通信が表現できる。
- 下位ネットがセーフな状態機械(有限オートマトン)であるような $PN^2$ をFSAN (Finite State Agent Nets) とよぶ。

# 狀態遷移

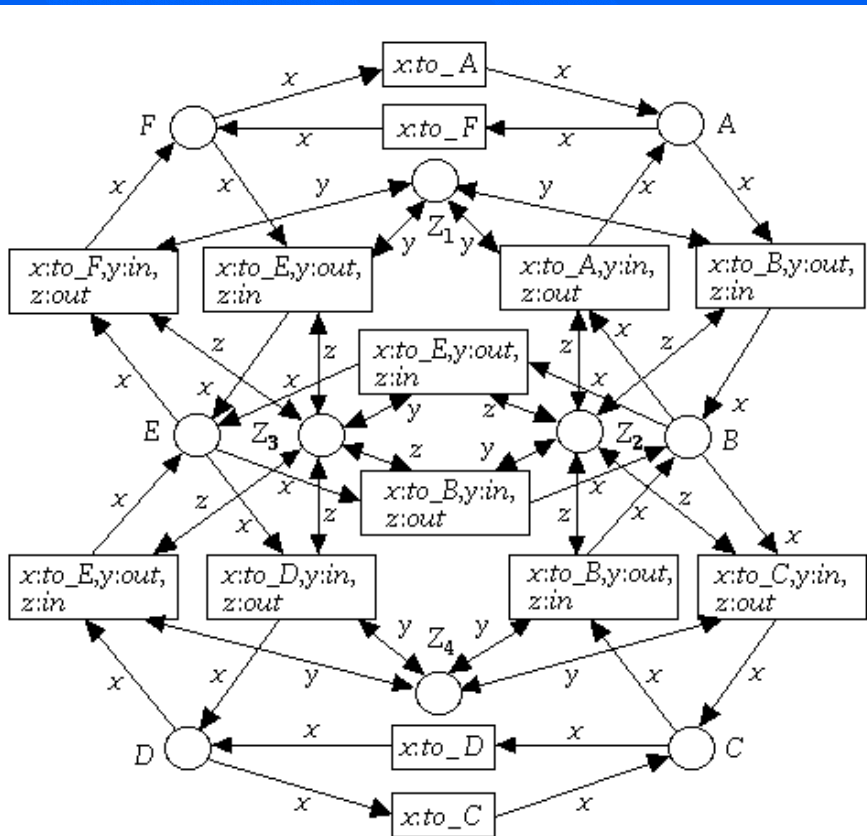


$$[\mu, \emptyset] \xrightarrow{a} [\emptyset, \mu' = \delta(\mu, a)] \xrightarrow{b} [\mu = \delta(\mu', b), \emptyset] \xrightarrow{a} \dots$$

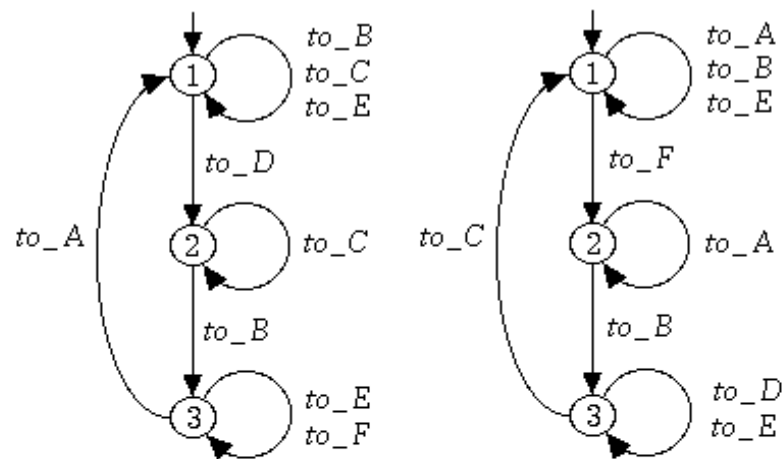


$$[\mu, \emptyset] \xrightarrow{a} [\emptyset, \mu' = \delta(\mu, a)] \xrightarrow{b} [\mu = \delta(\mu', b), \emptyset] \xrightarrow{a} \dots$$

# PN<sup>2</sup> によるAGVシステムのモデル化 (1)

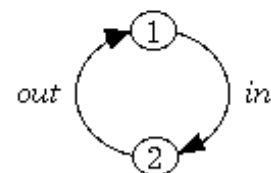


上位レベルネット



AGV1

AGV2

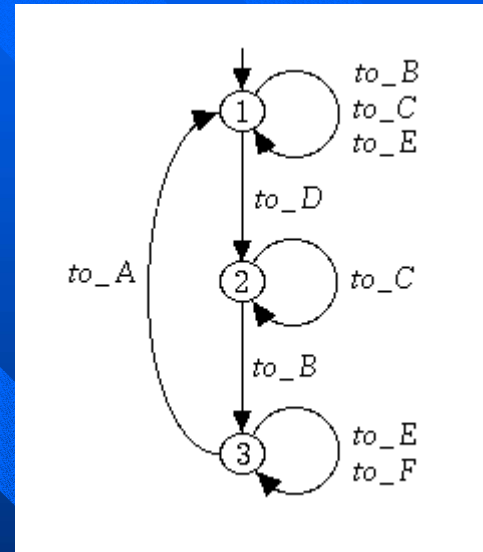
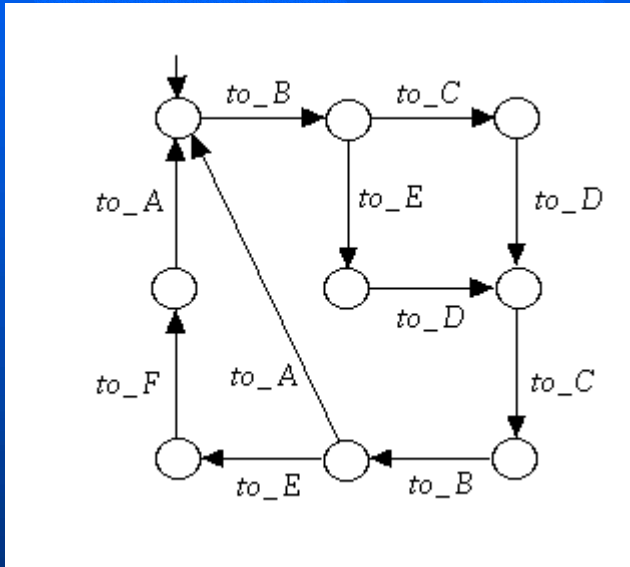


Zone  $i$  ( $i = 1, \dots, 4$ )

下位レベルネット

# PN<sup>2</sup> によるAGVシステムのモデル化 (2)

## 記述の最適化



$$AGV_1 = A(\rightarrow B \rightarrow (C | E) \rightarrow D \rightarrow C \rightarrow B \rightarrow (A | E \rightarrow F \rightarrow A))^*$$

# おわりに

- マルチエージェント・システムのモデル化に関する研究は、従来はどのようにシステムを記述するかが中心であった。今後は、記述したモデルが仕様どおりに動作するかを効率的に検証する問題など、解析面での研究の進展が望まれる。
- 移動エージェントを記述するための形式的モデルとして  $\pi$ -計算が提案されているが、ペトリネット・ベースのモデルとの関係についても考察する必要がある。
  - $\pi$ -計算では通信リンクを動的に変化させることによりエージェントの移動を表現する点でペトリネット・ベースのモデルと異なる。