

Хијерархијске ПМ

Хијерархијска ПМ

Омогућава приказивање декомпозиције процеса на различитим нивоима детаљности разлагањем великог модела на већи број мањих.

Хијерархијска ПН се увек може превести у одговарајућу нехијерархијску, и обрнуто.

Хијерархијска и њој еквивалентна нехијерархијска мрежа имају иста својства и исто понашање.

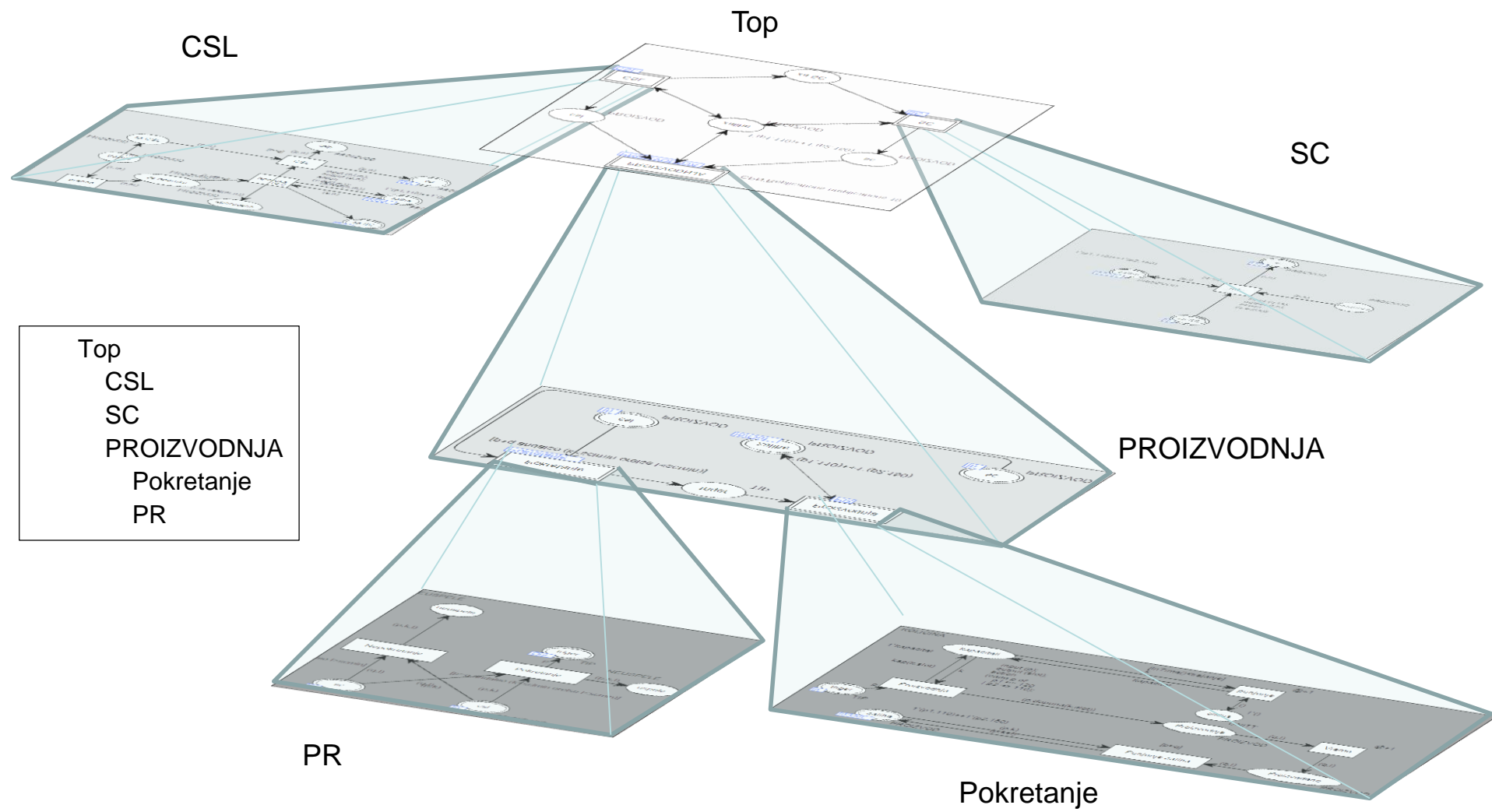
Хијерархијска ПМ

Хијерархијска ПН се формира заменом прелаза подмрежом или успостављањем хијерархије између постојећих мрежа.

Прелазу (*Substitution Transition*) и гранама које га окружују додељује се мрежа (подмрежа) која детаљније описује активност коју прелаз представља. Мрежа на којој је посматрани прелаз се назива надмрежа.

Подмрежа је повезана са остатком ПН истим местима са којима је повезан прелаз који је њен надчвор. Ова места се јављају на два места:

- у надмрежи, као *socket* места и
- у подмрежи, као везни (*port*) чворови. Везни чворови имају додатну ознаку *In*, *Out* или *In/Out*, у зависности од тога да ли су улазна, излазна или улазно-излазна места надчвора.



Формална дефиниција хијерархијске ОПМ

Хијерархијска ОПН је деветорка $HCPN = (S, SN, SA, PN, PT, PA, FS, FT, PP)$, где је:

(i) S скуп страна таквих да важи:

Свака страна $s \in S$ је нехијерархијска ОПМ, тј. деветорка $(\Sigma_s, P_s, T_s, A_s, N_s, C_s, G_s, E_s, M_{0s})$.

Скупови елемената су дисјунктни: $\forall s_1, s_2 \in S: [s_1 \neq s_2 \Rightarrow (P_{s_1} \cup T_{s_1} \cup A_{s_1}) \cap (P_{s_2} \cup T_{s_2} \cup A_{s_2}) = \emptyset]$.

(ii) $SN \subseteq T$ је скуп надчворова.

(iii) SA је скуп функција придруживања, које придружују стране из скупа S надчворовима из скупа SN , тако да ниједна страна не може да буде подстрана сама себи.

(iv) $PN \subseteq P$ повезивање се врши преко *port* места.

(v) PT – функција која дефинише тип *port* места {улазно, излазно, улазно/излазно}.

(vi) PA – функција придруживања: сваком *socket* месту из надстранице се додељује одговарајуће *port* место из подстранице, водећи рачуна да су места истог типа.

(vii) $FS \subseteq P_s$ коначан скуп скупова фузије тако да сви чланови једног скупа фузије имају исти тип података (*colour set*) и почетно маркирање.

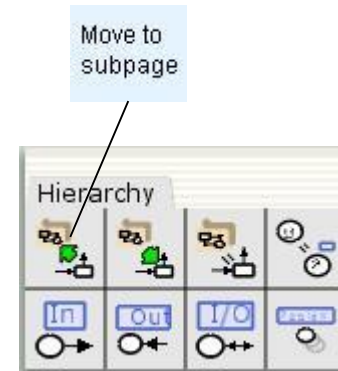
(viii) FT – функција која одређује тип фузије.

(ix) $PP \in S_{MS}$ је фамилија страна.

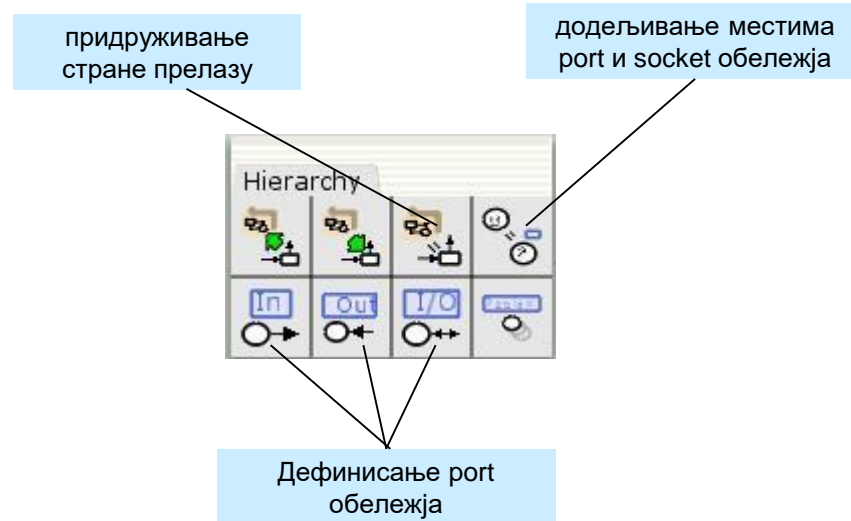
CPN Tools: Формирање хијерархијске ПМ

Два начина:

1. Додељивање нове подмреже (странице) прелазу



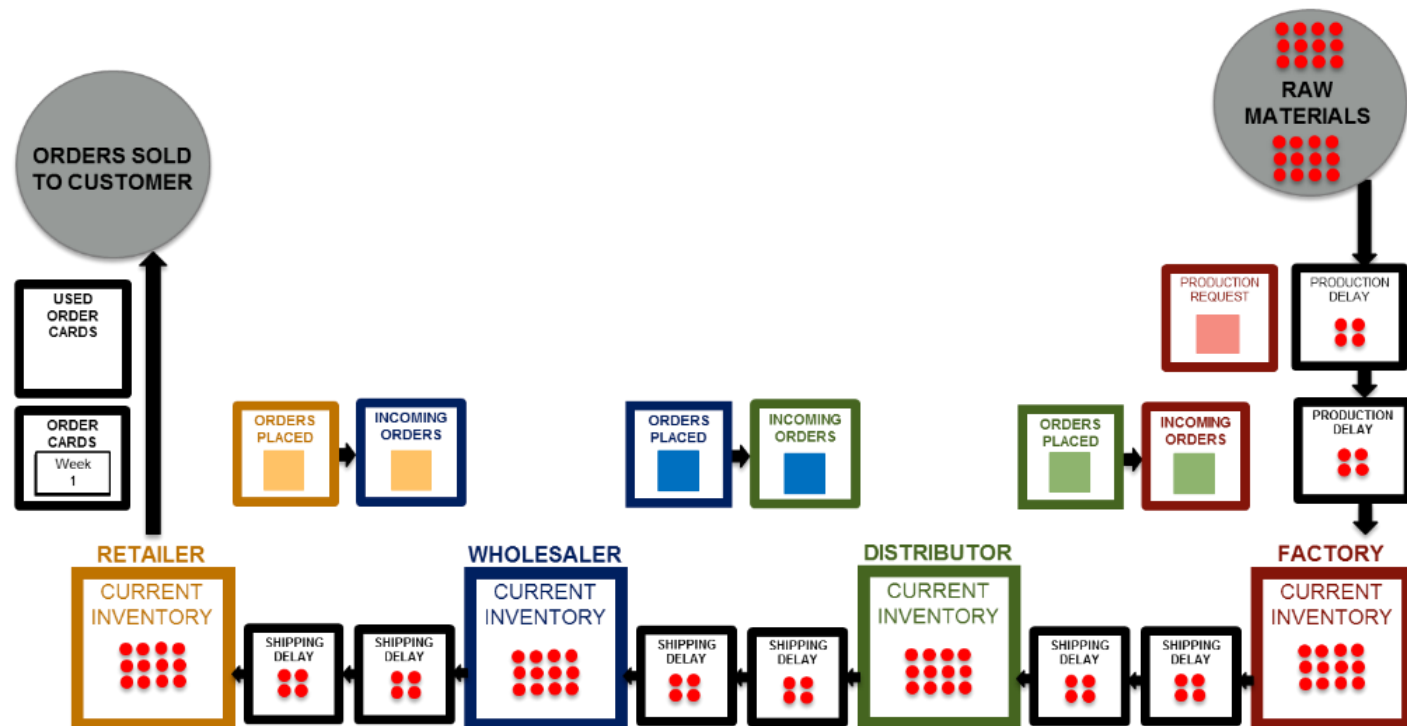
2. Успостављање хијерархије између постојећих мрежа (страна)



Пивска игра

Четири ешалона ЛС: малопродавац, veleпродавац, дистрибутер и произвођач.

Сваки учесник има на залихама 12 гајби; почетна тражња сваке фазе је 4 гајбе у прве три седмице. Након тога, тражња потрошача почиње да варира.



Пивска игра

Малопродавац настоји да задовољи тражњу купца из својих залиха.

Свака незадовољена поруџбина остаје за наредни период као дуг (*backorder*).

Малопродаја поручује од великопродаје која покушава да задовољи ову тражњу из својих залиха.

Велепродавац поручује од дистрибутера, дистрибутер од произвођача, а произвођач наручује сировине од снабдевача.

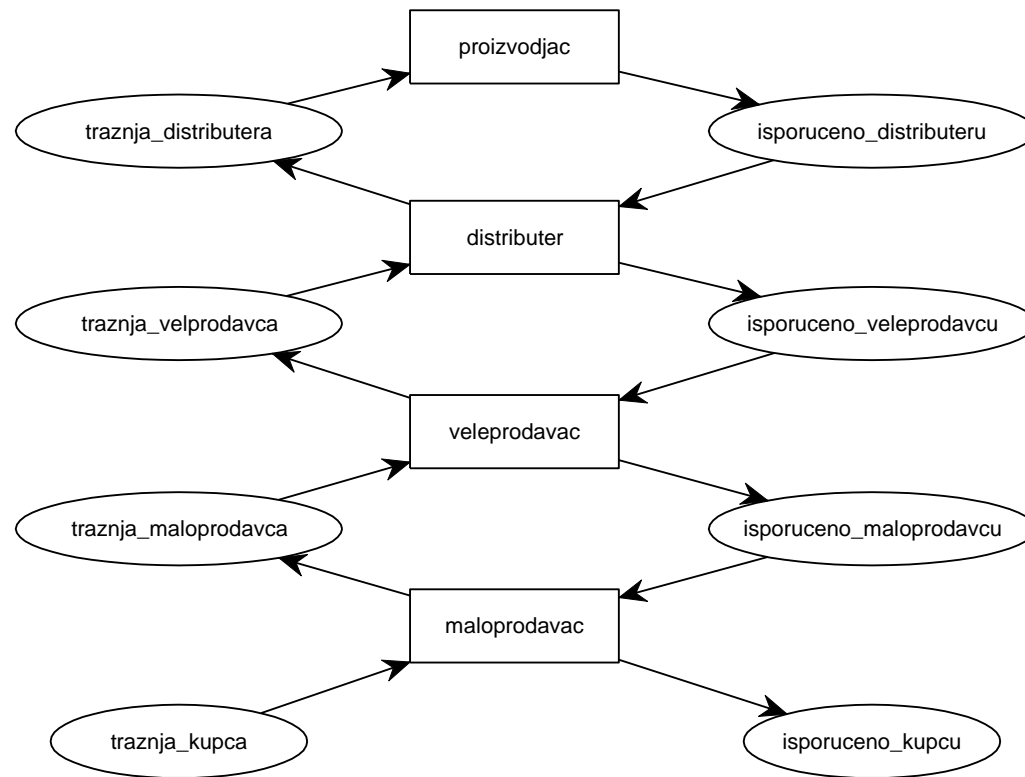
Поруџбине су тренутне а испорука између два ешалона траје две временске јединице.

Сваки учесник у ЛС предвиђа тражњу имајући у виду време испоруке, па на основу тога поручује.

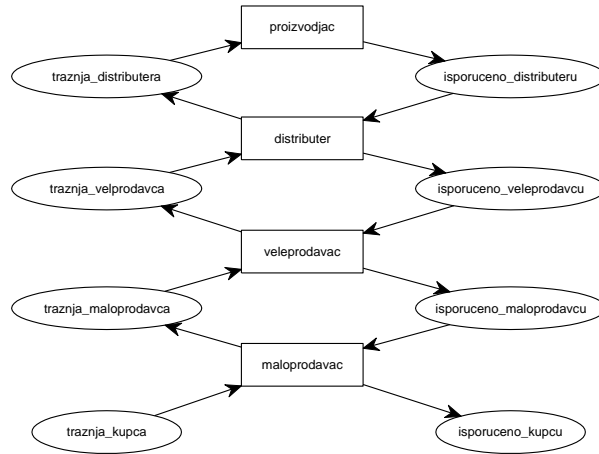
Игра траје 50 симулираних недеља.

Циљ: минимизација трошкова целог ЛС. Трошкови складиштења износе 0.50\$ по гајби недељно, а трошкови незадовољења тражње су 1\$ по гајби недељно.

ПМ ланца снабдевања на највишем нивоу хијерархије



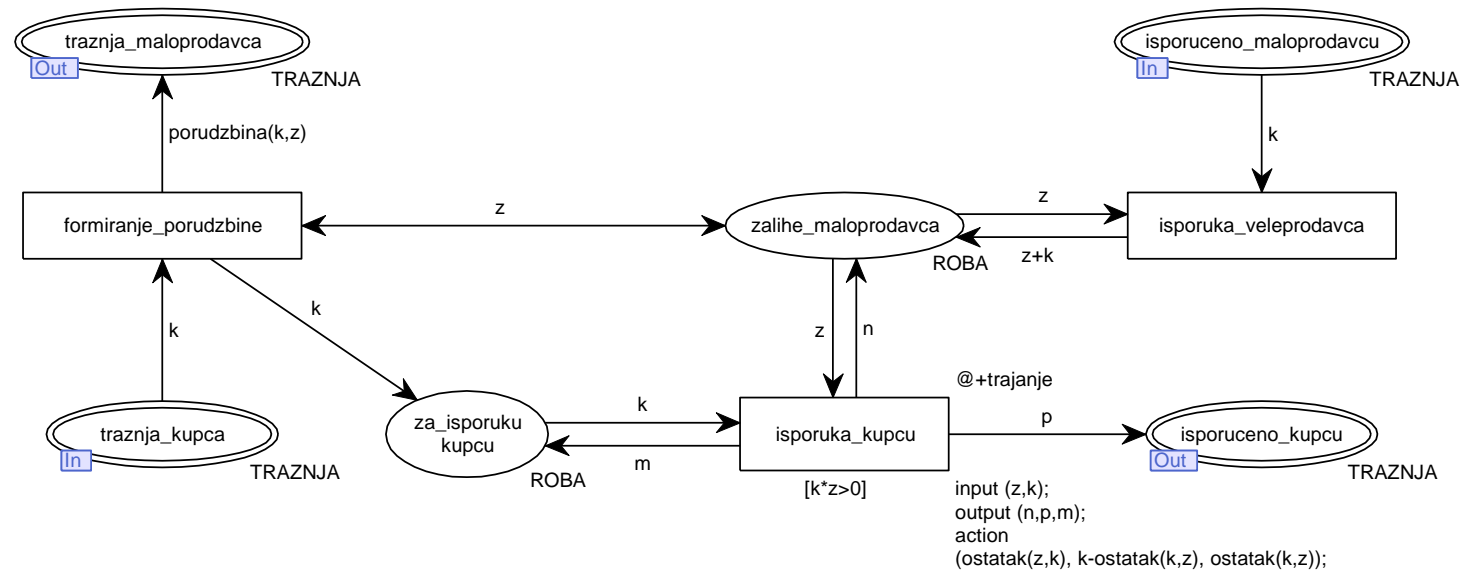
ПМ ланца снабдевања на највишем нивоу хијерархије и подмрежа придружена прелазу *maloprodavac*



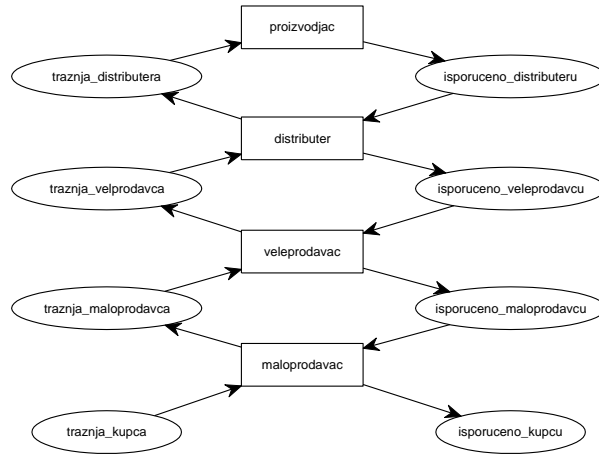
(* Standard declarations *)

```

colset INT = int;
colset TRAZNJA=int timed;
colset ROBA=int;
var k: TRAZNJA;
var z, n, m, p:ROBA;
fun ostatak(x,y)=if (x>=y) then x-y else 0;
fun porudzbina(x,y) = if y>=x then x else x+x-y;
val trajanje=2;
  
```

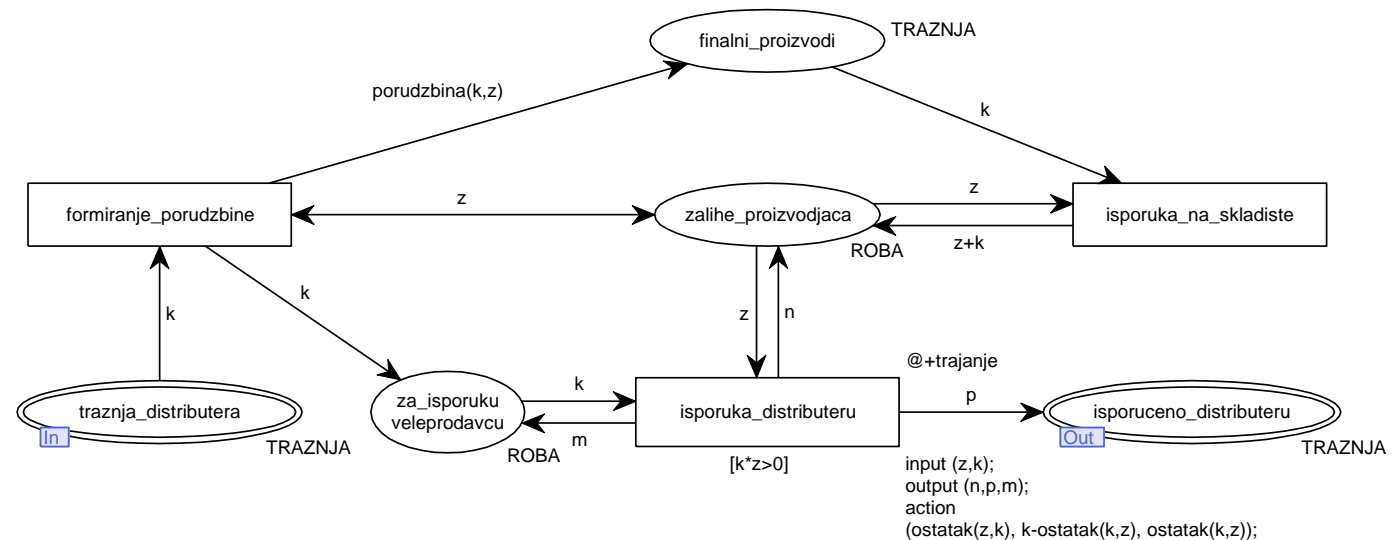


ПМ ланца снабдевања на највишем нивоу хијерархије и подмрежа придружена прелазу *proizvodjac*



(* Standard declarations *)

```
colset INT = int;
colset TRAZNJA=int timed;
colset ROBA=int;
var k: TRAZNJA;
var z, n, m, p:ROBA;
fun ostatak(x,y)=if (x>=y) then x-y else 0;
fun porudzbina(x,y) = if y>=x then x else x+x-y;
val trajanje=2;
```



Тражња купца

Нека је тражња у првих 10 недеља детерминистичка а у остатку периода нека подлеже нормалној расподели са математичким очекивањем 6 гајби и варијансом од 1 гајбе.

