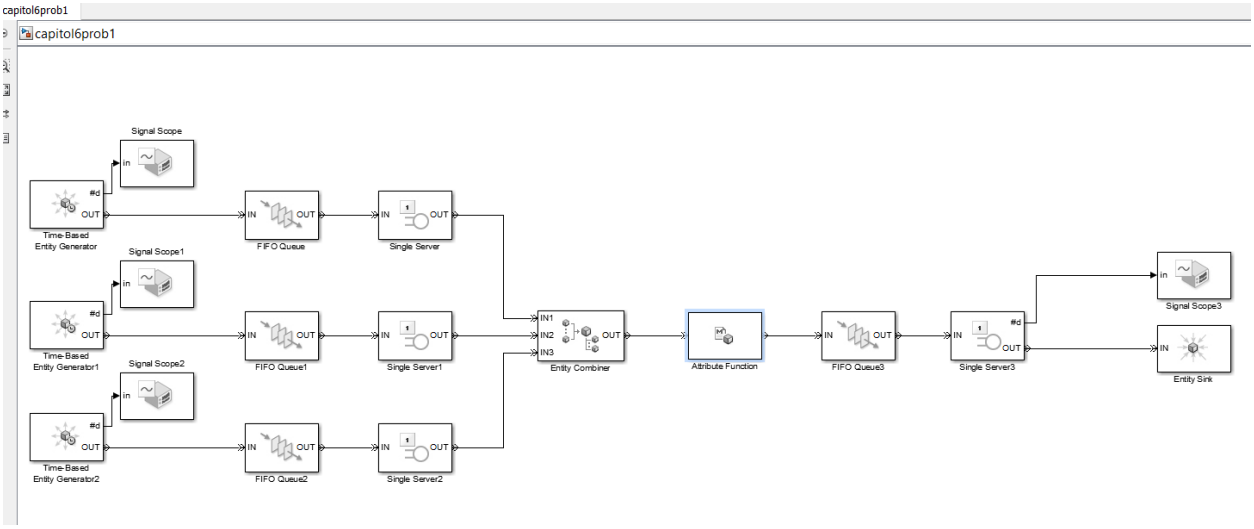
Draghici Andreea-Maria
CR 3.1B

Capitol 6 / Lucrarea 6

Gestionarea entitatilor

Draghici Andreea-Maria
CR 3.1B

Problema 1

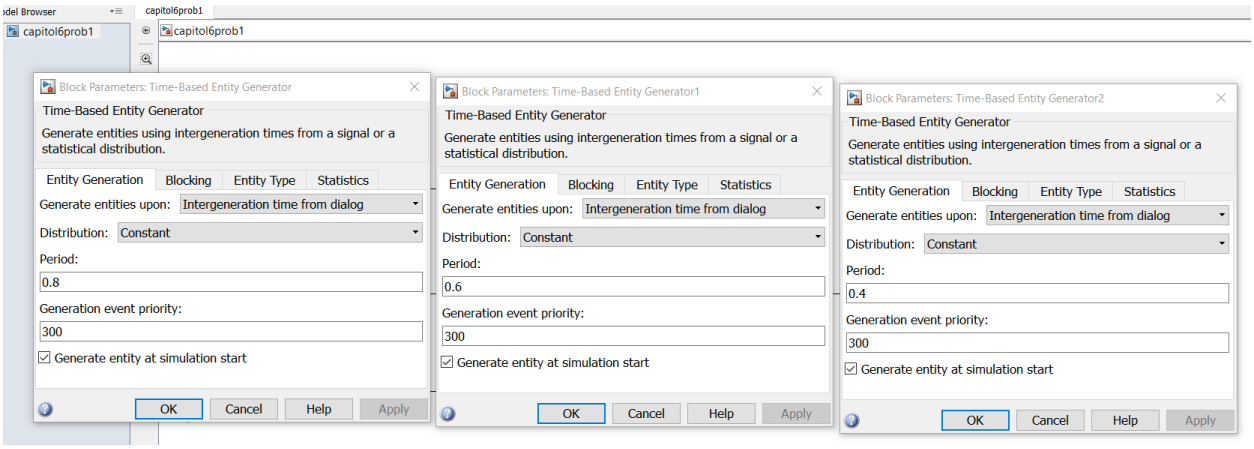


Fiecare ansamblu are propria statie de deservire. Fiecare entitate generata este stocata intr-o coada si ulterior “testata” intr-o statie de deservire. Dupa testari, piesele sunt combinate cu ajutorul blocului “Entity Combiner” si urmeaza o noua prelucrare ce reprezinta asamblarea pieselor.

Subansamblu	t_{gen} [h]	t_{test} [h]
Motor	0.8	0.6
Transmisie	0.6	0.4
Sasiu	0.4	0.4

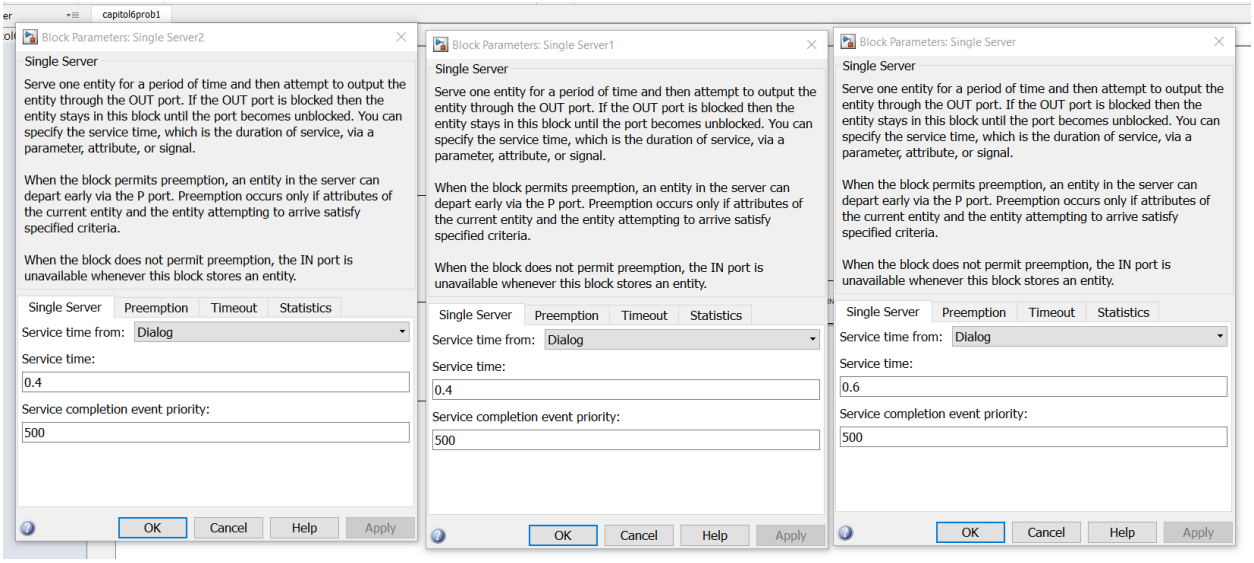
Tabelul 2. Date pentru linia de asamblare

Blocurile Time-Base Generator genereaza entitati de tipul : şasiu, motor și transmisie la interval de timp t_{gen}.

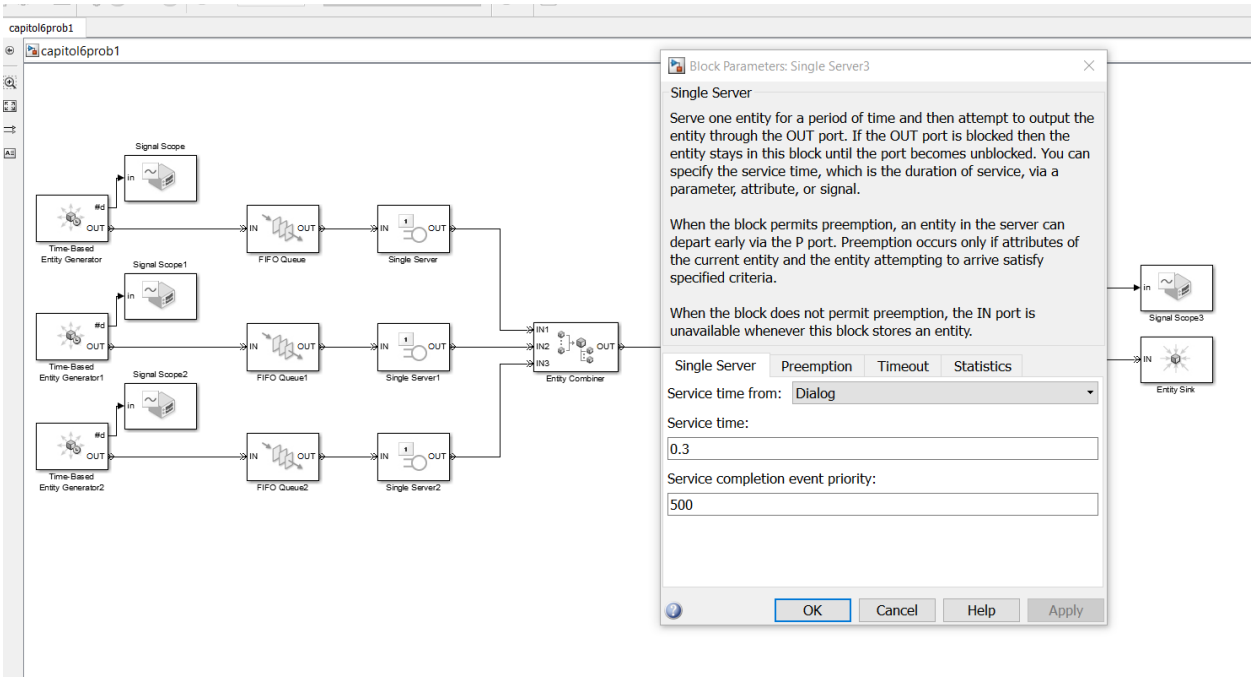


Draghici Andreea-Maria
CR 3.1B

Fiecare subansamblu este testat in statii un interval de timp t_{test}. Timpul de testare este reprezentat de timpul de deservire pentru blocul Single Server.

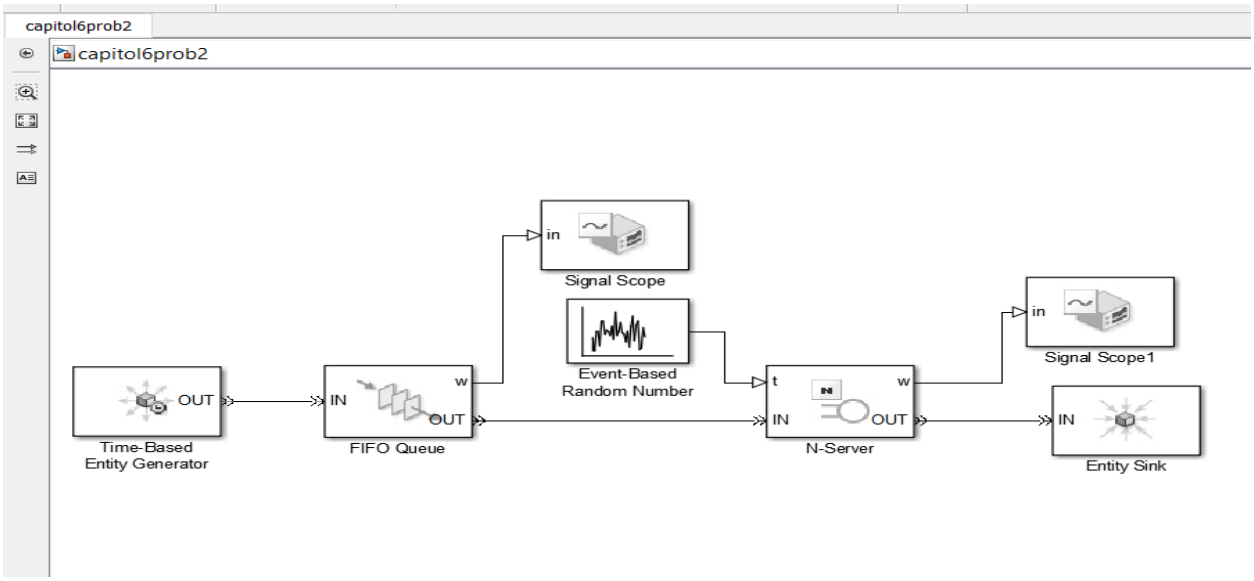


Asamblarea subansamblelor dureaza t_{ans}=0.3h



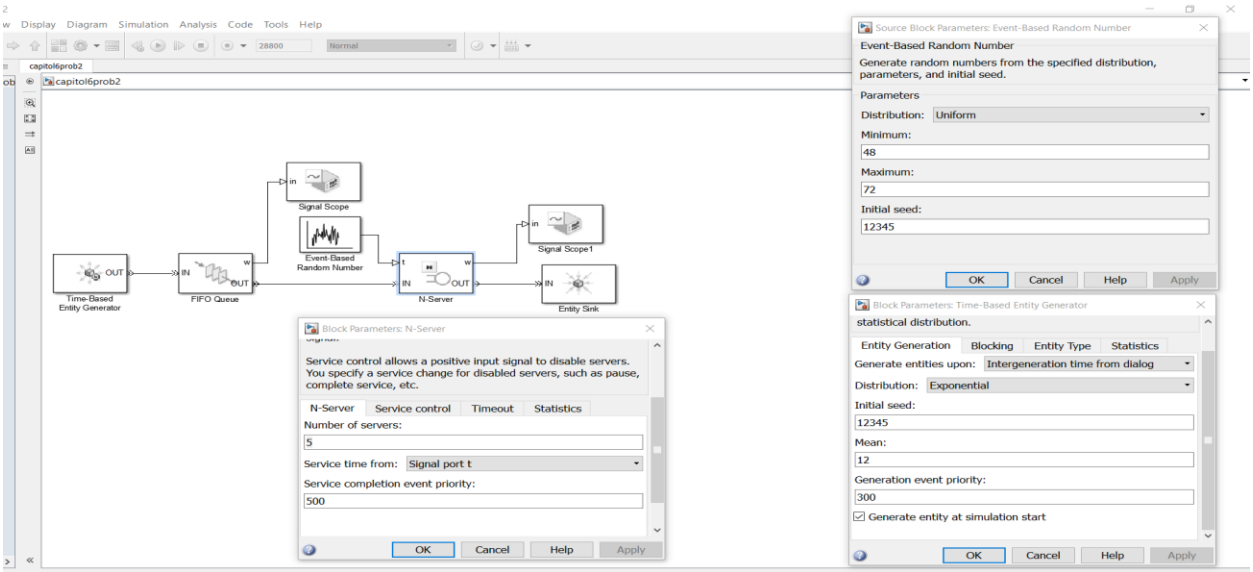
Draghici Andreea-Maria
CR 3.1B

Problema2



Cunoastem ca cererile apar la intervale cu distributie exponentiala cu valoare medie de 0.2 minute, deci intervalul de aparitie dintre doua entititati este egal cu 0.2 minute adica 12 secunde, iar distributia este exponentiala. Folosim un N-Server deoarece se cere un multiserver cu capacitatea de cinci. Cum distributia etse uniforma vom folosi un Event-Based Random Number pentru a seta timpul, durata convorbirii fiind de 1+(-)0.2, minimul este de 0.8 minute, adica 48 secunde, iar maximul este de 1.2 minute, adica 72 secunde.

In concluzie, timpul de prelucrare se asociaza ca si semnal de la blocul Event-Based Random Number pentru a avea distributie uniforma.

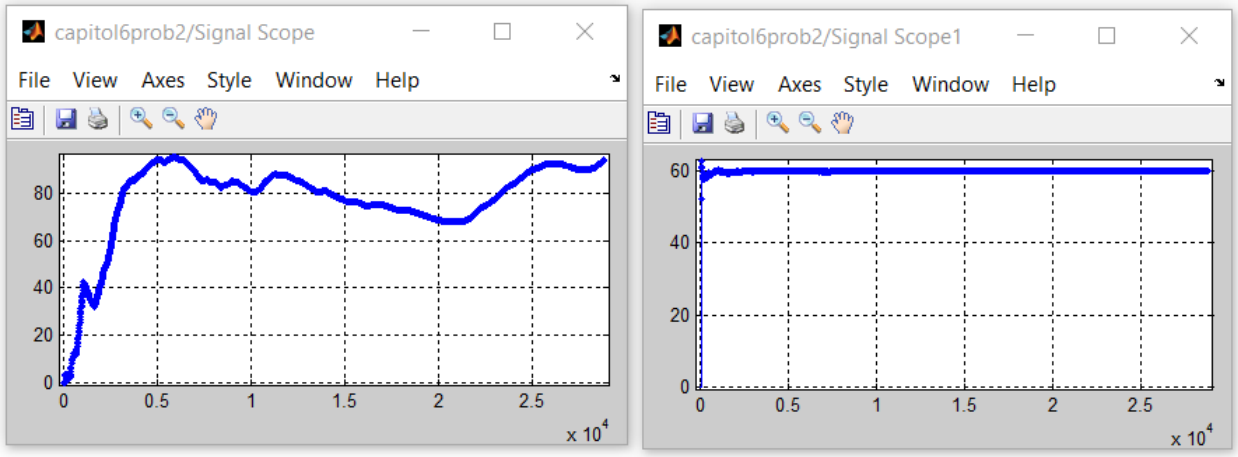


Blocul N-Server seteaza numarul operatorilor din compania Telefonica. (in cazul problemei, exista 5 operatori).

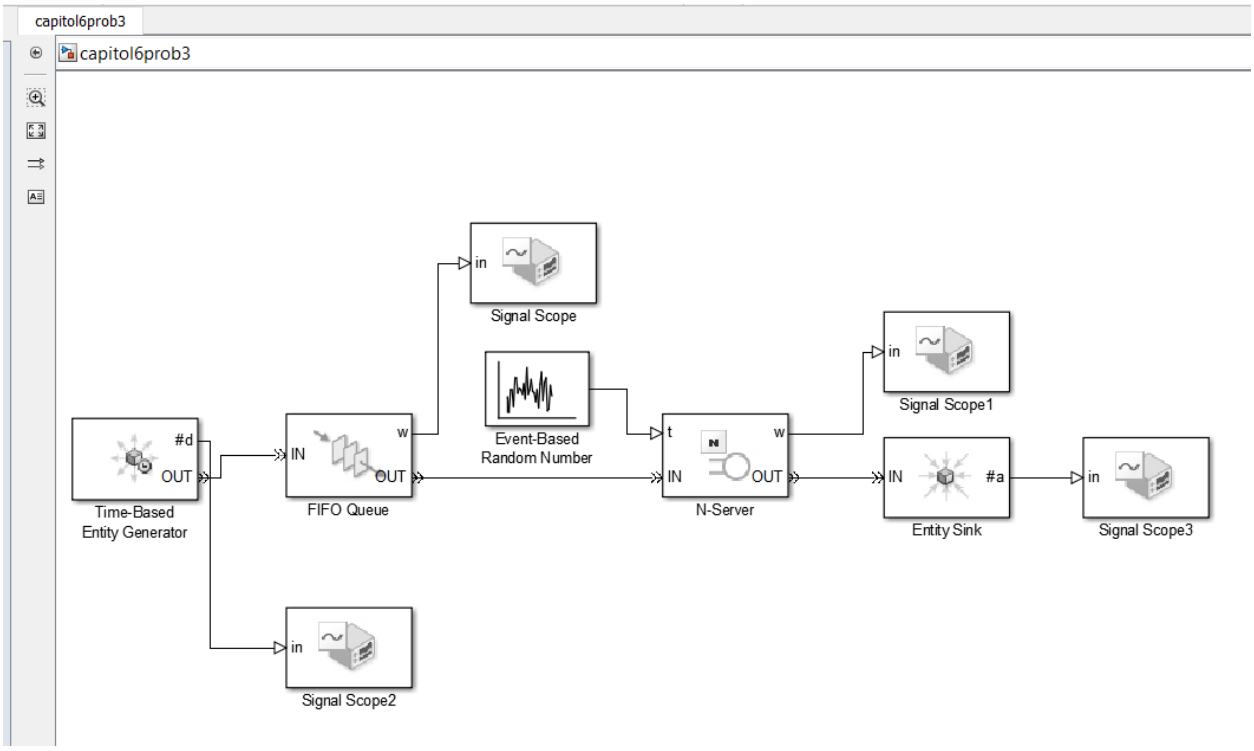
Draghici Andreea-Maria
CR 3.1B

Blocul Generator genereaza cererile cu o distributie exponentiala cu valoarea media 0.2, iar durata unei conversatii este data de blocul Random Number.

Rezultate obinute in urma simularii pe 8 ore, adica 28800 secunde:



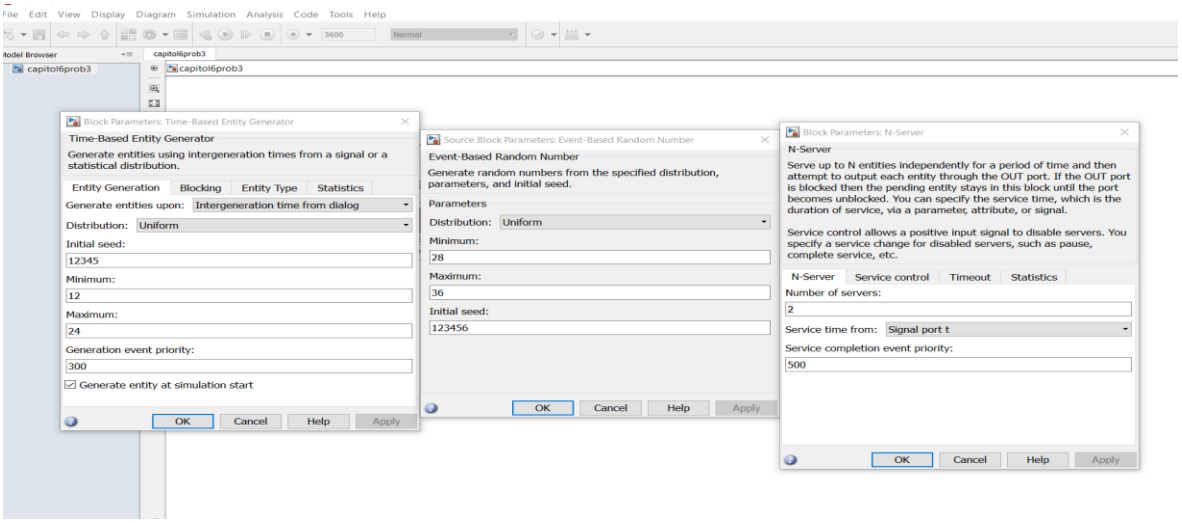
Problema 3



Blocul Time-Based Generator genereaza cele 2 tipuri de loturi cu distributie uniforma, iar prelucrarea unui lot este realizata de blocul Random Number. Piesele sunt generate in blocul Time-Based Entity Generator si stocate intr-o coada. Ulterior vor fi procesate in statia de deservire din figura de mai sus.

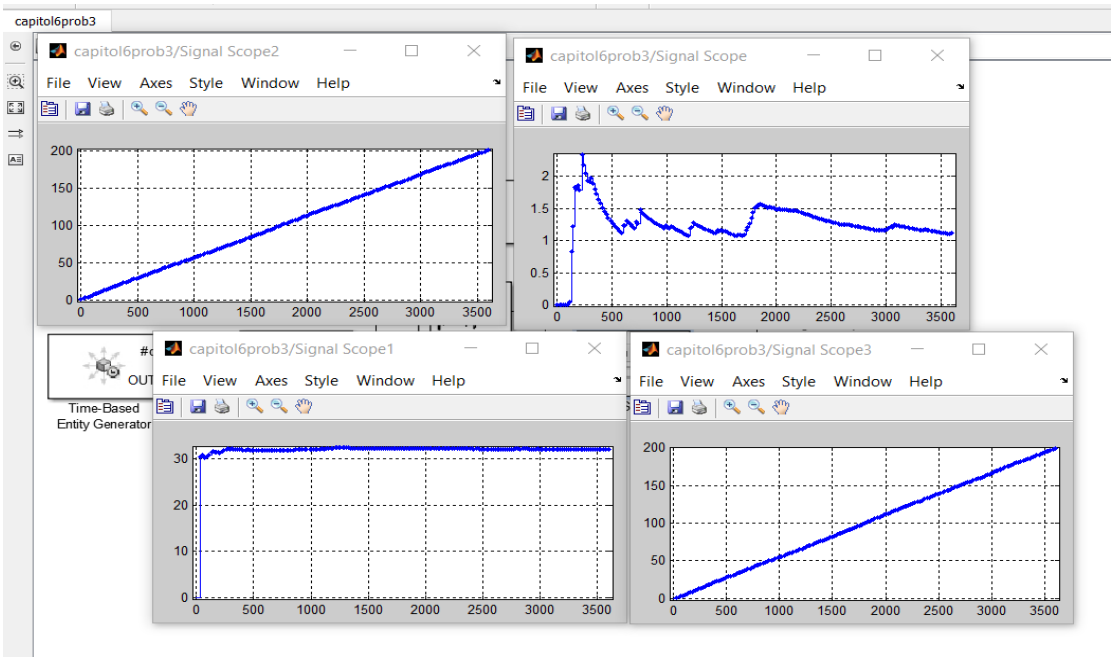
Draghici Andreea-Maria
CR 3.1B

Loturile sosesc la interval de 18+(-)6secunde cu distributie uniforma, deci intervalul de aparitie dintre 2 entitati va avea valori cuprinse intre 12 si 24 secunde. Prelucrarea unui lot dureaza 32+(-)4secunde cu distributie uniforma, deci intervalul de prelucrare va avea valori cuprinse intre 28 si 36 secunde. Loturile vor fi prelucrate de o statie cu capacitatea 2. Statia de preluare a loturilor este N-server care are capacitatea 2.



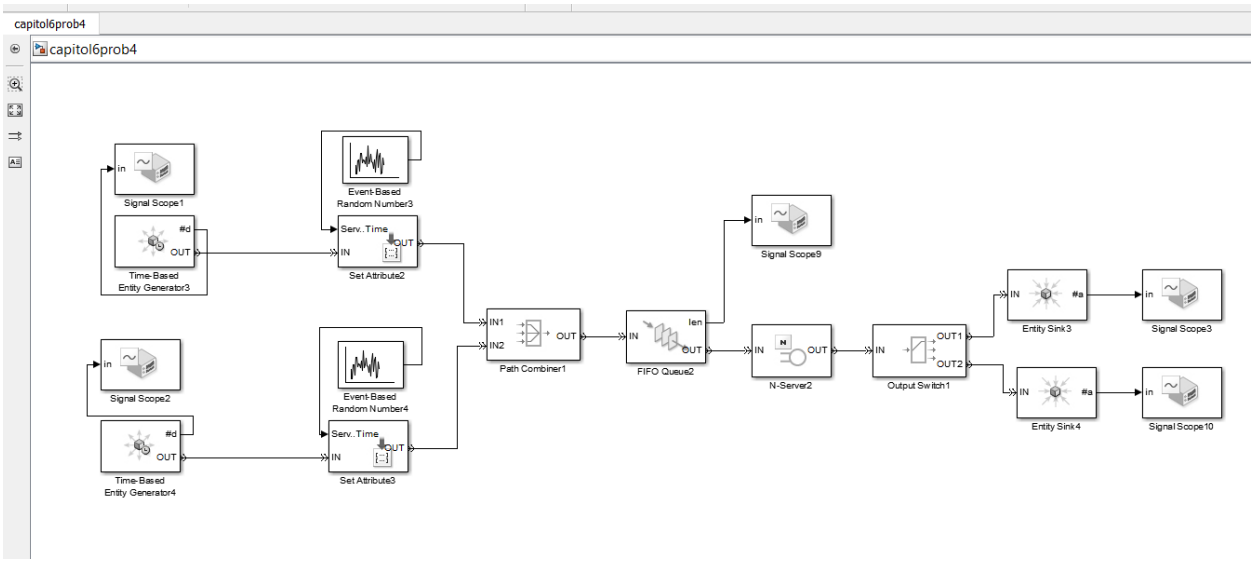
Rezultate obtinute in urma simularii pe o ora adica 3600 secunde:

Signal Scope2- afiseaza nr de entitati deservite;
Signal Scope-afiseaza timpul mediu de asteptare in coada;
Signal Scope1-afiseaza timpul mediu de deservire;
Signal scope3-afiseaza numarul de loturi prelucrate.

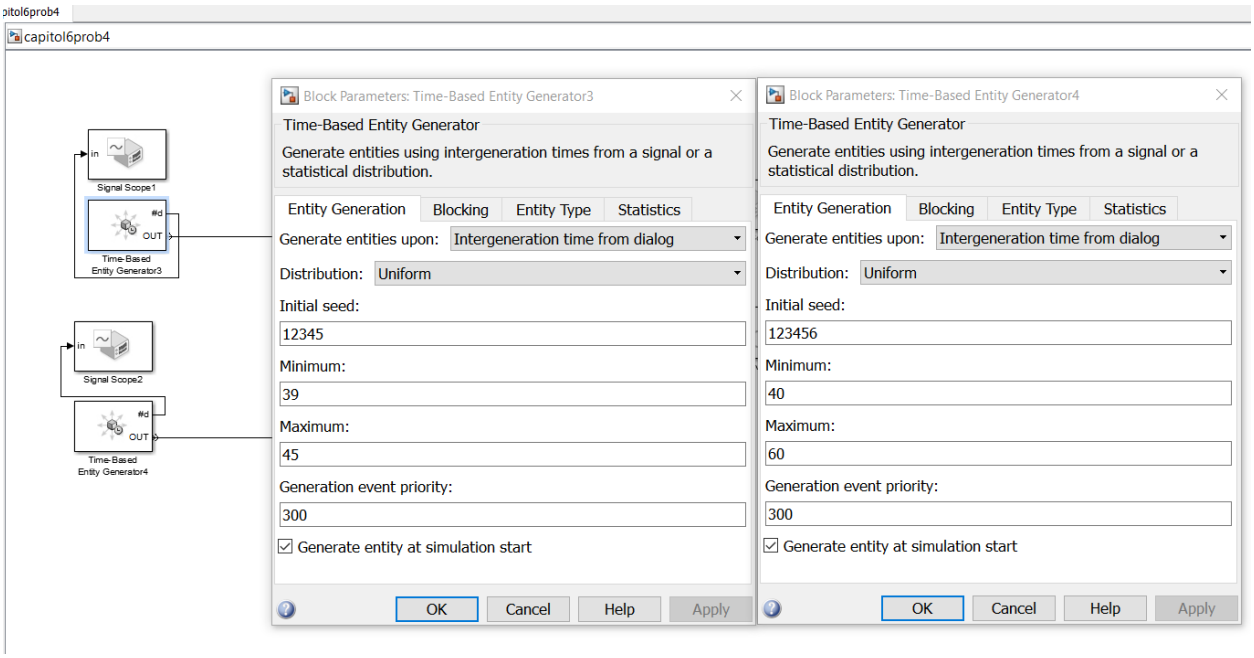


Draghici Andreea-Maria
CR 3.1B

Problema 4



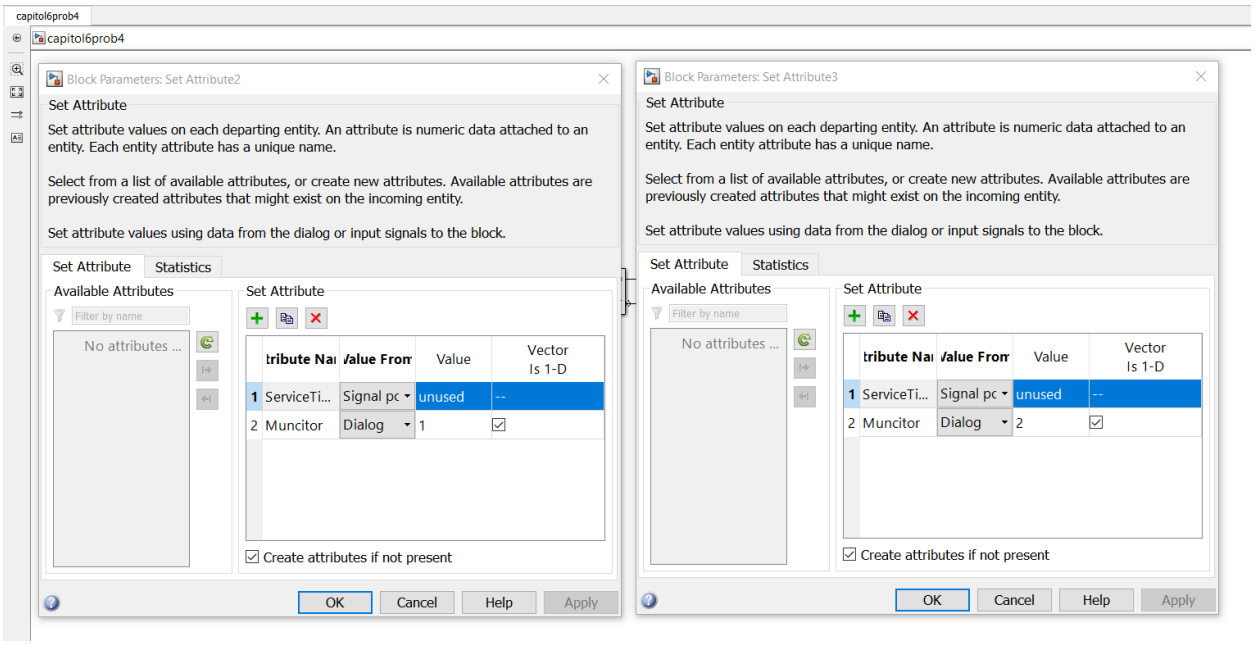
Cele 2 tipuri de muncitori sunt generate de cele 2 blocuri Generator , fiecare cu distributie uniforma, iar N-Server este statia de prelucrare acelor 2 tipuri de piese.



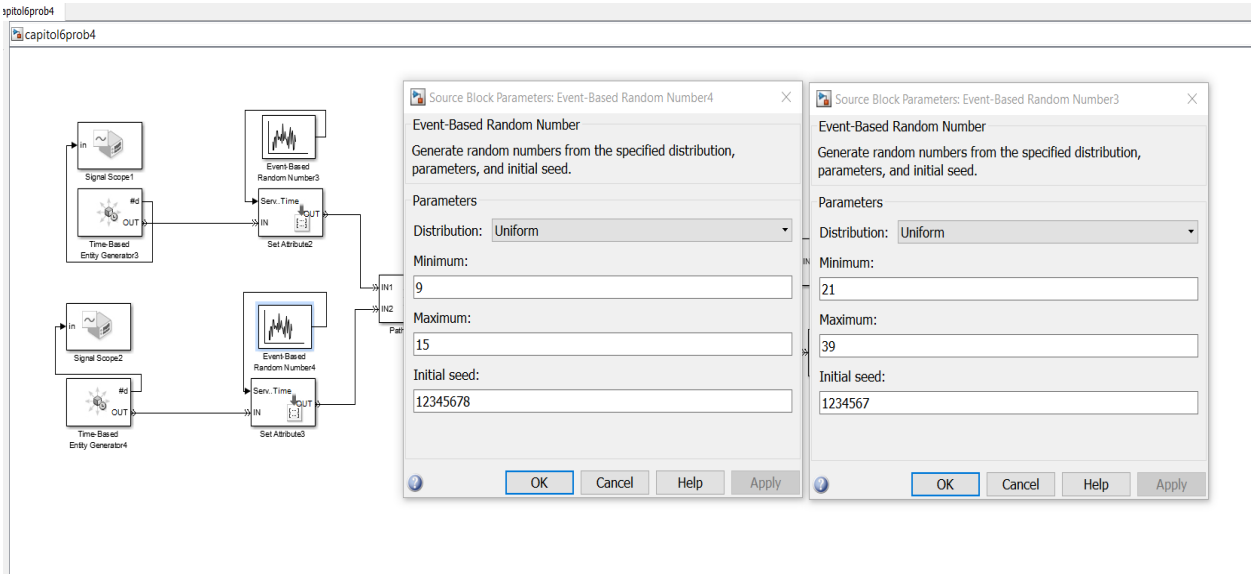
Pentru combinarea celeo 2 tipuri de piese am folosit blocul Path Combiner, iar pentru diferentierea lor dupa atributul Muncitor, am folosit blocul Output Switch.

Fiecare tip de muncitor are atribuit 2 atribute: Service time si Muncitor care poate avea valorile 1 sau 2 in functie de tipul muncitorului.

Draghici Andreea-Maria
CR 3.1B

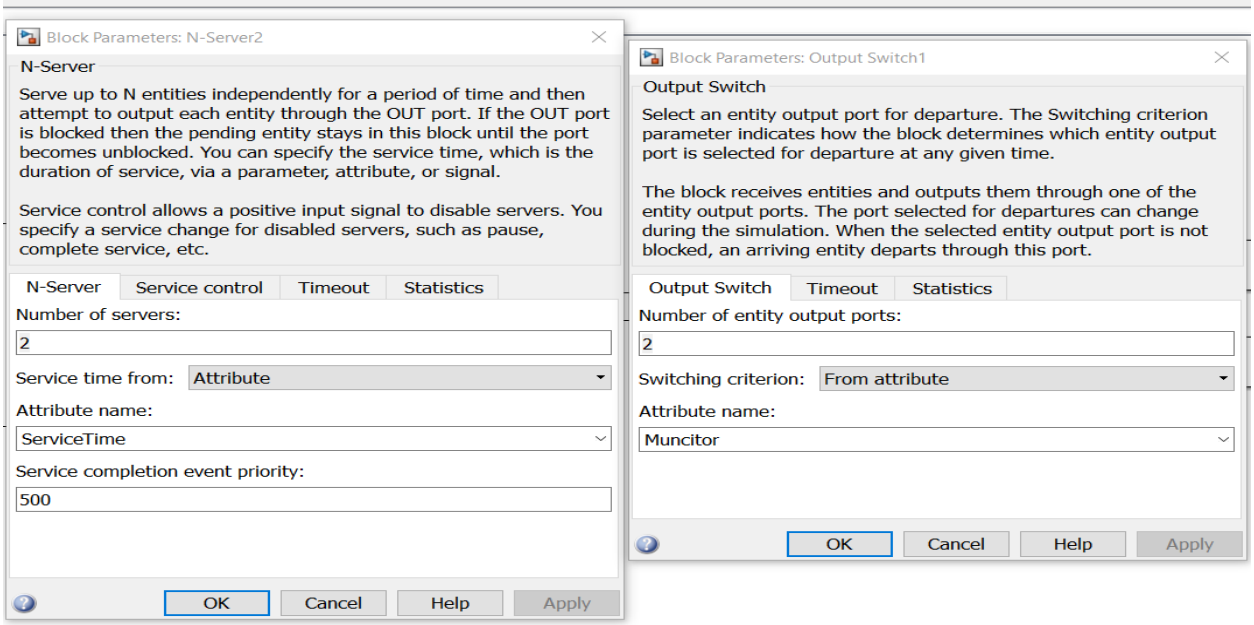


Pentru fiecare tip de muncitori exista un timp de deservire diferit, pentru asta setam cate un Service Time pentru fiecare tip folosind cate un bloc Event-Based Random Number cu distributie uniforma, acesta va fi transmis ca input in blocul Set Attribute. In acelasi timp setam si atributul cu cate o valoare pentru fiecare tip care mai tarziu va fi folosit pentru blocul Output Switch. Acest bloc dirijeaza entitatile dupa prioritate catre blocurile Entity Sink, separandu-le, criteriul de selectare a iesirii este valoarea atributului Muncitor.



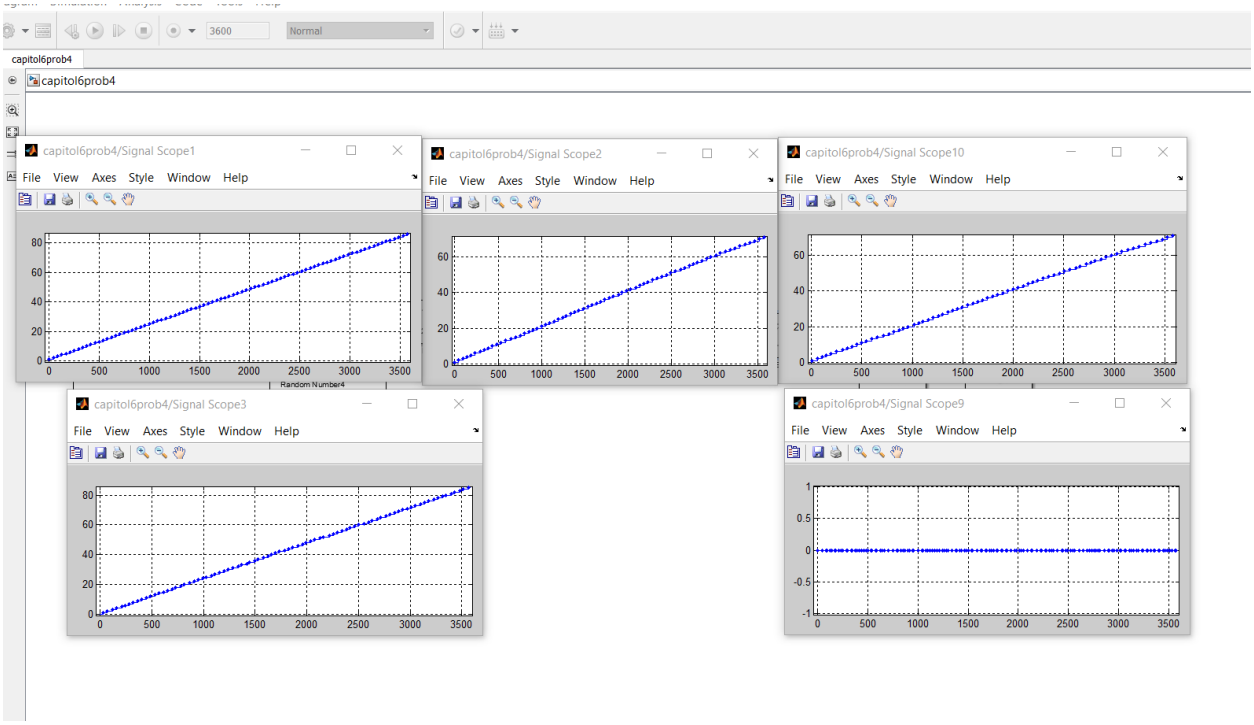
Folosim un bloc N-Server cu capacitatea doi pentru a putea deserve ambele tipuri de muncitori, iar pentru blocul Output Switch setam crieteriul pentru atribut, pentru a separa entitatile dupa atribut catre blocurile Entity Sink.

Draghici Andreea-Maria
CR 3.1B



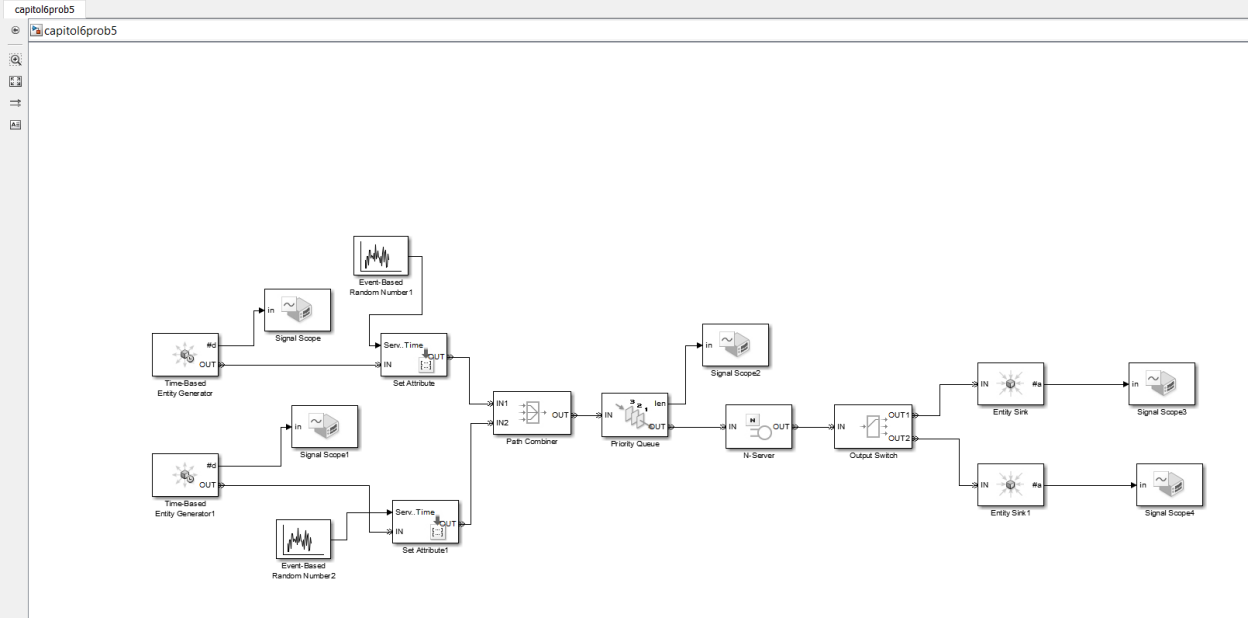
Rezultatele obtinute in urma simularii pe o ora, adica 3600 secunde:

Signal Scope 3 si 10 – afiseaza nr. muncitori deserviti de fiecare tip ;
Signal Scope9 - afiseaza lungimea cozii;
Signal scope 1 si 2 – nr muncitori de fiecare tip care vin spre aprovizionare



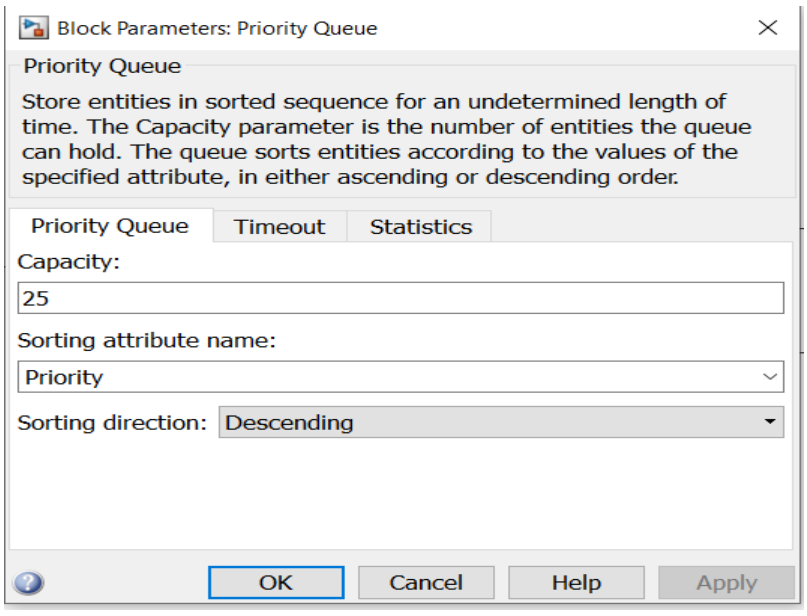
Draghici Andreea-Maria
CR 3.1B

Problema 5



Fiecarui muncitor i se atribuie 2 atribute, tipul si timpul de deservire. Tipul e folosit pentru separarea entitatilor de la finalul simularii, iar timpul de deservire va fi folosit la statia de deservire. Entitatile sunt stocate in aceeasi coada.

Daca muncitorii tip 2 sunt deserviti cu prioritate, vom modifica coada FIFO intr-un bloc Priority Queue astfel:



Diferenta intre problema 4 si 5 este data de reprezentarea blocului cozii, in cazul de fata pentru problema 5 am utilizat un bloc Priority Queue care sorteaza entitatile in ordine crescatoare sau

Draghici Andreea-Maria
CR 3.1B

descrescatoare dupa valorile unui atribut, prioritatea fiind un numar natural. Am setat timpii de deservire cu ajutorul a doua blocuri Event-Based Random Number ca si attribute. Entitatile sunt stocate in aceeasi coada si deservite cu prioritate pentru muncitorii de tipul 2, iar prelucrarea se face pe baza atributului ServiceTime, iar mai apoi entiatile sunt separate dupa atributul de prioritate Munctior pentru a vedea usor cate entitati din fiecare tip au ajuns la capatul simularii.

Rezultatele obtinute in urma simularii pe o ora, 3600 secunde: