

SZÁMÍTÓGÉPES SZIMULÁCIÓS PROGRAM BEMUTATÁSA

Buics László – Kovács Viktor

NEMZETI KÖZSZOLGÁLATI EGYETEM
BUDAPEST



SZÉCHENYI 2020



MAGYARORSZÁG
KORMÁNYA

Európai Unió
Európai Szociális
Alap



BEFEKTETÉS A JÖVŐBE

SZÁMÍTÓGÉPES SZIMULÁCIÓS PROGRAM BEMUTATÁSA

KISMONOGRÁFIA

Szerző:

Buics László
Kovács Viktor

Kiadó:

Nemzeti Közszolgálati Egyetem
Közigazgatási Továbbképzési Intézet
www.uni-nke.hu

Felelős kiadó:

Prof. Dr. Kis Norbert rektorhelyettes
Címe: 1083 Budapest, Üllői út 82.

A kiadvány a
KÖFOP-2.1.2-VEKOP-15
„A jó kormányzást megalapozó közszolgálat-fejlesztés”
című projekt keretében készült el és jelent meg.

© Buics László – Kovács Viktor, 2020

© Nemzeti Közszolgálati Egyetem
Közigazgatási Továbbképzési Intézet, 2020

A mű szerzői jogilag védett. Minden jog, így különösen a sokszorosítás, terjesztés és fordítás joga fenntartva. A mű a kiadó írásbeli hozzájárulása nélkül részeiben sem reprodukálható, elektronikus rendszerek felhasználásával nem dolgozható fel, azokban nem tárolható, azokkal nem sokszorosítható és nem terjeszthető.

TARTALOMJEGYZÉK

1. Folyamatszimulációs szoftver dokumentáció	13
Bevezető	13
1.1 Miért kell szimuláció?	13
1.2 Áttekintés	14
1.3 Telepítés	15
1.3.1 <i>Minimum követelmények</i>	15
1.3.2 <i>Windows</i>	15
1.3.3 <i>OS X</i>	15
1.3.4 <i>Linux</i>	16
2. A szimuláció részletes bemutatása	17
2.1 Kezelőfelület	17
2.2 A folyamatok építőelemei	22
2.2.1 <i>Szolgáltatási folyamatok</i>	23
2.2.2 <i>Termelési folyamatok</i>	27
2.3 További folyamatelemek	33
2.4 A folyamatszimuláció működése, szabályrendszer	33
2.4.1 <i>Szolgáltatástípusú folyamatok</i>	34
2.4.2 <i>Ticket machine (jegykiadó automata)</i>	34
2.4.3 <i>Waiting room (váróterem)</i>	34
2.4.4 <i>Administrator (ügyintéző)</i>	35
2.4.5 <i>Window (ablak)</i>	35
2.4.6 <i>Exit (kijárat)</i>	35
2.4.7 <i>Source (forrás)</i>	35
2.4.8 <i>Buffer (puffer)</i>	35
2.4.9 <i>Machine (gép)</i>	36
2.4.10 <i>Worker (munkás)</i>	36
2.4.11 <i>Sink (nyelő)</i>	36
2.4.12 <i>Splitter (daraboló)</i>	37
2.4.13 <i>Combiner (egyesítő)</i>	37
3. Valószínűségi változók, eloszlások	38
3.1 Statisztikák	40
3.1.1 <i>Várható érték statisztika</i>	40
3.1.2 <i>Standard szórás, variancia statisztika</i>	41
3.2 Egyenletes eloszlás	41
3.3 Normális eloszlás	42
3.4 Szimmetrikus háromszög eloszlás	43
3.5 Aszimmetrikus háromszög eloszlás	43

3.6	Exponenciális eloszlás	44
3.7	Előre meghatározott lista	46
3.8	Hisztogram	46
4.	Kimutatás, statisztika	47
4.1	Szolgáltatási kimutatás	48
4.2	Termelési kimutatás	49
5.	Introduction	52
5.1	Why is simulation needed?	52
5.2	Overview	53
5.3	Installation	54
5.3.1	<i>Minimum requirements</i>	54
5.3.2	<i>Windows</i>	54
5.3.3	<i>OSX</i>	54
5.3.4	<i>Linux</i>	54
6.	Detailed description of the simulation	55
6.1	User interface	55
6.2	Process elements	59
6.2.1	<i>Service process elements</i>	59
6.2.2	<i>Production processes</i>	64
6.3	Other process elements	69
6.4	The ruleset of the simulation	69
6.4.1	<i>Service processes</i>	70
6.4.2	<i>Ticket machine</i>	70
6.4.3	<i>Waiting room</i>	70
6.4.4	<i>Administrator</i>	70
6.4.5	<i>Window</i>	70
6.4.6	<i>Exit</i>	71
6.4.7	<i>Source</i>	71
6.4.8	<i>Buffer</i>	71
6.4.9	<i>Machine</i>	71
6.4.10	<i>Worker</i>	72
6.4.11	<i>Sink</i>	72
6.4.12	<i>Splitter</i>	72
6.4.13	<i>Combiner</i>	73
7.	Random variables, probability distributions	74
7.1	Statistics	76
7.1.1	<i>Expected value</i>	76
7.1.2	<i>Standard deviation, variance</i>	76
7.2	Uniform distribution	77
7.3	Normal distribution	78
7.4	Symmetric triangle distribution	79
7.5	Asymmetric triangle distribution	79
7.6	Exponential distribution	80

7.7	Predefined list	81
7.8	Histogram	81
8.	Reports, statistics	82
8.1	Service report	82
8.2	Production report	84
9.	Folyamatszimulációs szoftver — példák	87
	Bevezetés	87
10.	A szimulációról általánosságban	89
10.1	Szimuláció	89
10.2	Modell	90
10.3	Esemény	91
10.4	A szimuláció általános céljai	92
11.	A kormányablakokról röviden	93
11.1	A kormányablakok kialakítása	93
11.2	A kormányablakokban intézhető leggyakoribb ügyek típusai	96
11.3	A kormányablakokban leggyakrabban előforduló ügyek kategóriái	97
12.	Kormányablak szimulációs esetpéldák bemutatása	101
12.1	Első példa: egyszerű szimulációs modell felépítésének folyamata	104
12.1.1	<i>Bemutató</i>	104
12.1.2	<i>Felépítés menete</i>	104
12.1.3	<i>Futtatás</i>	109
12.1.4	<i>Eredmények értékelése</i>	117
12.2	Második példa: összetett szimulációs modell felépítésének folyamata	121
12.2.1	<i>Bemutató</i>	121
12.2.2	<i>Felépítés menete</i>	121
12.2.3	<i>Futtatás</i>	123
12.2.4	<i>Eredmények értékelése</i>	136
13.	Összefoglalás	142
	Irodalomjegyzék	143
	JOGSZABÁLYTÁR	145

ÁBRAJEGYZÉK

Folyamatszimulációs szoftver dokumentáció

1. ábra:	A szimulációs alkalmazás kezelőfelülete. 1) Modell rajzolófelület, 2) Az építőelemeket tartalmazó sáv, 3) Üzenet mező, 4) Vezérlő sáv	18
2. ábra:	Vezérlőpanel	19
3. ábra:	Szolgáltatásdefiníciós ablak	19
4. ábra:	Eloszlások megadására szolgáló ablak. 1) Az összes definiált eloszlás listája, 2) Új eloszlás hozzáadása, eltávolítása, ablak bezárása, 3) A kiválasztott eloszlás beállítása (típus és paraméterek), 4) Eloszlás sűrűségfüggvény előnézet	20
5. ábra:	Nyitvatartási idők definiálása	21
6. ábra:	Egy példa a termelési folyamat modelljére	22
7. ábra:	Jegykiadó automata piktogramja	23
8. ábra:	A jegykiadó automata tulajdonságai	23
9. ábra:	A váróterem piktogramja, a szimuláció közben folyamatosan látható a várakozók időbeli átlagos száma	24
10. ábra:	A váróterem beállítása	24
11. ábra:	Ügyintéző piktogramja, szimuláció során nyomon követhető a kihasználtság (utilization), illetve a kiszolgált ügyfelek száma	25
12. ábra:	Az ügyintéző beállításai	25
13. ábra:	Az ablak piktogramja	26
14. ábra:	Megadható az ablak azonosítószáma	26
15. ábra:	Egy példa az ablak-ügyintéző-kijárat elrendezésre	26
16. ábra:	A kijáráthoz tartozó piktogram	27
17. ábra:	A váróteremből kiesők egy második kijáráthoz vannak átirányítva	27
18. ábra:	A forrás piktogramja	28
19. ábra:	Az összes létrehozott munkadarab számának megadható nulla is, ebben az esetben folyamatosan biztosítja a munkadarabokat, míg a szimuláció meg nem áll.	28
20. ábra:	Puffer piktogram	29
21. ábra:	A puffer paraméterei	29
22. ábra:	A gép piktogramja, szimuláció közben nyomon követhető a feldolgozási folyamat, a dolgozó munkások, a kihasználtság	30
23. ábra:	A gép paraméterei	30
24. ábra:	A munkás piktogramja	31
25. ábra:	A munkásnál külön megadható, hogy bármelyik gépnél dolgozhat-e	31
26. ábra:	A daraboló piktogramja	31
27. ábra:	A daraboló beállítása	32

28. ábra:	Az egyesítő folyamatelem piktogramja	32
29. ábra:	A nyelő piktogramja	32
30. ábra:	Daraboló bekapcsolt beépített pufferral, a kettőből egy részmunkadarabot már kiadott	37
31. ábra:	Kettőből egy részmunkadarab már beérkezett	37
32. ábra:	Normális eloszlású eloszlásfüggvények	38
33. ábra:	Normális eloszlás sűrűségfüggvénye	39
34. ábra:	Egyenletes eloszlás eloszlásfüggvénye	41
35. ábra:	Egyenletes eloszlás sűrűségfüggvénye	42
36. ábra:	Aszimmetrikus háromszög eloszlás eloszlásfüggvénye	44
37. ábra:	Aszimmetrikus háromszög eloszlás sűrűségfüggvénye	44
38. ábra:	Exponenciális eloszlás eloszlásfüggvénye	45
39. ábra:	Exponenciális eloszlás sűrűségfüggvénye	45
40. ábra:	Ügyfelek beérkezésének eloszlása a nap folyamán (6-22 óra között), két kiválasztott szolgáltatás esetén	46

Figure 41:	User interface of the simulation application. 1) Model drawing region, 2) Ribbon for the process elements, 3) Message region, 4) Control area	55
Figure 42:	Main control panel.	56
Figure 43:	Service definition dialog window	56
Figure 44:	Distributions dialog window	57
Figure 45:	Open hours dialog	58
Figure 46:	an example production model	59
Figure 47:	Icon of the ticket machine	60
Figure 48:	Properties of the ticket machine element	60
Figure 49:	Icon of the waiting room	61
Figure 50:	settings of the waiting room	61
Figure 51:	Icon for the administrator	61
Figure 52:	Properties of the administrator	62
Figure 53:	Icon of the window	62
Figure 54:	The name and the identifier number can be set for a window	63
Figure 55:	An example of a window-administrator-exit layout	63
Figure 56:	Icon and real time statistics for the exit	63
Figure 57:	Example layout for handling churn clients by directing them to a different exit	64
Figure 58:	Icon of the source	64
Figure 59:	The number of generated items is unlimited by setting the value to zero	65
Figure 60:	Icon of the buffer, showing the actual workitems in the queue	65
Figure 61:	Settings of the buffer.	65
Figure 62:	During simulation the icon of the machine shows real time information	66
Figure 63:	Properties of the machine	66
Figure 64:	Icon of the worker	67
Figure 65:	Properties of the worker	67
Figure 66:	Icon of the splitter	67

<i>Figure 67:</i>	<i>Settings of the splitter</i>	68
<i>Figure 68:</i>	<i>Icon of the combiner</i>	68
<i>Figure 69:</i>	<i>Icon of the sink</i>	68
<i>Figure 70:</i>	<i>The splitter with activated output buffer. It can be seen that one part is still waiting for its receiver to be ready</i>	72
<i>Figure 71:</i>	<i>One out of two workitem parts have arrived to the combiner</i>	73
<i>Figure 72:</i>	<i>CDF of normal distributions</i>	74
<i>Figure 73:</i>	<i>PDF of normal distributions</i>	75
<i>Figure 74:</i>	<i>CDF of a uniform distribution</i>	77
<i>Figure 75:</i>	<i>PDF of a uniform distribution</i>	77
<i>Figure 76:</i>	<i>CDF of an asymmetric triangle distribution</i>	79
<i>Figure 77:</i>	<i>PDF of an asymmetric triangle CDF</i>	79
<i>Figure 78:</i>	<i>CDF of exponential distributions</i>	80
<i>Figure 79:</i>	<i>PDF of exponential distributions</i>	81
<i>Figure 80</i>	<i>Customer arrival time distribution throughout the day, for two selected services (between 6am and 22pm)</i>	81

Folyamatszimulációs szoftver példák

1. ábra:	Ticket machine (jegykiadó automata)	102
2. ábra:	Waiting room (Váróterem)	102
3. ábra:	Window (Kormányablak)	102
4. ábra:	Administrator (Ügyintéző)	103
5. ábra:	Exit (Kijárat)	103
6. ábra:	Connector (Összekötő) elem két objektum között	104
7. ábra:	A szimulációs szoftver kezelőfelülete	105
8. ábra:	Szolgáltatáshoz kapcsolódó objektumok eszköztára	105
9. ábra:	Ticket machine (Jegykiadó automata) objektum elhelyezése a szimulációs felületen	106
10. ábra:	Waiting room (Váróterem) objektum elhelyezése a szimulációs felületen	106
11. ábra:	Window (Kormányablak) objektum elhelyezése a szimulációs felületen	107
12. ábra:	Administrator (Ügyintéző) objektum elhelyezése a szimulációs felületen	107
13. ábra:	Exit (Kijárat) objektum elhelyezése és mozgatása a szimulációs felületen	108
14. ábra:	A szimulációs felületen lévő objektumok összekötése egymással Connectorok (Összekötő elemek) segítségével	109
15. ábra:	Hibaüzenetek megjelenítése a szimulációs modell hibás vagy hiányos alapbeállításai esetén (1)	109
16. ábra:	Hibaüzenetek megjelenítése a szimulációs modell hibás vagy hiányos alapbeállításai esetén (2)	110
17. ábra:	Szolgáltatás definiálása a jegykiadó automata (Ticket machine) objektum számára (1)	111

18. ábra:	Szolgáltatás definiálása a jegykiadó automata (Ticket machine) objektum számára (2)	111
19. ábra:	Nyitvatartási időintervallum definiálása a jegykiadó automata (Ticket machine) objektum számára (1)	112
20. ábra:	Nyitvatartási időintervallum definiálása a jegykiadó automata (Ticket machine) objektum számára (2)	113
21. ábra:	Egyedi elosztástípus definiálása a jegykiadó automata (Ticket machine) objektum számára (1)	114
22. ábra:	Egyedi elosztástípus definiálása a jegykiadó automata (Ticket machine) objektum számára (2)	114
23. ábra:	Egyedi elosztástípus definiálása a jegykiadó automata (Ticket machine) objektum számára (3)	115
24. ábra:	Hibaüzenetek megjelenítése a szimulációs modell hibás vagy hiányos alapbeállításai esetén (3)	115
25. ábra:	Egyedi elosztástípus definiálása az Administrator (Ügyintéző) objektum számára (1)	116
26. ábra:	Egyedi elosztástípus definiálása az Administrator (Ügyintéző) objektum számára (2)	117
27. ábra:	A szimuláció futtatása közben leolvasható információk az egyes objektumok esetén	118
28. ábra:	A szimuláció futtatása után leolvasható információk az egyes objektumok esetén	119
29. ábra:	Jelentés generálása az elvégzett szimuláció részletes adatainak áttekintéséhez és további elemzéséhez (1)	119
30. ábra:	Jelentés generálása az elvégzett szimuláció részletes adatainak áttekintéséhez és további elemzéséhez (2)	120
31. ábra:	Jelentés generálása az elvégzett szimuláció részletes adatainak áttekintéséhez és további elemzéséhez (3) - Az egyes szolgáltatásokra vonatkozó összesített adatok (átfutási idő átlagos hossza és szórása, minimum és maximum értékek)	120
32. ábra:	Jelentés generálása az elvégzett szimuláció részletes adatainak áttekintéséhez és további elemzéséhez (4) – Az Adminisztrátorokra vonatkozó összesített adatok (kiszolgált ügyek száma, ügyintézési idő átlagos hossza és szórása, minimum és maximum értékek)	120
33. ábra:	A második példában szereplő összetett modell vizuális megjelenése a szimulációs felületen az átrendezés után	123
34. ábra:	Szolgáltatások definiálása a jegykiadó automata (Ticket machine) objektumok számára (1)	124
35. ábra:	Szolgáltatások definiálása a jegykiadó automata (Ticket machine) objektumok számára (2)	124
36. ábra:	Szolgáltatások definiálása a jegykiadó automata (Ticket machine) objektumok számára (3)	125
37. ábra:	Nyitvatartási időintervallum definiálása a jegykiadó automata (Ticket machine) objektum számára (1)	126
38. ábra:	Nyitvatartási időintervallum definiálása a jegykiadó automata (Ticket machine) objektum számára (2)	126

39. ábra:	Egyedi eloszlás definiálása a Driving licence (jogosítvány igénylése) szolgáltatást generáló jegykiadó automata (Ticket machine) számára (1)	127
40. ábra:	Egyedi eloszlás definiálása a Driving licence (jogosítvány igénylése) szolgáltatást generáló jegykiadó automata (Ticket machine) számára (2)	128
41. ábra:	Egyedi eloszlás definiálása a Driving licence (jogosítvány igénylése) szolgáltatást generáló jegykiadó automata (Ticket machine) számára (3)	129
42. ábra:	Egyedi eloszlás definiálása az ID card (személyi igazolvány igénylése) szolgáltatást generáló jegykiadó automata (Ticket machine) számára	129
43. ábra:	Egyedi eloszlás definiálása a Passport (útlevél igénylése) szolgáltatást generáló jegykiadó automata (Ticket machine) számára	130
44. ábra:	Egyedi eloszlás definiálása a Client-gate registration (Ügyfélkapu regisztráció) szolgáltatást generáló jegykiadó automata (Ticket machine) számára	131
45. ábra:	Egyedi eloszlás definiálása a Taxation (adóügyek intézése) szolgáltatást generáló jegykiadó automata (Ticket machine) számára	131
46. ábra:	Az adminisztrátorok számára kiválasztható szolgáltatások listája a Properties (Tulajdonságok) menüpontban	132
47. ábra:	Egyedi eloszlás definiálása a jogosítvány igénylésével kapcsolatos ügyintézési időre (Administrator Driving licence distribution) az Ügyintézők számára	133
48. ábra:	Egyedi eloszlás definiálása a személyi igazolvány igénylésével kapcsolatos ügyintézési időre (Administrator ID card distribution) az Ügyintézők számára	134
49. ábra:	Egyedi eloszlás definiálása az útlevél igénylésével kapcsolatos ügyintézési időre (Administrator Passport distribution) az Ügyintézők számára	134
50. ábra:	Egyedi eloszlás definiálása az ügyfélkapu regisztrációval kapcsolatos ügyintézési időre (Administrator Client-gate registration distribution) az Ügyintézők számára	135
51. ábra:	Egyedi eloszlás definiálása az adóügyek ügyintézésével kapcsolatos ügyintézési időre (Administrator Taxation distribution) az Ügyintézők számára	136
52. ábra:	A szimuláció futtatása közben leolvasható információk az egyes objektumok esetén	137
53. ábra:	A szimuláció futtatása után leolvasható információk az egyes objektumok esetén	137
54. ábra:	Jelentés generálása az elvégzett szimuláció részletes adatainak áttekintéséhez és további elemzéséhez (1)	138
55. ábra:	Jelentés generálása az elvégzett szimuláció részletes adatainak áttekintéséhez és további elemzéséhez (2)	138

56. ábra: A szimuláció futásának részletes eredményei a táblázatkezelőben (1)
– Az egyes szolgáltatásokra vonatkozó összesített adatok (átfutási idő
átlagos hossza és szórása, minimum és maximum értékek) 139
57. ábra: A szimuláció futásának részletes eredményei a táblázatkezelőben (2)
– Az Adminisztrátorokra vonatkozó összesített adatok (kiszolgált ügyek
száma, ügyintézési idő átlagos hossza és szórása,
minimum és maximum értékek) 139
58. ábra: A szimuláció futásának részletes eredményei a táblázatkezelőben (3)
– Az Adminisztrátorokra vonatkozó részletes adatok (kiszolgált ügyek
száma szolgáltatásonként, ügyintézési idő átlagos hossza és szórása
szolgáltatásonként, minimum és maximum értékek) 140
59. ábra: A szimuláció futásának részletes eredményei a táblázatkezelőben (4)
– Az Adminisztrátorokra vonatkozó részletes
adatokat szolgáltatástípusonként csoportosítva 140
60. ábra: A szimuláció futásának részletes eredményei a táblázatkezelőben (4)
– Az egyes szolgáltatásokat igénybe vevők várakozási ideje
a Váróteremben kategóriánként (várakozási idő átlagos hossza
és szórása, minimum és maximum értékek) 141

1. FOLYAMATSZIMULÁCIÓS SZOFTVER DOKUMENTÁCIÓ

Bevezető

A kötetben bemutatunk egy egyszerű folyamatszimulációs szoftvert, melynek célja megismertetni a hallgatókkal a folyamatok építőelemeinek fogalmait, valamint lehetővé tenni, hogy egy egyszerűen használható felületen keresztül szabadon összeállíthassanak saját folyamatmodelleket. A modellek paraméterei szabadon hangolhatók, a hatásaik ezáltal látványosan megfigyelhetők, így a hallgatók az elméletet gyakorlatias, szemléletes módszerekkel sajátíthatják el.

A kötet első felében a folyamatszimulációs szoftver dokumentációját mutatjuk be részletesen magyar és angol nyelven, kitérve a program felépítésére, részeire, a benne használható elemekre és a velük végrehajtható műveletekre. Ezt követően a kötet második felében folyamatszimulációs példákat mutatunk be a szoftver segítségével, melyekkel a felhasználók elsajátíthatják a program használatát a részletesen felépített egyszerű és az összetett példák segítségével.

1.1. Miért kell szimuláció?

A szimuláció célja, hogy egy folyamat vagy tevékenység működését megértsük, valamint lehetővé teszi, hogy komolyabb (anyagi vagy biztonsági) kockázat nélkül megvizsgálhassuk, hogyan reagálna bizonyos eseményekre, változtatásokra. A szimuláció vezetésben betöltött szerepe (KOVÁCS, 2010):

- előrejelzés (mi lenne, ha...?),
- tervezés (mi lenne a legjobb...?),
- irányítás (hogyan kell(ene)...?),
- elemzés (hogyan kellett volna...?).

Általában elmondható, hogy a fenti területeken a döntésekkel kapcsolatos kockázatok értékelése és csökkentése a cél.

A szimulációhoz szükség van egy modellre, amely a célnak megfelelő részletességgel tartalmazza az egyes lépések leírását. A modellek a valóság egyszerűsített leírásai, a vizsgálat célja erősen befolyásolja, hogy milyen mértékben szükséges vagy megengedhető az egyszerűsítés.

A szimulációs folyamat első lépése a modellezés, ennek során hozzuk létre a szimuláció modelljét. A modellnek tartalmaznia kell azokat a kulcslépéseket, amelyek ténylegesen

befolyásolják a modell működtetése során vizsgálni kívánt információkat. A megtervezett modell nemcsak a szimulációs lépéseket írja le, hanem tartalmaz információt a bemenelekről, illetve a működéshez szükséges belső adatokról, eloszlásokról is. Ezen adatok, információk minősége kritikus a szimulációs eredmények szempontjából. Amennyiben egy valóságban már létező folyamat modellezéséről van szó, méréssel, adatgyűjtéssel lehet meghatározni. A mért értékek közvetlenül felhasználhatók, vagy regresszióval becsülhető egy eloszlás, ami jól közelíti a méréseket, ezzel lehetővé téve a méréshez képest sokkal több eset vizsgálatát a szimuláció során. Amennyiben a folyamat még nem létezik a valóságban (például még a tervezési fázisban), akkor a paramétereket becsülni kell, vagy esetleg más, például előírt paraméterekből kell visszakövetkeztetni. A szimulációs eredmények alapján tovább finomíthatók az értékek.

Az elkészült modell egy szimulációs keretrendszerben működtethető, futtatható. Ez a keretrendszer lehet egy szoftver, de működtethető akár manuálisan is, egy előre megadott szabályrendszer alapján, mint egy társasjáték. Az utóbbira jó példa a sörjáték, ami egy el-látási lánc résztvevőinek rendelési és raktározási folyamatait szimulálja, és ezen keresztül mutat rá rendkívül szemléletes módon a rendszer sajátosságaira, tanulságaira.

Ugyanerre a módszerre épülnek a szimulációs szoftverek is, melyek tartalmazzák a modell valamilyen leírását, illetve egy előre meghatározott szabályrendszer alapján lépésről lépésre frissítik a modell állapotát. Előnyük, hogy lényegesen kevesebb időt vesz igénybe a modell működtetése, ezért hosszabb időtartamok is szimulálhatók, kisebb a tévesztés valószínűsége, illetve egyszerűen megvizsgálható a modell viselkedése, stabilitása, robusztussága különböző paraméterek, valamint véletlen hatások esetén (Monte Carlo-szimuláció).

Tervezési fázisban szimulációval megbecsülhető a szükséges erőforrásigény, de meglévő folyamatok is felülvizsgálhatók, főleg mivel ilyenkor már pontosabb valós működési adatok is rendelkezésre állnak. Ebben az esetben meg lehet vizsgálni, miképpen lehet a folyamatot még hatékonyabbá tenni: mi történt volna, ha az igények megnövekedtek, de a kiszolgálási idők változatlanok maradtak volna.

Számos összetett folyamatszimulációs szoftver (Arena, VisSim, Siemens Tecnomatix, Simulink stb.) érhető el, melyekkel akár vállalatok teljes üzleti vagy gyártási folyamatai is modellezhetők. A kötet célja, hogy betekintést nyújtson e területbe egyszerű és könnyen elsajátítható eszközökkel, egy oktatási célból készült ingyenes, érthető, könnyen kezelhető vizuális szoftver segítségével.

1.2. Áttekintés

A szoftver két különböző folyamattípus szimulációjára szolgál. Az első folyamattípus a kormányablakok működését modellező *szolgáltatási folyamat*: sorra érkeznek ügyfelek (megadott eloszlás szerint), akik a jegykiadó automatán egy szolgáltatás igénybevételére jelentkezve kapnak egy sorszámot, ezután egy váróteremben foglalnak helyet. Az ablakoknál lévő ügyintézők különböző szolgáltatásokat tudnak nyújtani. Ha egy ügyintéző szabadabbá válik, hív egy újabb várakozó ügyfelet, akit ki tud szolgálni. A kiszolgálás ideje beállítható eloszlás szerint történik, végül az ügyfél elégedetten távozik a kijáraton.

A szoftver lehetővé teszi különböző modellek felépítését: tetszőleges szolgáltatástípusok definiálhatók, megadható az ügyfelek beérkezési időközének eloszlása szolgáltatásonként, a kiszolgálók képesítései, hogy mely szolgáltatások elvégzésére képesek, ezen

kiszolgálási idők eloszlása, az ablakok száma stb. A paraméterek szabadon állíthatók, könnyen meg lehet vizsgálni, hol vannak a folyamatban szűk keresztmetszetek, és meg lehet vizsgálni, hogy azok miképpen oldhatók fel. Modellezhető, hogy mi történne, ha egy adott ügyintéző több szolgáltatást is el tudna végezni, mi lenne, ha még egy ügyintéző bekerülne a folyamatba, vagy egy ablak bezárásra milyen hatással lenne a sor átlagos hosszára vagy a várakozási időre.

A másik folyamatípus a *termelési folyamat*. Ennek során munkadarabok érkeznek egy forrásból, melyeket adott eloszlási idő szerint gépek dolgoznak fel. A gépek működtetéséhez munkások szükségesek. A munkafolyamatok összefűzhetők, illetve elágazhatnak. Megvizsgálható, hogy mikor melyik gép lesz a szűk keresztmetszet, hogyan javítható a kihasználtság, miképpen hat az átfutási időkre a pufferek mérete és a feldolgozási idők bizonytalansága stb.

Az alkalmazás felhasználói felülete angol nyelvű. A felületen található beviteli mezőkben sok esetben megadhatók számok, értékek. Az eltérő konfigurációk és az egyértelműség miatt a program az angol formátumot használja, a tizedesjel a „.” (pont) karakter. Az eredmények megjelenítése is ilyen formázással történik.

1.3. Telepítés

A folyamatszimulációs szoftver „.NET”, illetve „Eto.Forms” keretrendszerekre épül. A két keretrendszer lehetővé teszi, hogy az alkalmazás többféle platformon is futtatható legyen, és az egyes platformok saját natív kezelőfelülete vizuális szempontból hasonló legyen. A szoftver legfrissebb verziója a <http://servitization.uni-nke.hu> oldalon érhető el.

1.3.1. Minimum követelmények

A szoftver futtatható Windows 7–10, OS X (Mac), illetve Linux operációs rendszereken. A javasolt minimum képernyőfelbontás 1280x720 pixel.

1.3.2. Windows

A futtatáshoz a .NET keretrendszer 4.6-os változata szükséges, ez Windows 10 esetén már külön telepítés nélkül elérhető.

1.3.3. OS X

OS X-en (Mac OS) történő futtatáshoz a következő címen található Mono csomagot szükséges telepíteni:

<https://www.mono-project.com/download/stable/#download-mac>
vagy
<https://goo.gl/KLpgJ9>

A szoftver dmg archívumként érhető el, megnyitás után az ikonra való dupla kattintással indítható.

1.3.4. *Linux*

Linux disztribúciók esetében is szükség van a Mono futtatókörnyezetre, mely a következő címen érhető el:

```
https://www.mono-project.com/download/stable/#download-lin  
sudo apt-get install mono-runtime  
sudo apt-get install gtk-sharp2
```


2. A SZIMULÁCIÓ RÉSZLETES BEMUTATÁSA

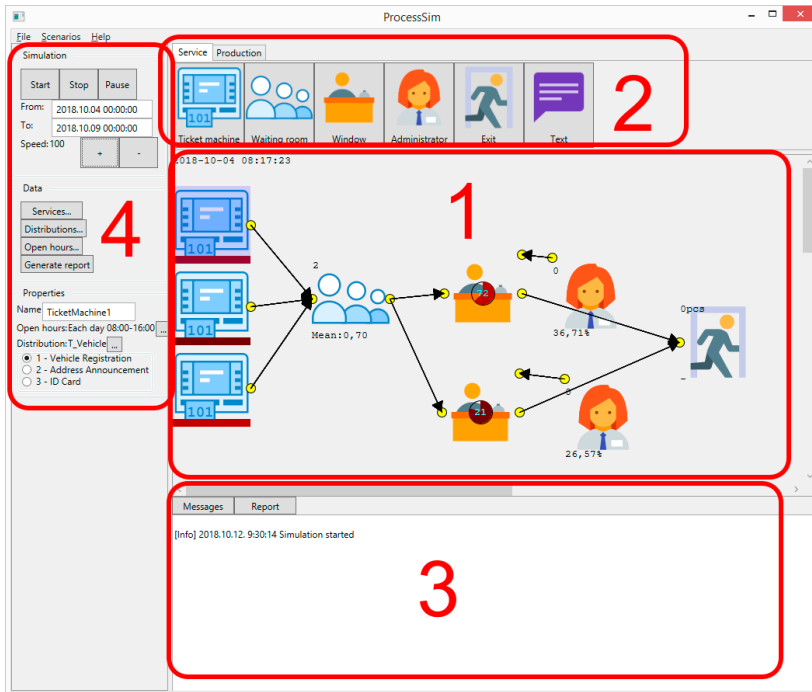
Ebben a fejezetben bemutatjuk a szimulációban használható elemeket, blokkokat, melyekből a modell felépíthető, valamint ismertetjük a szimuláció szabályrendszerét.

2.1. Kezelőfelület

Az alkalmazás kezelőfelülete az 1. ábrán látható. A felület központi része a modell összeállítására szolgáló terület (1). A modell feletti részen találhatóak az építőelemek (2) egy füles elrendezésben: a „Service” fülön vannak a szolgáltatási esethez tartozó blokkok, a „Production” fülön pedig a termelési scenárióhoz kapcsolódó elemek. A modell alatti rész (3) összecsukható, itt jeleníthetők meg a működési üzenetek („Messages” gombbal nyitható, illetve összezárható), illetve a kimutatások („Report”). A felület bal oldalán található sáv (4) a működésre, beállításra vonatkozó vezérlőket tartalmazza.

A legfelső menüsor tartalmazza a szokásos fájl menüt („File”), itt lehet új projektet kezdeni („New”), meglévőt fájlból megnyitni („Load”), vagy az elkészített projektet elmenteni („Save”, „Save as”). Az „Import” paranccsal lehetőség van táblázatos adatokat importálni.

A „Scenarios” menüpont előre elkészített scenáriókat tartalmaz, melyek a kiválasztás után azonnal megnyílnak. Ezek az esetek remekül használhatók kiindulási pontként, az elvégzett módosítások az egyéb projektekhez hasonlóan fájlba menthetők, ám a scenáriók eredeti verziója nem írható felül.



1. sz. ábra: a szimulációs alkalmazás kezelőfelülete
 1) Modell rajzolófelület, 2) Az építőelemeket tartalmazó sáv,
 3) Üzenet mező, 4) Vezérlő sáv

Az alkalmazás bal oldalán található vezérlőpanel (2. ábra) tetején lévő „Simulation” csoport tartalmazza a szimuláció indítására, illetve kézi megállítására vonatkozó vezérlőket. Itt adható meg, hogy a szimuláció milyen időtartamon fusson, valamint a szimuláció sebessége növelhető, illetve csökkenthető a „+” és „-” gombokkal.

Simulation

Start Stop Pause

From: 2018.10.04 00:00:00

To: 2018.10.09 00:00:00

Speed: 500 + -

Data

Services...

Distributions...

Open hours...

Generate report

Properties

Name: TicketMachine1

Open hours: Each day 08:00-16:00 ...

Distribution: T_Vehicle ...

1 - Vehicle Registration

2 - Address Announcement

3 - ID Card

2. sz. ábra: vezérlőpanel

A „Data” csoportban adhatók meg az alapadatok: a „Services” gombbal nyílik meg a szolgáltatásdefiníciós ablak (3. ábra). Itt állíthatjuk be, hogy milyen szolgáltatásokat szeretnénk használni a szimuláció során. Ennek a funkciónak kizárólag a szolgáltatás típusú szcenárióknál van jelentősége. Minden definiált szolgáltatásnak rendelkeznie kell egy egyedi pozitív egész számú azonosítóval, valamint egy névvel. A „0” azonosítóval megadott, illetve meg nem nevezett szolgáltatásokat az alkalmazás automatikusan törli. A táblázathoz új sor a „New” parancssal adható.

Services

Number	Name
1	Vehicle Registration
2	Address Announcement
3	ID Card
0	

Save

Cancel

New

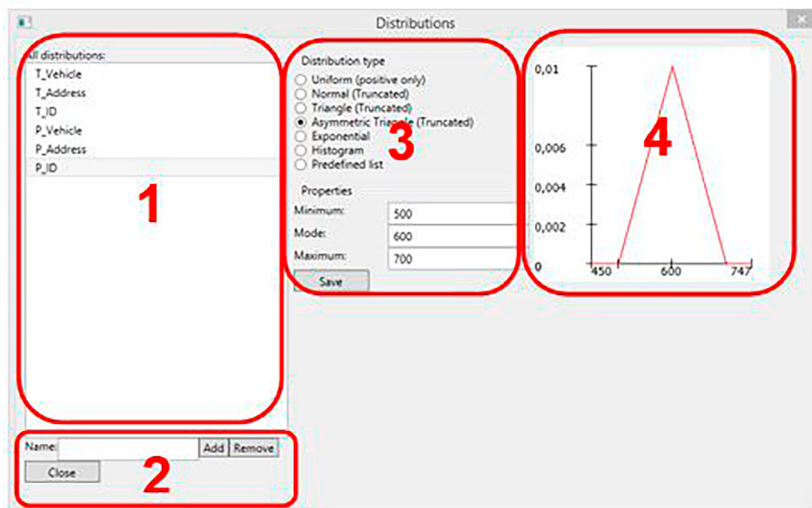
3. sz. ábra: szolgáltatásdefiníciós ablak

A főablakon lévő baloldali vezérlőpanel „Distributions” gombja nyitja meg az eloszlások megadására szolgáló ablakot (4. ábra). A megnyílt ablak bal oldalán (1) található egy lista az eddig definiált eloszlásokkal. Az alsó részen (2) adható meg egy új eloszlás neve, amely az „Add” gombbal hozzáadható a listához, míg a „Remove” gombbal eltávolítható onnan. Az eloszlás típusa beállítható a „Distribution type” rádiógombok egyikével (3). A típusok alatt látható az adott eloszlás beállításait tartalmazó panel. Fontos, hogy ha az eloszlás típusa megváltozik, azt a rendszer automatikusan elmenti. Az eloszlás paramétereinek módosítása esetén a mentéshez meg kell nyomni a „Save” vezérlőt. A beállításoktól jobbra látható egy előnézeti ábra (4), amely a kiválasztott eloszlás paramétereinek alapján jeleníti meg az eloszlás sűrűségfüggvényét. Az eloszlás bizonyos paramétereit megváltoztathatjuk az egérmutató vízszintes vagy függőleges húzásával (lenyomás mellett): például normális eloszlás esetén vízszintes húzással a várható érték, függőlegessel a szórás módosítható.

Az eloszlás a következő definiált típusok közül választható:

- egyenletes (uniform),
- csonkított normális,
- háromszög (szimmetrikus),
- aszimmetrikus háromszög,
- exponenciális,
- hisztogram,
- véletlen, vagy tapasztalati számok listája.

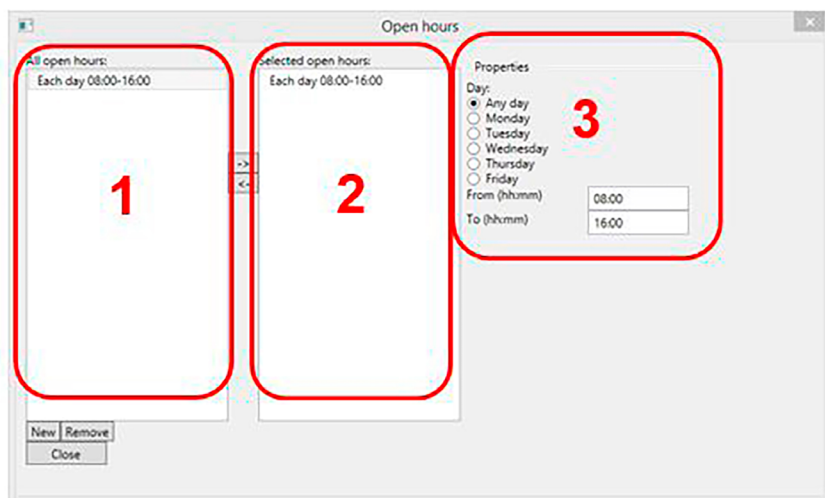
Az eloszlások részletes bemutatása a 3. fejezetben történik.



4. sz. ábra: eloszlások megadására szolgáló ablak

1) az összes definiált eloszlás listája, 2) új eloszlás hozzáadása, eltávolítása, ablak bezárása, 3) a kiválasztott eloszlás beállítása (típus és paraméterek), 4) az eloszlás sűrűségfüggvényének előnézete

A főablakon lévő baloldali vezérlőpanel „Open hours” gombjával nyitható meg a nyitvatartási időket definiáló ablak (55. ábra). A nyitvatartási idők a szolgáltatás típusú szimulációkban használhatók, a termelési szimulációk során nem relevánsak. Az ablak bal oldalán (1) található az összes definiált nyitvatartási idő, alul adható hozzá új, vagy távolítható el egy meglévő az „Add”, ill. „Remove” gombokkal. Az ablak középső részén (2) láthatók a kiválasztott nyitvatartási idők. Ezeknek nincs funkciója az alapadatok szerkesztésekor, azonban ha egy adminisztrátor vagy egy jegykiadó automata beállításait szerkesztjük, itt választhatók ki az alkalmazandó nyitvatartási idők az összes közül (akár több is). Az összes listából a „→” gombbal lehet hozzáadni a kiválasztottakhoz, míg a „←” gombbal lehet eltávolítani egy kiválasztottat. Az ablak jobboldali részén (3) lehet beállítani a nyitvatartási időt. Megadható, hogy melyik munkanapra (vagy az összesre: „Any day”) milyen nyitási és zárási idő vonatkozzon. Az idő megadásának formátuma: óra:perc.



5. sz. ábra: nyitvatartási idők definiálása

A főablak „Generate report” gombjával lehet a szimuláció kezdetétől az adott időpillanatig tartó kimutatást generálni. Az elkészült kimutatás a főablak alsó részén jelenik meg egy szövegmezőben, melyet a jobb láthatóság és elemzés érdekében érdemes vágólapon keresztül egy táblázatkezelő alkalmazásba átmásolni. Minden kijelölése a Ctrl+A, másolás a Ctrl+C, beszúrás a Ctrl+V billentyűkombinációkkal történik. A kimutatásokról bővebben a 4. fejezetben lesz szó.

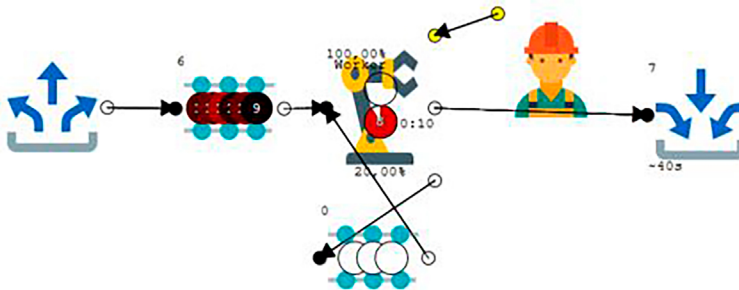
A főablak bal oldalán lévő vezérlőpanel alján láthatók az éppen kiválasztott építőelem beállításai („Properties” rész). Amennyiben egy olyan alapadatot módosítunk, amely hatással van az elem tulajdonságaira, érdemes az elem újra kijelölésével frissíteni a tulajdonság ablakot. Például, ha ki van jelölve egy jegykiadó automata, majd szerkesztjük a szolgáltatások listáját, a jegykiadó automata beállításainál csak akkor frissül a szolgáltatások listája, ha újra kijelöljük a jegykiadót.

2.2. A folyamatok építőelemei

A folyamatmodellek építőelemek összekapcsolásával hozhatók létre. A felhasználható építőelemekről a következő alfejezetekben lesz szó. Egy építőelem beszűrhető, ha az elérhető elemek listájában rákattintunk. Egy elem kijelölhető, ha a modellben a képére kattintunk. Kijelölt elem törölhető a „Delete” billentyű megnyomásával. Az elemek lenyomott egérgombbal mozgathatók.

Minden építőelem rendelkezhet bemenetekkel vagy kimenetekkel, az elemek ilyen kapcsolódási pontokon keresztül köthetők össze. A kimenetekhez más elemek bemenetei kapcsolhatók, azonos kapcsolódási pontok nem csatlakoztathatók. A bemenetek általában az elemek bal oldalán, a kimenetek a jobb oldalán helyezkednek el, de vannak kivételek. A bemenetek kis körökként jelennek meg a blokkokon, színük a típusuk függvényében változik. Az alapértelmezett bemenetek fekete színűek, míg a kimenetek fehérek. A speciális csatlakozási pontok (pl. adminisztrátorok, munkások bekötése) színei ettől eltérnek.

Csatlakozások az egérmutató segítségével hozhatók létre: a ki- és bemenetet reprezentáló ponton az egérmutatót lenyomva, majd húzva egy összekötő nyíl jelenik meg. Összeköttetés úgy hozható létre, ha egy csatlakozási pont felett elengedjük a mutatót. Azonos típusú pontok nem köthetők össze. Amennyiben az egérmutatót magán az elemen engedjük el, az összeköttetés az alapértelmezett be- vagy kimenettel jön létre. Az összeköttetések törölhetők az összekötés megismétlésével, vagy a nyílra kattintva (ami színessé válik) és a „Delete” billentyű megnyomásával. A 6. ábra illusztrálja a folyamatelemek közötti összeköttetéseket.



6. sz. ábra: egy példa a termelési folyamat modelljére

Egy kijelölt építőelem tulajdonságai a főablak bal alsó részén jelennek meg. A tulajdonságok típustól függően eltérőek, de közös bennük, hogy itt adható meg az elemek neve is. A név a kimutatás készítésénél érdekes: egyedi nevek beállításával egyértelműen beazonosíthatóvá válnak az egyes elemek.

2.2.1. Szolgáltatási folyamatok

Szolgáltatási folyamatok a következő elemekből állíthatók össze:

- Ticket Machine (jegykiadó automata),
- Waiting room (váróterem),
- Window (ablak, pl. kormányablak),
- Administrator (adminisztrátor, kiszolgáló munkatárs),
- Exit (kijárat).

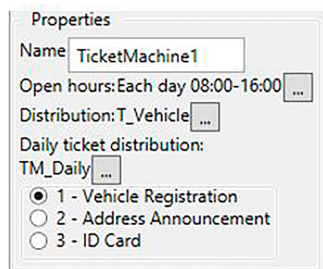
A következőkben az egyes építőelemek részletes bemutatása történik.

2.2.1.1. Ticket Machine (jegykiadó automata)



7. sz. ábra: jegykiadó automata piktogramja

A jegykiadó automata („Ticket Machine”) felel a szimulációban az ügyfelek „létrehozásáért”. Kiválasztott nyitvatartási időben a beállított eloszlás szerint generál ügyfeleket, akik egy megadott szolgáltatást kívánnak igénybe venni. Amennyiben a szimulációban több szolgáltatást is kezelni szeretnénk, több jegykiadó automatát kell elhelyezni. Természetesen a valóságban egy jegykiadó automata ad sorszámokat különböző szolgáltatásokhoz, azonban a szimulációt a választott megoldás átláthatóbbá teszi.



8. sz. ábra: a jegykiadó automata tulajdonságai

A 8. ábrán láthatók a jegykiadó automata beállításai. Az „Open hours” melletti gombbal lehet választani nyitvatartási időt (akár többet is) a már megismert ablakban. Amennyiben nincs nyitvatartási idő kiválasztva, az alapértelmezett lesz aktív. Javasoljuk, hogy az egyértelműség miatt mindig legyen megadva nyitvatartási idő.

A „Distribution” melletti gombbal lehet kiválasztani egy eloszlást, ez alapján generál a rendszer bejövő ügyféligenyeket. Alapértelmezés szerint az eloszlások által generált véletlenszámok a megjelenő ügyfelek közötti időket adják meg másodpercben. Azaz egy

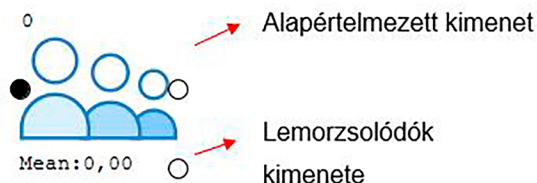
exponenciális eloszlás 60 várható értékkel átlagosan percenként produkál egy bejövő igényt. Kivétel ez alól a hisztogram, amely azt adja meg, hogy a nap adott időszakában (órában) milyen arányban jelennek meg bejövő igények.

A „Daily ticket distribution” mezőben megadható, hogy a naponta megjelenő ügyfelek beérkezési időközének eloszlása hogyan alakuljon. Ez azért fontos, mert naponta változhat, hogy hány ügyfél jelenik meg. Csak abban az esetben van jelentősége, ha a kiválasztott eloszlás nem határozza meg egyértelműen (pl. a hisztogram csak azt határozza meg, hogy a nap folyamán milyen eloszlással érkeznek ügyfelek, de azt nem, hogy mennyien, míg, ha a beérkezések időközét adjuk meg eloszlással, egyértelműen adódik, mikor érkeznek az ügyfelek). A működés részletes leírása a 2.4.2 részben található.

Végül az utolsó mezőben a rádiógombokkal ki lehet választani, hogy az adott jegykiadó automata milyen szolgáltatásra adjon ki sorszámokat.

A jegykiadó automata piktogramjának alsó sávjában a szolgáltatásokhoz kötődő egyedi szín jelenik meg. Az ügyfelek megjelenítése is ezzel a színnel történik a beazonosíthatóság érdekében.

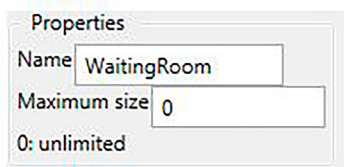
2.2.1.2. Waiting room (váróterem)



9. sz. ábra: a váróterem piktogramja. A szimuláció közben folyamatosan látható a várakozók időbeli átlagos száma

A váróteremben az ügyfelek kiszolgálásra várnak. A teremnek két kimenete van. Az alapértelmezett (1) kimeneten a várakozók, míg a második kimeneten (2) a lemorzsolódók távoznak.

A váróterem kapacitása a „Maximum size” paraméterrel állítható be. A kitüntetett „0” méret jelzi, hogy a kapacitása nem korlátozott (0 kapacitásnak nem is lenne értelme). Amennyiben a váróterem betelt és újabb ügyfél érkezne, két lehetőség van. Ha az alternatív kimenet nincs sehova sem bekötve, a váróterem blokkolt állapotba kerülhet. Ha a bemeneten egy jegykiadó automata van, akkor a szimuláció hibával leáll. Amennyiben egy másik elem (jellemzően egy kijárat) bemenetére van kötve a váróterem lemorzsolódó kimenete, ide kerül az ügyfél, aki már nem fért be a váróterembe. (A háttérben a lemorzsolódók számára egy korlátlan puffer áll rendelkezésre.)

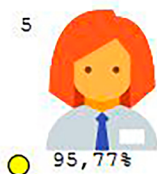


10. sz. ábra: a váróterem beállítása

A szimuláció futtatása közben folyamatosan nyomon követhető az indítástól számított átlagos várakozószám (a sor átlagos hossza).

A váróterem nem tisztán FIFO (first in first out) sor, hanem szolgáltatásonként FIFO. Azaz csak ugyanazon adott szolgáltatást igénybe vevő ügyfelekre igaz, hogy az elsőként érkezőt elsőként szolgíttják.

2.2.1.3. Administrator (ügyintéző)



11. sz. ábra: ügyintéző piktogramja. Szimuláció során nyomon követhető a kihasználtság (utilization), illetve a kiszolgált ügyfelek száma

Az ügyintézőt reprezentáló építőelem egy speciális csatlakozó ponttal rendelkezik, ezzel adható meg, hogy mely ablakoknál dolgozhat. Egy ügyintéző megadott szolgáltatásokat igénybe vevő ügyfeleket képes kiszolgálni.

Properties

Name Administrator2

Services

1-Vehicle Registration D:-

2-Address Announcement D:P_Address

3-ID Card D:P_ID

Open hours

Each day 08:00-16:00

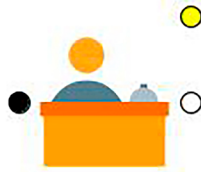
12. sz. ábra: az ügyintéző beállításai

A beállítások panelen lehet megadni, hogy az összes rendelkezésre álló szolgáltatástípus közül az ügyintéző miket tud kiszolgálni. A szolgáltatás előtt lévő pipa jelenti, hogy azt a típusú szolgáltatást el tudja végezni. Amikor egy szolgáltatást engedélyezünk, felugrik az eloszlást kiválasztó dialógus ablak. Itt adható meg, hogy egy adott szolgáltatást milyen eloszlási idővel szolgál ki. A kiválasztott eloszlás neve meg is jelenik a szolgáltatás neve mellett a beállítások panelen. Ez azért lényeges, mert a különböző szolgáltatásokat igénybe vevők kiszolgálási ideje jelentősen eltérhet, továbbá a kiszolgálási idő különbözhet egy tapasztalt ügyintéző és egy éppen betanuló esetén is. (Természetesen az ügyfélen is múlik a kiszolgálási idő, ezért használunk véletlen változókat.)

A szolgáltatások és kiszolgálási idők mellett megadható az ügyintéző munkaideje is. A szimuláció során az algoritmus minden nap elején az ügyintézőket ablakokhoz rendeli. Hogy melyik ügyintéző melyik ablaknál foglalhat helyet, beállítható a piktogram sarkában lévő sárga jelzés és az ablakok hasonló jelzéseinek összekötésével.

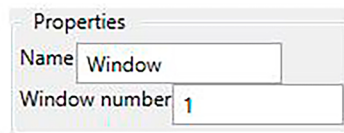
A szimuláció során a modellező felületen folyamatosan megjelenik az ügyintéző által kiszolgált ügyfelek száma (bal oldalon, felül), valamint az ügyintéző kihasználtsága is (alul).

2.2.1.4. Window (ablak)



13. sz. ábra: az ablak piktogramja

Az ablaknál történik az ügyfelek kiszolgálása. Csak akkor szolgálható ki egy ügyfél, ha van szabad ablak, továbbá az ablakhoz olyan ügyintéző van rendelve, aki az adott szolgáltatást nyújtani tudja.

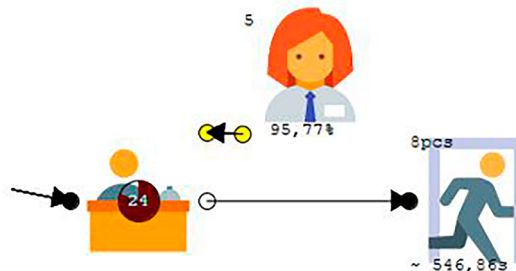


14. ábra. Megadható az ablak azonosítószáma

Az ablak beállításainál megadható az ablak neve és azonosítószáma. Utóbbi a szimulációra nincsen hatással, a kimutatásban jelenhet meg, illetve adatok importálásakor segít az ablak azonosításában.

Alapértelmezés szerint az ügyfelek egy kimenet felé távoznak a kiszolgálás után. Természetesen a modellben felfűzhetők egymás után ablakok, de kérdés, hogy ez mennyire reprezentálja a valós folyamatokat.

Szimuláció futása közben a program a kiszolgálás alatt lévő ügyfelet egy koronggal vizualizálja, mely a kiszolgálás ideje alatt folyamatosan telik meg, ahogy az óra jár körbe.



15. sz. ábra: egy példa az ablak-ügyintéző-kijárat elrendezésre

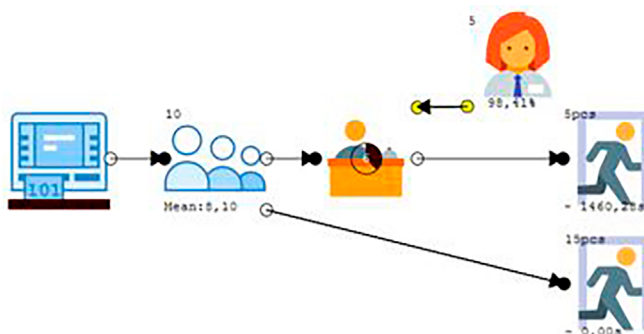
2.2.1.5. Exit (kijárat)

A kijárat szolgál a kiszolgált ügyfelek összegyűjtésére, a kimutatásban kijáratonként jelennek meg statisztikák. Egy bemenete van, nincs kimenete. Szimuláció közben a piktogram felső sarkában megjelenik a folyamat során kiszolgált ügyfelek száma, valamint az átlagos átfutási idő („Flow time”) másodpercben. Ez a statisztika már egyesítve tartalmazza az összes ügyfél idejét, nincs szolgáltatásokra bontva.



16. ábra. A kijárhoz tartozó piktogram

Amennyiben a váróterem kapacitása végesre van beállítva, érdemes egy második kijáratot is elhelyezni, valamint ebbe a kijáratba küldeni a váróteremből kieső ügyfeleket (17. ábra).



17. sz. ábra: a váróteremből kiesők egy második kijárhoz vannak átirányítva

2.2.2. Termelési folyamatok

A termelési folyamatok esetén munkadarabok haladnak egy forrásból a nyelő felé. A szolgáltatás scenárióban az ügyfeleken történt feldolgozás az ablakoknál, a termelés során gépek végeznek műveleteket a munkadarabokon. A gépek működtetéséhez munkásra van szükség.

2.2.2.1. Source (forrás)

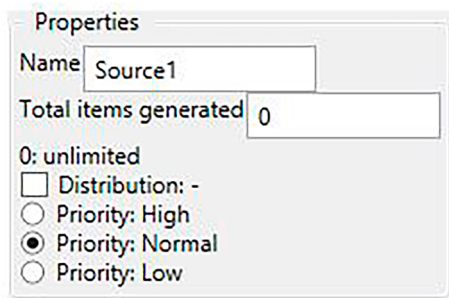


18. ábra. A forrás piktogramja

A munkadarabokat a forrás hozza létre, így kerülnek a folyamatba. A forrásnak nincs bemenete, csak kimenete. Alapértelmezés szerint a forrás folyamatosan bocsátja ki az új munkadarabokat, amikor a következő elem nincs blokkolva.

A forrásnál megadható, hogy korlátlan vagy korlátozott mennyiségű munkadarabot állítson elő. A létrehozandó munkadarabok száma beállítható a „Total items generated” paraméterrel. „0” esetén korlátlanul állítja elő a munkadarabokat.

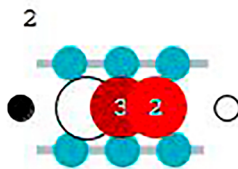
Amennyiben a forrás létrehozta a megadott számú munkadarabot és nincs további tevékenység a modellben, a szimuláció automatikusan leáll. Ez a funkció abban az esetben különösen hasznos, ha a modell különböző paramétereinek hatásait szeretnénk vizsgálni, mert lehetőség nyílik egyenlő számú munkadarabot előállítani és végigfuttatni a rendszeren, a kimutatásban pedig egyszerűen összehasonlíthatók a működési paraméterek (pl. a szimuláció időtartama).



19. sz. ábra. Az összes létrehozott munkadarab számának megadható nulla is, ebben az esetben folyamatosan biztosítja a munkadarabokat, míg a szimuláció meg nem áll

A kibocsátás módja megváltoztatható egy eloszlás megadásával. Ebben az esetben a forrás adott eloszlás szerint generálja a munkadarabokat, valamint aktiválódik egy beépített kimeneti puffer, ami azokat a munkadarabokat tárolja, amik az eloszlás alapján létrejöttek, de nem tudta azokat kibocsátani.

2.2.2.2. Buffer (puffer)

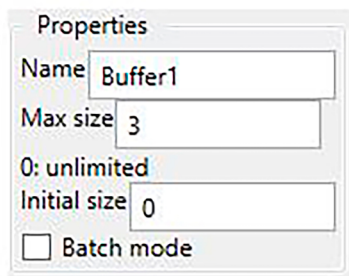


20. sz. ábra: Puffer piktogram

A puffer átmeneti tárolóként funkcionál a munkadarabok számára, egy bemenettel és egy kimenettel rendelkezik. A puffer mérete állítható, „0” estén végtelennek (nagy) tekinthető. Amennyiben a puffer tele van, blokkol, nem fogad több munkadarabot. A puffer egy tisztán FIFO (first in first out) várakozó sort valósít meg, az a munkadarab távozik elsőként, amelyik először érkezett be.

Elsősorban gépek közé érdemes elhelyezni, hogy csökkentse az éheztetési vagy blokkolási szituációkat, növelve a kihasználtságot.

A puffer paramétereinél lehetőség van beállítani egy kezdeti értéket. Ilyenkor a szimuláció elején a rendszer létrehoz adott mennyiségű munkadarabot és elhelyezi a pufferben. Így a szimuláció során hamarabb elérhető az állandósult állapot (amennyiben van), illetve vizsgálható a rendszer válasza „váratlan” eseményekre. Például, ha a várttal ellentétben egy nagyobb leállítás esetén felhalmozódik nagyobb számú munkadarab, képes-e a folyamat feldolgozni és visszatérni a normál működési állapotba.



21. sz. ábra. A puffer paramétereit

A „Batch” mód aktiválásával megváltozik a puffer működése. A puffer mindaddig folyamatosan fogad munkadarabokat (és nem bocsát ki), amíg tele lesz. Ekkor elkezd a kibocsátást, míg kifogy. Kibocsátás közben nem fogad egyetlen munkadarabot sem. Ezzel a lehetőséggel szimulálható a „batch” vagy kötegelt anyagmozgatás. A batch puffer előtt és után további pufferek elhelyezésével megoldható több köteg munkadarab mozgatása, tárolása is.

2.2.2.3. Machine (gép)



22. sz. ábra: a gép piktogramja. Szimuláció közben nyomon követhető a feldolgozási folyamat, a dolgozó munkások, a kihasználtság

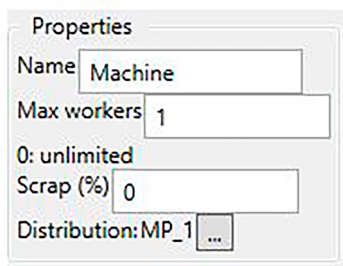
A termelési folyamatok esetén gépek végeznek műveleteket a munkadarabokon. A műveletek végrehajtási ideje tetszőleges eloszlás szerinti lehet. Adott valószínűséggel képes a gép selejtet kibocsátani, ilyenkor az ábra jobb alsó sarkában, az alternatív (selejt) kimeneten jelenik meg a munkadarab.

A gép reprezentálhat késleltetést, műveletvégzést, de akár egy forrás után helyezve a munkadarabok tetszőleges (időbeli) eloszlását is produkálhatja.

Egy gépnél több munkahely is beállítható. A munkahelyek betöltéséhez különböző munkásokra van szükség, egy munkás egyszerre egy munkadarabon dolgozhat. A munkahelyek növelésével átlagosan csökkenthető a feldolgozási idő, ugyanis párhuzamosan több munkadarab feldolgozása lehetséges. Alapértelmezés szerint addig kötött egy munkás egy munkadarabhoz, amíg a munkadarab nem hagyja el a munkahelyet. Ez a szabály önmagában holtpont helyzetekhez vezethet, ezért szükség esetén a munkás elhagyhatja a munkahelyet. Holtpont akkor alakul ki, amikor egységek egymásra várnak, és egyik sem tudja megkezdeni a munkát, míg a másik nem végez. Ilyen szituációban a szimuláció leáll.

A holtpontok elkerülése érdekében érdemes minden géphez dedikálni egy munkást. Amennyiben univerzális munkással vagy munkásokkal szeretnénk a hatékonyságot növelni, érdemes a gépeket több munkahelyűnek beállítani.

A gépekhez a munkások a sárga ponton keresztül rendelhetők hozzá.



23. sz. ábra: a gép paramétereit

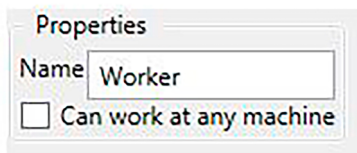
2.2.2.4. Worker (munkás)



24. sz. ábra: a munkás piktogramja

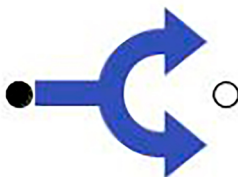
A munkás üzemelteti a gépet, amely egy munkadarabon dolgozik. Több munkahelyes gépek esetén egyszerre több munkás is dolgozhat, de különböző munkadarabokon. Egy munkás több géphez is hozzárendelhető (univerzális, „cross trained”), egyszerre azonban csak egy gépnél tud dolgozni. A helyváltogatás nem jár költséggel, nulla idő alatt megtörténik. A munkás és a gépek összerendeléséről egy algoritmus dönt.

Az egyszerűség kedvéért, hogy ne kelljen adott munkást minden géppel összekötni (ha mindegyiknél dolgozhat), elég a paramétereinél beállítani, hogy minden gépnél dolgozhat („Can work at any machine”).



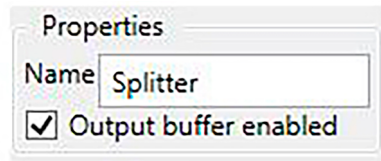
25. sz. ábra: a munkásnál külön megadható, hogy bármelyik gépnél dolgozhat-e

2.2.2.5. Splitter (daraboló)



26. sz. ábra: a daraboló piktogramja

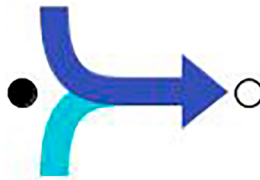
A daraboló folyamategység egy munkadarabot több munkadarab-töredékké alakít, melyek a további folyamatok során teljes munkadarabként funkcionálnak. Minden elem, ami a kimenetére van kötve, kap egy munkadarab-töredéket. Később egy „Combiner” egyesítő egység tudja újra teljes munkadarabbá alakítani. A folyamatelem azt reprezentálja, hogy egy munkadarab szétválasztásra kerül, a különböző részegységei különböző folyamatokon haladhatnak keresztül, végül újra egyesülnek.



27. sz. ábra: a daraboló beállítása

A folyamatlem egy paraméterrel rendelkezik („Output buffer enabled”), amely azt befolyásolja, hogy az egységbe legyen-e beépítve egy puffer.

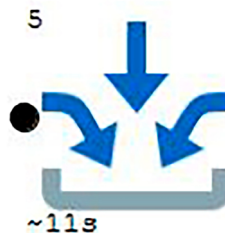
2.2.2.6. Combiner (egyesítő)



28. sz. ábra: az egyesítő folyamatlem piktogramja

Az egyesítő több munkadarab-töredék egyesítésére szolgál, amelyeket egy korábbi lépésben a daraboló egység hozott létre. Az egyesítő minden bemenetén bevár egy-egy munkadarab-töredéket, majd a kimenetén egyesítve kiadja. A daraboló és a kombinálók helyes használata a folyamatmodellben a felhasználó felelőssége.

2.2.2.7. Sink (nyelő)



29. sz. ábra: a nyelő piktogramja

A nyelő a bemenetén fogadja a késznek tekinthető, azaz a folyamatból távozó munkadarabokat, ennek megfelelően nincs kimenete. Célja, hogy a kimutatásban egyértelműsíteni lehessen, hogy mely egységekből távozó munkadarabok statisztikájára vagyunk kíváncsiak.

A szimuláció futtatása közben folyamatosan frissül a piktogram, ábrázolva az összes fogadott munkadarabot, illetve az átlagos átfutási időt. A kimutatásban további statisztikák is megjelennek.

2.3. További folyamatelemek

A *Text* (szöveg) mezővel feliratok helyezhetők el a modell mellett. A szövegmezők nem befolyásolják a szimuláció működését. Hasznos munkafolyamatok mellett magyarázatként elhelyezni.

2.4. A folyamatszimuláció működése, szabályrendszer

A modellezendő folyamat szempontjából egy szimuláció történhet rövid, egyenlő hosszúságú diszkrét időegységekre (kvantumokra) bontva, valamint eseményalapúan, mely során az események tetszőleges időpontokban történhetnek. Az első megoldást abban az esetben célszerű alkalmazni, ha a folyamat matematikai leírással modellezhető, valamint a be- és kimenetek közötti szabályrendszer állandó (pl. fizikai: elektromos vagy mechanikai rendszer). A szimuláció pontosságát alapvetően meghatározza a választott időbeli lépésköz, amellyel a szimulátor folyamatosan frissíti a modell állapotát.

A kötet a folyamatszimulációs szoftverrel foglalkozik, mely az eseményalapú megvalósítást alkalmazza. Ez a megközelítés hatékonyabb, mint a fix időegység-alapú szimuláció, ugyanis a folyamat működtetése során nagyon sok az állásidő, az állásokat pedig események választják el. Pazarlás lenne rövid időegységenként (ezredmásodperc, tizedmásodperc) újraszámolni a modellt, miközben percekig nem történik semmi változás. Bár a rajzolás (megjelenítés) miatt bizonyos szempontból történik időnként újraszámolás, ez nem a szimuláció szerves része, valamint a rajzolási időköz és a szimuláció sebessége nagyon tág keretek között állítható.

Az eseményalapú megközelítés során a szimulátor lépésenként vizsgálja meg, hogy melyik egységnek mikor várható eseménye (pl. elkészült munkadarab, kiszolgált ügyfél stb.). Amelyik egységnek a leghamarabb következik be eseménye, az teszi meg a következő lépést a szimuláció során, illetve ez az idő lesz a szimuláció következő aktuális ideje is. Természetesen előfordulhat, hogy két egység azonos időben jelentkezik eseménnyel. Ez nem probléma, véletlen sorrendben minden egység elvégezheti a feladatát. Az sem okoz gondot, ha egy esemény újabbakat vált ki, akár azonos időben is, ekkor minden esemény kezelése végre lesz hajtva, azonban az eseményeket kizárólag időrendben szabad kiszolgálni.

A következő alfejezetekben végigtekintjük az egyes egységek részletes működését, szabályrendszereiket.

- Minden egység képes a következő kérdésekre választ adni, valamint tevékenységeket végrehajtani.
- Képes-e fogadni egy munkadarabot vagy adott szolgáltatásra jelentkezett ügyfelet a bemenetén, ha van bemenete?
- Készen áll-e kiadni egy munkadarabot vagy ügyfelet (adott szolgáltatással) a kimenetén, ha van kimenete?
- Mikor lesz a következő esemény? (Ha nincs adat, a végtelen sokára a válasz.)
- Fogadjon a bemenetén egy bizonyos munkadarabot vagy ügyfelet.
- Adjon ki a kimenetén egy bizonyos munkadarabot vagy ügyfelet.

2.4.1. Szolgáltatástípusú folyamatok

A szolgáltatástípusú folyamatok alapvetően egyszerű – egy kiszolgálót, de több szolgáltatást tartalmazó – láncok szimulációjára szolgálnak. A modellben jellemzően több jegykiadó automata helyezhető el (különböző szolgáltatásokkal), egy közös váróterem, valamint számos ablak és ügyintéző, akik eltérő szolgáltatásportfóliókkal rendelkeznek, végül egy vagy több kijárat a statisztikai céloknak megfelelően. A modell lehetővé teszi, hogy több ablak is elhelyezhető legyen egymás után, azonban a következőkben definiált működést érdemes figyelembe venni.

2.4.2. Ticket machine (jegykiadó automata)

A jegykiadó automata minden szimulációs nap elején előre legenerálja az aznap érkező ügyfelek sorszámaint a beállított nyitvatartási időben. Ez a módszer biztosítja, hogy mind az érkezési idők közötti eloszlás, mind a nap folyamán érkezők eloszlásával megadhatók legyenek az érkezések.

A jegykiadó automatának nincs bemenete, ezért nem értelmezett, hogy a bemenetén képes-e fogadni ügyfeleket. Alapértelmezés szerint a kimenete mindig készen áll.

A szimuláció során a jegykiadó automata kimenete nem kerülhet blokkolt állapotba. Ebben az esetben a szimuláció hibával leáll. Ez az eljárás szándékos. A jegykiadó automata kimenete jellemzően a váróterembe megy, mely pufferként funkcionál. A váróterem abban az esetben képes blokkolt állapotba kerülni, amennyiben korlátozott a kapacitása és nincs kezelve a lemorzsolódók kezelésére szolgáló kimenete.

A következő esemény idejét a legkorábbi, még ki nem bocsátott ügyfél érkezési ideje adja.

2.4.3. Waiting room (váróterem)

A váróterem mindig képes fogadni ügyfeleket (bármilyen szolgáltatásra érkezőket), akkor is, ha a maximális keret betelt. Ha van még hely, a normál várakozók közé kerül az ügyfél, ha tele van, egy átmeneti tárolóba, amelyből minél hamarabb ki kell hozni. Erre a külön tárolóra azért van szükség, hogy a váróterem bemenete folyamatosan tudjon ügyfeleket fogadni. Az átmeneti tárolóból a lemorzsolódók az alternatív kimeneten távoznak.

Amennyiben van ügyfél akár a normál, akár a lemorzsolódókat tartalmazó tárolóban és a hozzá tartozó megfelelő kimeneten bármilyen egység kész fogadni ügyfelet, a következő esemény ideje az aktuális szimulációs idő (most), ellenkező esetben végtelen sokára (nincs belátható időn belül esemény).

Ha a váróterem kapacitása limitált és nincs bekötve a lemorzsolódók számára a kimenet, akkor a váróterem bemenete blokkol. Ez szándékosan a szimuláció leállításához vezet, hogy ne keletkezzen félreérthető kimutatás a működésről. Limitált váróterem-kapacitásnál ajánlott mindig kezelni a lemorzsolódók esetét.

2.4.4. *Administrator (ügyintéző)*

Minden ügyintéző két eseménnyel foglalkozik: a beállított munkaidő kezdetével és végével. A kettő között számolja a rendszer a teljes munkaidőt, ez szükséges a kihasználtság meghatározásához.

2.4.5. *Window (ablak)*

Az ablaknál történik az ügyfél kiszolgálása. Egy ablak abban az esetben működik, ha van hozzárendelve ügyintéző. Az ügyintéző képesítése határozza meg, hogy milyen szolgáltatásokra jelentkezt ügyfelet képes kiszolgálni (illetve azt is, hogy mennyi idő alatt). Az ügyfél fogadásának a pillanatában a megadott eloszlás alapján meghatározza a rendszer a kiszolgálási időt.

Amennyiben nincs éppen kiszolgált ügyfél egy ablaknál, vagy van, de a kimenet nem képes fogadni, akkor az ablak következő eseményének az ideje végtelen. Ha van kiszolgált ügyfél és a kimeneten lévő egység képes fogadni, akkor a kiszolgálási idő vége lesz a következő esemény időpontja. Jellemzően a kiszolgálás után az ügyfél a kijárathoz fárad, ezért általában nincs blokkolás a kiszolgálás végén.

2.4.6. *Exit (kijárat)*

A kijárat bemenete mindig képes fogadni ügyfeleket, kimenete és eseménye nincs, elsősorban statisztikai célokat szolgál.

2.4.7. *Source (forrás)*

A forrás alapértelmezés szerint folyamatosan hozza létre a munkadarabokat. Akkor nem jön létre új munkadarab, amikor a kimenete blokkolva van. Ezért nem szabad végtelen méretű puffert helyezni a forrás kimenetére, mert az folyamatosan képes fogadni, így a szimuláció megáll ebben az időpontban és nagyon gyorsan tölti a puffert.

Ha a forrás létrehozta az összes beállított munkadarabot vagy nincs a kimenetére semmi sem kötve vagy a kimenetén egy fogadó sem kész a fogadásra, akkor az esemény ideje ismeretlen (várakozik), ha van fogadásra kész egység a kimenetén, akkor a forrás azonnal kéri az esemény kiszolgálását.

2.4.8. *Buffer (puffer)*

Ha a puffer üres vagy a kimenetén egyik egység sem tud fogadni, nem kér esemény kiszolgálást. Ha a kimenetén tud fogadni munkadarabot egy másik egység, akkor az esemény ideje azonnali. Ha a puffer megtelt, akkor a bemenete nem áll készen fogadásra. A kimenete kész kibocsátásra, ha a puffer nem üres.

„Batch” módban a ki- és bemeneti rendelkezésre állás az aktív állapotnak megfelelően változik. Fogadó módban, amíg nincs tele, fogadja a bemenetén a munkadarabokat. Amikor tele lett, átvált kibocsátó módba, ilyenkor nem fogad, csak a kimenete áll készen, amíg van benne munkadarab. Amikor kiürül, ismét átvált fogadó módba.

2.4.9. *Machine (gép)*

A gép végzi a munkadarabok feldolgozását, ez a folyamat működése szempontjából késleltetést jelent. Legalább egy gépnek lennie kell a folyamat típusú modellben. Munkadarab áramlás szempontjából egy bemenete és két kimenete van.

A gép akkor tud munkadarabot fogadni a bemenetén, ha van olyan munkahelye, amelyhez lett munkás hozzárendelve és üres, nincs feldolgozás folyamatban.

A gép nem jelez eseményt (nincs előrelátható történés a közeljövőben), ha nincs folyamatban feldolgozás, vagy nem áll készen a kimenet fogadni egy készülő vagy elkészült munkadarabot. Ha egy munkadarab feldolgozása folyamatban van és képes a kimenetén egy másik egység fogadni, akkor a következő esemény ideje a munkadarab elkészültének ideje. Ha több munkadarab feldolgozása is folyamatban van, akkor az esemény ideje ezek közül a legkorábbi. Amennyiben egy elkészült és várakozó munkadarabot tud fogadni egy másik egység a bemenetén, akkor az aktuális szimulációs idő lesz az esemény ideje.

A gépeknél a feldolgozási idő a gép jellemzője, nem a munkásé.

2.4.10. *Worker (munkás)*

A gépeknél dolgozó munkást reprezentáló elemnek nincs anyagáramlás szempontjából bemenete vagy kimenete. A munkás gépekhez rendelhető, vagyis megadható, hogy mely gépeknél végezhet munkát. A munkás nincs hatással a munkadarab feldolgozási idejére.

Egy munkás egy időben egy géphez lehet hozzárendelve. Alapértelmezésként a munkás nem hagyhatja el a munkahelyét, amíg a munkadarab nem távozott a gépből.

Amíg nincsenek univerzális („cross trained”) munkások, akik különböző gépeknél is dolgozhatnak, nem alakulhat ki holtpon. Azonban, ha „A” gép nem tudja kiadni a munkadarabot, mert „B” gép nem tudja fogadni, ugyanis „A” gépnél le van kötve a munkás, aki a „B” gépet üzemeltetné, holtpon alakulhat ki. Ezért a szoftver ilyen esetben áthelyezi a munkást a blokkolt géptől az éhező („starving”) géphez.

A munkások gépekhez rendelése során először azokat a munkásokat rendeli gépekhez, akik egy gépnél dolgozhatnak (nem univerzálisak). Az univerzális („cross trained”) munkások gépekhez rendelése a pillanatnyi állapotok alapján történik egy algoritmus alapján. Ez a hozzárendelés befolyásolhatja a szimuláció eredményeit.

2.4.11. *Sink (nyelő)*

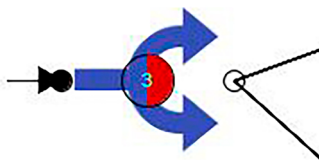
A nyelőnek egy bemenete van, ide érkeznek a munkadarabok, kimenete nincs. Érdemes több nyelőt elhelyezni, ha a kimutatásban szeretnénk a különböző utakat bejáró munkadarabokról külön statisztikát készíteni.

A nyelő bemenete mindig képes munkadarabokat fogadni, külön eseményt nem vált ki, a statisztikában vesz részt.

2.4.12. Splitter (daraboló)

A daraboló egy munkadarabot több részmunkadarabbá képes szétválasztani és azokat különböző kimenetekre továbbítani. Segítségével egy teljes munkadarab részei különböző feldolgozási folyamatokon haladhatnak végig.

Az egységen beállítható, hogy rendelkezzen-e beépített kimeneti pufferrel. Ha nincs beépített kimeneti puffer, akkor abban az esetben adja ki a kimenetein a munkadarab részeit, ha az egységek az összes kimenetén készek fogadni. Beépített kimeneti puffer esetén tárolja a munkadarab részeket, míg az egyes kimenetek fogadni tudják. Ez az opció jelentős hatással lehet a folyamatra, a munkadarabok átfutási idejére, mert jelentős blokkolást okozhat.

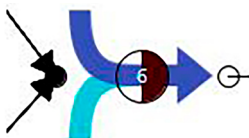


30. sz. ábra: daraboló bekapcsolt beépített pufferrel.
A kettőből egy részmunkadarabot már kiadott

2.4.13. Combiner (egyesítő)

Az egyesítő blokk a részmunkadarabok – amelyeket a daraboló hozott létre – teljes munkadarabokká történő visszaállítására szolgál. Szinkronizálási funkciója is van, ugyanis csak akkor hozza létre a kimenetén a teljes munkadarabot, ha minden részmunkadarab beérkezett.

Az egyesítő rendelkezik beépített pufferrel minden egyes bemenetéhez: minden bemenetről képes fogadni egy (rész)munkadarabot, majd amikor az összes bemenettől rendelkezésre áll, megpróbálja a kimenetre továbbítani. Amíg a kimenete blokkolva van, további bemeneteket sem tud fogadni.



31. sz. ábra: kettőből egy részmunkadarab már beérkezett

3. VALÓSZÍNŰSÉGI VÁLTOZÓK, ELOSZLÁSOK

A szoftverben felépített modell számos paramétere megadható valószínűségi változókkal, ezek részletes bemutatása következik ebben a fejezetben.

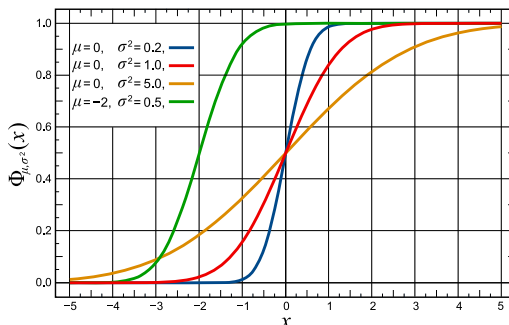
Valószínűségi változóknak nevezünk olyan mérhető fogalmakat (változókat), melyek értéke nem egyértelmű, különböző időpontokban vagy különböző szituációkban más-más értéket vehet fel. Ilyen lehet egy dobókocka dobás kimenetele (értéke), vagy vizsgálatunkhoz kapcsolódóan egy ügyfél kiszolgálási ideje. Egy bizonyos ügyfél kiszolgálási ideje természetesen egy valós időtartam, azonban ha a kiszolgálási időt, mint fogalmat tekintjük, értéke ügyfélről ügyfélre változik. A kiszolgálási időre hatással lehet az ügy (szolgáltatás) jellege, az ügyintéző és az ügyfél személye, de még az időjárás is.

Valószínűségi változók az eloszlásukkal adhatóak meg. A valószínűségi változók felvehetnek diszkrét értékeket (pl. kockadobás kimenetelei), vagy folytonos értékeket (pl. kiszolgálási idő). Az eloszlás megadható a (kumulatív) eloszlásfüggvénnyel (CDF: Cumulative Distribution Function) vagy az eloszlás sűrűségfüggvénnyel (PDF: Probability Density Function).

$$F: \mathbb{R} \rightarrow [0,1] \qquad F(x) := \mathbf{P}(X \leq x)$$

Az F eloszlásfüggvény egy olyan valós függvény, amely minden értékhez megadja, hogy milyen valószínűséggel vesz fel valószínűségi változó -nél kisebb vagy egyenlő értéket:

A P valószínűségi mérték definíciója szerint az az esemény, amely sosem következik be 0 valószínűségű, a biztos esemény valószínűsége 1. A teljes eseménytér valószínűsége ezért 1 (minden lehetséges kimenetel közül az egyik biztosan be fog következni). Az eloszlásfüggvény értéke ezért 0-tól 1-hez tart a lehetséges értékek felett.



32. sz. ábra: Normális eloszlású eloszlásfüggvények

(Forrás: https://commons.wikimedia.org/wiki/File:Normal_Distribution_CDF.svg)

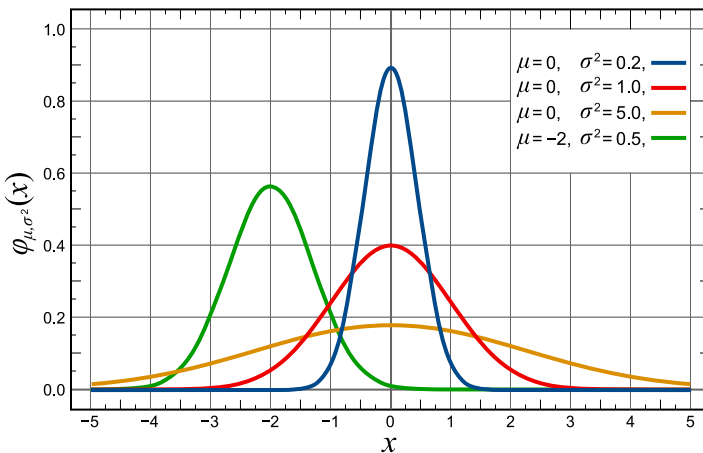
Az eloszlás f sűrűségfüggvénye is definiálja az eloszlást. Az eloszlás sűrűségfüggvény az eloszlásfüggvény deriváltja, azaz minden pontban az eloszlásfüggvény meredekségét adja meg. Fordítva is igaz, a sűrűségfüggvény integráltja adja meg a kumulatív eloszlásfüggvényt:

$$F(x) = \int_{-\infty}^x f(t) dt$$

A sűrűségfüggvény egy pontjánál felvett érték neve „likelihood”, ami csak diszkrét esetben igaz, folytonos esetben valószínűségnek félrevezető lenne nevezni, mert folytonos valószínűségi változó minden értékét 0 valószínűséggel veszi fel. Abban az esetben beszélhetünk valószínűségről, ha két pont között, a sűrűségfüggvény alatti területet nézzük: ez annak a valószínűsége, hogy a valószínűségi változó a két érték (a, b) közötti értéket veszi fel.

$$P(a, b) = \int_a^b f(t) dt$$

A sűrűségfüggvény integráltja 1. Azaz annak a valószínűsége, hogy bármely lehetséges értéket felveszi a valószínűségi változó, biztos esemény.



33. ábra. Normális eloszlás sűrűségfüggvénye
(Forrás: https://commons.wikimedia.org/wiki/File:Normal_Distribution_PDF.svg)

Amennyiben sorra megmérjük az egyes ügyfelek kiszolgálási idejét, meghatározhatjuk a kiszolgálási idő tapasztalati eloszlását. Fontos megjegyezni, hogy ez nem egyezik meg a valószínűségi változó elméleti eloszlásával. Minél több adatot gyűjtünk, annál közelebb kerülhetünk hozzá, de a kettő sosem fog megegyezni, hiszen hiába gyűjtöttünk már rengeteg adatot, még végtelen számú további mérés hiányzik, hogy megkapjuk a valódi elméleti eloszlást, ami egyébként általában nem is ismert.

Folytonos eloszlás esetén felmerül a probléma, hogy bármilyen értéket felvehet egy esemény jellemzője. (A mérésünk felbontásától, pontosságától függ, hogy két mérést egyenlőnek mérünk, azonban annak a valószínűsége, hogy két mért érték valóban megegyez-

zen, nulla.) A tapasztalati eloszlásfüggvény esetén ezért szükséges diszkrétizálni, vagyis meghatározni azon lehetséges kimenetek számát (és mindegyik tartományát), amelyeket megkülönböztetünk. Ez alapján minden értéktartomány (osztályköz) esetén a mérések alapján meghatározhatjuk, hányszor fordult elő az esemény. Ezt hívjuk hisztogramnak, tapasztalati gyakoriságfüggvénynek. Amennyiben az előfordulások számát normalizáljuk (minden előfordulásszámot elosztunk a mérések számával), az összes tartomány relatív gyakoriságának összege 1-et ad eredményül. Ez a relatív gyakoriságfüggvény jó becslője a valódi (elméleti) eloszlás sűrűségfüggvénynek. Bár a hisztogramból is származtatható, a (kumulált) tapasztalati eloszlásfüggvény felvételéhez nincs feltétlenül szükség osztályközökre. A lépcsős tapasztalati eloszlásfüggvény felvehető úgy is, hogy az értékek növekvő sorrendében – tipikusan $1/n$ -nel – növeljük a függvény értékét, ahol n az adatok (mintaelemek) száma. Ez finomabb képet ad az eloszlásról, mint az osztályközös relatív gyakoriságok kumulálása.

Diszkrét eloszlás esetén a (relatív) gyakoriságok a valószínűségi változó által felvett értékekhez közvetlenül rendelhetők, ezért nincs szükség osztályközökre a sűrűség megjelenítéséhez.

3.1. Statisztikák

Számos nevezetes eloszlás létezik, melyek jól leírják bizonyos jellemzőkkel rendelkező valószínűségi változókat. Ilyen pl. az egyenletes, a normális és az exponenciális eloszlás. Ezek tárgyalása a következő alfejezetekben következik.

Az eloszlások, valószínűségi változók számos különböző mutatóval (statisztikával) jellemezhetők – még abban az esetben is, ha ezek nem határozzák meg egyértelműen az eloszlást.

3.1.1. Várható érték statisztika

Egy valószínűségi változó $E[X]$ várható értéke az az érték, amelyet a valószínűségi változó legnagyobb valószínűséggel felvesz, egy átlagos érték: a lehetséges értékek súlyozott átlaga a hozzájuk tartozó likelihood értékekkel. A valószínűségi változó nem is feltétlenül veheti fel ezt az értéket, pl. kockadobások esetén az érték 3,5.

Diszkrét eloszlás esetén:

$$E[X] = x_1 p_1 + x_2 p_2 + \dots + x_n p_n = \sum_{i=1}^n x_i p_i$$

A becslése: $\sum x_i / n$.

Mivel $\sum_i p_i = 1$, tekinthető súlyozott átlagnak is. Egyenletes eloszlás esetén a kifejezés egyszerű számtani átlagra egyszerűsödik: $p_i = 1/n$.

Adott eloszlás sűrűségfüggvény esetén:

$$E[X] = \int_{-\infty}^{\infty} f(t) t dt$$

3.1.2. Standard szórás, variancia statisztika

A szórásnégyzet vagy variancia statisztika az értékek átlagos négyzetes hibáját adja meg. Más szavakkal az átlagtól való eltérések négyzeteinek átlaga vagy az átlagtól való négyzetes eltérés várható értéke. A szórás (σ) a szórásnégyzet (variancia, $\text{Var}[X]$) négyzetgyöke.

$$\sigma^2 = \text{Var}[X] = E[(X - E[X])^2] = E[X^2] - E^2[X]$$

$$\sigma = \sqrt{\text{Var}[X]} = \sqrt{E[(X - E[X])^2]} = E\sqrt{[X^2] - E^2[X]}$$

A valószínűségi változó szórása a minták alapján becsülhető, de sosem határozható meg pontosan. Minél több minta áll rendelkezésünkre, annál megbízhatóbb a becslés. Ezt a becslést hívjuk tapasztalati (empirikus) szórásnak ($\hat{\sigma}$).

$$\hat{\sigma}^2 = \frac{1}{N-1} \sum_{i=1}^N (x_i - \bar{x})^2$$

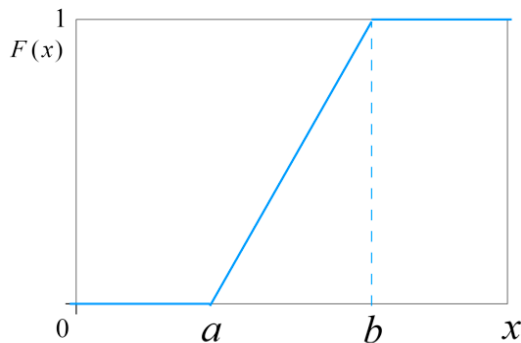
3.2. Egyenletes eloszlás

Egyenletes eloszlásról (uniform distribution) abban az esetben beszélhetünk, ha nincs olyan körülmény, amely miatt az egyik kimenetel valószínűsége nagyobb lenne, mint egy másik. Ilyen például egy kockadobás kimenetele: a kocka bármelyik oldala kerülhet felfelé, a szabályos dobókockának nincs kitüntetett lapja.

Az eloszlásnak két paramétere van (két értékkel egyértelműen megadható): (a, b) megadja a minimális és maximális értéket, amelyet a valószínűségi változó felvehet. Az (a, b) pontok között a likelihood értéke $1/(b-a)$, ezen kívül mindenhol 0.

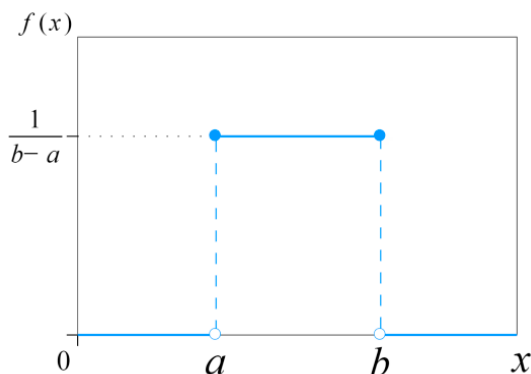
Az eloszlás várható értéke: $E[X] = (b-a)/2$.

Variancia: $\sigma^2 = (b-a)^2/12$.



34. sz. ábra: egyenletes eloszlás eloszlásfüggvénye

(Forrás: [https://en.wikipedia.org/wiki/Uniform_distribution_\(continuous\)#/media/File:Uniform_cdf.svg](https://en.wikipedia.org/wiki/Uniform_distribution_(continuous)#/media/File:Uniform_cdf.svg))



35. sz. ábra: egyenletes eloszlás sűrűségfüggvénye

(Forrás: [https://en.wikipedia.org/wiki/Uniform_distribution_\(continuous\)#/media/File:Uniform_Distribution_PDF_SVG.svg](https://en.wikipedia.org/wiki/Uniform_distribution_(continuous)#/media/File:Uniform_Distribution_PDF_SVG.svg))

3.3. Normális eloszlás

A normális eloszlás két paraméterrel jellemezhető: a várható értékkel és a szórással. Természetesen a paraméterek megegyeznek a várható érték és a szórási statisztikák értékeivel. Speciális esete a standard normális eloszlás, amelynek a várható értéke 0 és a szórási (és szórásnégyzete) 1. Jelölése:

$$X \sim N(m, \sigma^2)$$

A normális eloszlás jól használható olyan esetek modellezése során, ahol a valószínűségi változó nagyszámú, független, azonos eloszlású változók összegeként (vagy átlagaként) jelenik meg. (A centrális határeloszlás-tétel alapján kis szórási független hatások összege normális eloszlású.) Ez az eloszlás nagyon gyakori a természetben, ahol sok kis véletlen tényező hatása összeadódik, például nagyszámú pénzdobás vagy dobókocka dobás átlagos értékének eloszlása.

Az eloszlásfüggvénye nem adható meg zárt alakban, csak a sűrűségfüggvénye. Az eloszlásfüggvény integrálással meghatározható, gyakran táblázatos módon is rendelkezésre áll.

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-m)^2}{2\sigma^2}}$$

Mivel a sűrűségfüggvénye szimmetrikus, egy maximuma van, módusza és mediánja is a várható értékkel egyezik meg.

Az eloszlásfüggvényből látható, hogy a normális eloszlású véletlen változó bármely valós számot felveheti (végtelen tartományú eloszlás): a várható értéktől távolodva egyre kisebb, de nem nulla valószínűséggel. Nem csak pozitív, de negatív számokat is felvehet. Gyakran használt megoldás, ha olyan véletlen változókat modellezünk normális eloszlással,

amelyekről fizikailag tudjuk, hogy nem vehetnek fel negatív értéket (vagy tetszőleges alsó vagy felső határt), akkor csonkoljuk az eloszlást. Ez a csonkolás megváltoztatja az eloszlásfüggvényt is. A szimulációs szoftverben is ez történik, ugyanis semmilyen feldolgozási idő nem lehet negatív.

Az eloszlás várható értéke nagyon beszédes paraméter, hiszen ez lesz a leggyakoribb érték, ekörül fog a valószínűségi változó értéke mozogni. Hogy mennyire, azt a szórás határozza meg. A szórás értelmezésében a következő táblázat nyújt segítséget: megmutatja, hogy a véletlen változó milyen valószínűséggel lesz adott tartományokon belül, m és s konkrét értékétől függetlenül.

Tartomány	$p_{\text{belül}}$	$p_{\text{kívül}}$
$m-\sigma..m+\sigma$	68,27%	31,73%
$m-2\sigma..m+2\sigma$	95,45%	4,55%
$m-3\sigma..m+3\sigma$	99,73%	0,27%
$m-4\sigma..m+4\sigma$	>99,99%	<0,01%

1. sz. táblázat: a véletlen változó előfordulási valószínűsége

3.4. Szimmetrikus háromszög eloszlás

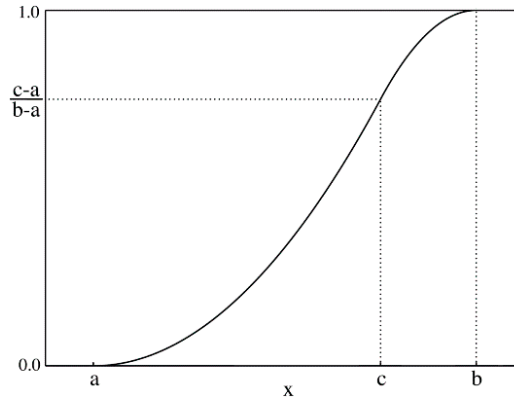
A szimmetrikus háromszög eloszlás két paraméterrel definiálható: a várható érték és a szélesség. Az eloszlásfüggvénye jellegre hasonlít a normális eloszláshoz, azonban érték-tartománya nem végtelen, és ezért jobban behatárolható.

Szimulációs célból jól használható a normális eloszlás helyett például kiszolgálási vagy feldolgozási idők modellezésére.

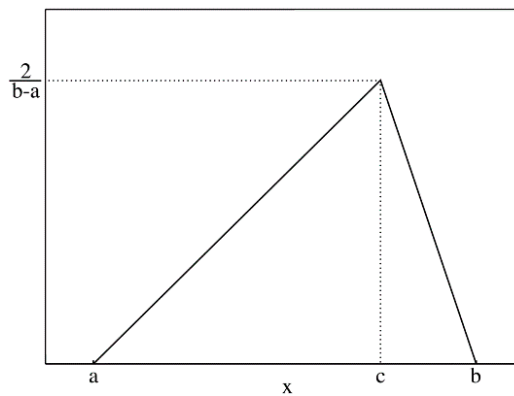
3.5. Aszimmetrikus háromszög eloszlás

Az aszimmetrikus háromszög eloszlás a szimmetrikus általános esete, három paraméterrel adható meg: minimum érték, módusz és maximum érték.

Az eloszlás lehetővé teszi a minimum és a maximum irányába mutató lefutás finomhangolását. Mivel a módusz már nem feltétlenül a minimum és maximum érték átlaga (mint a szimmetrikus esetben), megadható olyan eloszlás, ahol a minimum és a módusz viszonylag közel vannak, de a maximum jóval távolabbra is eshet.



36. sz. ábra: aszimmetrikus háromszög eloszlás eloszlásfüggvénye
(Forrás: https://commons.wikimedia.org/wiki/File:Triangular_distribution_CMF.png)



37. sz. ábra: szimmetrikus háromszög eloszlás sűrűségfüggvénye
(Forrás: https://en.wikipedia.org/wiki/Triangular_distribution#/media/File:Triangular_distribution_PMF.png)

3.6. Exponenciális eloszlás

Az exponenciális eloszlás egyváltozós eloszlás. A paramétert (λ) hívhatjuk rátának is. λ paraméter reciproka a várható értékkel, illetve a szórással egyenlő.

Memóriamentes (memoryless) vagy örökifjú folyamatok leírására szolgál, ilyen például egy üvegpohár túlélési ideje. Az üvegpohár, amíg össze nem törik, jó közelítéssel éppen olyan jól használható, mint újkorában. Az esemény (hogy a pohár összetörik) bármikor ugyanúgy bekövetkezhet (akár új a pohár, akár régi), így a pohár túlélési esélyei folyamatosan csökkennek. A gépek két meghibásodás közötti hibamentes működési ideje is jól közelíthető exponenciális eloszlással, ha csak véletlen hatások érvényesülnek.

Az exponenciális eloszlás szoros rokonságot mutat a Poisson-eloszlással, Poisson-folyamatokkal. A Poisson-folyamatok adott időnként bekövetkező események számát modellezik. Az események azonos valószínűséggel és függetlenül következnek be. A Poisson-eloszlás két időpont között bekövetkezett események számának eloszlását adja meg. Az

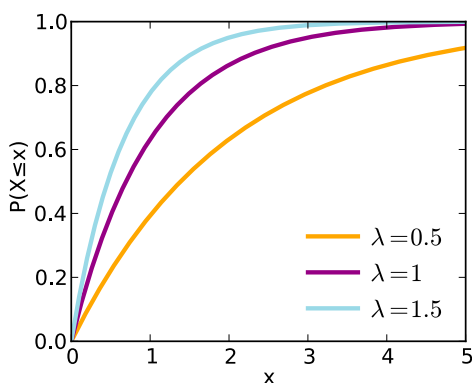
egy esemény között eltelt időt exponenciális eloszlás írja le. Ha az ügyfelek beérkezését eseményeknek tekintjük, exponenciális eloszlással jól modellezhető az ügyfelek beérkezése között eltelt idő. Természetesen ez abban az esetben igaz, ha nincs szezonális: a nyitva tartás bármely időpontjában azonos valószínűséggel érkezhetnek. A valóságban azonban nyitáskor lényegesen több ügyfél érkezik, mint a nap további részében.

Tekintsük független eseményeknek az ügyfelek beérkezését, átlagosan m időnként. Két esemény között eltelt idő exponenciális eloszlású, ahol $\lambda=1/m$.

$$f(x) = \lambda e^{-\lambda x}$$

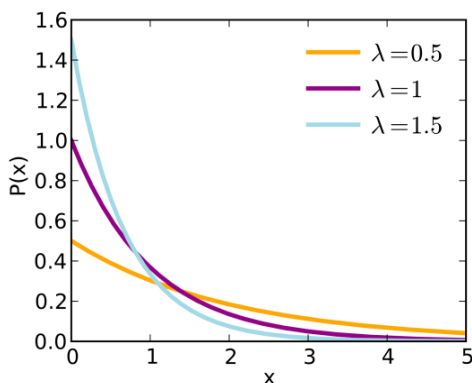
$$F(x) = 1 - e^{-\lambda x}$$

$$\text{Var}[X] = \frac{1}{\lambda^2}$$



38. sz. ábra: exponenciális eloszlás eloszlásfüggvénye

(Forrás: https://en.wikipedia.org/wiki/Exponential_distribution#/media/File:Exponential_cdf.svg)



39. sz. ábra: exponenciális eloszlás sűrűségfüggvénye

(Forrás: https://en.wikipedia.org/wiki/Exponential_distribution#/media/File:Exponential_pdf.svg)

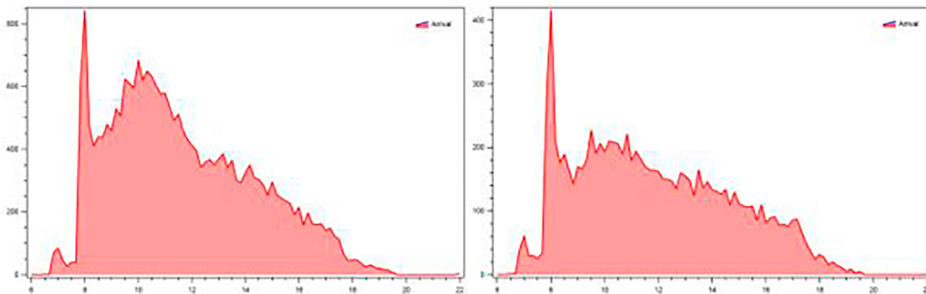
Ahogy az az összefüggésekből látható, az exponenciális eloszlású valószínűségi változó relatív szórása 1.

3.7. Előre meghatározott lista

A szimulációs szoftverben lehetőség van megadni egy valós számokból álló listát, melyből a program egymás után választja ki a számokat. Így lehetőség van tetszőleges eloszlású valószínűségi változók megadására.

3.8. Hisztogram

A hisztogram valószínűségi változók sűrűségfüggvényének becslésére szolgál. Egy táblázat, amelyben nyilvántartjuk, melyik érték milyen gyakorisággal fordult elő.



40. sz. ábra: ügyfelek beérkezésének eloszlása a nap folyamán (6-22 óra között), két kiválasztott szolgáltatás esetén
(Forrás: saját szerkesztés.)

A szimulációs szoftverben lehetőség van megadni ilyen gyakoriság táblázatokat, ezáltal modellezhetők tetszőleges eloszlások. Így akár a valóságban megfigyelhető ügyféligények szezonalitása is megvalósítható a modellben. A beérkező ügyfelek száma ábrázolható a nap folyamán adott intervallumokban, pl. 10 perces egységekben. A beérkező ügyfelek száma a nap folyamán így tekinthető egy valószínűségi változónak, mely nem hasonlít nevezetes eloszlásokra: a nyitás időpontjáig nulla ügyfél érkezik. A nyitástól kb. fél-egy órán át megfigyelhető egy csúcs, majd ez visszaesik, a nap folyamán állandó marad, zárás előtt pedig visszaesik nullára.

4. KIMUTATÁS, STATISZTIKA

A szimuláció futása közben vagy a befejezése után lehetőség van kimutatást, beszámolót generálni a „Generate report” gomb megnyomásával. A megnyomás után kinyílik az alsó sáv „Report” mezője, ahol megjelenik a kimutatás. A jobb olvashatóság érdekében célszerű a szöveget egy táblázatkezelő alkalmazásba vágólapon keresztül átmásolni (Ctrl+A mindent kijelöl, Ctrl+C másol, majd a táblázatkezelőbe beillesztés Ctrl+V kombinációkkal).

A kimutatás tartalmazza a szimuláció kezdetétől kialakult mutatók numerikus értékeit. A statisztika kiterjed a várakozási időkre, a kiszolgált ügyfelekre, feldolgozási időkre, kihasználtságokra stb. Külön kimutatások állnak rendelkezésre a szolgáltatási, illetve a termelési esetekre.

A kimutatás az egyértelműség és az eltérő nyelvi beállítások miatt minden tizedes törtet az angol írásmódban jelenít meg, azaz a tizedestört jele a „.” (pont). A másolás során beállítástól függően érdemes lehet lecserélni a pontot vesszőre, vagy a táblázatkezelőben beállítani a tizedesjelet.

Microsoft Excel (2013) esetén a Fájl menü Beállítások, Speciális ablakában a „Rendszerbeállítások szerinti elválasztók használata” kikapcsolásával és a „.” beírásával állítható be a decimális elválasztó karakter, vagy CTRL+H paranccsal elvégezhető a „.” és „,” cseréje. Utóbbi megoldás akkor célszerű, ha a táblázatok magyar nyelven kerülnek publikálásra.

A kimutatás során előfordulhat, hogy kismértékű eltérések lehetnek a várttól, pl. a százalékok összege nem adja ki a 100%-ot. Ez azért van, mert bizonyos paraméterek minden időpillanatban meghatározhatók, míg mások csak adott időnként, eseményekhez kötve frissülnek. Továbbá előfordulhat nem 100%-os kihasználtság, ugyanis a szimuláció beállításaitól függően időbe telhet, mire a rendszerben a munkadarabok a későbbi folyamatrészekhez is eljutnak.

A csonkolt normális eloszlás esetén a statisztikában csalóka értékek jelenhetnek meg. Mivel ez az eloszlás már nem valódi normális eloszlás, ezért az empirikus várható érték és szórás nem az eredeti nem csonkolt eloszlás paramétereit adja meg. Pl. csonkolt (negatív értékek levágva) normális $N(m=50; \sigma=100)$ eloszlás alapján a statisztika kb. $m \approx 100$, $\sigma \approx 70$ statisztikát ad.

A szimuláció futtatása közben nincs akadálya bizonyos paraméterek megváltoztatásának, azonban ennek hatása van a kimutatásra is. A futás közbeni módosítás egyrészt okozhat nem várt hibákat, valamint a kimutatás nem a végső modellt reprezentálja. Javasoljuk, hogy kimutatás készítése érdekében módosítás nélkül fusson szimuláció.

4.1. Szolgáltatási kimutatás

A szolgáltatási kimutatás az elején tartalmazza a szimuláció futási idejére vonatkozó paramétereket. A „Simulation start” mező a szimuláció kezdési időpontjára vonatkozik, a „Simulation end” pedig a szimuláció befejezési idejére. Ezek az idők nem azt mondják meg, hogy mikor futtattuk a szimulációt, hanem, hogy a szimulált virtuális idő mettől meddig tartott. A „Simulation duration” sor pedig a szimuláció időtartamát adja meg különböző formátumokban: nap.óra:perc:másodperc: törtmásodperc, illetve összes másodperc, összes óra.

A „Lead time” rész szolgáltatásonként és kijáratonként tartalmazza a teljes átfutási időket, sorban állással, kiszolgálással együtt:

- Szolgáltatás neve,
- Szolgáltatás száma,
- Kijárat név,
- Várható érték (s),
- Szórás (s),
- Minimum (s),
- Maximum (s).

Az „Administrator utilization” rész tartalmazza az ügyintézők kihasználtsági mutatóit:

- Ügyintéző név,
- Ügyintéző száma,
- Kiszolgált ügyfelek száma,
- Kihasználtság (%),
- Blokkolva (%),
- Hasznos idő (s),
- Blokkolt idő (s).

Az „Administrator processing times” tartalmazza az ügyintézők és a szolgáltatások kiszolgálási időinek statisztikáját:

- Ügyintéző név,
- Ügyintéző száma,
- Szolgáltatás név,
- Szolgáltatás szám,
- Összes kiszolgált ügyfél,
- Átlagos kiszolgálási idő (s),
- Kiszolgálási idő szórása (s),
- Minimum (s),
- Maximum (s).

A „Queue lengths” rész tartalmazza a sorban állási statisztikákat:

- Váróterem név,
- Átlagos sor hossz.
- A „Waiting times (total)” az összesített várakozási időket tartalmazza:
- Átlagos sorban állási idő (s),

- Szórás (s),
- Minimum (s),
- Maximum (s).

A „Waiting times (per waiting room)” tartalmazza az egyes várótermekre vonatkozó statisztikákat:

- Váróterem név,
- Átlagos sor hossz,
- Átlagos várakozási idő (s),
- Szórás (s),
- Minimum (s),
- Maximum (s).

A „Waiting times (per waiting room, per service)” rész tartalmazza a várakozási időket adott várótermek esetén adott szolgáltatásokra:

- Váróterem név,
- Szolgáltatás,
- Szolgáltatás szám,
- Átlagos hossz,
- Átlagos várakozási idő (s),
- Várakozási idő szórás (s),
- Minimum (s),
- Maximum (s).

4.2. Termelési kimutatás

A termelési kimutatás az elején tartalmazza a szimuláció futási idejére vonatkozó paramétereket. A „Simulation start” mező a szimuláció kezdési időpontjára vonatkozik, a „Simulation end” pedig a szimuláció befejezési ideje. Ezek az idők nem azt mondják meg, hogy mikor futtattuk a szimulációt, hanem, hogy a szimulált virtuális idő mettől meddig tartott. A „Simulation duration” sor pedig a szimuláció időtartamát adja meg különböző formátumokban: nap.óra :perc:másodperc:törtmásodperc, összes másodperc, összes óra.

A „Flow time” rész tartalmazza az egyes munkadarabok rendszeren történő átfutási idejét: a forrásuktól kezdve az egyes kimenetekig. Minden kimenet esetén leolvashatók a következő értékek:

- Feldolgozott munkadarabok,
- Átlagos átfutási idő (s),
- Szórás (s),
- Minimum (s),
- Maximum (s),
- Átlagos db/perc,
- Átlagos db/óra.

A „worker utilization” rész soronként tartalmazza az egyes munkások kihasználtsági adatait:

- Elkészített munkadarabok,
- Kihasználtság (%),
- Blokkolva (%),
- Hasznos idő (s),
- Blokkolt idő (s),
- Teljes idő (s).

A kihasználtság százalékos értéke azt adja meg, hogy a teljes szimulációs idő mekkora részében végzett a munkás tényleges munkát. Akkor nem végez hasznos munkát, ha munkadarabra és gépre vár, vagy blokkolva van. Blokkolt állapot abban az esetben lép fel, amikor elkészült egy munkadarabbal, de azt nem tudja a gépből kiadni (tele van a fogadó puffer, dolgozik a következő gép stb). Ilyenkor képes lenne munkát végezni, de a körülmények nem teszik lehetővé.

A „Worker utilization by machine” részben az egyes munkások gépenkénti munkaidejének paramétereit tartalmazza:

- Munkás név,
- Gép név,
- Kész munkadarabok,
- Kihasználtság (%),
- Blokkolva (%),
- Hasznos idő (s),
- Blokkolt idő (s),
- Teljes idő (s).

A „Worker operating at machines (%)” rész soronként tartalmazza, hogy melyik munkás, mely gépeknél, milyen arányban töltötte az idejét.

A „Worker processing times” tartalmazza a feldolgozási időkre vonatkozó értékeket:

- Munkás név,
- Összes munkadarab,
- Átlagos feldolgozás idő (s),
- Szórás (s),
- Minimum (s),
- Maximum (s).

Bár a munkások nincsenek hatással a feldolgozási időkre, csak a gépek, a munkások dolgozhatnak különböző gépeknél, így kialakul a munkások empirikus feldolgozási ideje.

Fontos megjegyezni, hogy szabályos normális eloszlású feldolgozási idők esetén az eloszlás paramétereit közelítik a tapasztalt feldolgozási idők várható értékei és szórásai. Azonban csonkolt, és főleg – a nagy relatív szórás miatt – jelentősen csonkolt normális eloszlás esetén a paraméterek és a meghatározott várható értékek, szórások jelentősen eltérhetnek, ez nem hiba, hiszen nem szabályos normális eloszlásból származnak a feldolgozási idők.

A „waiting times (total)” rész összesítve tartalmazza a várakozási időket, amelyeket a munkadarabok a pufferekben töltenek:

- Átlagos várakozás (s),
- Szórás (s),
- Minimum (s),
- Maximum (s).

A „waiting times (per buffer)” rész tartalmazza az egyes pufferekre vonatkozó részletes várakozási adatokat:

- Puffer név,
- Átlagos hossz,
- Átlagos várakozás (s),

Szórás (Kisebb érték esetén a várakozás kiszámíthatóbb, egyenletesebb. Nagyobb érték esetén nagy a fluktuáció, a puffer képes jól ellátni a feladatát, csökkenti a blokkolások hatását.),

- Minimum (s),
- Maximum (s).

A „Machine processing times” rész tartalmazza az egyes gépek feldolgozási idejének statisztikáját:

- Gép név,
- Átlagos feldolgozási idő (s),
- Szórás (s),
- Minimum (s),
- Maximum (s).

A „Machine utilization” rész tartalmazza az egyes gépek kihasználtságára vonatkozó értékeket:

- Gép név,
- Kész munkadarabok,
- Kihasználtság (%),
- Blokkolva (%),
- Hasznos idő (s),
- Blokkolt idő (s),
- Teljes idő (s).
- A „Machine scrap” a selejtekre vonatkozó adatokat tartalmazza:
- Munkadarabok (összes),
- Nem selejtek száma,
- Selejt szám,
- Selejt arány (%).

5. INTRODUCTION

In this manual we introduce a simple desktop software for process simulation, in order to demonstrate the concepts and building blocks of process models by letting students and users build simple process models with an easy to use interface. The models are built from a set of process elements or blocks and are defined by their connections. Each building block contains several parameters that can be easily tuned and their effects observed allowing students to understand theory with a more practical and visual way.

5.1. Why is simulation needed?

The purpose of simulation is to understand the features and essence of processes, and it allows us to answer the question to “What would happen if...?” without any financial or safety risks. We can change parameters and try different setups in order to find the optimal solution during design or a review phase.

Simulation requires a model that properly describes the steps, relations and features of the process. The level of detail and what “properly” means is always a valid question. These models are simplified descriptions of reality and as a thumb rule, the model should be simple but also detailed enough to grasp the general and studied behaviors of the real process. There is no need to describe details that have no effect on the studied features. As the model not only describes the relations and the steps of the processes, but features of the steps are also described. These features involve parameters such as processing times, input rate and so on. In case of modeling an existing real world process, such parameters must be measured or estimated as distributions in order to simulate significantly more cases than measured. In case of planning a new process, the parameters need be estimated, perhaps based on other (predefined) parameters. Later the estimations can be refined based on the results of the simulations.

The way the model is described is based on the framework that is used. Usually simulation frameworks provide ways and tools to describe the models, while also allowing to “operate” or “run” these models. This framework, the set of rules that describe how to operate the model, can be executed manually or more optimally by software. An example for manual operation can be any board game or the so called “Beer Game” that simulates the activities of parties in a supply chain such as inventory management, ordering and transportation. By focusing only on key activities, it is still able to highlight the features of the real system in an exceptionally illustrative way.

Simulation software are based on this very same principal: they offer a set of rules, the description of the process as the model (that is based on the provided rules) and simulator, the framework to operate the process. The advantage compared to the manual operation is that it requires far less time to simulate thus much longer times of more complex models can

be studied while reducing the likelihood of mistakes. Also the simulation can be automated to test several parameters under stochastic circumstances (Monte Carlo simulation) in order to evaluate the sensitivity and robustness of systems.

During planning phase of processes, the resource requirements can be estimated by simulation, but also existing systems can be revised to optimize. Existing processes provide real operating data, such as ordering and processing times or time distributions that can be incorporated in the simulation offering more realistic estimations of “what would happen if...” scenarios. For example: what would have happened if the demands increased but service times remain the same.

Numerous process simulation software are available (Arena, VisSim, Siemens Tecnomatix, Simulink etc.), that can be used to model and simulate whole business or production processes. The purpose of this manual is to provide an introduction to the topic, with a simple, freely available desktop software.

5.2. Overview

The software offers the modeling and simulation of two process types. The first type is the *service process*, which can be used to model a government window. In this scenario clients arrive to ticket machines in order to select a service offered by the provider. The client receives a ticket and waits in the waiting room for his or her turn at the window. The client is served at a window by an administrator, this takes service (processing) time, defined by a custom distribution. Finally, the satisfied client leaves through the exit.

The software allows several customization of the model: arbitrary services can be defined, each administrator can serve any sets of services, each at a definable random time distribution. Customers arrive at selectable random times for each service while also opening hours can be set up for ticket machines and working hours for administrators. Bottlenecks of the processes can easily be identified, solutions to resolve them can be evaluated. What would happen if we removed or added a window, or trained an administrator to serve more services: would it decrease the waiting times significantly?

The second model type is the *production process*. In this scenario work items arrive from a source, which are processed by machines that are operated by workers. Machine operations are executed in a given processing time (distribution) and can be concatenated after each other. The flow can branch and merge, allowing to model more complex scenarios. It can easily be evaluated how random distributions affect the flow, where bottlenecks are formed, how buffers ease fluctuation, how utilization can be improved, or how cross training benefits the flow time.

The user interface of the application is in English. Input boxes can be used to manually enter numbers or values. For clarity, English formatting is used for both input and output formatting despite system settings: the decimal point is denoted by the “.” (point) character.

5.3. Installation

The process simulator application is developed using the “.NET” and the “Eto.Forms” frameworks, allowing it to run on multiple platforms, while the user interface and the speed remain similar across operating systems as the native graphical user interface is used on each platform. The latest version of the software can be found at: <http://servitization.uni-nke.hu>

5.3.1. *Minimum requirements*

The software can run on Windows (7-10), OSX (Mac) and most Linux distributions. A minimum display resolution of 1280x720 is recommended.

5.3.2. *Windows*

In order to run the application, .NET Framework 4.6 must be installed. This is included in recent Windows versions by default (Windows 10).

5.3.3. *OSX*

On OSX (Mac OS) Mono is required to run “.NET” applications. Mono is available at:

<https://www.mono-project.com/download/stable/#download-mac>

or

<https://goo.gl/KLpgJ9>

The software is provided as a .dmg directory image. After opening the image, the application is started by double clicking on the icon.

5.3.4. *Linux*

On Linux distributions Mono package is required to run the application. Mono is available at:

<https://www.mono-project.com/download/stable/#download-lin>

```
sudo apt-get install mono-runtime
```

```
sudo apt-get install gtk-sharp2
```

6. DETAILED DESCRIPTION OF THE SIMULATION

In this chapter we introduce interface of the application, the building blocks that can be used to create a model and also describe the simulation execution rules.

6.1. User interface

The application interface is shown in Figure 41. The main region where the model is built (1) by dragging and dropping with the mouse cursor. The building blocks (2) can be found above the model on tab panels for different process types: Service, Production. The region under the main area can be collapsed or opened to show messages or reports (3).

The top menubar contains the usual items in the File menu: New is used to reset the project to the startup state. A project can be loaded from file with the “Load” menu item. Projects can be saved to file using the “Save” or “Save as” items. Data from a spreadsheet file can be imported using the “Import” menu.

The “Scenarios” menu contains predefined projects that can be used as startup models and can be customized further. A scenario loads by simply selecting it. Modifications can be saved to file, the original scenarios cannot be overwritten.

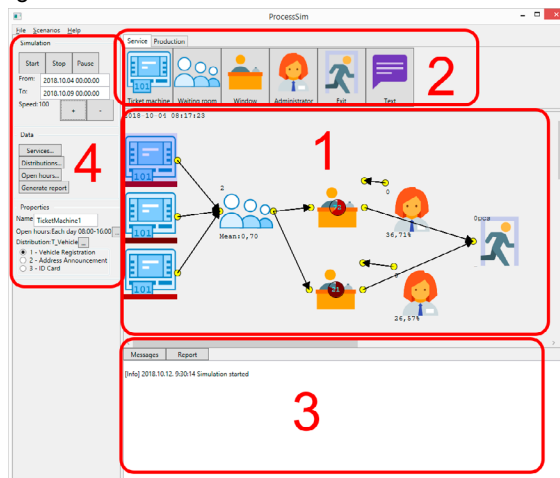


Figure 41: user interface of the simulation application
1) Model drawing region, 2) Ribbon for the process elements,
3) Message region, 4) Control area

On the left side of the user interface the control area (4) can be found. The first groupbox (“Simulation”) contains controls and settings for the simulation. The Start, Stop, Pause buttons can be used to control the simulation after the model is ready. The starting and ending time of the simulation can be given as a direct date and time. The simulation speed can be increased by the “+” button or decreased by the “-” button.

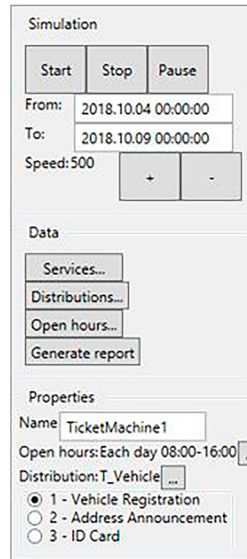


Figure 42: main control panel

The “Data” section is used to set up the base data used throughout the project. The “Services” button opens a dialog box (Figure 43) that is used to define arbitrary service names with identifier numbers. This function is used only in service model types, irrelevant in production models. All the given services must be associated with a unique positive integer numbers and non-empty service names. Any line with a zero identifier or an empty name will be removed on closing the window. New lines can be added by the “New” button.

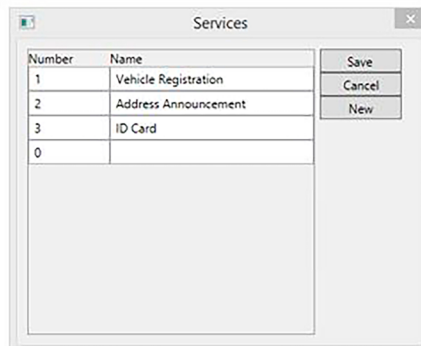


Figure 43: Service definition dialog window

Distribution used throughout the project can be defined by opening the “Distributions” dialog window (Figure 44). All the defined distributions are listed in the box on the left of the window (1). The bottom controls (2) can be used to add a new distribution or delete the selected one. Each distribution must have a valid name. It is advised to make the name talkative: “T_ID”: Ticket distribution for personal IDs, or “P_Address”: Processing time distribution for Place of residence declaration, etc. Selecting a defined distribution by its name allows the user to modify its type and properties (3). The type can be selected by the radio buttons, under the radio buttons properties can be set for each type of distribution. By changing the distribution type the settings reset to default and this is saved. In order to make sure the parameters are saved as well press the “Save” button. On the right side of the window a preview (4) of the probability density function is shown for each distribution. The properties of some distributions can be changed by dragging the mouse button horizontally or vertically in this preview area: the mean value and the standard deviation can be changed by dragging the mouse cursor horizontally or vertically.

Distribution types can be selected from a given list:

- Uniform,
- Truncated normal,
- Triangle distribution (symmetric),
- Asymmetric triangle,
- Exponential,
- Histogram (empirical probability density function),
- List of predefined numbers.
- A detailed description of the distributions is given in chapter 7.

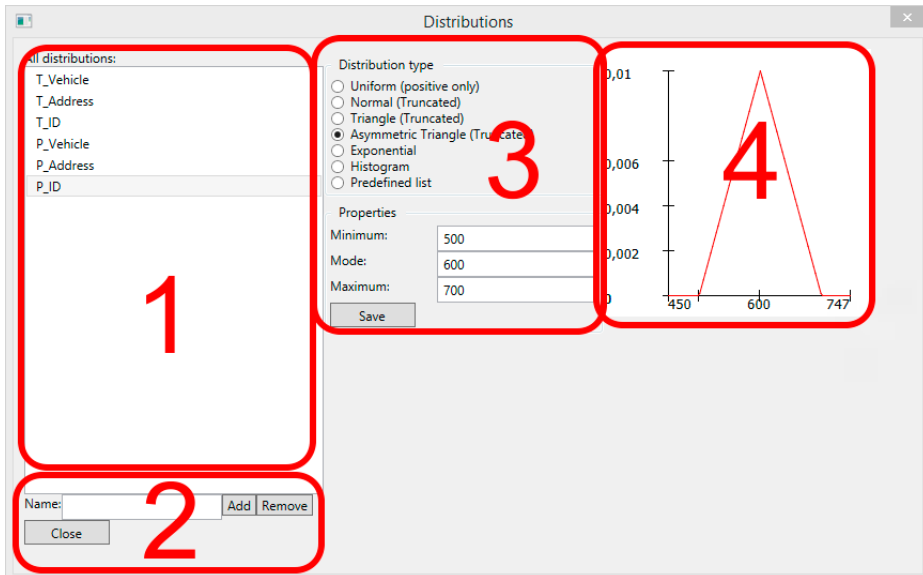


Figure 44: distributions dialog window

A list of open hours can be set up by opening the dialog box (Figure 45) by clicking the “Open hours” button in the main control panel. On the left side (1) of the window all the defined open hours are shown. New ones can be added, or selected ones can be removed by the bottom “Add” and “Remove” buttons. The middle section (2), “Selected open hours” are only used if the window is opened for the settings of a given control. Multiple open hours can be applied for a given control, by moving the line to the selected region by the “→” button. Selected ones can be removed by the “←” button. Settings can be set up on the right side of the window (3): a given day of week can be selected or any day, which applies to all the weekdays. Also a start and end time must be defined in “hour:minute” format, assuming 24 hours format. Names are automatically generated from the settings.

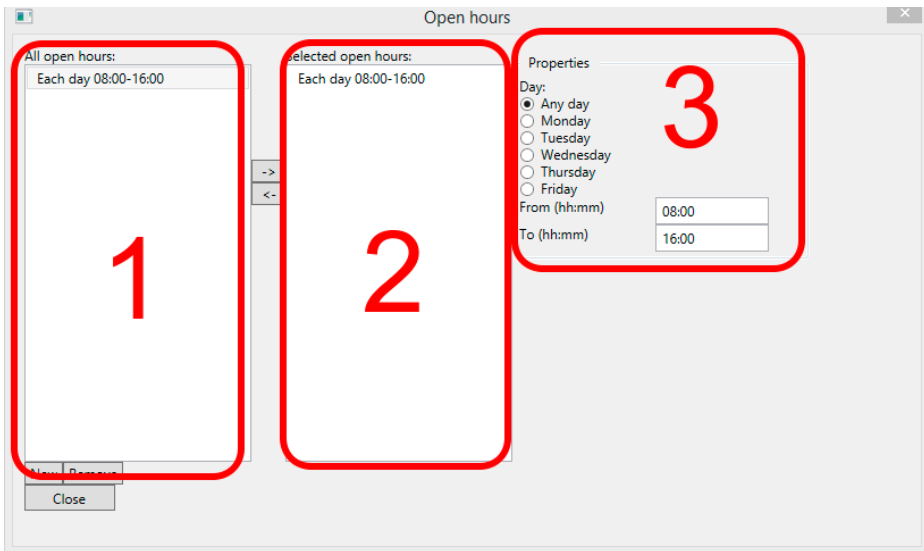


Figure 45: open hours dialog

The “Generate report” button on the main control panel is used to generate an output report of the simulation. This report is based on the data from the beginning of the simulation up to the time it is pressed, or if the simulation already ended: to the end. The report text is shown in the bottom region. It can be closed or opened by the “Report” button. It is advised to copy the report to a spreadsheet application, as the text format is not the most suitable to present the data. Also further calculations can be done in spreadsheets. The clipboard can be easily used for this purpose. Simply select all the text (Ctrl+A), copy (Ctrl+C) and paste it in the spreadsheet (Ctrl+V). Reports are described in details in chapter 4.

At the bottom of the main control panel the properties of the selected process element can be found. Any changes here are applied immediately to the process element. In case some base data is changed while the properties are open, please reselect the control to update settings panel.

For example if a ticket machine is selected and the service list is modified, the ticket machine must be selected again to refresh the service list.

6.2. Process elements

Process elements are the building blocks of the process models. Models are defined by the blocks, their connections and settings. In this section the process elements are presented: their main features and settings.

Elements can be inserted to the model editor region by clicking on the appropriate button in the ribbon. An element can be selected by simply clicking on it in the editor region. The selected element is highlighted and can be deleted by the “Delete” key. The elements can be moved by dragging them with the mouse pointed while pressing the mouse button.

Each element may contain inputs and outputs. Inputs (left side, black) can be connected to outputs (right side, white), outputs to inputs. This can be done by pressing and holding the mouse button on an input (or output) and dragging it to a connection point where it is ought to be connected. Connecting to default input or outputs is even simpler: press on an input or output, drag and release on the process element itself. Special connections, such as where administrators or workers connect are colored different, but also behave similarly.

Connections are visualized by straight lines ending in an arrow. A connection can be removed by applying the same connection again, or selecting it (color changes from black) and pressing the “Delete” key.

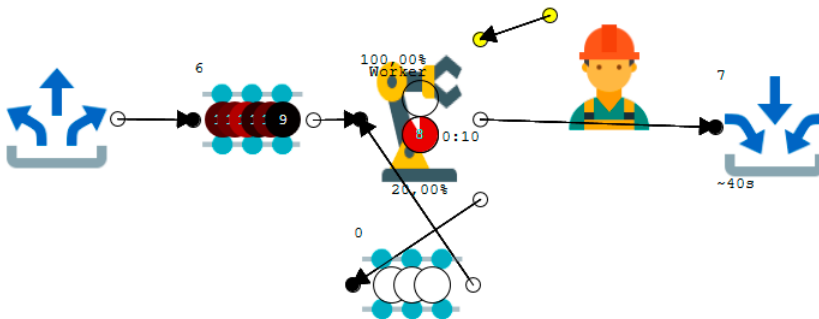


Figure 46: an example production model

The settings of a selected element is shown at the bottom of the main control panel. The available settings differ for each element type, but one is common: the name of the element. The name is used in the reports to make identification of each element easier. It is advised to give a unique name for each block.

6.2.1. Service process elements

Service type of processes can be created using the following blocks:

- Ticket Machine,
- Waiting room,
- Window,
- Administrator,
- Exit.

In service processes clients with their demands appear, they wait until they can be served, finally they leave. In the following sections we introduce the main features of these elements.

6.2.1.1. Ticket machine



Figure 47: icon of the ticket machine

The ticket machine is responsible for generating the clients or their “demands”. It generates clients for a selected service, throughout the open hours with a given distribution. In order to set up the ticket machine correctly, open hours, distribution, daily distribution and a service must be selected.

Figure 48: Properties of the ticket machine element

Figure 8 shows the settings of the ticket machine. By clicking the “...” button next to “Open hours” the previously described open hours dialog window appears. Here open hours must be selected from the available ones by the “→” button, or even new ones can be defined. The ticket machine will generate tickets only in the selected time intervals.

A distribution can be selected by the “...” button next to it. The selected distribution defines how often clients arrive to the system. Daily ticket distribution applies only if the distribution itself doesn’t determine the number of daily clients (ie. the Histogram). The operation of the ticket machine is described in more details in section 6.4.2. By default, the distribution defines the time difference of customers arriving for a given service. By selecting exponential distribution with an expected value of 60 (seconds), customers arrive every minute by average, some arrive closer to the other, sometimes more than a minute later. This random number is given by the distribution.

Finally, a service type must be selected. In reality one ticket machine is responsible for giving out numbers for all the services, however for better visualization and easier setup, one ticket machine is responsible here for the demands for one service only. Naturally any num-

ber of ticket machines can be added to the model, even multiple ticket machines providing the same service. Each service is associated with a color, the bottom of the ticket machine is colored according to the selected service.

6.2.1.2. Waiting room

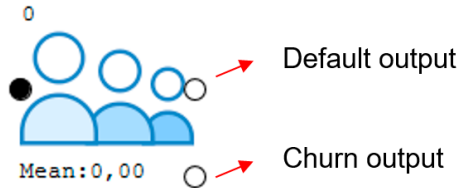


Figure 49: Icon of the waiting room

The waiting room is used to store the tickets (clients) waiting to be served at windows by administrators. It has two outputs: default and churn. The default output is where clients leave normally (to windows). The churn output is where churn clients leave. In case the size of the waiting room is limited, any incoming client goes to the internal (unlimited) churn buffer and will leave through this output. If no input is connected to the churn output, the waiting room blocks incoming clients on its input.

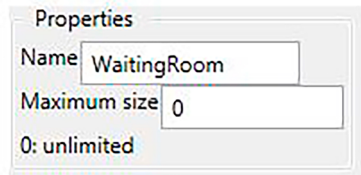


Figure 50: settings of the waiting room

During simulation the mean waiting time and the mean queue length is shown for each waiting room element.

The waiting room is not a pure FIFO buffer, it is FIFO by services. When a window becomes free, the first customer is called who can be served at that window.

6.2.1.3. Administrator

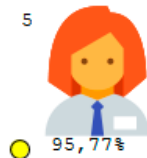


Figure 51: icon for the administrator

The process element describing the administrator does not have a real flow-type of input or output. However, it still has a connection point that can be connected to windows. This determines at which windows is he or she allowed to take place.

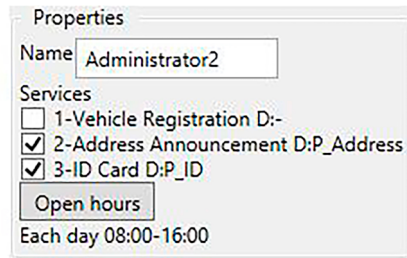


Figure 52: Properties of the administrator

For each administrator a set of services can be selected that the administrator can serve. For each service, a service time can also be specified. When the checkbox is ticked, the administrator is able to serve a selected service. Upon checking a box the distribution selector dialog window appear, where the service time can be selected. It is important to differentiate services by processing times, as different services may have significantly different service times. Also, an experienced administrator may handle clients faster than the one still learning the process. (Naturally the service time depends on the client as well, hence the random variables.)

The administrator also has a work time, denoted by open hours. Administrators only accept clients during these active times. At the start of each administrator's work time the simulator assigns a window (from the associated windows) to the administrator.

During simulation the number of handled clients and the utilization is shown in real time.

6.2.1.4. Window



Figure 53: icon of the window

Clients are served at windows by administrators. If no administrator is associated to a window, that window does not accept clients. A window only accepts clients who can be served by the associated administrator.

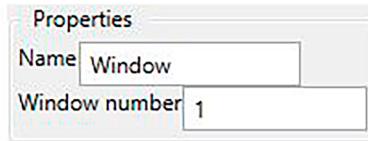


Figure 54: the name and the identifier number can be set for a window

The window only acts as a workplace for the administrator where it accepts clients, it does not have any effects on the interaction between the administrator and the client. The name and the identifier number helps identification in the reports.

By default, clients go to and exit after finishing at a window. It is possible to link multiple windows after each other, but it must be justifiable to do so to represent real processes.

During simulation a circle represents the client, the processing time is visualized as this circle is being filled in a clockwise style.

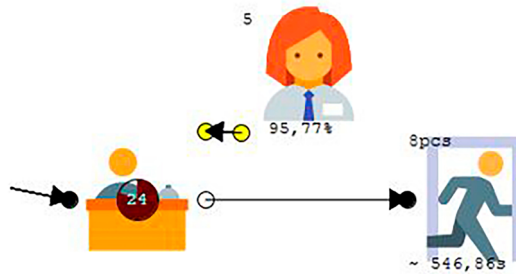


Figure 55: an example of a window-administrator-exit layout

6.2.1.5. Exit

Exits are used to collect served customers. It has one input and no outputs. In order to differentiate in the reports, multiple exit elements may be added to the model. One example could be to have a normal exit for customers who were served and on other exit for churn customers who gave up waiting as the waiting room was full.

During simulation the number of customers and the mean flow time in seconds is shown in the icon of the exit. This statistics combines all the clients who entered this exit, it is not differentiated by services.



Figure 56: icon and real time statistics for the exit

It is advised to direct churn customers to a second exit not to distort statistics of the exit. An example layout is given in Figure 57.

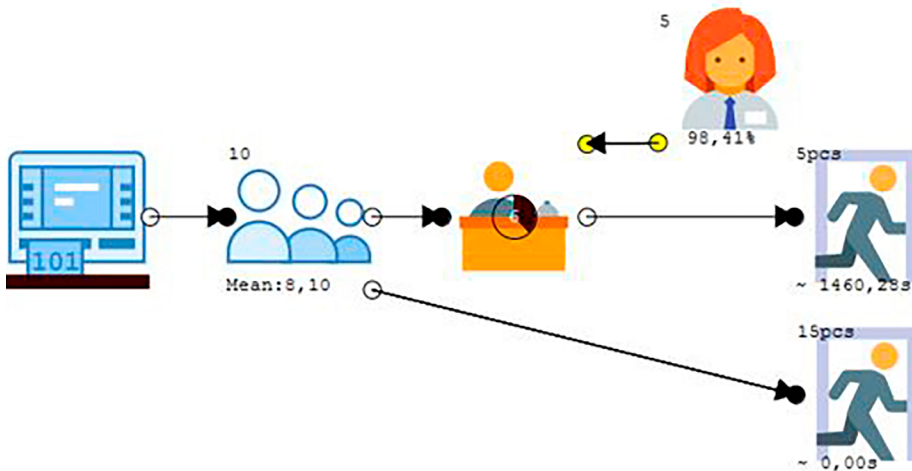


Figure 57: example layout for handling churn clients by directing them to a different exit

6.2.2. Production processes

In production type of processes work items flow from sources to sinks. In service processes clients were served at windows, here machines process work items. Workers are needed to operate machines.

6.2.2.1. Source



Figure 58: icon of the source

The source generates work items that flow through the model of the process. It has no inputs and one output.

By default, the source outputs work items whenever the input of a connected process element is not blocked (when it is able to receive a work item).

For each source it can be configured to output a given number of work items ("Total items generated"), setting this value to zero will produce unlimited number of items.

In case of limited work items are generated and all the items were processed, the simulation automatically stops. This is useful to compare situations with different parameters or setups. If the number of items is not limited, the simulation end time marks the end or the user may stop the simulation manually.

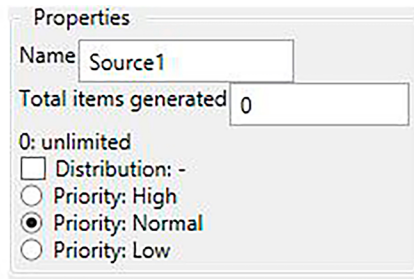


Figure 59: the number of generated items is unlimited by setting the value to zero

The default output mode can be changed by selecting an output distribution. This way items are only output with a defined distribution not always when a receiver can accept it. The source in this mode has an internal buffer to store the work items that were generated but not yet output.

6.2.2.2. Buffer

The buffer serves as a temporary FIFO (first in first out) storage location (inventory) of work items. The allowed maximum number of items can be configured to a finite number or it can be unlimited (denoted by the size of zero). By default, buffers are located between machines, in order to reduce blocked times and compensate the nature of random processing times, thus increasing utilization.

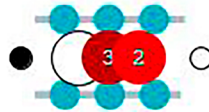


Figure 60: icon of the buffer, showing the actual workitems in the queue

It is also possible to define an initial value for the buffer. At the beginning of the simulation a given number of work items are generated and put to the buffers. This allows the simulation to reach a steady state (if it exists) faster, or it can be used to study the properties of the model, whether it can overcome the situation of a completely filled buffer and return to a normal operating state. Buffers usually fill up in reality when machines break and cannot process items. The last operating machine produces inventory.

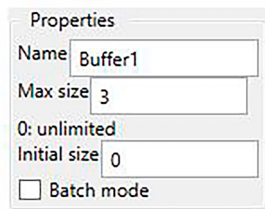


Figure 61: settings of the buffer.

Batch mode can be selected for a buffer. This means that the buffer collects work items until it is full. It starts to output work items until it is empty, then it starts collecting again. Other (larger) buffers can be connected to both sides of the batch buffer to transfer multiple batches.

6.2.2.3. Machine

Machines operate on work items. The operation takes time, essentially machines introduce delay to the system. These items have one input, where work items are accepted, have a default output, a scrap output and a connection point where workers may be connected.

A worker is needed to operate the machine. The process time can be determined by a distribution. By a given ratio the output may arrive to the scrap output instead of the default one. The scrap may be handled by other process branches.



Figure 62: during simulation the icon of the machine shows real time information

The machine may be used directly after the source to output work items by a given time distribution, but the source has settings for the same purpose.

A machine may incorporate multiple workplaces, allowing multiple workers to process multiple work items at the same time, resulting in higher throughput. Multiple workers do not reduce the processing time of a single work item. By default, workers are locked to a workplace until a finished work item is output, however this may lead to deadlock in some situations. Deadlock is when multiple units are waiting for the other one, thus cannot proceed with their operations. In order to avoid deadlock situations, it is worth having one worker dedicated to each machine. For adding cross-trained workers to improve throughput use multiple-workspace machines.

Workers can be associated to machines by the yellow connection point.

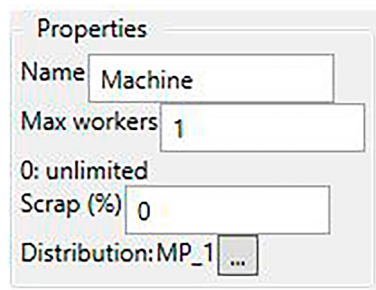


Figure 63: properties of the machine

During simulation the image of the machine is updated with real time information: which work item is under operation, what the progress is of the process, which workers are associated to the machine, what the utilization is.

6.2.2.4. Worker



Figure 64: icon of the worker

The worker operates machines. Workers can be associated to dedicated machines or can be cross-trained to be able to work on multiple (or even all of the) machines. At machines that offer multiple workplaces, multiple workers can operate on the same number of work items. An algorithm determines the association of workers and machines. Moving the workers between machines has no cost in time nor value.

For simplification, there is no need to connect a cross-trained worker with all the machines: at the settings of the worker it can be explicitly stated that the worker can work at any machine.

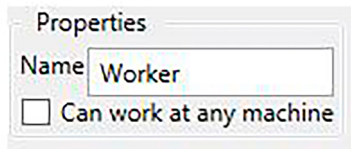


Figure 65: Properties of the worker

6.2.2.5. Splitter

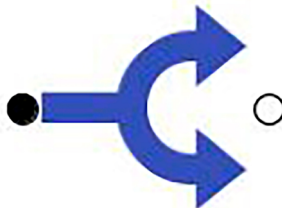


Figure 66: icon of the splitter

The splitter can split a work item into multiple work item-parts, which can flow on different work item paths and can be combined back to complete work items using the Combiner process element. Work item parts also behave as complete work items, machines can operate on each one of them, can be put into buffers and so on. Each process element on its output receives one work item-part.

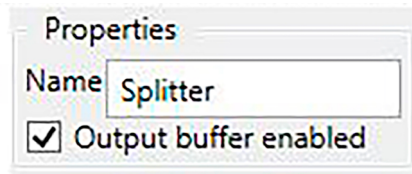


Figure 67: settings of the splitter

The splitter has one setting, whether it should have an internal output buffer or not. If it is not checked, it has no buffer and will input a work item only if all of the process elements on its output is ready to receive. If it has an internal buffer enabled, it receives a work item, splits it to pieces and outputs to the appropriate outputs when they are ready. It receives a work item if this buffer is completely empty.

6.2.2.6. Combiner



Figure 68: icon of the combiner

The combiner collects work item-parts from its input and combines them to a complete work item. It is used in conjunction with the splitter. It does not have any special settings.

6.2.2.7. Sink

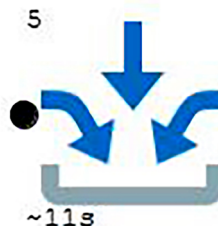


Figure 69: icon of the sink

The sink is used to collect work items that are ready to leave the simulation. Statistics from each sink in the simulation is presented in the report.

During the simulation real time statistics are shown: number of work items arrived, mean flow time of the accepted work items.

6.3. Other process elements

The “Text” item can be used to add text fields to the model. These elements have no effect on the simulation, but can be used to add comments or directions for the user.

6.4. The ruleset of the simulation

Depending on the modeled process, the simulation can happen based on short, equal time steps (quantums), or it can be event based, where events can happen at arbitrary times. The first solution is used if the process can be modeled mathematically, and the relation between the inputs and outputs are constant, such as in physical systems, for example electronic or mechanical networks. The precision of the simulation is defined by the time quantums used to refresh the state of the systems.

The simulator described in this manual uses the latter, event based solution. This provides a more optimal solution: most of the times there is no change in the state of the process, most process elements are waiting for something to be available on their inputs, even machines are waiting until the work items are finished. There is no need to recalculate the state of the system so often, only when something actually happens. Although the drawing of the model during simulation happens periodically it is only loosely connected to the simulation itself. This approach allows to have very fast simulation times.

The simulator asks each process element when their next event would happen: such as end of the client serving time, end of machine processing time etc. The earliest time will be the next actual time and one of the process elements requesting this event time will be allowed to process the event. If multiple elements report the same event time, a randomly picked one will be served. This may result in new elements requesting new or the same event times, these will be served in the next round. The rule is to process events only in time order.

In the following sections, a detailed description is given how the process elements are simulated.

- Each element can provide the following information at any time, or execute their events:
- Is the process element able to receive a work item (or a client ticket) on its input (if it has an input)?
- Is the process element ready to output a work item, or a client ticket (if it has an output)?
- When will the next event happen for the process element? (Unknown can be an answer as well.)
- Accept a work item or a client ticket on its input.
- Output a ticket or a workitem on its output to an input of a different element.

6.4.1. Service processes

Service processes are designed to model a simple one time serving scenarios, with multiple services, ticket machines, a waiting room, multiple windows and an exit. The administrators can be trained to serve multiple types of services. It is possible to add multiple serving steps (windows) after each other, but the following operation rules must be taken into account.

6.4.2. Ticket machine

The ticket machine generates a set of tickets for the day at the opening time. This allows to use multiple types of distributions. From the simulation's point of view there is no difference if these are precalculated values. Tickets are generated based on the selected open hours.

During simulation the output of the ticket machine must not be blocked, otherwise the simulation stops with an error. This is by design. Normally the output of the ticket machine leads to a waiting room, which serves as a buffer. However if the number of waiting customers is limited in the waiting room, its input can be blocked.

Ticket machines do not have inputs, the next event will always be the time of the ticket that is not yet output.

6.4.3. Waiting room

By default the waiting room is always ready to input client tickets, applying to any service. If the maximum size of the waiting room is limited, it is still able to input to its churn buffer, but only if a process element is connected to its churn output. If the waiting room is full and churn clients are not handled, the input is blocked. This situation may lead to (intentional) simulation errors. It is intentional in order not to create false reports about how well the model is operating.

When a client is in the waiting room and there is a process element (window) that is able to accept the client, the next event time will be "now" otherwise "unknown".

6.4.4. Administrator

The administrator does not have a normal flow-type of input or output, however it plays an important role in the model. The training of the administrator determines what sort of services he or she can serve and with what time distributions.

The administrator deals with two events: the beginning and the end of the workday. The default total work time is based on these two in order to calculate the utilization.

6.4.5. Window

Clients are served at windows. A window is operating only if a worker is associated. The training of the worker determines what services are accepted at a window (also by what time distribution). The exact service time is determined when a client is accepted. The next event time is the time when the client finishes. If no client is at the window, the next event time is "unknown". Usually clients head to the exit, which is always ready to accept clients, thus blocking rarely happens.

6.4.6. *Exit*

The exit is the most simple process element. It is always ready for input and does not need any events to be served. It is used for statistical purposes.

6.4.7. *Source*

The source creates work items for the process. By default it creates and outputs new work items whenever a receiver is ready to accept, not blocked. This is why it is not suggested to add an unlimited buffer on the output of a source.

The number of work items to be created can be limited to a specific value. As long as this number is not reached and any element is ready to receive, the requested event time of the source will be “now” otherwise “unknown”.

6.4.8. *Buffer*

The buffer provides a temporary FIFO (first in first out) storage of work items. By default the input is ready if the storage is not full and the output is ready if there is at least one element in the buffer.

If an element on the output is ready to receive, the requested event time will be “now” (to output an item), otherwise it will be unknown.

In case of batch mode, input and output ready state is different: first the buffer is in receive mode, the input is ready to receive as long as it is not full. The output is not ready even if there are items in the buffer. When the buffer becomes full it switches to output mode: the input is not ready, the output is ready as long as there are items in the buffer. When it becomes empty it switches to receive mode again.

6.4.9. *Machine*

The machine provides delay in the model: the delay time it takes to process a work item. In order to operate the machine a worker is needed. The processing time and whether the item will be scrap is determined on the input of a work item. The scrap leaves through the scrap output.

The input of the machine is ready if there is a worker associated with the machine and the workplace does not contain a work item.

The output of the machine is ready if a processed work item is available at a workplace and a worker is associated with the workplace.

The machine cannot request an event if there is no active processing or an element is not ready to input a work item from the machine. If the processing of a work item is in progress but there is element on its output ready to receive, the next event time will be the time if the end of the processing. In case of multiple work items in progress, the soonest will be the time of the event. If the operation is completed and there is an element ready to receive, the event time will be “now”.

The processing time distribution is a feature of the machine, not the worker – in contrast to the service scenarios.

6.4.10. Worker

The worker does not have flow type of inputs or outputs but they can be associated to machines. Multiple workers may work at machines, processing multiple work items in parallel.

A worker can only be associated to one machine at a time. By default the worker cannot leave the workplace until the work item is output. This can change in some situations, but a worker is needed to output a work item.

As long as there are no cross-trained workers, this works without question. However locking a cross-trained worker to machines until work items are output may lead to deadlocks. Imagine a cross-trained worker who is able to operate multiple machines: A and B. Machine A finishes a work item, but cannot output it to machine B, as there is no worker associated to it yet to receive the item. In order to reduce the deadlock situation in such cases workers are allowed to move. In case of a deadlock situation the simulation stops and notifies the user in the message box.

The association of workers to machines always starts with workers who are not cross trained. The association of cross-trained workers are based on momentary states. The association algorithm determines which machines are starving, blocking thus may have impact on the simulation results.

6.4.11. Sink

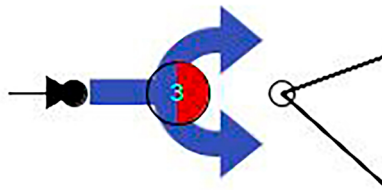
The sink has one input, no outputs. It is always ready to receive work items. It is mostly used for statistical purposes. It may be worth adding different sinks for different routes for work items. It does not request any events.

6.4.12. Splitter

The splitter creates work item parts from an input work item and outputs them to all the elements connected to its output.

It can be configured whether it should be equipped with an output buffer or not. It has an input buffer by default.

If it does not have an output buffer, the splitter only outputs work item parts only if all of the elements on its output are ready to receive. If it has an output buffer, it outputs where it can and waits for all of the parts to be output. Then it can receive the next workitem. This option may have a huge impact on the flow time, thus this is up to the user to activate it or not.



*Figure 70: the splitter with activated output buffer.
It can be seen that one part is still waiting for its receiver to be ready*

6.4.13. Combiner

The combiner block is used to combine workitem parts created by the splitter. The combiner waits for parts on all of its inputs and outputs a complete workitem. It has input buffers by default, thus it can input one part from all of the elements on its input. It also serves a synchronization function between the branches of the workitem parts: it only outputs if all the parts are ready: the slowest branch determines the flow time of all the branches.

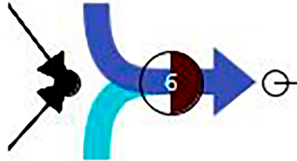


Figure 71: one out of two workitem parts have arrived to the combiner

7. RANDOM VARIABLES, PROBABILITY DISTRIBUTIONS

The application allows to define several parameters as random variables by random probability distributions. In this chapter, these random variables are described in details.

Random variables are the concept of measurable variables, which may have different values, outcomes at different times or situations. Such a concept can be the outcome of a dice throw, or related to the topic of process simulation: the time it takes to serve a customer. The time it takes differs depending on the case, the administrator, the customer or even the weather. It cannot be determined exactly, however the possible time range can be estimated.

Random variables can be defined by probability distributions. Such variables may have discrete values (such as a dice throw – 6 different outcomes are possible), or continuous values (such as service time). The probability distribution can be given by the cumulative distribution function (CDF) or the probability density function (PDF).

The F cumulative distribution function is a real valued function, such that for each x value it returns the probability of X random variable being less than x :

$$F: \mathbb{R} \rightarrow [0,1] \qquad F(x) := P(X \leq x)$$

The P probability measure defines that an event which would never happen (impossible event) has a probability of 0, a sure event that happens every time has a probability of 1. The probability of an event happening over the complete event space is 1: one event of all the possibilities will surely happen. Thus the cumulative probability function is an increasing function from 0 to 1 over the possible outcomes.

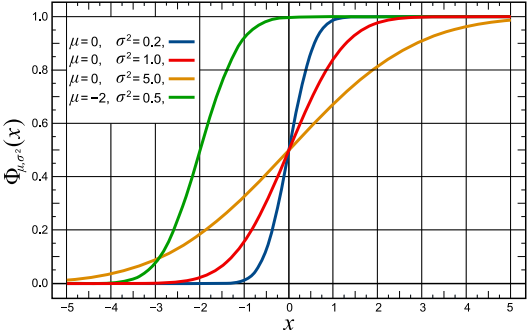


Figure 72: CDF of normal distributions
(Source: https://commons.wikimedia.org/wiki/File:Normal_Distribution_CDF.svg)

The probability density function also defines the probability distribution. The density function is the derivate of the CDF, each point in the PDF is associated to the slope of the CDF at the same point. Also, the other way around, the integral of the PDF gives the CDF:

$$F(x) = \int_{-\infty}^x f(t) dt$$

The value at each point of the PDF is denoted the likelihood (not probability!). Probability is the area under the PDF: it gives the probability of an outcome being in the range of (a, b) , out of all the possible outcomes:

$$P(a, b) = \int_a^b f(t) dt$$

The integral of the PDF is 1: the probability of the outcome being in the range of all the possible values is always sure, "1".

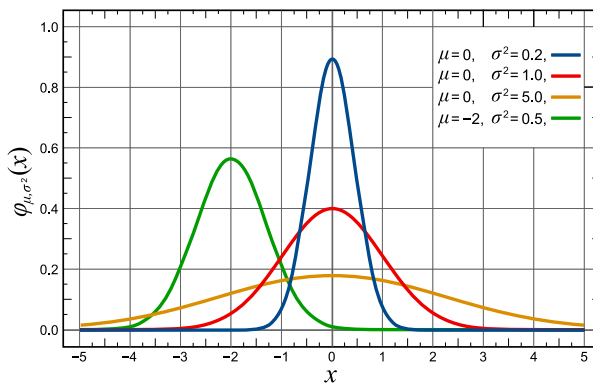


Figure 73: PDF of normal distributions

(Source: https://commons.wikimedia.org/wiki/File:Normal_Distribution_PDF.svg)

In case we measured many of the service times for clients, the empirical probability distribution can be determined. This does not equal the "real" or "theoretical" probability function, but as more and more measurements are done the closer they get (there would always be infinite more measurements that are missing).

In case of continuous distribution functions the outcomes may have any possible value in the range defined by the function. (The accuracy and resolution of our measurement determines if we consider two measurements equal, however in reality the probability of two outcomes being the same is zero.) The empirical distribution function should be discretized: the number of possible outcomes must be limited by defining ranges (bins) in which the values are considered equal. For each bin the number of events can be associated. This is called the histogram. In case the numbers in the bins (frequency) are normalized, such that the value stored in each bin is divided by the total number of measurements (giving the relative frequency), it would provide a good estimation of the PDF. The more measurements are available, the better the estimation is. The width of the bins must be chosen carefully.

7.1. Statistics

There are numerous common distribution functions, which describe different real processes and have special properties. Such distributions include uniform, normal, exponential etc. In the next sections the most common distributions are introduced.

Distribution functions can be characterized by several features, statistics, even if these statistics don't define the distributions completely.

7.1.1. Expected value

The expected value of a random variable $E[X]$ is the value that can be described as an "average" in the long run: values weighted by the likelihood of the values. The random variable may not even be able to take this value: the expected value of dice throws is 3.5, although there is no such value on the dice.

In case of discrete distributions:

$$E[X] = x_1p_1 + x_2p_2 + \dots + x_np_n = \sum_{i=1}^n x_i p_i$$

As $\sum_i p_i = 1$ the formula can be considered as the weighted average, where the weights are defined by the probability of each value. For uniform distributions each weight is equal: $p_i = 1/n$, resulting in a simple average formula.

For a general, continuous PDF the expected value is given by the formula:

$$E[X] = \int_{-\infty}^{\infty} f(t) t dt$$

7.1.2. Standard deviation, variance

The variance or squared standard deviation is given by the mean of the squared deviations from the mean. The standard deviation is denoted by sigma (σ), the variance ($Var[X]$) is the square of the standard deviation.

$$\begin{aligned} \sigma^2 &= Var[X] = E[(X - E[X])^2] = E[X^2] - E^2[X] \\ \sigma &= \sqrt{Var[X]} = \sqrt{E[(X - E[X])^2]} = E\sqrt{[X^2] - E^2[X]} \end{aligned}$$

The variance of a set of samples can be estimated, but is never going to be equal with the variance of the random variable. The more samples are available the closer the estimation will be. Estimation of the standard deviation is called the empirical deviation: ($\hat{\sigma}$).

$$\hat{\sigma}^2 = \frac{1}{N-1} \sum_{i=1}^N (x_i - \bar{x})^2$$

7.2. Uniform distribution

Uniform distribution models processes where the probability of each outcome is equal, when there is no circumstance that would make one value more likely than an other. A regular dice throw is a good example. All sides are equal, it can land on any with the same probability.

$$E[X] = \frac{b - a}{2}$$

The uniform distribution has two parameters: (a, b) that determine the minimal and maximal values the random variable may have. The likelihood between (a, b) points is $1/(b-a)$, 0 otherwise.

The expected value of the uniform distribution is:

$$\sigma^2 = \frac{b - a}{2}$$

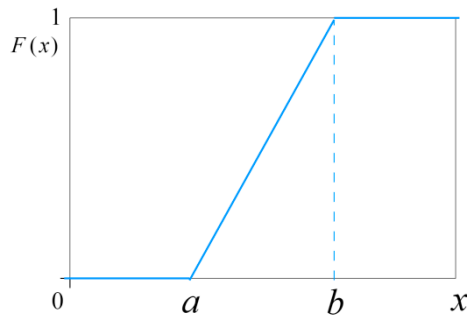


Figure 74: CDF of a uniform distribution

(Source: [https://en.wikipedia.org/wiki/Uniform_distribution_\(continuous\)#/media/File:Uniform_cdf.svg](https://en.wikipedia.org/wiki/Uniform_distribution_(continuous)#/media/File:Uniform_cdf.svg))

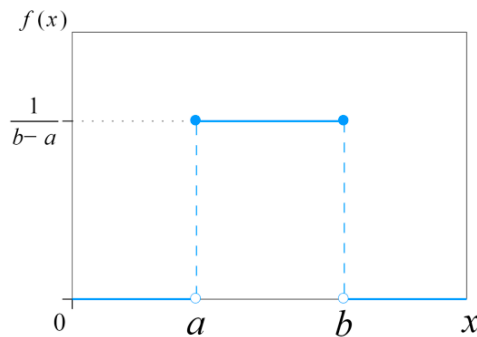


Figure 75: PDF of a uniform distribution

(Source: [https://en.wikipedia.org/wiki/Uniform_distribution_\(continuous\)#/media/File:Uniform_Distribution_PDF_SVG.svg](https://en.wikipedia.org/wiki/Uniform_distribution_(continuous)#/media/File:Uniform_Distribution_PDF_SVG.svg))

7.3. Normal distribution

The normal distribution is defined with two parameters: the expected value (or mean) and the standard deviation. Naturally the same statistics for the distribution yield the same values as the parameters.

$$X \sim N(m, \sigma^2)$$

A special case, the standard normal distribution is given by the parameters: $X \sim N(0,1)$.

The normal distribution is used to model random variables that are the sum (or average) of a large number of independent random variables of the same type. (See central limit theorem.) This distribution is very common in nature when a large number of identical small random factors contribute to a random variable. Such examples are the mean value (or the sum) of coin flips or the mean value of many dice throws.

The CDF of the normal distribution cannot be given in a closed form. Usually precalculated tables or estimations are available. However the PDF can be given:

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-m)^2}{2\sigma^2}}$$

As the PDF is symmetric and has only one maximum, the mode, median and the expected value all yield the same.

It can be seen from the PDF that a normal distribution may yield any real number: less and less probable as it is further from the mean, but the likelihood is never zero. It is common to model random variables with normal distribution, however if it is known that a value cannot have a given range (due to physical limitations), truncated normal distribution is used. Such as when a variable (service time) is modeled as normal distribution, it is important to limit the variable not to yield negative numbers. Service times cannot be negative. In reality this truncation changes the distribution function (it will not be normal any more) and the parameters (mean and variance) will also be distorted.

The mean parameter of the normal distribution is very self-explanatory: the outcomes of the random variable will be around this value, but how far? It is less and less likely that the outcome will be far from the mean value. This is defined by the variance or the standard deviation. The following table describes the likelihood of a random variable inside the following regions:

Region	p_{inside}	p_{outside}
$m-\sigma..m+\sigma$	68,27%	31,73%
$m-2\sigma..m+2\sigma$	95,45%	4,55%
$m-3\sigma..m+3\sigma$	99,73%	0,27%
$m-4\sigma..m+4\sigma$	>99,99%	<0,01%

7.4. Symmetric triangle distribution

The symmetric triangle distribution can be described with two parameters: expected (mean) value and width. The distribution resembles the normal distribution, however the range of outcomes are limited (unlike the normal) and it does not have so many special features as the normal distribution.

For simulation purposes it can be used as a random service or processing times, just like the normal distribution.

7.5. Asymmetric triangle distribution

The asymmetric triangle distribution is a generalized triangle distribution, given by 3 parameters: minimum value (a), mode (c) and the maximum value (b). It can be used to fine tune the falloff of the high and low sides of the PDF. As the mode is not necessary the mean of the minimum and maximum value, it can define distributions where the minimum and the mode are close, but the maximum value may be further away.

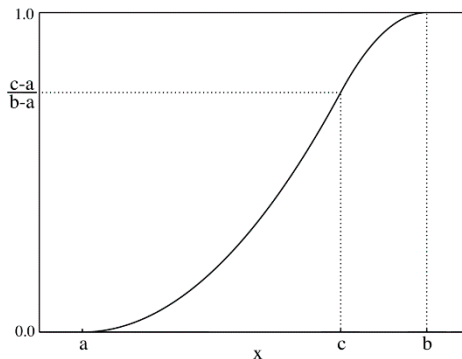


Figure 76: CDF of an asymmetric triangle distribution

(Source: https://commons.wikimedia.org/wiki/File:Triangular_distribution_CMF.png)

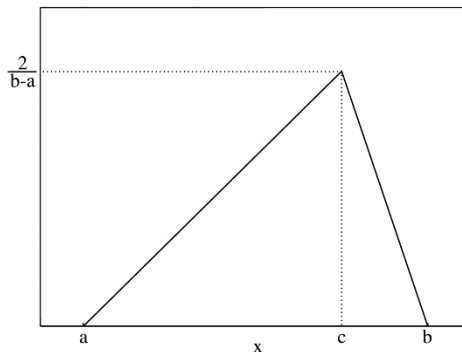


Figure 77: PDF of an asymmetric triangle CDF

(Source: https://en.wikipedia.org/wiki/Triangular_distribution#/media/File:Triangular_distribution_PMF.png)

7.6. Exponential distribution

The exponential distribution is defined by one parameter: rate (λ). The multiplicative inverse ($\beta=1/\lambda$) can also be used to define the distribution, it is called the scale. The value of scale equals the expected value ($E[X]=1/(\lambda)=\beta$) and the standard deviation ($\sigma=E[X]=1/(\lambda)=\beta$) as well.

Exponential distribution is used to model memoryless processes such as the lifetime of a glass. The probability of breaking the glass is constant throughout its lifetime, but can happen any time. The probability of survival decreases as an exponential function. Service times may be also modeled well as exponential distributions.

The exponential distribution has a close relationship with the Poisson process and distribution: the Poisson process models the number of events occurring between fixed time intervals. These events occur at the same probability and independently. The Poisson distribution gives the likelihood of the number of events occurred in a given time period. The time between events is given as an exponential distribution. Client arrival to government windows can be thought as events: thus the exponential distribution models the time between clients well. However in reality the probability of arrival changes throughout the opening hours: more clients tend to arrive in the morning when the office opens.

Let us consider the arrival of clients as independent events every m times average in a given time period. The time between two arrivals is given as an exponential distribution where $\lambda=1/m$.

$$f(x) = \lambda e^{-\lambda x}$$

$$F(x) = 1 - e^{-\lambda x}$$

$$Var[X] = \frac{1}{\lambda^2}$$

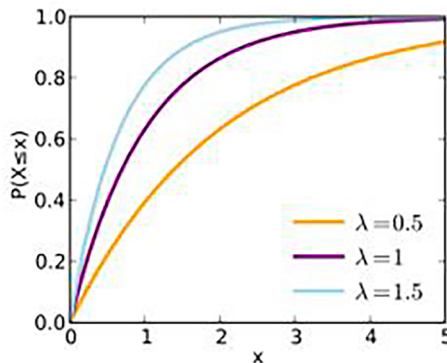


Figure 78: CDF of exponential distributions

(Source: https://en.wikipedia.org/wiki/Exponential_distribution#/media/File:Exponential_cdf.svg)

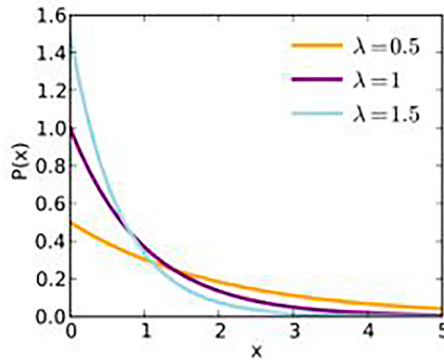


Figure 79: PDF of exponential distributions

(Source: https://en.wikipedia.org/wiki/Exponential_distribution#/media/File:Exponential_pdf.svg)

7.7. Predefined list

It is possible to select set up predefined lists of real numbers in the application as a source of random variables. This enables the user to use custom distributions.

7.8. Histogram

Histograms are used as empirical estimation of discretized probability density functions. It is essentially a table containing the relative number of frequency of different events. Events denote the value of the random variable being in given ranges.

In the simulation application this enables the user to create and use custom distributions. It also enables the system to handle seasonality of client arrivals: considering a random variable to be the number of clients arriving at a given time throughout a day (at a given resolution, such as in 10 minute periods) has distribution function: zero clients arrive until opening hours. There is a peak of customers in the first hour or so, then it falls down significantly. This value is kept throughout the day, finally it falls down nearing the closing time.

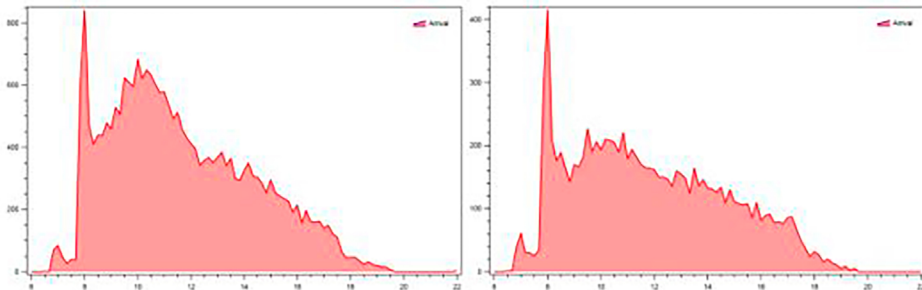


Figure 80: customer arrival time distribution throughout the day, for two selected services between 6am and 22pm (Author's source.)

8. REPORTS, STATISTICS

Reports can be generated during or after the simulation by pressing the “Generate report” button. After the button press the report box opens at the bottom. For better readability it is suggested to copy the text to a spreadsheet application through the clipboard (select all: Ctrl+A, copy: Ctrl+C and paste it to a worksheet in Excel: Ctrl+V).

The report contains statistics since the beginning of the simulation until the time the button is pressed (or to the end of the simulation if it has been stopped). These statistics provide information about service or processing times, utilization, etc. The detailed description of the contents of the reports are presented in this chapter. Different reports are generated for service and production scenarios.

Due to confusion by different system settings the numerical values in the reports are presented in the English formatting of fractional numbers (“.” dot as the decimal sign). In Microsoft Excel, the dot can be set manually as the decimal sign: File|Settings|Special, turn of use systems settings and enter the “.” Symbol manually.

Due to rounding or because some statistics update in real time while others only at events, it may happen that the sum of percents do not exactly add up to 100%.

Statistics (mean and standard deviation) of truncated normal distribution may be misleading. It is due to the fact that a truncated normal distribution is not normal anymore and statistics do not give the parameters of the original distribution. As an example: truncated (negative values cut off) normal $N(m=50;\sigma=100)$ distribution gives a statistics of about $m\approx 100, \sigma\approx 70$.

8.1. Service report

The service report starts with runtime statistics: “Simulation start” and “Simulation end” shows the start time of the simulation and the end time or the report generation time (both virtual times). “Simulation duration” is available in time format (day.hours:minutes: seconds.fraction) and total seconds, total hours as well.

“Lead time” part contains information based on workitem generation time and the time workitems entered exits. It involves waiting times, service times etc. Statistics are grouped by services:

- Service name,
- Service number,
- Exit,
- Mean (s),
- Standard deviation (s),
- Minimum (s),
- Maximum (s).

“Administrator utilization” contains statistics about work times of the administrators:

- Name,
- Number,
- Number of served,
- Utilization (%),
- Blocked (%),
- Useful time (s),
- Blocked time (s).

“Administrator processing times” provides information about the serving time of clients – by service types:

- Name,
- Number,
- Service name,
- Service number,
- Total count,
- Mean processing time (s),
- Standard deviation (s),
- Minimum (s),
- Maximum (s).

The “Queue lengths” section provide statistics of the waiting rooms:

- Name,
- Mean length.

„Waiting times (total)” section provides summarized statistics of the waiting times:

- Mean (s),
- Standard deviation (s),
- Minimum (s),
- Maximum (s).

„Waiting times (per waiting room)” shows statistics for the waiting time of clients per individual waiting room:

- Name,
- Mean length,
- Mean waiting time (s),
- Standard deviation waiting time (s),
- Minimum waiting time (s),
- Maximum waiting time (s).

Most detailed statistics of the waiting time is given in the section „Waiting times (per waiting room, per service)“:

- Name,
- Service name,
- Service number,
- Mean length,
- Mean waiting time (s),
- Standard deviation waiting time (s),
- Minimum waiting time (s),
- Maximum waiting time (s),

8.2. Production report

The production report begins with information about the runtime of the simulation: „Simulation start“ and „Simulation end“ shows the virtual start and end times. „Simulation duration“ presents information about the virtual runtime in different formats: day.hours:minutes:seconds.fraction, total seconds, total hours.

A „Flow time“ contains information about the time workitems spent in the simulation. For each sink, the following statistics is given:

- Workitems completed,
- Mean (s),
- Standard deviation (s),
- Minimum (s),
- Maximum (s),
- Mean pcs/min,
- Mean pcs/h.

„Worker utilization“ part contains information about work time, operational and blocked states of each worker:

- Workitems completed,
- Utilization (%),
- Blocked (%),
- Useful time (s),
- Blocked time (s),
- Total time (s).

Utilization percent denotes the fraction of the total simulation time that is spent by actually operating machines that process workitems. After finishing a workitem the worker may end up being blocked by not being able to output a finished workitem from the workplace of the machine: The receiver machine or buffer may be full. This time is integrated in the blocked time statistics.

The „Worker utilization by machine” section shows statistics of workers at each machine:

- Name,
- Machine,
- Workitems completed,
- Utilization (%),
- Blocked (%),
- Useful time (s),
- Blocked time (s),
- Total time (s).

The section „Worker operating at machines (%)” gives information about what each cross trained worker spent at different machines.

„Worker processing times” presents information about the processing time of workers.

- Worker name,
- Total count,
- Mean processing time (s),
- Standard deviation (s),
- Minimum (s),
- Maximum (s).

Processing times depend on machines, not workers, however workers may operate different machines, thus machine operating time affect worker operating times.

It is important that in case normal distribution is used as processing time, the mean and standard deviation of the empirical processing times approach the parameters of the theoretical distribution set for the machine. However if the normal distribution is significantly truncated, the empirical statistics and the theoretical parameters differ.

The „waiting times (total)” section provides cumulative statistics about time spent in buffers:

- Mean (s),
- Standard deviation (s),
- Minimum (s),
- Maximum (s).

„Waiting times (per buffer)” shows waiting time statistics for individual buffers:

- Name,
- Mean length,
- Mean waiting time (s),
- Standard deviation waiting time (s),
- Minimum waiting time (s),
- Maximum waiting time (s).

The „machine processing times” section provides statistics of the processing time of machines. After processing more and more workitems these statistics should approach the theoretical mean and standard deviation statistics of the selected distribution:

- Name,
- Mean processing time (s),
- Standard deviation processing time (s),
- Minimum processing time (s),
- Maximum processing time (s).

The „machine utilization” section provides information about the operating and blocked time of the machines:

- Name,
- Workitems completed,
- Utilization (%),
- Blocked (%),
- Useful time (s),
- Blocked time (s),
- Total time (s).

„Machine scrap” section provides information of the scraps created by machines:

- Workitems completed,
- Not scrap workitems,
- Scrap workitems,
- Scrap percent (%).

9. FOLYAMATSZIMULÁCIÓS SZOFTVER — PÉLDÁK

Bevezetés

Jelen kötet arra hivatott, hogy bemutassa az Olvasó számára, miként épül fel és miként használható a szimulációs szoftver a kormányablakok felépítésének és működésének megértéséhez, hatékonyabb átlátásához.

A kormányablakok egy kormányzati reformprogram (Magyar Zoltán Közigazgatás-fejlesztési Program) lépéseként jöttek létre, amelyeket a közigazgatási rendszer olcsóbb és hatékonyabb működtetésének érdekében hoztak létre 2013-ban.

Egy olyan nagyméretű és összetett rendszer működtetése, mint amilyen az államigazgatás, minden kormányt komoly kihívások elé állít, kiváltképp napjainkban. A bürokratikus terhek csökkentése, az ügyintézés hatékonyabbá tétele, az adminisztratív folyamatok egyszerűsítése, átláthatóságának és közérthetőségének biztosítása és a technológiai fejlődésben rejlő lehetőségek kiaknázása egytől egyig kiemelt fontosságú célok ezen a területen. A közigazgatás fejlesztése és hatékonyságának növelése azonban csak hosszú távon valósítható meg, amit mi sem bizonyít jobban, mint hogy a területi államigazgatási szervek korszerűsítése több évtizede, gyakorlatilag a rendszerváltás óta napirenden szerepel a magyar kormányok programjaiban. 2010 előtt az államigazgatás minden szintjére a széttagoltság és az átláthatatlanság volt jellemző, ám az elmúlt években ezekhez az állapotokhoz képest a magyar közigazgatási rendszer központi és területi szinten is számos jelentős változáson ment keresztül.

A 2010-ben meghirdetett Magyar Zoltán Közigazgatás-fejlesztési Program (Magyar Program) keretében indított reformhullám célja az ügyfelek igényeit és érdekeit figyelembe vevő, ügyfélközpontú szolgáltató működés kialakítása volt (magaryprogram.kormany.hu). A Magyar Programban az eljárások egyszerűsítését, az ügyfélterhek csökkentését és egyégesen magas színvonalú, mindenki által hozzáférhető szolgáltatások kialakítását tűzték ki célként a jogalkotók. Annak érdekében, hogy emeljék a közszolgáltatások színvonalát, létrehoztak egy többcsatornás kormányzati ügyfélszolgálati rendszert, amely megteremtette az egyablakos ügyintézés lehetőségét az állampolgárok számára az egyablakos ügyfélszolgálatok, az úgynevezett one stop shopok kialakítása által (Kovács, 2016).

A Magyar Program (magaryprogram.kormany.hu) stratégiai célként jelölte meg a közigazgatás működésének hatékonyabbá tételét és a közszolgáltatások színvonalának emelését. A területi államigazgatás átfogó átszervezésének első állomása a fővárosi és megyei kormányhivatalok létrehozása, valamint ezzel párhuzamosan a kormányhivatalok integrált ügyfélszolgálati irodáinak, a kormányablakoknak a kiépítése volt. A területi reform második szakasza a járási rendszer kialakítása, melyhez kapcsolódóan a járási székhelyeken 300

új kormányablak kialakítása volt a cél országszerte. A kormányablakok kialakítása során azt a célt tűzték ki maguk elé a jogalkotók, hogy ne csak megyei, hanem alsóbb szinten is megvalósulhasson az egyablakos ügyintézés az állampolgárok számára, megkönnyítve és meggyorsítva ezáltal a hatósági folyamatokat.

Ezen kötetben bemutatott szimulációs szoftver célja az, hogy a kormányablakok folyamatszempontú működését megismertesse az olvasóval, miközben egyszerű és összetett példákon keresztül szintén bemutatja a program működését és átadja a hatékony használatához szükséges alapvető ismereteket. Jelen szakaszban az olvasó megismerkedhet a fokozatosan összetettebbé váló modell-példákkal, melyek minden esetben a példa rövid bemutatásával kezdődnek, majd a könyv végigvezeti az olvasót a felépítés menetén, a futtatáshoz szükséges alapvető beállítások használatán, majd pedig bemutatja a szimuláció eredményeit és értékeli a kapott adatokat.

Elsőként a szoftver használatához és a látottak megértéséhez szükséges alapvető szimulációval kapcsolatos ismeretek és a kormányablakok fő felépítésére és működésére vonatkozó információkat mutatom be, mely előismeretek szükségesek a szoftver működési elvének könnyű megértéséhez és a használatának hatékony elsajátításához.

10. A SZIMULÁCIÓRÓL ÁLTALÁNOSÁGBAN

Egy szimulációs szoftver alapvető működésének megértéséhez négy fontos fogalommal kell tisztában lennünk.

10.1. Szimuláció

Az első és legfontosabb fogalom, amellyel meg kell ismerkednünk az maga a szimuláció. A szimuláció egy valóságos folyamat vagy rendszer működésének utánzata az idő múlásával. Ez magában foglalja a rendszer mesterséges történetének megteremtését, és ennek a mesterséges történelemnek a megfigyelését, hogy következtetéseket vonhasson le az alkotó az ábrázolt valódi rendszer működési jellemzőire vonatkozóan (BANKS et al., 2013; BERNARD et al., 2000; BOHÁCS, 2012; GARRIDO, 2009; KELTON et al., 2015; MARTIJN., 2017; PRATEEK, 2015).

A szimuláció olyan vizsgálat, amikor egy rendszer, ill. folyamat fizikai vagy számítógépes modelljén tanulmányozzák a rendszer várható, illetve valódi viselkedését. Ennek megfelelően a szimulációk a valóság leegyszerűsítései, amelyek inkább a rendszer egészére és kevésbé annak részleteire koncentrálnak. A szimuláció során tehát egy valós jelenséget utánzunk a rendelkezésre álló alternatív technológiai megoldások segítségével, amely alatt a leggyakrabban egy erre a célra kifejlesztett szimulációs szoftvert értünk (BANKS et al., 2013; BERNARD et al., 2000; BOHÁCS, 2012; GARRIDO, 2009; KELTON et al., 2015; MARTIJN., 2017; PRATEEK, 2015).

A szimuláció célja, hogy a modell szintjén az utánzott jelenséggel azonos, vagy ahhoz nagyon hasonló körülményeket teremtsen a virtuális környezetben a felhasználók számára. Így lehetővé teszi, hogy az eredeti rendszer működését utánzó virtuális környezetet használjanak egy adott feladat megvalósításához, ami nagymértékben megkönnyíti például a különféle hatékonyságnövelési és optimalizálási törekvéseket (BANKS et al., 2013; BERNARD et al., 2000; BOHÁCS, 2012; GARRIDO, 2009; KELTON et al., 2015; MARTIJN., 2017; PRATEEK, 2015).

A szimuláció fogalmán belül megkülönböztetünk gép-ember és ember-ember szimulációt is. Az ember-ember szimuláció esetén személyek egy csoportja testesíti meg a szimulált valóságot. Ennek során valós társadalmi élethelyzeteket, szituációkat lehet szimulálni modellkörnyezetben, ami alkalmas például a tanításra és a lezajló folyamatok megértésére (BANKS et al., 2013; BERNARD et al., 2000; BOHÁCS, 2012; GARRIDO, 2009; KELTON et al., 2015; MARTIJN., 2017; PRATEEK, 2015). Ezeket másképpen játéknak is nevezik (KOVÁCS, KOVÁCS, 2015a, b). A gép-ember szimuláció esetében a szimulált valóságot a számítógép közvetíti a felhasználók számára és ezzel lépnek kapcsolatba. Például egy gyártósor vagy összetett szolgáltatási folyamat működését szolgáló szimulátorok alkalmasak ezeknek a

folyamatoknak a megértésére és átlátására, de a számítógépen futtatott szimulációba be lehet táplálni egy adott szituáció valós mért adatait is, amelyeket felhasználva létező problémák okainak feltárására és lehetséges megoldási alternatíváira kereshetünk válaszokat a szimuláció segítségével (BANKS et al., 2013; BERNARD et al., 2000; BOHÁCS, 2012; GARRIDO, 2009; KELTON et al., 2015; MARTIJN., 2017; PRATEEK, 2015).

10.2. Modell

A második fontos fogalom, amellyel meg kell ismerkednünk, az a modell. A modell egy valódi rendszer ábrázolása. Az egyik legfontosabb dolog, amit a modellalkotás során el kell döntenünk, hogy hol húzzuk meg a határt az összetettség tekintetében. Mi az a limit, ami már elég összetett ahhoz, hogy valóságghűen jellemezze a rendszer működését a szimuláció során, ám ugyanakkor még nem túlságosan összetett ahhoz, hogy átláthatatlanná tegye az egész rendszert. A modellnek elég összetettnek kell lennie ahhoz, hogy megválaszolja a vele szemben támasztott kérdéseket, de nem szabad, hogy túl bonyolult legyen, tehát csak az adott szituációban számunkra ténylegesen releváns jelentéssel bíró elemeket szabad kiemelnünk a valóságból (BANKS et al., 2013; BERNARD et al., 2000; BOHÁCS, 2012; GARRIDO, 2009; KELTON et al., 2015; MARTIJN, 2017; PRATEEK, 2015).

Egy adott rendszer alapvető matematikai modelljének megalkotásához három elméleti módszer létezik, a white-box eljárás, a black-box eljárás és a grey-box eljárás. (POKORÁDI, 2008).

A white-box eljárás esetén a modell felállításához rendelkezünk előzetes információval a rendszerről, így a modellalkotás fizikai megfontolásokra, törvényszerűségekre támaszkodva megy végbe. A white-box módszer nagy előnye, hogy a modell fizikai paramétereinek valós tartalma, jelentése van, hátránya viszont, hogy a modell felépítése általában rendkívül bonyolult, hiszen a létrehozáshoz szükséges pontos információk beszerzése időigényes és körülményes is lehet az adott esetben. Ezt a módszert a leggyakrabban a mérnöki gyakorlatban alkalmazzák (POKORÁDI, 2008).

A black-box eljárás ezzel szemben a másik végletet képviseli, ekkor a modell felállításához és létrehozásához csak kísérletekkel és mérésekkel lehet információkat szerezni. A white-box módszerrel szemben a black-box módszer egyik nagy előnye, hogy viszonylagosan egyszerű a modell megalkotása. Viszont hátránya is van, mivel ebben az esetben a paramétereknek, adott esetben, nincs valós fizikai jelentése. Azonban bizonyos esetekben éppen az egyszerűségükből fakadóan ezek a modellek is nagyon jól alkalmazhatók. A black-box modelleket szokás input/output modelleknek is nevezni, mivel általában az ilyen modelleket az adott esetben vizsgált rendszer bemeneti és kimeneti jellemzői alapján hozzuk létre. (POKORÁDI, 2008)

A harmadik modellalkotási módszer az úgynevezett grey-box eljárás. Ez a fajta megközelítés lényegében az előző két módszer, azaz a white-box eljárás és a black-box kombinációjából jött létre. A valóságban a szimulációs modellek megalkotása során a legtöbb esetben ezt a típusú eljárást alkalmazzák mind a műszaki, mind pedig a társadalomtudományok folyamatainak szimulációja esetén. Mivel általában egy adott műszaki probléma megoldása során rendelkezésünkre áll egy bizonyos mennyiségű adat és információ, így nem kell teljesen a sötétben tapogatóznunk, mindig vannak kapaszkodók, ugyanakkor vannak nem ismert fekete foltok is (POKORÁDI, 2008).

A megalkotás módszerétől függetlenül egy szimulációs modellt számos cél elérése érdekében megalkothatunk, legyen szó akár tervezés alatt álló rendszerek lehetséges működésének vizsgálatáról, létező rendszerek további optimalizálásáról, egy adott rendszerben fellépő hibák okainak feltárásáról, elhárításáról és jövőbeli kiküszöböléséről, hatékonyságvizsgálatról, diagnosztikáról vagy tervezett állapotfelmérésről (POKORÁDI, 2008).

10.3. Esemény

Az esemény alapvetően egy olyan momentum, amely megváltoztatja a rendszer adott pillanatban létező állapotát. A szimulációs szoftvereken belül megkülönböztetünk úgynevezett diszkrét esemény szimulációt és folytonos esemény szimulációt, ennek megfelelően pedig beszélhetünk úgynevezett folytonos paraméterű és diszkrét paraméterű szimulációs modellekről. A folytonos paraméterű modellekben a változók egy adott értékhatáron vagy tartományon belül bármilyen értéket felvehetnek. A diszkrét paraméterű modellek esetében viszont a változók csak diszkrét értékeket vehetnek fel (POKORÁDI, 2008; BANKS et al., 2013; BERNARD et al., 2000; BOHÁCS, 2012; GARRIDO, 2009; KELTON et al., 2015; MARTIJN, 2017; PRATEEK, 2015).

Amikor bizonyos események zajlanak le, a modellek egyes összetevői megváltoztatják állapotukat, és így szabályozzák a szimulációt. A műben tárgyalt szimulációs szoftver ezeket az eseményeket diszkrétén, lépésről lépésre figyelembe veszi.

A diszkrét események szimulációs modelljét úgy definiáljuk, hogy az állapotváltozók csak azokban a diszkrét pontokban változnak meg, amelyekben az események előfordulnak. A diszkrét esemény szimuláció azoknak a rendszereknek a modellezése, amelyekben az állapotváltozó csak időbeli pontok diszkrét sorában változik (BANKS et al., 2013; BERNARD et al., 2000; BOHÁCS, 2012; GARRIDO, 2009; KELTON et al., 2015; MARTIJN, 2017; PRATEEK, 2015).

Ennek a megközelítésnek az a fő előnye, hogy a szoftver kihagyja az események közötti időt, így előre meghatározott sűrűségű időpillanatokban követhetjük nyomon az adott rendszer elemeinek változását. A teljesen objektumorientált környezet és programozás pedig lehetővé teszi számunkra, hogy a nagy, összetett rendszereket szervezzük és átlátható módon modellezzük (BANKS et al., 2013; BERNARD et al., 2000; BOHÁCS, 2012; GARRIDO, 2009; KELTON et al., 2015; MARTIJN, 2017; PRATEEK, 2015).

Az általunk tárgyalt szimulációs szoftver alapvető objektumokat tartalmaz, amelyek olyan jellemzőkkel rendelkeznek, amelyek lehetővé teszik számunkra, hogy közvetlenül használjuk őket szimulációs modellben. Így a szoftver segítségével diszkrét eseményszimulációs és statisztikai elemzési lehetőségeket alkalmazhatunk a kormányablakok esetében például azok folyamatainak szimulálása, valamint ezen folyamatok megértése, hatékonyságnövelése és a munkaerőigény optimalizálása érdekében. Ez lehetővé teszi számunkra, hogy gyorsan ellenőrizzük a szűk keresztmetszeteket, átfutási időket és időről időre megnézzük az erőforrás-felhasználást és az egyes elemek kihasználtságának mértékét.

A diszkrét események szimulációja nyomon követi a modellelemek állapotváltozását az adott változások idején. Ellentétben a folyamatos szimulációval, ahol az óra folyamatosan fut, a diszkrét események szimulációjának órája egy eseményről azonnal a következő ütemezett eseményre ugrik. Ezek az események ütemezhetnek más eseményeket, például a gépbe való belépést, amely egy adott eseményre ütemezi a gép elhagyását (BANKS et al.,

2013; BERNARD et al., 2000; BOHÁCS, 2012; GARRIDO, 2009; KELTON et al., 2015; MARTIJN, 2017; PRATEEK, 2015).

A diszkrét események szimulációja csak a modellelemek állapotváltozását mutatja az idő bizonyos pontjaiban, nem folyamatosan az idő múlásával. Amikor bizonyos események zajlanak le, egyes modellek összetevői megváltoztatják állapotukat, és így szabályozzák a szimulációt. A szimulációs szoftver ezeket az eseményeket diszkrétén, lépésről lépésre figyelembe veszi (BANKS et al., 2013; BERNARD et al., 2000; BOHÁCS, 2012; GARRIDO, 2009; KELTON et al., 2015; MARTIJN, 2017; PRATEEK, 2015).

10.4. A szimuláció általános céljai

Egy szimulációs modell megalkotása számos célt szolgálhat, amelyek közül a leggyakoribbak közé sorolhatók a következők (BANKS et al., 2013; BERNARD et al., 2000; BOHÁCS, 2012; GARRIDO, 2009; KELTON et al., 2015; MARTIJN, 2017; PRATEEK, 2015):

- Bizonyos szempontok szerinti növekedés elérése
 - Teljesítménynövelés
 - Erőforrás-felhasználás növelése
 - A létesítmény kihasználtságának növelése
- Bizonyos típusú kulcsfontosságú értékek meghatározása
 - Puffer optimális méretének meghatározása
 - Szállítóeszközök és szállítójárművek optimális számának meghatározása
 - Alapanyag-tárolók (munkadarabok) optimális számának meghatározása
 - A megfelelő termelési menetrendek és ütemtervek meghatározása és kialakítása
- Bizonyos típusú jellemzők értékeinek csökkentése, optimalizálása
 - Átadási idők csökkentése, optimalizálása
 - Szükséges erőforrások mértékének csökkentése, optimalizálása
 - Megfelelő tárolási követelmények meghatározása, kialakítása
- Egyéb fontos tulajdonságok meghatározása, értékelése
 - A tervezés folyamatának validálása
 - szűk keresztmetszetek felismerése
 - A WIP (work in process – termelésközi készletek) mértékének csökkentése
 - Tőkebefektetések vagy folyamatokban bekövetkező változások értékelése
 - Szabályozási stratégiák optimalizálása
 - Tervezési hibák elkerülése
 - A folyamatban lévő vagy jövőbeli beruházások védelme

11. A KORMÁNYABLAKOKRÓL RÖVIDEN

A kormányablakok létrehozása az úgynevezett „One Stop Shop” modellt vagyis az egyablakos ügyintézését vezette be a magyar közigazgatásba, mely a skandináv országokban már elterjedt ügyintézési megoldásként létezik. (NAVRACSICS, 2013; KOVÁCS, 2016; VIRÁG, 2012; VIRÁG, 2014) A rendszer lényege, hogy az ügyfelek egy ügyintézési ponton egyszerre akár azonnal több ügyüket is elintézhetik, összetettebb eljárás esetén az eljárás megindítását kezdeményezhetik, vagy tájékoztatást kaphatnak. (www.jaras.info.hu, www.kormanyablak.hu, www.jarasikormanyablak.hu) Az egyablakos ügyintézés alkalmas eszköz arra, hogy mind a vállalkozások, a lakosság és a közszféra oldalán felmerülő adminisztratív terheket csökkenteni lehessen. (TEMESI, LINDER, 2015; www.kormany.hu; magyaryprogram.kormany.hu)

11.1. A kormányablakok kialakítása

2014-ben az okmányirodák bázisán kezdték el kialakítani a kormányablak rendszert. A területi közigazgatási reform távlati célja az volt, hogy biztosítsa az egyablakos ügyintézését az ügyfelek számára, hogy az állampolgárok egyetlen helyen, a kormányablakoknál tudják hivatalos ügyeiket intézni. A járási államigazgatási rendszer létrehozásának keretében ezáltal kialakult az ügyfélszolgálati irodák országos hálózata, amely az állampolgárok számára tartalomban és színvonalban is magas szintű szolgáltatást biztosít (www.jaras.info.hu, www.kormanyablak.hu, www.jarasikormanyablak.hu).

A kormányablakok alapvető jellemzői (www.jaras.info.hu, www.kormanyablak.hu, www.jarasikormanyablak.hu):

- Egységesen kialakított külső és belső megjelenés, ami megkönnyíti az állampolgárok számára az eligazodást, függetlenül attól, hogy az ország területén melyik kormányablakba térnek be intézni az ügyeiket.
- Az egységesen jellemző professzionális szakmai felkészültség, ami az ügyintézők hozzáállását és jártasságát illeti.
- Az egységes nyitvatartási időkeret.
- A fejlett számítástechnikai háttér és támogatás, ami lehetővé teszi a hatékony ügyintézését a különböző rendszerekhez való hozzáféréseket igénylő ügyek esetében is.

A jól szervezett közigazgatás jogos igényként jelenik meg az állammal kapcsolatba kerülő ügyfelek részéről, ennek kielégítése érdekében pedig nem csak integrációt valósítottak meg az átalakítások során, hanem a könnyű elérhetőség követelményét is teljesítették, hiszen

mindenki a lakóhelyéhez legközelebb, vagy bármely, a számára legkönnyebben elérhető kormányablakban, munkaidőn kívül is el tudja intézni a hivatalos ügyeit (www.jaras.info.hu, www.kormanyablak.hu, www.jarasikormanyablak.hu).

A kormányablakok a területi közigazgatás gerincét alkotó megyei kormányhivatalok, illetve a kormányhivatalok alá tartozó járási hivatalok ügyfélszolgálati irodái, amelyekben jelenleg az állampolgárok több, mint 1500 féle különböző ügykör esetén intézhetik a felmerülő ügyeiket, problémákat, kezdeményezhetnek folyamatokat, illetve kaphatnak tájékoztatást az adott eljárások pontos menetéről. A kormányablakok számát 2013-tól növelték fokozatosan, 2014-ben pedig nem csak a megyeszékhelyeken és a megyei jogú városokban, hanem a járási székhelyeken is kialakították az integrált ügyfélszolgálati irodákat. A folyamat végére mintegy 360 kormányablak jött létre az országban (www.jaras.info.hu, www.kormanyablak.hu, www.jarasikormanyablak.hu).

A járási hivatalok alá rendelt és általuk működtetett kormányablakok a vonatkozó jogszabályoknak megfelelően az egész ország területére kiterjedő illetékességgel rendelkeznek, képesek ellátni a jogszabályokban hatáskörükbe utalt feladatokat, ebből fakadóan az állampolgárok az ország területén belül bármelyik járás kormányablakába betérhetnek, hogy a felmerülő ügyeiket elintézzék (www.jaras.info.hu, www.kormanyablak.hu, www.jarasikormanyablak.hu). Az egyes kormányablakokban intézhető ügyek a 515/2013. (XII. 30.) kormányrendelet szerint több kategóriába sorolhatók (TEMESI, LINDER, 2015; KOVÁCS, 2016; VIRÁG, 2012; VIRÁG, 2014).

A kormányablakokról szóló 515/2013. (XII. 30.) kormányrendelet mellékletei tartalmazzák azoknak a főbb ügyköröknek a felsorolását, amelyek a kormányablakokban elérhetőek az állampolgárok számára (www.jaras.info.hu, www.kormanyablak.hu, www.jarasikormanyablak.hu):

- Azonnal intézhető ügyek (pl. új személyazonosítóról és lakcímről szóló hatósági igazolvány kiállítása).
- Saját hatáskörében intézhető ügyek (pl. vezetői engedély kiállítása, magánútlevél kiadása, cseréje iránti kérelem).
- Továbbítás céljából előterjeszhető beadványok (pl. az anyakönyvi kivonat kiállítása iránti kérelem).
- Tájékoztatásnyújtás (pl. házassági szándék, bejegyzett élettársi kapcsolat létesítése).
- Kiegészítő szolgáltatások (pl. ügyfélkapu regisztráció).
- Hivatalból intézhető ügyek (pl. egyéni vállalkozók hatósági ellenőrzése).

A kormányablakokban kizárólag speciálisan képezett ügyintézők dolgozhatnak, akik széles körű ismeretekkel rendelkeznek. Ezen ismeretek közé tartozik a magas fokú szakmai felkészültség, a segítőkészség, a problémakezelő képesség és az udvarias viselkedés (www.jaras.info.hu, www.kormanyablak.hu, www.jarasikormanyablak.hu).

Amint az látható és érzékelhető is volt az elmúlt évek során, a magyar államigazgatási rendszer jelentős átalakításokon ment keresztül, melyek mind a hatékonyságnövelést tűzték ki célul. A kialakított új rendszer a szerkezeti átalakítások után már jóval átláthatóbbá és központosítottabbá vált, mint amilyen az elődje volt (www.kormany.hu, magyaryprogram.kormany.hu). A korábban már kialakított kormányhivatalok megyei szinten szabályozták a közigazgatási rendszereket, a járási hivatalokat pedig a kormányhivatalok területileg illetékes

alegységeiként definiálták a törvényalkotók. A járási hivatal szervezetrendszere magában foglalja a korábban eltérő irányítások alatt, de azonos illetékességi területen működő közigazgatási rendszereket, ezáltal egységes és átláthatóbb szervezeti felépítés jött létre, ahol már nincsenek jelen a korábban jellemző átfedések és hatásköri különbségek (www.jaras.info.hu, www.kormanyablak.hu, www.jarasikormanyablak.hu).

Az állampolgárok szempontjából a közigazgatás átalakításának igazán érzékelhető hatása viszont a kormányablakok rendszerének kialakítása és ezzel az egyablakos ügyintézés tényleges megvalósítása volt (KOVÁCS, 2016; VIRÁG, 2012; VIRÁG, 2014). Ezáltal a kormányablakok a magyar közigazgatás kiemelten fontos intézményeivé váltak, hiszen az állampolgárok a kormányablakokon keresztül kerülnek közvetlen kapcsolatba az államapparátus gépezetével, és azon keresztül érzékelik annak hatékonyságát (www.kormany.hu, magaryprogram.kormany.hu). Ebből következően pedig rendkívüli fontossággal bír egy olyan modern, letisztult és ügyfélorientált szervezeti struktúra létrehozása, mint amilyenek a kormányablakok (www.jaras.info.hu, www.kormanyablak.hu, www.jarasikormanyablak.hu).

A Magyar Zoltán Közigazgatás-fejlesztési Programban testet öltő egyértelmű kormányzati szándék az volt, hogy létrejöjjön egy olyan letisztult, egységes arculattal, azonos felkészültségű és minőségű személyi állománnyal, ugyanolyan ügyfélfogadási rendben történő szolgáltatásnyújtásnak a platformja, amely ugyanazon standardok szerint működik az ország teljes területén, hatékony, ügyfélorientált beállítottságú, és képes megfelelni azoknak a modern elvárásoknak, amiket az állampolgárok a 21. században támasztanak az állami ügyintézés kialakításával szemben (www.jaras.info.hu, www.kormanyablak.hu, www.jarasikormanyablak.hu, www.kormany.hu, magaryprogram.kormany.hu).

A kormányablak-hálózat kialakításával ezek a törekvések megvalósultak, létrejött egy olyan egységes kialakítású intézményrendszer, amely az ügyfelek számára könnyen elérhető, megközelíthető, és lehetővé teszi az egy helyen történő ügyintézését. Ezáltal az ügyintézésre fordított költségek (idő, energia, pénz) csökkennek, hiszen a one stop shopok létrehozása révén megszűnt a több hivatalnál való sorbaállási kényszer és a közigazgatási struktúra átalakítása és központosítása révén rövidültek az egyes eljárásokhoz kapcsolódó ügyintézési idők is (www.jaras.info.hu, www.kormanyablak.hu, www.jarasikormanyablak.hu).

Az egyes szakigazgatási szervekhez és főosztályokhoz tartozó kormányablakokban elintézhető ügyek száma még nem végleges, folyamatos bővítés alatt áll, melynek végcélja, hogy a kormányablakokban elintézhető ügyek köre a szakigazgatási területek teljes horizontját átfogja végül. Így tehát a reform teljes végbemenetelét követően megközelítőleg 2500 különböző hatósági ügytípusban tudnak majd az ügyfelek segítséget kapni egy helyen, a kormányablakokban (www.jaras.info.hu, www.kormanyablak.hu, www.jarasikormanyablak.hu).

11.2. A kormányablakokban intézhető leggyakoribb ügyek típusai

A kormányablakok által nyújtott szolgáltatások az egyes ügykörök típusától függően öt főbb kategóriába sorolhatók be.

Tájékoztatás igényelhető egy adott ügy várható ügymenetéről, valamint az ügyfél jogairól és kötelezettségeiről

A kormányablakok szolgáltatásainak első szintje a tájékoztatás, információnyújtás, amikor az ügyintéző a megfelelő hatóság irányába irányítja az ügyfelet, tájékoztatja arról, hogy mely hatósághoz fordulhat az ügyével, amennyiben valóban közigazgatási problémáról van szó, és milyen típusú eljárásban. Itt a kormányablak ügyintézője kizárólag tájékoztatást nyújt az állampolgárok számára (www.jaras.info.hu, www.kormanyablak.hu, www.jarasikormanyablak.hu).

Kérelmek terjeszthetők elő és bejelentések tehetők meg

A kormányablakok szolgáltatásainak második szintje, amikor a tájékoztatáson túl az ügyfél a helyszínen előterjeszti a kérelmét, azaz a kormányablaknál indítja meg az eljárást. Ezeknél az ügyeknél a kormányablak ügyintézője a csatolt dokumentumokkal együtt haladéktalanul, de legkésőbb a benyújtást követő munkanapon továbbítja a kérelmet a döntési jogkörrel rendelkező főosztályhoz a további ügyintézés céljából (www.jaras.info.hu, www.kormanyablak.hu, www.jarasikormanyablak.hu).

Azonnal intézhető és saját hatáskörben intézhető ügyek kapcsán az ügyintézés végrehajtása

Ezen kategóriába tartoznak azok az ügykörök, amelyekben az ügyintéző hivatalból megindíthatja az ügyintézés folyamatát. Ide tartoznak a leggyakrabban előforduló közigazgatási ügyek, hatósági ügyek és problémák, amelyek az állampolgárok részéről felmerülhetnek. A szolgáltatás ezen szintjén az ügyintézők teljes körű ügyintézacskét végeznek. Ezekben az ügyekben a kormányablak az eljárási hatáskörrel és illetékességgel rendelkező szakhatóság helyett jár el. Az ilyen típusú ügyintézacskét vagy azonnal teljesíteni tudja az ügyintéző, vagy a jogszabályban az eljárásra meghatározott ügyintézacskési határidőn belül saját hatáskörben, helyben teszi meg a szükséges eljárási cselekményeket, hozza meg az érdemi döntést (www.jaras.info.hu, www.kormanyablak.hu, www.jarasikormanyablak.hu).

A kormányablakban azonnal intézhető leggyakoribbak ügyek közül néhány példa:

- új személyi azonosítóról és laccímről szóló hatósági igazolvány kiállítása iránti kérelem,
- személyazonosító igazolvány elvesztésének, megtalálásának bejelentése,
- laccímbejelentés, tartózkodási hely bejelentés,
- járművekkel kapcsolatos ügyintézés (pl. forgalmival, törzskönyvvel kapcsolatos ügyintéзések, eladó bejelentése a jármű tulajdonjogának változásáról, átírás, üzembentartó bejelentés, parkolási igazolvány stb.),
- hiteles tulajdoni lap, térképmásolat, földhasználati lap,
- családtámogatással kapcsolatos feladatok (www.jaras.info.hu, www.kormanyablak.hu, www.jarasikormanyablak.hu).

Szakmai, informatikai támogatás az egyedi ügyek intézéséhez.

A kormányablakok ügyintézői a megfelelő technikai feltételek megléte esetén hatósági szolgáltatásként egyéb ügynevezett kiegészítő szolgáltatásokat is biztosítanak az állampolgárok számára. Ezek közül a szolgáltatások közül az egyik leggyakrabban igénybe vett szolgáltatástípus például az ügyfélkapu-regisztráció és a cégkapu-regisztráció elvégzése (www.jaras.info.hu, www.kormanyablak.hu, www.jarasikormanyablak.hu).

Iratbetekintési és iratmásolati szolgáltatás a szakhatósági eljárásokkal összefüggésben

Ebben az esetben a kormányablakok ügyintézői az egyedi hatósági ügyek intézéséhez szükséges iratanyagokkal összefüggésben lehetőséget biztosítanak az állampolgároknak a vonatkozó anyagok beszerzésére. Amennyiben ők maguk nem rendelkeznek ezen lehetőségekkel, informatikai segítséget is nyújthatnak az adott esettől függően (www.jaras.info.hu, www.kormanyablak.hu, www.jarasikormanyablak.hu).

11.3. A kormányablakokban leggyakrabban előforduló ügyek kategóriái

Ebben a fejezetben az Olvasó megtalálhatja a kormányablakokban az ügyintézők által a leggyakrabban igénybe vett szolgáltatások listáját kategóriákra bontva (Kovács, 2014).

- **A születés kapcsán leggyakrabban előforduló igénybe vett szolgáltatások**
 - Anyakönyvi kivonat kiállítása
 - Családtámogatások, ellátások igénylése gyermek(ek) után
 - Teljes hatályú apai elismerő nyilatkozat felvétele iránti kérelem
- **Az iskolakezdés kapcsán leggyakrabban előforduló igénybe vett szolgáltatások**
 - Diákigazolvány igénylése
 - Iskoláztatási támogatásra való jogosultság megállapítása (családi pótlék)
 - Sajátos nevelési igényű gyermek iskoláztatása

- **A betegségek és a balesetek kapcsán leggyakrabban előforduló igénybe vett szolgáltatások**
 - Fogyatékosági támogatásra való jogosultság
 - Társadalombiztosítási Azonosító Jelet igazoló hatósági igazolvány kiállítása iránti kérelem
 - Alanyi és normatív közgyógyellátás megállapítása
 - Utazási költségtérítés iránti kérelem egészségbiztosítási ellátás igénybevétele kapcsán
- **A munkahelykeresés kapcsán leggyakrabban előforduló igénybe vett szolgáltatások**
 - Aktív korú álláskeresők támogatása
 - Nyugdíjkorhatár eléréséhez közelálló álláskeresők támogatása
 - Foglalkoztatást helyettesítő támogatás
- **A külföldi munkavállalás kapcsán leggyakrabban előforduló igénybe vett szolgáltatások**
 - Magyarország területét külföldi letelepedés szándékával történő elhagyás bejelentése
 - Külföldi munkavállaló bejelentése
- **Az ipari és a kereskedelmi tevékenység kapcsán leggyakrabban előforduló igénybe vett szolgáltatások**
 - Rendezvénytartási engedély kiadása
 - Szálláshely üzemeltetésének feltételei és engedélyeztetése
 - Üzlet működési engedélyének visszavonása
 - Panasz és közérdekű bejelentés
- **A vállalkozóvá válás kapcsán leggyakrabban előforduló igénybe vett szolgáltatások**
 - Egyéni vállalkozói tevékenység megkezdésének bejelentése és nyilvántartásba vétele
 - Egyéni vállalkozói tevékenység szüneteltetése
- **Az építkezés és az otthonteremtés kapcsán leggyakrabban előforduló igénybe vett szolgáltatások**
 - Tulajdoni lap szolgáltatási kérelem és ingatlannyilvántartási térképi adatbázisból kiadott hiteles térképmásolat iránti kérelem
 - Lakcímváltozás bejelentése
 - Építési engedély, használatba vételi engedély kiadása iránti kérelem
 - Lakásfenntartási támogatás, adósságkezelési szolgáltatás
- **A szociális ellátás kapcsán leggyakrabban előforduló igénybe vett szolgáltatások**
 - Időskorúak járadéka
 - Megváltozott munkaképességű személyek
 - Ápolási díj
- **A házasságkötés kapcsán leggyakrabban előforduló igénybe vett szolgáltatások**
 - Házassági szándék bejelentése és házasság hivatali helyiségen kívüli megkötésének engedélyezése iránti kérelem

- Bejegyzett élettársi kapcsolat létesítése és az erre vonatkozó szándék bejelentése
- **A névváltoztatás kapcsán leggyakrabban előforduló igénybe vett szolgáltatások**
 - Házassági névviselési forma módosítása
 - Személyazonosító igazolvány cseréje névváltozás miatt
- **A haláleset és az öröklés kapcsán leggyakrabban előforduló igénybe vett szolgáltatások**
 - Temetkezéssel összefüggő szociális ügyek
 - Hagyatéki eljárás megindításának kezdeményezése, hagyatéki ügyintézés
- **A gépjármű üzemben tartása kapcsán leggyakrabban előforduló igénybe vett szolgáltatások**
 - Forgalmi engedély
 - Vezetői engedély első alkalommal történő kiállítása
 - Gépjármű végleges forgalomból történő kivonása
 - Parkolási igazolvány
- **A személyi nyilvántartás és okmányok igénylése, elvesztése, pótlása kapcsán leggyakrabban előforduló igénybe vett szolgáltatások**
 - Személyazonosító igazolvány
 - Új lakcímgazolvány kiadása
 - A magánútlevelel pótlása
- **A bűncselekmények kapcsán leggyakrabban előforduló igénybe vett szolgáltatások**
 - Lopás bejelentése, azonnali pénzügyi segély iránti kérelem
 - Okmányok letiltása és pótlása
 - Állandó személyazonosító igazolvány eltulajdonításának bejelentése és pótlása lopás miatt
 - TAJ-kártya pótlása lopás miatt
- **A közúti közlekedési szolgáltatás végzése kapcsán leggyakrabban előforduló igénybe vett szolgáltatások**
 - Személy-taxi szolgáltatás végzésének engedélyeztetése
- **Az utazások kapcsán leggyakrabban előforduló igénybe vett szolgáltatások**
 - A magánútlevelel igénylése
 - Utazásszervezői, utazásközvetítői tevékenység engedélyezése
- **A letelepedés és az állampolgárság igénylése kapcsán leggyakrabban előforduló igénybe vett szolgáltatások**
 - Egyszerűsített honosítási kérelem
 - Letelepedett, menekült és oltalmazott jogállású személy nyilvántartásba vétele, valamint okmányok kiállítása az idegenrendészeti és menekültügyi hatóságnál indult kérelem alapján
 - Külföldről hazatérő magyar állampolgár bejelentése a személyiadat- és lakcímnnyilvántartásba

- **A nyugdíjba vonulás kapcsán leggyakrabban előforduló igénybe vett szolgáltatások**
 - Nyugdíjbiztosítási igazgatóságnál nyilvántartott adatokba történő betekintés
 - Az öregségi nyugdíj méltányosságból történő felemelése
 - Nyugdíjfolyósításhoz kapcsolódó bejelentés

12. KORMÁNYABLAK SZIMULÁCIÓS ESETPÉLDÁK BEMUTATÁSA

Amint azzal már az előző fejezetekben megismerkedhettünk, a szimulációs szoftverek alkalmazása kiváló eszköz arra, hogy egy adott folyamatot (legyen az akár egyszerű vagy összetett) virtuális környezetben modellezzünk és tanulmányozzunk. A legtöbb esetben a valóságban rendkívül körülményes lenne a vizsgálat tényleges alanyát képző rendszert hosszasan tanulmányozni a különféle inputok és outputok hatásait vizsgálva. Ez nem csak a vizsgálatot nehezítené meg, de a rendszer működését is zavarnánk vele. Noha a modell-építés során a rendszeridentifikációhoz szükség van adatgyűjtésre, és így a zavar okozása nehezen elkerülhető, azonban ezt követően a virtuális környezetben létrehozott és a valóság leképzésén alapuló modellt már kötöttségek nélkül, sokkal könnyebben vizsgálhatjuk, tesztelhetjük, anélkül, hogy a valódi rendszer működésében további zavart keltenénk.

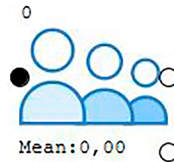
A kormányablakok működésének szimulálása során öt objektummal kell megismerkednünk, amelyekről a kismonográfia használati útmutatójában részletesebb információk találhatóak.

Az első elem, amely nélkülözhetetlen a szimuláció megépítéséhez, az a forrás szerepét betöltő objektum. Ennek az a feladata, hogy generálja azokat a mozgó egységeket, vagyis esetünkben a kormányablakba betérő állampolgárokat és a hozzájuk kapcsolt, igénybe vett szolgáltatásokat, amelyeknek a viselkedését és tulajdonságait tanulmányozni kívánjuk a szimulációs folyamat során. A kormányablakok szimulációjában a forrás szerepét az úgynevezett „ticket machine” (1. ábra), azaz a jegykiadó automata tölti be. Egy kormányablakba belépve az állampolgárok első útja a jegykiadó automatához vezet, ahol kiválaszthatják, hogy milyen szolgáltatás kapcsán kívánják igénybe venni a kormányablakban dolgozó ügyintézők segítségét. A valóságban a legtöbb esetben egyetlen jegykiadó automata üzemel az adott kormányablakokban, azonban a szimulációs felület ebben eltér és egy jegykiadó automata csak egyetlen szolgáltatást igénybe vevő ügyféltípust tud generálni. Így tehát ahány típusú szolgáltatást szeretnénk a szimulációban tesztelni, annyi jegykiadó automatát kell elhelyeznünk a felületen. Ezt azon megfontolásokból alakítottuk ki, hogy átláthatóbbá váljanak a különféle szolgáltatások és a hozzájuk kapcsolódó olyan egyedi jellemzők, mint például a kiválasztott eloszlás típusa. A jegy generálása, vagyis a megfelelő szolgáltatás kiválasztása után az állampolgárok a következő elemhez kerülnek továbbításra. A ticket machine elem működését még részletesebben írja le a szoftver felhasználói útmutatója.



1. sz. ábra: ticket machine (jegykiadó automata)

A második objektum, amelynek a működési jellemzőivel és a használatának módjával tisztában kell lennünk, az a puffer szerepét betöltő objektum. A szimulációk során a pufferek azok az elemek, amelyek átmeneti tárolóegységként funkcionálnak a folyamatok végbemenelete alatt. Bennük kerülnek tárolásra az elemek mindaddig, amíg a szimuláció következő állomására tovább nem tudnak lépni. A kormányablakok működésének szimulációjában a puffer szerepét a „waiting room” (2. ábra) elem, vagyis a váróterem tölti be. A váróterem kapacitása minden esetben állítható; lehet végtelen, vagy akár korlátozhatjuk is a méretét attól függően, hogy milyen paramétereknek megfelelő szimulációs modellt szeretnénk felépíteni. Az állampolgárok várakozása kétféle módon érhet véget. Vagy tovább lépnek a következő objektumhoz, vagy pedig lemorzsolódnak, vagyis távoznak a váróteremből anélkül, hogy az általuk kiválasztott szolgáltatás igénybevétele ténylegesen megtörtént volna. Ez az eset akkor áll fent, ha például a váróteremben tartózkodó személy vagy személyek nem tudják kivárni a sor végét és inkább távoznak, hogy aztán később visszatérve újra sorszámot húzzanak. A waiting room elem működéséről szintén találhatunk részletesebb bemutatást a szoftver felhasználói útmutatójában.



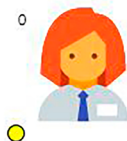
2. sz. ábra: waiting room (váróterem)

A harmadik objektum, amely szintén kiemelt jelentőséggel bír a kormányablakok működésének szimulálásában, az a munkaállomás, amely a számára definiált feladatokat végre tudja hajtani az előre megadott paramétereknek megfelelően. A kormányablakok szimulációja során a munkaállomás szerepét betöltő objektum maga a „window” (3. ábra), vagyis az ügyfélszolgálati ablak lesz. A váróteremből az ablakhoz érkező ügyfeleket itt szolgálják ki az előre megadott feltételeknek megfelelően. A window elem működéséről szintén találunk részletesebb bemutatást a szoftver felhasználói útmutatójában.



3. sz. ábra: window (kormányablak)

A negyedik és talán mindegyik közül a legfontosabb objektum az úgynevezett munkás, ami a kormányablakok szimulációjában nem más, mint az „administrator” (4. ábra) objektum. Ezen elem szerepe a szimulációban kiemelt jelentőségű. Az adminisztrátorok, vagyis az ügyintézők esetén egyedileg be lehet állítani, hogy az előre definiált szolgáltatástípusok közül melyek végrehajtására van jogosítványuk, vagyis, hogy milyen típusú ügyeket képesek elintézni. Ezen felül az is beállítható, hogy az adott szolgáltatástípushoz ügyintézőnként egyedileg milyen elosztást rendelünk hozzá, így befolyásolva az elvégzési időket, és téve a szimulációt még inkább valóságosabbá. Ezen felül a további finomítások érdekében definiálhatjuk az adott ablaknál az ügyintéző munkaidejének hosszát is. Az administrator elem működéséről szintén olvashatunk részletesebben a szoftver felhasználói útmutatójában.



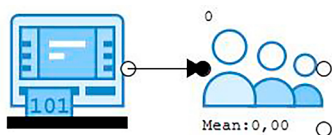
4. sz. ábra: administrator (ügyintéző)

A kormányablakok működésének szimulációjában az utolsó objektum az úgynevezett nyelő, aminek a szerepét a szoftverben az „exit” elem (5. ábra) tölti be. Ez az elem a szimuláció végpontja, vagyis a kijárat, ahol az ügyfelek elhagyják a kormányablakot, miután az általuk igénybe vett szolgáltatással kapcsolatos munka befejeződött. Kijáratból elhelyezhető egy vagy akár több is a szimulációban, attól függően, hogy milyen paraméterek mentén és mit szeretnénk részletesebben vizsgálni a folyamatok megfigyelése során. Az exit elem működését még részletesebben taglalja a szoftver felhasználói útmutatója.



5. sz. ábra: Exit (Kijárat)

Végül, de nem utolsósorban ahhoz, hogy az előbbieken felsorolt különálló elemeket egyetlen egységként tudjuk folyamatszimulációra használni, a felületen történő elhelyezésük után össze is kell kötnünk egymással. Az összeköttetést jelző úgynevezett „connector” elem (6. ábra) nem tekinthető önálló objektumnak. Létrehozásához nem kell mást tennünk, mint az egyik elem végpontjára kattintva áthúzni a kurzort a másik elem bemeneti pontjára, ezt követően pedig az összeköttetés már ki is alakult a két elem között. Az összeköttetés megszüntetésének több módja is van; az újbóli összekötés a két pont között, vagy az összeköttetést szimbolizáló és vizuálisan megjelenítő nyíl elem kijelölése és törlése egyaránt hatékony módja a kapcsolat megszüntetésének.



6. sz. ábra: connector (összekötő) elem két objektum között

A kismonográfia következő szakaszában olyan egyszerű és összetett példákkal ismerkedhet meg az Olvasó, amelyek lépésenként elmagyarázzák a szoftver használatát és bemutatják a program használatának különféle aspektusait a kormányablakokra jellemző környezetben.

12.1. Első példa: egyszerű szimulációs modell felépítésének folyamata

Egyetlen bemenettel és egyetlen kimenettel rendelkező folyamat modellezése, amelyben egy ügyintéző egy szolgáltatást nyújt.

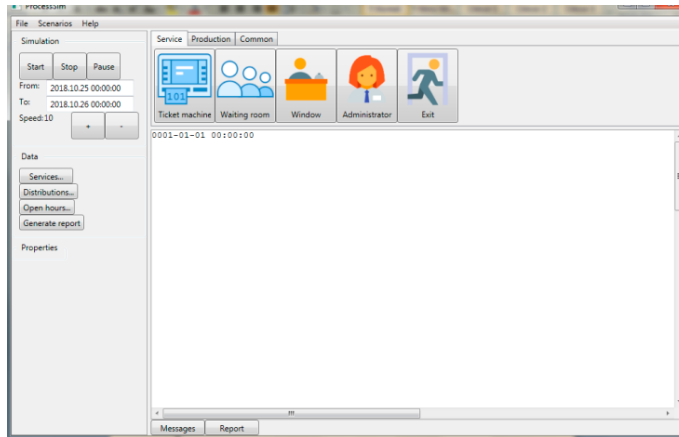
12.1.1. Bemutató

Az első példa egy egyszerű modell felépítését fogja bemutatni a szoftverben, amely egyetlen jegykiadó automatából, egyetlen váróteremből, egyetlen ügyfélszolgálati ablakból, egyetlen ügyintézőből és egyetlen kijáratból fog állni. Az ügyintéző egyetlen szolgáltatás elvégzéséhez szükséges jogosítvánnyal fog rendelkezni.

Ennek a bemutató példának az a célja, hogy a felhasználókat megismertesse az alapvető modellépítés és működtetés menetével és elengedhetetlenül szükséges feltételeivel. Ezen alapulva később már komplexebb modellek felépítése is lehetségessé fog válni az ezt követő példák keretében.

12.1.2. Felépítés menete

A szoftver elindítása után megjelenik előttünk az alap kezelőfelület (7. ábra), amely egy központi üres részből, a felette található eszköztárból, valamint a baloldalon található kezelőfelületből és az alul található üzenetpanelből áll. A szoftver egyaránt használható gyártási folyamatok és szolgáltatási folyamatok szimulációjára, alapértelmezés szerint a szolgáltatások szimulálásához szükséges service oldalt jeleníti meg elsőként a bekapcsolás után.



7. sz. ábra: a szimulációs szoftver kezelőfelülete

Az egyes objektumokat, amelyekre az adott szimulációs folyamat létrehozásához van szükségünk a szolgáltatások modellezéséhez használatos service fülön található eszköz-készletből (8. ábra) tudjuk kiválasztani és a szimulációs felületre helyezni.

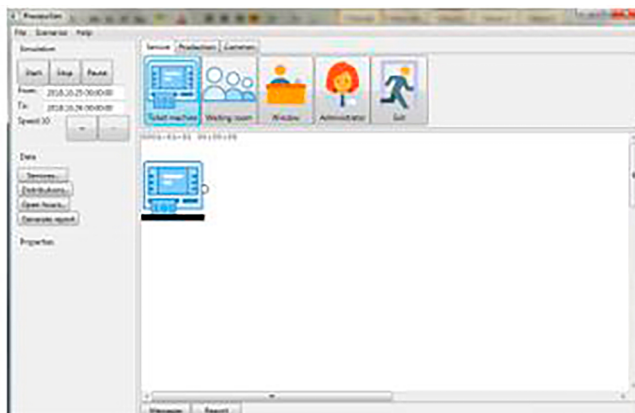


8. sz. ábra: szolgáltatáshoz kapcsolódó objektumok eszköztára

Ahhoz, hogy egy objektumot a szimulációs felületre felvigyünk, nem kell mást tennünk, mint rákattintani az eszköztárban az általunk felvinni kívánt objektum gombjára. Ezt követően az adott objektum megjelenik a kezelőfelületen. Amennyiben az objektumot a megjelenési helytől eltérő helyre szeretnénk áthelyezni, ehhez elég megfognunk a megjelenített objektumot és arrébb húzunk azt. A mozgatáshoz az objektumot ki kell jelölnünk, majd a bal egérgombot nyomva tartva tudjuk azt a felület egy másik pontjára átmozgatni.

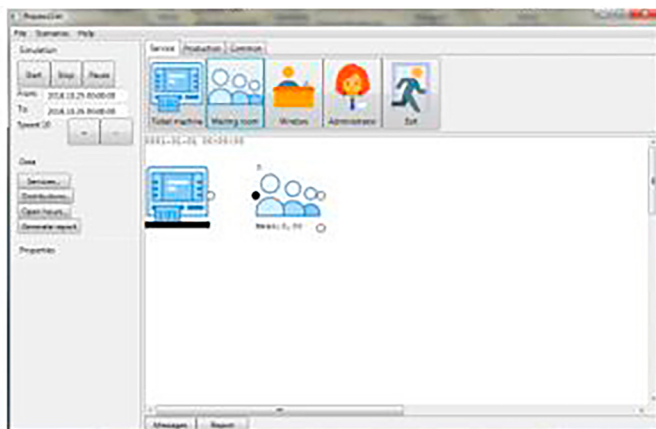
Az egyszerű szimulációs modellek esetén az objektumok átrendezése nem feltétlenül szükséges, mivel ahogy látni fogjuk, ha az alapértelmezett megjelenési pontjaikon maradnak, akkor is átlátható marad a szimulációs folyamat a felületen. Azonban egy összetettebb, több elemből álló szimulációs modell esetén már szükségessé válhat az objektumok átrendezése, csoportosítása, hogy átláthatóbb folyamatrendszert kapjunk eredményül.

Az egyszerű példamodell megépítésének első lépéseként szükségünk van egy forrás objektumra, azaz jelen esetben egy jegykiadó automatára (ticket machine), amely a service eszköztár első eleme a kezelőfelületen. A gomb megnyomása után a jegykiadó automata objektum piktogramja megjelenik a szimulációs felület bal oldalán (9. ábra).



9. sz. ábra: ticket machine (jegykiadó automata) objektum elhelyezése a szimulációs felületen

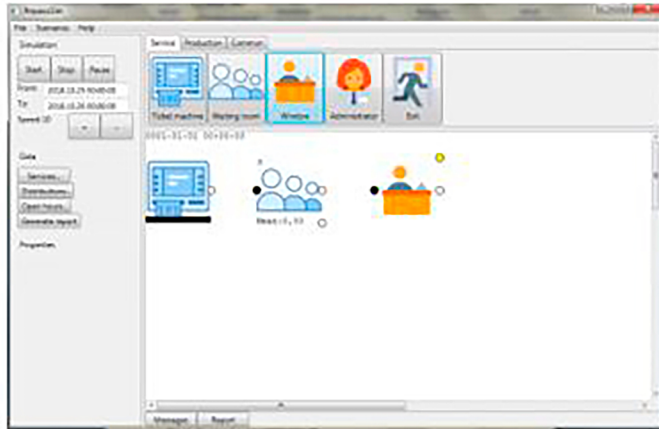
A példamodell megépítésének második lépéseként el kell helyoznünk a szimulációs felületen a váróterem (waiting room) objektumot, amely a második elem a service eszköztár kezelőfelületén. A várótermet jelképező gomb megnyomása után az objektum megjelenik közvetlenül a jegykiadó automata objektuma mögött (10. ábra). A váróterem a szimulációs modellben a puffer szerepét tölti be, funkcióját tekintve egy tárolóként érdemes tekintenünk rá, amiben a jegykiadó automatától érkező elemek (állampolgárok) várakoznak mindaddig, amíg sorra nem kerülhetnek valamelyik kormányablakban és megkezdődhet a tényleges ügyintézés.



10. sz. ábra: waiting room (váróterem) objektum elhelyezése a szimulációs felületen

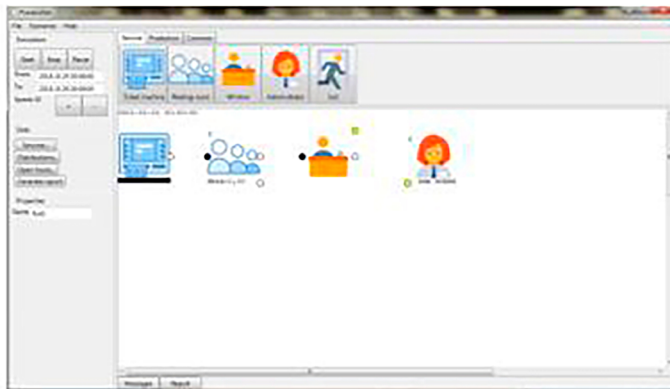
A példamodell megépítésének harmadik elemeként a szimulációs felületen el kell helyoznünk egy Kormányablak (Window) objektumot, ahol majd az igénybe vett szolgáltatás vagy szolgáltatások tényleges ügyintézése végrehajtásra kerül. A kormányablakot jelképező gomb megnyomása után az objektum megjelenik közvetlenül a váróterem objektuma mögött (11. ábra). A kormányablak objektum tölti be a szimulációban a munkaeszköz (munkagép)

szerepét, ez az objektum képes arra, hogy egy adott típusú feladatot, vagy előre definiált feladatok különböző típusait végrehajtsa, ami által a beérkező állampolgár által igénybe vett szolgáltatási funkció megvalósul.



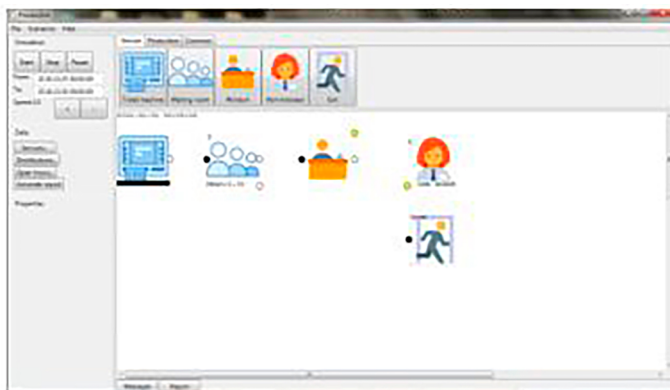
11. sz. ábra: window (kormányablak) objektum elhelyezése a szimulációs felületen

A példamodell megépítésének negyedik elemeként a jegykiadó automata, a váróterem és a kormányablak objektum után szükségünk van egy ügyintéző (administrator) objektumra is. Az eszköztáron az ügyintézőt jelképező piktogram megnyomása után az ügyintéző objektum is megjelenik a szimulációs felületen közvetlenül a kormányablak objektum után (12. ábra). Az ügyintéző objektum és a kormányablak objektum csak együtt működőképesek, ügyintéző nélkül a feladatok nem lesznek végrehajtva, a szimulációs folyamatban ő tölti be a munkát ténylegesen elvégző munkás szerepét.



12. sz. ábra: administrator (ügyintéző) objektum elhelyezése a szimulációs felületen

Ahhoz, hogy a szimuláció sikeresen megvalósuljon a példamodellben, utolsó elemként szükségünk van egy kimeneti objektumra is, amelyen keresztül a szolgáltatásokat igénybe vevő állampolgárok az ügyintézés befejezése után elhagyhatják a kormányablakot. Ez a kimeneti objektum a kijárat (exit) objektum, ami a kezelőfelület eszköztárának legutolsó eleme. Miután rákattintunk az exit objektum gombjára, az szintén megjelenik a szimulációs felületen közvetlenül az ügyintéző után (13. ábra). A későbbi jobb átláthatóság végett ezt az objektumot célszerű a korábban leírt módszer segítségével egy kicsit áthelyezni (13. ábra).



13. sz. ábra: exit (kijárat) objektum elhelyezése és mozgatása a szimulációs felületen

A szimulációs folyamat megépítésének utolsó lépéseként ahhoz, hogy a szimulációs modell működőképes legyen, a szimulációs felületen elhelyezett objektumokat mindenképpen össze kell kötnünk egymással. A korábban már ismertetett módon, egymás után sorban haladva először kössük össze a jegykiadó automata (ticket machine) kimenetét a váróterem (waiting room) bemenetével. Amint az látható, a váróteremnek két kimenete is van, ezek közül a felső az elsődleges kimenet, ezt kössük össze a kormányablak (window) bemenetével. A váróterem másodlagos kimenete a lemorzsolódás kezelésére szolgál és további kijárat (exit) objektum illeszthető be és köthető össze vele, azonban ez nem része az első példa megvalósítási tervének. A váróterem objektum működési elvéről bővebb információ található meg a szimulációs szoftver felhasználói útmutatójában. A kormányablak objektum egy kimenettel rendelkezik, ezt szintén kössük össze a kijárat objektum bemenetével. Utolsó lépésként az ügyintéző (administrator) objektumot kell még összekötnünk a kormányablakkal, hogy megteremtjük a két elem között az összeköttetést és ezáltal biztosítsuk a beérkező szolgáltatási igények megfelelő kezelését (14. ábra).



14. sz. ábra: a szimulációs felületen lévő objektumok összekötése egymással connectorok (összekötő elemek) segítségével

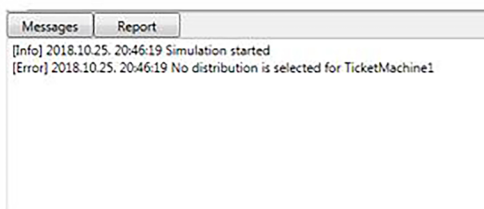
12.1.3. Futtatás

A szimulációs felületen elhelyezett objektumok összekötését követően a szimulációs modell lényegi részének felépítésével végeztünk, a folyamatára jól látható és az elemek egyértelműen elkülönülnek egymástól. Azonban ahhoz, hogy a szimulációt ténylegesen elkezdhesük futtatni, számos további beállítást kell alkalmaznunk az egyes elemeken.

A jelenlegi bemutató példában egyetlen szolgáltatás esetén fogjuk megvizsgálni a szimuláció lefutásának menetét. Amennyiben a szükséges beállítások nélkül próbálnánk meg futtatni a szimulációs modellt, úgy a beépített parancsok azonnal egy hibaüzenetet jelenítenének meg számunkra a szimulációs felület alatt felugró részen, amely egyértelműen közli a felhasználóval, hogy milyen beállítások megtétele szükséges a szimuláció sikeres elindításához (15. és 16. ábra).



15. sz. ábra: hibaüzenetek megjelenítése a szimulációs modell hibás vagy hiányos alapbeállításai esetén (1)

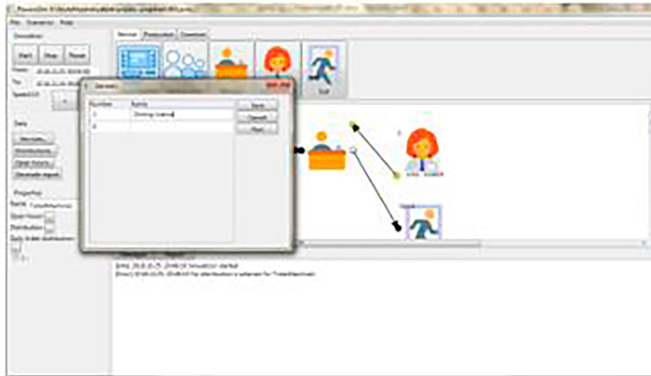


16. sz. ábra: hibaüzenetek megjelenítése a szimulációs modell hibás vagy hiányos alapbeállításai esetén (2)

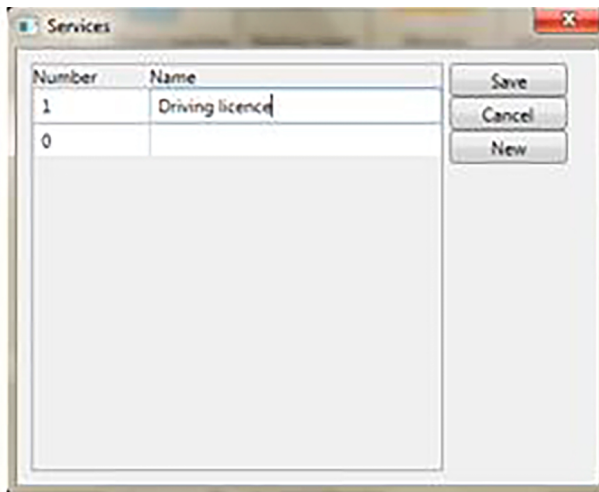
Amint az a 16. ábrán jól látható, ebben az esetben a bal oldali kezelőfelületen található start gomb megnyomása után a szimulációs szoftver közli velünk, hogy a szimulációs folyamat elindult ugyan, azonban egyből hibába is ütközött, mivel a folyamat bemeneteként funkcionáló jegykiadó automata (ticket machine) esetén nem választottunk ki elosztási típust, ami jellemezhetné a beérkező állampolgárok időbeli mértékének alakulását. Miután a szimulációs szoftver által közölt információk alapján megállapítottuk, hogy mi a hiba forrása, a hibákat felsoroló ablakot a messages (üzenetek) gomb megnyomásával tudjuk lekicsinyíteni, és ugyanezt a gombot lenyomva tudjuk később bármikor újra megjeleníteni is.

A következő lépésünk tehát még a szimuláció elindítása előtt az kell, hogy legyen, hogy mindegyik objektum esetén elvégezzük a szükséges további beállításokat és adatbevitelt. Természetesen ez a lépés történhet az objektumok létrehozásával párhuzamosan is külön-külön, vagy az itt látható módon az objektumok elhelyezése után. A modell futtatási leírásának következő szakaszában ezeket a lépéseket fogjuk sorra venni, elsőként a jegykiadó automata esetében.

A jegykiadó automata esetén első lépésként ki kell választanunk, hogy ez az adott objektum a jelenlegi példában milyen szolgáltatást igénybe vevő ügyfeleket generáljon a szimulációban. Ehhez a szimulációs szoftver bal oldali kezelőfelületének data (adatok) szekciójában először definiálnunk kell egy általunk választott szolgáltatástípust. Ahhoz, hogy ezt megtehessek, elsőként rá kell kattintanunk a data szekció services (szolgáltatások) gombjára, ami után egy felugró ablakban vihetjük be az általunk kívánt szolgáltatás vagy szolgáltatások neveit, valamint sorszámozhatjuk is őket. Ebben a példában egyetlen szolgáltatást fogunk definiálni, ami a jogosítvány igénylése lesz (driving licence). A szolgáltatás nevének megadása után rá kell kattintanunk a mentés (save) gombra, hogy az újonnan definiált szolgáltatás neve hozzárendelődjön a jegykiadó automatához, majd ezt követően, ha újból rákattintunk a jegykiadóra, akkor láthatjuk a kezelőfelület tulajdonságok (properties) szakaszánál, hogy a driving licence felirat megjelenik legalul, mint kiválasztható szolgáltatás. (17. és 18. ábra) Több szolgáltatást is definiálhatunk a fent leírt módszerrel, azonban a tetszőleges számú szolgáltatások közül egy jegykiadó automata csak egy szolgáltatást fog tudni kezelni. A valóságban természetesen egyetlen automata kezel minden szolgáltatást és ad hozzájuk jegyeket, azonban a szimulációs szoftver ezt praktikussági és átláthatósági okok miatt külön kezeli.



17. sz. ábra: szolgáltatás definiálása a jegykiadó automata (ticket machine) objektum számára (1)



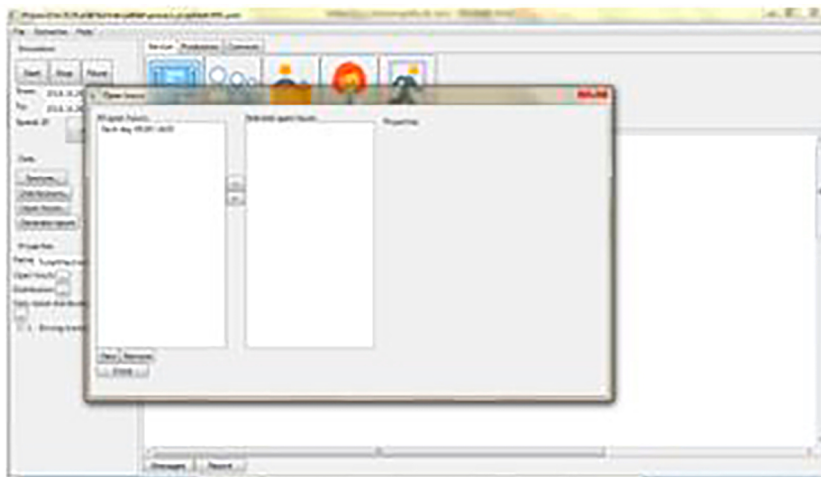
18. sz. ábra: szolgáltatás definiálása a jegykiadó automata (Ticket machine) objektum számára (2)

Miután a definiált szolgáltatás nevét beírtuk és a mentés (save) gomb segítségével elmentettük, elengedhetetlenül szükséges, hogy a megadott szolgáltatást ténylegesen hozzárendeljünk a jegykiadó automatához. Ehhez nem kell mást tennünk, mint a properties szekcióban kiválasztani és kijelölni a driving licence szolgáltatás előtt látható checkboxot. Amennyiben ezt véletlenül elmulasztanánk, a szoftver beépített hibaelLENŐRZŐ felülete fel fogja rá hívni a figyelmünket egy erre vonatkozó üzenet formájában.

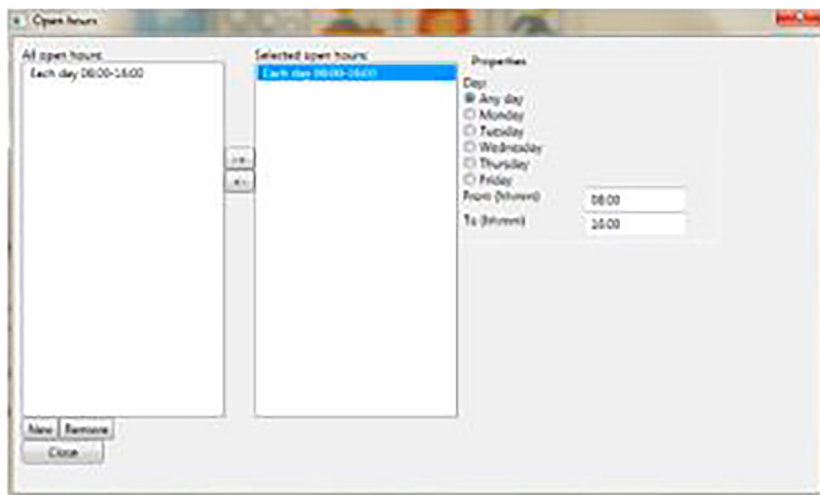
A jegykiadó automata beállítása során a következő lépés, amit mindenképpen meg kell tennünk, az a nyitvatartási idő definiálása. A szimuláció számára a bal oldali kezelőfelület felső részében megadhatók a szimuláció futási idejére vonatkozó paraméterek, így például eldönthetjük, hogy csak egy adott napot szeretnénk definiálni és vizsgálni, vagy akár egy egész hétnyi vagy hónapnyi intervallumban szeretnénk vizsgálatokat folytatni.

Legyen szó akár egyetlen napról, vagy ennél hosszabb időintervallumról, a jegykiadó automata számára meg kell adni egy adott napon belül a munkaidő hosszát. Hosszabb szimulációk esetén a jegykiadó csak ebben a napi időintervallumban fog szolgáltatásokat igénybe vevő ügyfeleket generálni. A ticket machine objektum kijelölése után a kezelőfelület properties részén az open hours (nyitvatartási órák) gombra kattintva ismét egy új ablak ugrik fel. Ennek az ablaknak a bal oldalán láthatók a már definiált nyitvatartási idők. Az alapértelmezett nyitvatartási időintervallum reggel nyolctól délután négy óráig terjed, kiválaszthatjuk ezt, vagy akár generálhatunk egy új nyitvatartási időintervallumot is a szimuláció számára tetszés szerint. Jelen példánkban az előre definiált reggel nyolctól délután négyig terjedő időintervallumot fogjuk alkalmazni, a további intervallumok létrehozásáról bővebb információkat az olvasó a szimulációs szoftver kezelési útmutatójában találhat.

A ticket machine számára a nyitvatartási idő megadása úgy történik, hogy a már definiált időintervallumot a bal oldali oszlopban kiválasztjuk, majd pedig rákattintunk a két oszlop között található nyilak közül a jobbra, vagyis az üres oszlopra mutatóra. Ezt követően a nyitvatartási idő hozzáadásra kerül a jobb oldalon található oszlophoz (19. és 20. ábra). A felületen további kiegészítő lehetőségként beállítható, hogy az adott nyitvatartási időintervallum bármelyik napra (any day) vagy csak egy adott napra vonatkozzon, így akár az eltérő napokra eltérő nyitvatartási időket is definiálhatunk egy összetett szimulációs vizsgálat során. Az ablak bezárása után láthatjuk, hogy a properties (tulajdonságok) szakaszban az open hours felirat mellett feltűnt a kiválasztott nyitvatartási időintervallum.



19. sz. ábra: nyitvatartási időintervallum definiálása a jegykiadó automata (ticket machine) objektum számára (1)

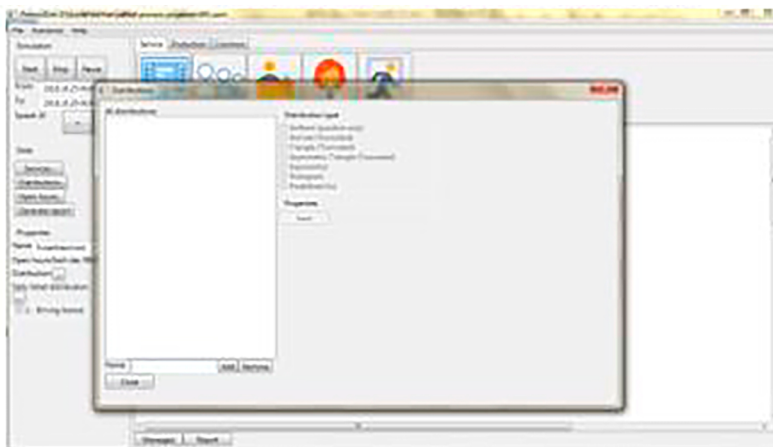


20. sz. ábra: nyitvatartási időintervallum definiálása a jegykiadó automata (ticket machine) objektum számára (2)

Végül, de nem utolsó sorban a nyitvatartási idő intervallumának megadása után ki kell választanunk egy eloszlást, amit az adott objektum használni fog majd a kiválasztott szolgáltatást igénybe vevő állampolgárok generálására. A szimulációs szoftverben számos előre beépített eloszlási függvény közül választhatunk, amelyeket tetszés szerint változtathatunk is, vagy a függvény egérrel történő húzásával történő módosításával, vagy akár konkrét egyedi értékekkel is szabhatjuk őket testre.

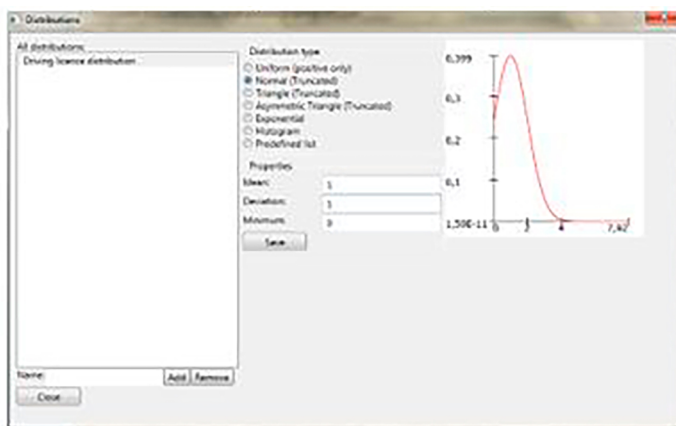
Az adott objektumhoz kétféle módon definiálhatunk eloszlási értékeket, vagy a kezelőfelület data (adatok) szakaszán kattintunk rá a distributions (eloszlás) gombra, vagy a properties (tulajdonságok) szakaszon választjuk ki szintén a distributions gombot. Ez utóbbit csak akkor tehetjük meg, ha a jegykiadó automata (ticket machine) ki van jelölve a szimulációs felületen, mivel a properties szakaszon csak így jeleníti meg a szoftver az adott objektumhoz tartozó egyedi adatokat és tulajdonságokat.

A distributions gombra való rákattintás után egy új ablakfelület jelenik meg, amely két fő részből áll, egy jobboldali üres fehér oszlopból (amennyiben még nem definiáltunk korábban a szimulációhoz kötődő eloszlásokat) és egy jobb oldali listából, amely felsorolja a lehetséges választható eloszlástípusokat. Ez utóbbi mindaddig szürke marad, amíg egy egyedi névvel ellátott eloszlástípust nem hozunk létre a felületen. Ehhez nem kell mást tennünk, mint az ablak bal alsó sarkában található name (név) mezőbe beírni egy általunk tetszőlegesen választott vagy kitalált nevet, majd ezt követően rá kell kattintanunk az add (hozzáadás) gombra, amely által az általunk generált eloszlás elnevezése megjelenik majd a bal oldali fehér oszlopban. A mi példánk esetében ez az elnevezés driving licence distribution (jogsítvány eloszlás) lesz. Az elnevezést és a hozzáadást követően az elnevezés megjelenik a listában, majd erre rákattintva már választhatóvá válnak az egyes eloszlástípusok, és így mi magunk választhatjuk ki, hogy milyen eloszlást szeretnénk hozzárendelni a jegykiadó automatánkhöz (21. és 22. ábra).

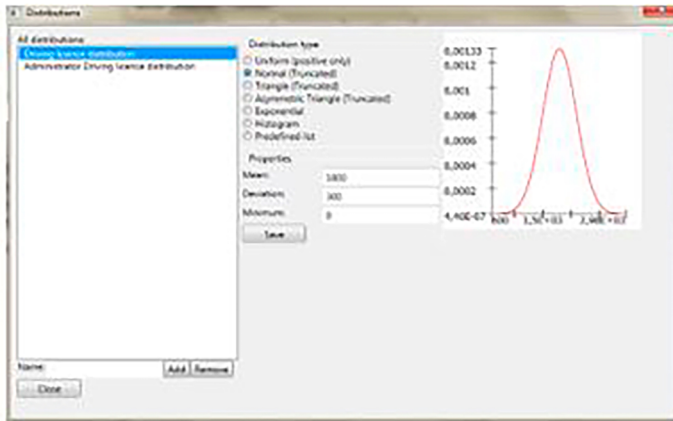


21. sz. ábra: egyedi eloszlástípus definiálása a jegykiadó automata (ticket machine) objektum számára (1)

A valóságban az adatgyűjtés során szerzett adatokból megállapítható az eloszlás típusa, ezt pedig betáplálhatjuk a szimulációba. Ebben a példában viszont az egyszerűség kedvéért az előre definiált normális eloszlást fogjuk kiválasztani az általunk generált eloszlás típusának, egyedi értékeknek pedig be fogjuk állítani, hogy átlagosan (mean) harminc percnként érkezzon egy ügyfél, öt perces szórással. A szoftverben az időértékeket másodpercben kell megadni, tehát a harminc perc ezernyolcszáz másodpercnek felel meg, míg az öt perc háromszáz másodpercnek. A megfelelő időértékek megadása után a save (mentés) gomb lenyomásával véglegesítjük a műveletet. Az adatok megadása után a felület jobb oldalán található grafikonon is azonnal nyomon követhető a függvény görbéjének és elhelyezkedésének a megváltozása (22. és 23. ábra).



22. sz. ábra: egyedi eloszlástípus definiálása a jegykiadó automata (ticket machine) objektum számára (2)



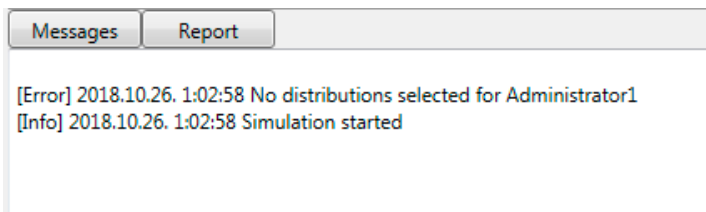
23. sz. ábra: egyedi eloszlástípus definiálása a jegykiadó automata (ticket machine) objektum számára (3)

Az egyes eloszlástípusok főbb jellemzőiről és a beállítások további módjairól részletes leírás található meg a szimulációs szoftver felhasználói útmutatójában.

A jegykiadó automata működéséhez szükséges szolgáltatás, nyitvatartási idő és eloszlástípus definiálása után immáron elvégeztük a teljes körű alapbeállításokat az objektummal kapcsolatban, az eloszlások beállítását segítő ablak bezárása után láthatjuk, hogy a jegykiadó automata tulajdonságai között ott szerepel az általunk kiválasztott és definiált eloszlás elnevezése.

Ezt követően, ha egyből ismét a start gombra kattintunk, hogy futtassuk a szimulációt, ismét egy hibaüzenetbe fogunk ütközni, amit ezúttal ügyintéző (administrator) objektum okoz. A rendszerüzeneteket tartalmazó alsó felület felugrása után látható válik, hogy a hiba forrása az ügyintéző objektum alapvető beállításainak hiánya (23. ábra).

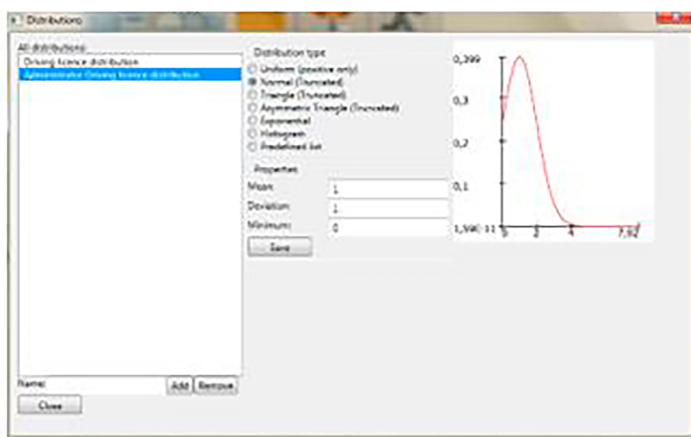
A szimulációs szoftver futtatásához az egyes ügyintézők számára is definiálnunk kell egy eloszlás típust, valamint elengedhetetlenül szükséges az is, hogy kijelöljük, hogy az adott ügyintézők milyen típusú szolgáltatást vagy szolgáltatásokat tudnak kezelni a kormányablakokban, hiszen ebből tudják majd a váróteremben várakozó ügyfelek eldönteni, hogy melyik ablakhoz menjenek tovább. Egy valódi adatgyűjtés során ezek az információk is mind meghatározhatók lesznek a begyűjtött adatokból statisztikai eszközök segítségével, akár csak a szimuláció többi eleme esetében.



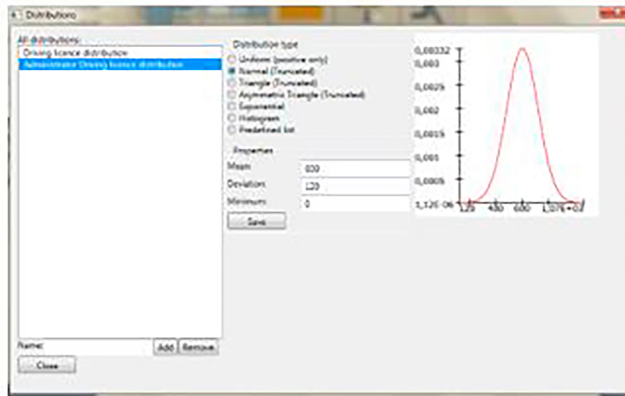
24. sz. ábra: hibaüzenetek megjelenítése a szimulációs modell hibás vagy hiányos alapbeállításai esetén (3)

Az Ügyintéző objektum alapbeállításainak első lépéseként jelöljük ki az objektumot a szimulációs felületen. A kijelölést követően láthatóvá válnak a bal oldali kezelőfelület properties (tulajdonságok) szakaszán az adott objektum egyedi értékei, ezek között pedig felsorolva láthatjuk az összes olyan szolgáltatás elnevezését, amelyet definiáltunk már korábban. Jelen esetben egyetlen kiválasztható szolgáltatástípus van, méghozzá a jogosítvány ügyintézése (driving licence), amit ki is választunk. A neve előtti checkboxba pipát téve azonnal felugrik az eloszlások definiálására szolgáló ismerős ablak, ahol láthatjuk a már korábban általunk definiált eloszlásokat. Jelen példánk esetében látható a jegykiadó automata számára a jogosítványok esetén létrehozott normális típusú eloszlás.

Az ügyintézők esetében tehát külön-külön is definiálható, hogy az egyes szolgáltatásokat milyen típusú eloszlásfüggvénnyel jellemezhető módon képesek kiszolgálni. Ez alapulhat valós mért adatokon, vagy ismételten használhatunk előre definiált formulákat is. Ebben a példában az ügyintéző számára is egy normális típusú eloszlást fogunk generálni, azonban ennek az eloszlásnak mások lesznek az átlagos és szórás értékei. Elsőként a felugró ablak alsó részén ismét megadunk egy általunk kiválasztott elnevezést, ami jelen esetben administrator driving licence distribution (ügyintéző jogosítvánnyal kapcsolatos munkavégzésének eloszlása) nevet fog kapni (24. és 25. ábra).



25. sz. ábra: egyedi eloszlástípus definiálása az administrator (ügyintéző) objektum számára (1)



26. sz. ábra: egyedi eloszlástípus definiálása az administrator (ügyintéző) objektum számára (2)

Az ügyintézőhöz rendelt normális típusú eloszlás átlaga az előzőtől eltérően tíz perc lesz, míg a szórása kettő perc, tehát az ügyintéző átlagosan tíz perc alatt tudja majd ez a szolgáltatást végrehajtani, kettő perces szórással. Nem szabad elfelejtenünk, hogy az időadatokat ismételtelen másodpercekben kell definiálnunk, tehát a tíz perc hatszáz másodpercnek felel meg, a két perc pedig százhusz másodpercnek felel meg. Az adatok megadása után pedig a felület jobb oldalán található grafikonon is ismét azonnal nyomon követhető a függvény görbéjének és elhelyezkedésének a megváltozása (24. és 25. ábra).

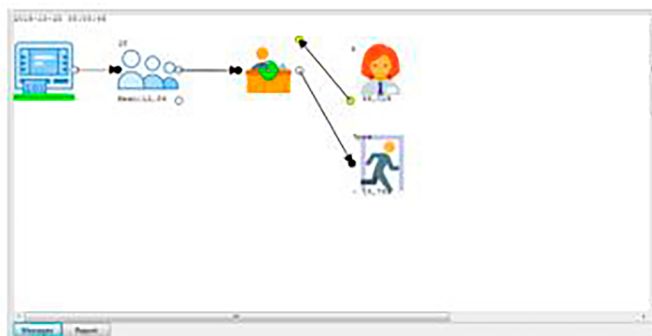
A distributions ablak bezárása után immáron láthatjuk, hogy az ügyintézőhöz kiválasztott driving licence (jogosítvány) szolgáltatás mellett megjelenik a hozzá rendelt eloszlás neve, vagyis az administrator driving licence distribution felirat, amelyet az imént definiáltunk az eloszlások megadásának felületén.

Az ügyintéző alapbeállításainak megfelelő elvégzése után nincs más dolgunk, mint ismét elindítani a szimulációs modellt futtatását a bal oldali kezelőfelület felső részén található start (indítás) gomb segítségével. Amennyiben minden beállítást az előbb leírtaknak megfelelően végeztünk el mind a jegykiadó automata, mind pedig az ügyintéző objektumok esetén, úgy ezúttal a szimulációs modell futásának már gond nélkül el kell indulnia.

Amint az a gomb megnyomása után látható a folyamatszimuláció ezúttal már gond nélkül elindul, a jegykiadó automata elkezd generálni a driving licence (jogosítvány igénylése) szolgáltatást igénybe vevő ügyfeleket, akik ezt követően bekerülnek a váróterembe, ahol mindaddig várnak, amíg sorra nem tudnak kerülni a kormányablakban, ahol az ügyintéző segítségével a szolgáltatás lebonyolítása végbe megy, majd a szolgáltatás befejezését követően a kijáraton át távoznak.

12.1.4. Eredmények értékelése

Amint az a szimulációs felületen elhelyezett objektumokon is jól látható, a szimulációs folyamat elindítása után az egyes objektumok állapotában változás áll be. Amint az alábbi ábrán is látható, az adott pillanatban a pufferként funkcionáló váróteremben várakozó elemek száma leolvasható a waiting room objektum bal felső sarkában, alatta pedig láthatjuk az átlagosan várakozók számát valós időben nyomon követhető (26. ábra).



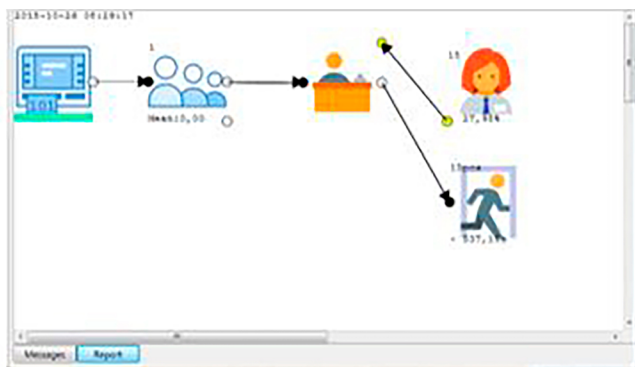
27. sz. ábra: a szimuláció futtatása közben leolvasható információk az egyes objektumok esetén

A kormányablak objektum esetén az objektum közepén látható számlálóról leolvashatjuk, hogy éppen hányadik ügy végrehajtása van folyamatban. Több hozzárendelt szolgáltatástípus esetén az egyes típusok eltérő színnel kerülnek megjelenítésre, ami vizuálisan is megkönnyíti a modellen látható információk leolvasását, ezzel kapcsolatban a szoftver felhasználói útmutatójában részletesebb tájékoztatás is található (27. ábra).

Az ügyintéző objektumáról szintén leolvashatók valós idejű adatok, itt láthatjuk a bal felső sarokban a már befejezett ügyek számát, vagyis az adott ügyintéző által kiszolgált állampolgári igények mennyiségét. Az objektum alsó részén látható százalékos érték az adott ügyintéző kihasználtságáról nyújt valós idejű tájékoztatást (27. ábra).

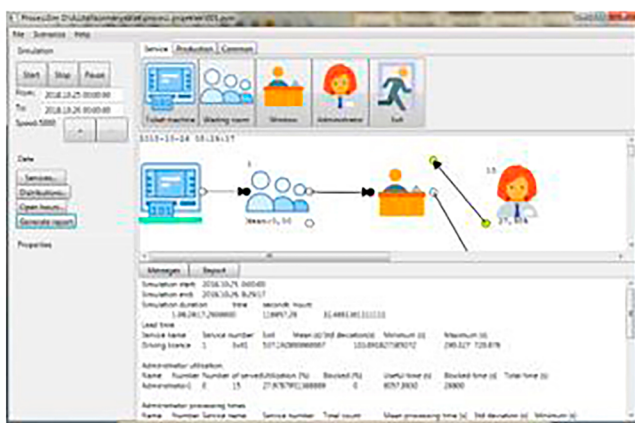
Az előző objektumokhoz hasonlóan a kijárat objektumról is olvashatunk le információt, a bal felső sarokban láthatjuk a távozott ügyfelek számát, míg az objektum alján látható számláló az átlagos átfutási idő mértékéről nyújt valós idejű tájékoztatást. (27. ábra)

A szimulációs modell hosszabb ideig történő futtatása esetén jól látható, hogy miként alakultak az egyes objektumok értékei, a szimuláció leállítását követően pedig elemezhetjük az addig végbement folyamatok adatait részletesen. Amint az ábrán is jól látható, a szimuláció leállításának pillanatáig az ügyintéző tizenöt beérkezett ügyfelet szolgált ki, a váróteremben pedig egy ügyfél maradt várakozóként, akire a nyitvatartási idő lejártáig már nem került sor (28. ábra). A leállítás történhet kézzel a megfelelő gombok segítségével a kezelőfelületen, vagy ugyanezt megadhatunk egy előre definiált időintervallumot is, amin belül futtatni szeretnénk az adott szimulációs modellt. Ebben az esetben is az előre definiált időintervallumot választottuk, a szimuláció egy egész szimulált nap folyamatait vette végig a kormányablak kinyitásától egészen a definiált munkaidő hosszának lejártáig.



28. sz. ábra: a szimuláció futtatása után leolvasható információk az egyes objektumok esetén

A végrehajtott szimulációs folyamat eredményeinek részletes megtekintéséhez az adatokról egy jelentést kell generálnunk, amit úgy tehetünk meg, ha a bal oldali kezelőfelületen rákattintunk a generate report (jelentés generálása) gombra. Ezt követően a szimulációs felület alatt elhelyezkedő üzenőpanel ismét megjelenik és láthatóvá válik rajta a report fül alatt a részletes jelentés minden adata (29. és 30. ábra).



29. sz. ábra: jelentés generálása az elvégzett szimuláció részletes adatainak áttekintéséhez és további elemzéséhez (1)

Ezeket az adatokat a könnyebb áttekinthetőség és a további elemezhetőség érdekében közvetlenül átmásolhatjuk egy táblázatkezelő program felületére úgy, hogy az adatok kijelölése után azokat másoljuk, majd pedig a táblázatkezelő program megfelelő gombja segítségével beillesztjük. Ezt követően az adott példa esetén az alábbi táblázatba rendezett eredményeket kapjuk.

Messages		Report	
Simulation start: 2018.10.25. 0:00:00			
Simulation end: 2018.10.26. 8:29:17			
Simulation duration			
time	seconds	hours	
1:08:29:17.2900000	116997,29	32,4881361111111	
Lead time			
Service name	Service number	Exit	Mean (s) Std deviation(s) Minimum (s) Maximum (s)
Driving licence	1	Exit1	537,192866666667 103,891827385072 290,327 720,876
Administrator utilization			
Name	Number of served	Utilization (%)	Blocked (%) Useful time (s) Blocked time (s) Total time (s)
Administrator1	0 15	27,9787951388889	0 8057,8930 28800
Administrator processing times			
Name	Number	Service name	Service number Total count Mean processing time (s) Std deviation (s) Minimum (s)

30. sz. ábra: jelentés generálása az elvégzett szimuláció részletes adatainak áttekintéséhez és további elemzéséhez (2)

Simulation duration	time	seconds	hours			
	1.08:26:41.9750000	116801,98	32,4449931			
Lead time						
Service name	Service number	Exit	Mean (s)	Std deviation(s)	Minimum (s)	Maximum (s)
Driving licence	1	Exit1	590,875875	97,5114226108066	352,851	739,434

31. sz. ábra: jelentés generálása az elvégzett szimuláció részletes adatainak áttekintéséhez és további elemzéséhez (3). Az egyes szolgáltatásokra vonatkozó összesített adatok (átfutási idő átlagos hossza és szórása, minimum és maximum értékek)

Az így kapott jelentés alkalmas arra, hogy kiolvassuk belőle az egyes szolgáltatásokra vonatkozó átfutási időket, átlagokat, szórásokat, az ügyintézők kihasználtságát, az egyes szolgáltatások igénybe vevőinek számát, a sorban állási idők átlagát és szórását és még sok minden mást (31. ábra). A jelentés tartalmáról és lehetséges további felhasználásáról az olvasó bővebb információkat találhat a szoftver működését részletesen tartalmazó felhasználói útmutatóban.

Administrator utilization							
Name	Number of served	Utilization (%)	Blocked (%)	Useful time (s)	Blocked time (s)	Total time (s)	
Administrator1	16	32,8264375	0	9454,014	0	28800	
Administrator processing times							
Name	Service name	Service number	Total count	Mean processing time (s)	Std deviation (s)	Minimum (s)	Maximum(s)
Administrator1	Driving licence	1	16	590,875875	97,5114226108066	352,851	739,434

32. sz. ábra: jelentés generálása az elvégzett szimuláció részletes adatainak áttekintéséhez és további elemzéséhez (4). Az adminisztrátorokra vonatkozó összesített adatok (kiszolgált ügyek száma, ügyintézési idő átlagos hossza és szórása, minimum és maximum értékek)

Az így kapott jelentésből az adott példa esetén megállapítható, hogy összesen tizenhat állampolgár vette igénybe a driving licence, azaz a jogosítvány megújítását jelképező szolgáltatást a szimuláció futási ideje alatt, átlagosan harminc percenként érkeztek. Továbbá megállapítható, hogy a kormányablakhoz rendelt adminisztrátor az előre definiált eloszlási paramétereknek megfelelően átlagosan tíz percet töltött egy ügyfél szolgáltatási igényének a kielégítésével, két perces szórással, ez idő alatt pedig a kihasználtsága közel harminchárom százalékos volt (32. ábra).

12.2. Második példa: összetett szimulációs modell felépítésének folyamata

Egyetlen bemenettel és egyetlen kimenettel rendelkező folyamat modellezése, amelyben öt szolgáltatást szolgál ki három ügyintéző.

12.2.1. Bemutató

A második példa egy, az előző egyszerű modellnél jóval összetettebb szimulációs modell felépítését fogja bemutatni a szoftverben, amely öt jegykiadó automatából, egyetlen váróteremből, három ügyfélszolgálati ablakból, három ügyintézőből és egyetlen kijáratból fog állni. Az egyes kormányablakok ügyintézői eltérő számú szolgáltatás elvégzéséhez szükséges jogosítvánnyal fog rendelkezni, azaz csak részben lesz átfedés az ő ablakaiknál intézhető ügyek halmazai között.

Ennek a bemutató példának az a célja, hogy a felhasználókat megismertesse az előző példában bemutatott alapvető modellépítés és működtetés menetéhez elengedhetetlenül szükséges feltételeken túl egy, az előzőnél nagyságrendileg összetettebb szimulációs modell felépítésének menetével, melynek keretében lehetővé válik egy sokkal komplexebb modell felépítése és átlátása.

12.2.2. Felépítés menete

Az összetett modell felépítésének menete ugyanúgy indul, ahogy a korábban bemutatott egyszerű példamodell esetén is csináltuk, így az alapvető lépések bemutatásáról részletesebb útmutató megtalálható az egyszerű modell felépítési menetéről szóló korábbi fejezetben.

Az összetett példamodell megépítésének első lépéseként szükségünk van egy forrás objektumokra, azaz jelen esetben jegykiadó automatákra (ticket machine), amely a service eszköztár első eleme a kezelőfelületen. A gombra kattintva a jegykiadó automata objektum piktogramja megjelenik a szimulációs felület bal oldalán. A szimulációs szoftver sajátos felépítéséből fakadóan egy jegykiadó automata egyetlen szolgáltatásért lesz felelős és az azt igénybe vevő ügyfeleket fogja majd generálni, így tehát öt szolgáltatás esetén öt jegykiadó automatára lesz szükségünk a szimulációs modell megépítéséhez.

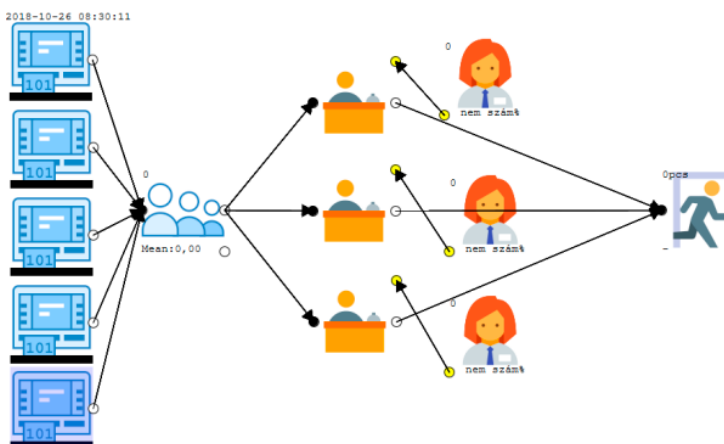
A példamodell megépítésének második lépéseként el kell helyeznünk a szimulációs felületen a váróterem (waiting room) objektumot, amely a második elem a service eszköztár kezelőfelületén. A várótermet jelképező gomb megnyomása után az objektum megjelenik közvetlenül a jegykiadó automata objektuma mellett. Ahogyan azt már korábban is tárgyaltuk,

ez az objektum a szimulációs modellben a puffer szerepét tölti be, funkcióját tekintve egy tárolóként érdemes tekintenünk rá, amiben a jegykiadó automatától érkező elemek (állampolgárok) várakoznak mindaddig, amíg sorra nem kerülhetnek valamelyik kormányablakban és megkezdődhet a tényleges ügyintézés.

A példamodell megépítésének harmadik lépéseként a szimulációs felületen el kell helyeznünk három kormányablak (window) objektumot, ahol majd az igénybe vett szolgáltatás vagy szolgáltatások tényleges ügyintézése kerül végrehajtásra az ügyintézők által. Az eszköztáron a kormányablakot jelképező gomb megnyomása után az objektum megjelenik közvetlenül a váróterem objektuma mögött. Háromszori lenyomás után a három kormányablak objektum egymás alatt fog megjelenni a szimulációs felületen. A kormányablak objektumok töltik be a szimulációban a munkaeszközök (munkagépek) szerepét, ez az objektum képes arra, hogy egy adott típusú feladatot, vagy előre definiált feladatok különböző típusait végrehajtsa, ami által a beérkező állampolgár által igénybe vett szolgáltatási funkció megvalósul.

A példamodell megépítésének negyedik lépéseként a jegykiadó automata, a váróterem és a kormányablak objektumok után szükségünk van három ügyintéző (administrator) objektumra is. Az eszköztáron az ügyintézőt jelképező piktogram megnyomása után az ügyintéző objektum is megjelenik a szimulációs felületen közvetlenül a kormányablak objektum után. Háromszori lenyomás után a három ügyintéző objektum egymás alatt fog megjelenni a szimulációs felületen. Ahogyan azt már korábban is tisztáztuk az ügyintéző objektumok és a kormányablak objektumok csak együtt működőképesek, ügyintézők nélkül a feladatok nem lesznek végrehajtva a kormányablakokban, a szimulációs folyamatban ők töltik be a munkát ténylegesen elvégző munkások szerepét.

Ahhoz, hogy a szimuláció sikeresen megvalósuljon a példamodellben, utolsó elemként szükségünk van egy kimeneti objektumra is, amelyen keresztül a szolgáltatásokat igénybe vevő állampolgárok az ügyintézés befejezése után elhagyhatják a kormányablakot. Ez a kimeneti objektum a kijárat (exit) objektum, ami a kezelőfelület szolgáltatás eszköztárának legutolsó eleme. Miután rákattintunk az exit objektum gombjára, az szintén megjelenik a szimulációs felületen közvetlenül az ügyintéző után. A szimulációs folyamat megépítésének utolsó lépéseként ahhoz, hogy szimulációs modell működőképes legyen, a szimulációs felületen elhelyezett objektumokat mindenképpen össze kell kötnünk egymással, ahogyan azt az előző példában is megtettük, majd a későbbi jobb átláthatóság végett ezeket az objektumokat célszerű a korábban szintén leírt módszer segítségével egy kicsit áthelyezni. A modell megépítésének eredményeként a szimulációs felületen az alábbi szimulációs folyamatot kaptuk (33. ábra).



33. sz. ábra: a második példában szereplő összetett modell vizuális megjelenése a szimulációs felületen az átrendezés után

12.2.3. Futtatás

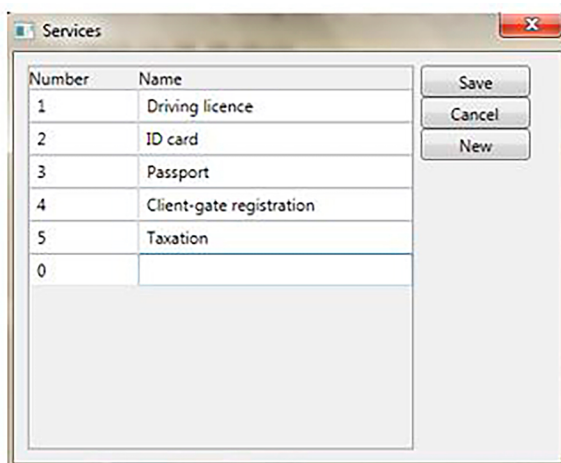
A szimulációs felületen elhelyezett objektumok összekötését követően a szimulációs modell lényegi részének felépítésével végeztünk, a folyamatára jól látható és az elemek egyértelműen elkülönülnek egymástól. Azonban ahhoz, hogy a szimulációt ténylegesen elkezdhesük futtatni, számos további beállítást kell alkalmaznunk az egyes elemeken.

A második bemutató példában a szimuláció lefutásának a menetét öt különböző szolgáltatás esetén fogjuk vizsgálni, azonban mielőtt még a szimuláció futtatását megkezdzenénk, szükséges az egyes szolgáltatások definiálását és a hozzájuk rendelt eloszlások bevitelét végrehajtanunk, ugyanis ezeknek az alapbeállításoknak a megtétele, amint az már az első példában is láthattuk, elengedhetetlen részét képezik a szimulációs folyamat lefutásának.

Amennyiben a szükséges beállítások nélkül próbálnánk meg futtatni a szimulációs modellt, úgy a beépített parancsok azonnal egy hibaüzenetet jelenítenének meg számunkra a szimulációs felület alatt felugró részen, amely egyértelműen közli a felhasználóval, hogy milyen beállítások megtétele szükséges még a szimuláció sikeres elindítása előtt.

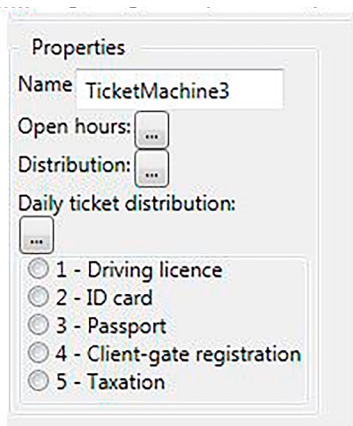
Elsőként tehát a jegykiadó automaták esetén ki kell választanunk, hogy ez az adott objektum a jelenlegi példában milyen szolgáltatást igénybe vevő ügyfeleket generáljon a szimulációban. Ehhez a szimulációs szoftver bal oldali kezelőfelületének data (adatok) szekciójában először definiálnunk kell az általunk választott szolgáltatástípusokat. Ahhoz, hogy ezt megtehesük, elsőként rá kell kattintanunk a data szekció services (szolgáltatások) gombjára, ami után egy felugró ablakban vihetjük be az általunk kívánt szolgáltatások neveit, valamint sorszámozhatjuk is őket.

Ebben a példában öt szolgáltatást fogunk definiálni, amik a jogosítvány igénylése (driving licence), a személyi igazolvány (ID card) megújítása, az útlevél (passport) igénylése, az ügyfélfelkapu-regisztráció (Client-gate registration), és az adóügyek intézése (taxation) lesznek. Jelen esetben, hogy elég beviteli mező álljon a rendelkezésünkre, a felület new (új) gombjára kattintva adhatunk hozzá további sorokat, hogy kellő számú beviteli mező a rendelkezésünkre álljon (34. ábra).



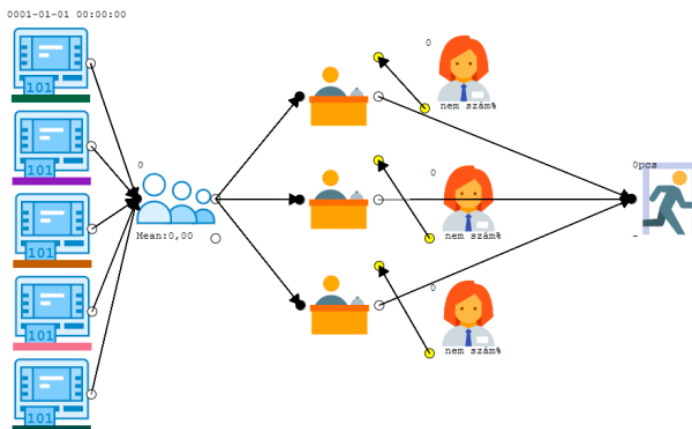
34. sz. ábra: szolgáltatások definiálása a jegykiadó automata (ticket machine) objektumok számára (1)

A szolgáltatások nevének megadása után rá kell kattintanunk a mentés (save) gombra, hogy az újonnan definiált szolgáltatások kiválaszthatóvá váljanak az egyes jegykiadó automataknak esetén. Ezt követően, ha rákattintunk egy jegykiadóra, akkor láthatjuk a kezelőfelület properties (tulajdonságok) szakaszánál, hogy a driving licence, ID card, passport, client-gate registration és taxation feliratok megjelennek legalul, mint kiválasztható szolgáltatások. A tetszőleges számú szolgáltatások közül egy jegykiadó automata csak egy szolgáltatást fog tudni kezelni, tehát ezt követően mindegyik jegykiadó automata esetén ki kell választanunk egy kezelt szolgáltatást a név mellett található checkbox segítségével (35. ábra).



35. sz. ábra: szolgáltatások definiálása a jegykiadó automata (ticket machine) objektumok számára (2)

A valóságban természetesen egyetlen automata kezel minden szolgáltatást és ad hozzájuk jegyeket, azonban a szimulációs szoftver ezt praktikussági és átláthatósági okok miatt külön kezeli.



36. sz. ábra: szolgáltatások definiálása a jegykiadó automata (ticket machine) objektumok számára (3)

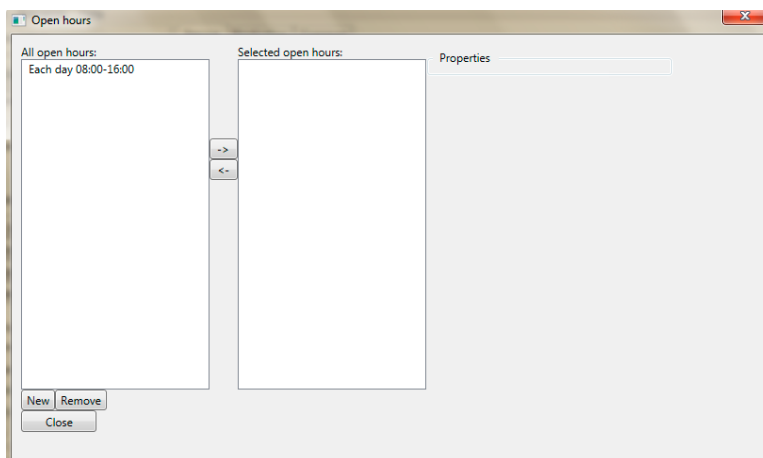
Amint az szintén látható a szimulációs felületen, miután minden egyes jegykiadó automatahoz hozzárendeltünk egy szolgáltatást, az automaták alatt korábban látható fekete sáv eltérő színűvé változik, a szimulációs szoftver ezen tulajdonsága szintén az átláthatóságot hivatott elősegíteni a folyamatok nyomon követése és elemzése során (36. ábra).

A jegykiadó automaták beállítása során a következő lépés, amit mindenképpen meg kell tennünk, az a nyitvatartási idő definiálása. A szimuláció számára a baloldali kezelőfelület felső részében megadhatók a szimuláció futási idejére vonatkozó paraméterek, így például eldönthetjük, hogy csak egy adott napot szeretnénk definiálni és vizsgálni, vagy akár egy egész hétnyi vagy hónapnyi intervallumban szeretnénk vizsgálatokat folytatni.

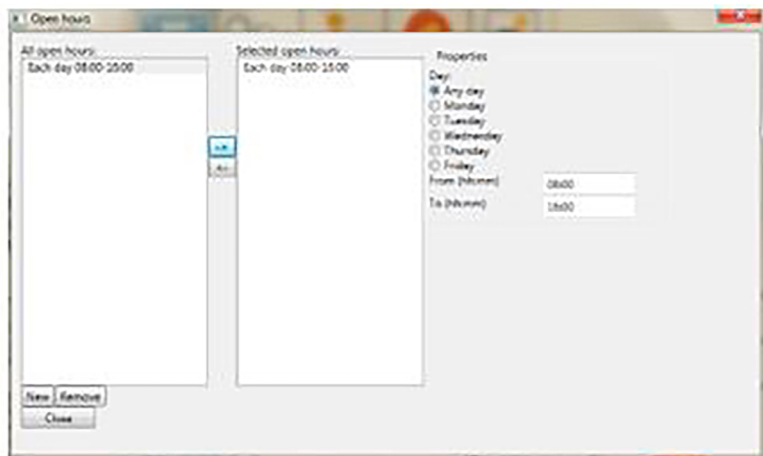
Jelen esetben, ahogy az előző példa esetében is, egyetlen nap folyamatát szeretnénk szimulálni, azonban, a jegykiadó automata számára meg kell adni egy adott napon belül a munkaidő hosszát. Hosszabb szimulációk esetén a jegykiadó csak ebben a napi időintervallumban fog szolgáltatásokat igénybe vevő ügyfeleket generálni. Az első ticket machine objektum kijelölése után a kezelőfelület properties részén az open hours (nyitvatartási órák) gombra kattintva ismét egy új ablak ugrik fel. Ennek az ablaknak a bal oldalán láthatóak a már definiált nyitvatartási idők. Az alapértelmezett nyitvatartási időintervallum reggel nyolctól délután négy óráig terjed, kiválaszthatjuk ezt, vagy akár generálhatunk egy új nyitvatartási időintervallumot is a szimuláció számára tetszés szerint. Jelen példánkban ismét az előre definiált reggel nyolctól délután négyig terjedő időintervallumot fogjuk alkalmazni, a további intervallumok létrehozásáról bővebb információkat az olvasó a szimulációs szoftver kezelési útmutatójában találhat.

A ticket machine számára a nyitvatartási idő megadása úgy történik, hogy a már definiált időintervallumot a bal oldali oszlopban kiválasztjuk, majd pedig rákattintunk a két oszlop között található nyilak közül a jobbra, vagyis az üres oszlopra mutatóra. Ezt követően a nyitvatartási idő hozzáadásra kerül a jobb oldalon található oszlophoz (37. és 38. ábra).

A felületen további kiegészítő lehetőségként beállítható, hogy az adott nyitvatartási időintervallum bármelyik napra (any day) vagy csak egy adott napra vonatkozzon, így akár az eltérő napokra eltérő nyitvatartási időket is definiálhatunk egy összetett szimulációs vizsgálat során. Az ablak bezárása után láthatjuk, hogy a properties (tulajdonságok) szakaszban az open hours felirat mellett feltűnt a kiválasztott nyitvatartási időintervallum.



37. sz. ábra: nyitvatartási időintervallum definiálása a jegykiadó automata (ticket machine) objektum számára (1)



38. sz. ábra: nyitvatartási időintervallum definiálása a jegykiadó automata (Ticket machine) objektum számára (2)

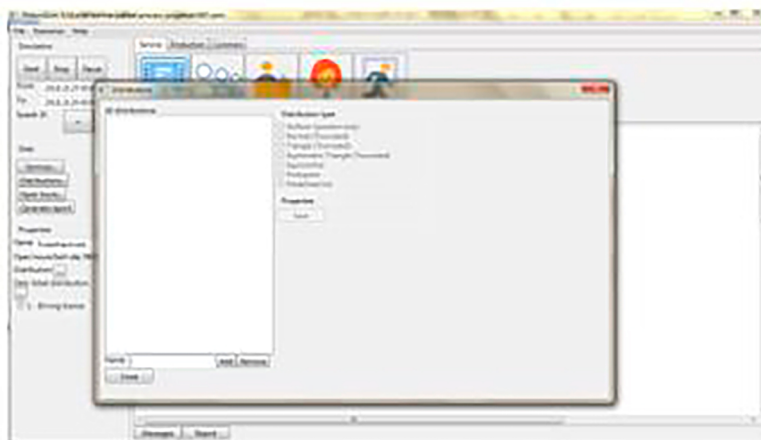
A leírt módon a nyitvatartási időket mind az öt jegykiadó automata esetén egyesével kell definiálnunk és hozzárendelnünk ahhoz, hogy a szimulációs szoftver működése zavartalan legyen.

Miután a jegykiadó automaták beállításával végeztünk, ki kell választanunk az eloszlásokat, amiket az adott jegykiadó objektumok használni fognak majd a kiválasztott szolgáltatást igénybe vevő állampolgárok generálására. Ahogyan azt már az első példa esetén is láthattuk, a szimulációs szoftverben számos előre beépített eloszlási függvény közül választhatunk, amelyeket tetszés szerint változtathatunk is, vagy a függvény egérrel történő húzásával történő módosításával, vagy akár konkrét egyedi értékekkel is szabhatjuk őket testre.

Az adott objektumhoz kétféle módon definiálhatunk eloszlási értékeket, vagy a kezelőfelület data (adatok) szakaszán kattintunk rá a distributions (eloszlás) gombra, vagy a properties (tulajdonságok) szakaszon választjuk ki szintén a distributions gombot. Ez utóbbit csak akkor tehetjük meg, ha a jegykiadó automata (ticket machine) ki van jelölve a szimulációs felületen, mivel a properties szakaszon csak így jeleníti meg a szoftver az adott objektumhoz tartozó egyedi adatokat és tulajdonságokat.

A distributions gombra való kattintás után egy új ablakfelület jelenik meg, amely két fő részből áll: egy jobboldali üres fehér oszlopból (amennyiben még nem definiáltunk korábban a szimulációhoz kötődő eloszlásokat) és egy jobb oldali listából, amely felsorolja a lehetséges választható eloszlástípusokat. Ez utóbbi mindaddig szürke marad, amíg egy egyedi névvel ellátott eloszlástípust nem hozunk létre a felületen. Ehhez nem kell mást tennünk, mint az ablak bal alsó sarkában található name (név) mezőbe beírni egy általunk tetszőlegesen választott vagy kitalált nevet, majd ezt követően rá kell kattintanunk az add (hozzáadás) gombra, amely által az általunk generált eloszlás elnevezése megjelenik majd a bal oldali fehér oszlopban.

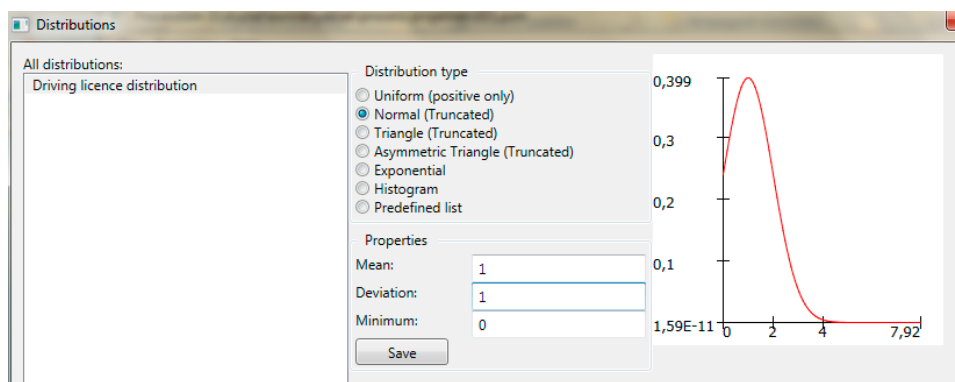
Az összetett példánk esetén öt különböző szolgáltatáshoz kell eloszlást definiálnunk, melyek közül az első a jogosítvány igénylésének gyakoriságát leíró elnevezés driving licence distribution (jogosítvány eloszlás) lesz. Az elnevezést és a hozzáadást követően az elnevezés megjelenik a listában, majd erre rákattintva már választhatóvá válnak az egyes eloszlástípusok, és így mi magunk választhatjuk ki, hogy milyen eloszlást szeretnénk hozzárendelni a jegykiadó automatánkhoz (39. és 40. ábra).



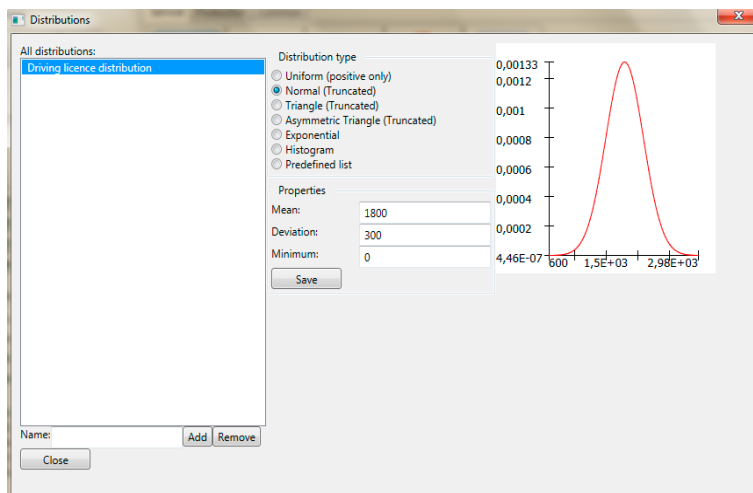
39. sz. ábra: egyedi eloszlás definiálása a driving licence (jogosítvány igénylése) szolgáltatást generáló jegykiadó automata (ticket machine) számára (1)

Ebben a példában mindegyik szolgáltatás esetén az előre definiált normális eloszlást fogjuk kiválasztani az általunk generált eloszlás típusának, egyedi értékeknek pedig be fogjuk állítani, hogy átlagosan (mean) hány percenként érkezzen egy adott szolgáltatást igénybe vevő ügyfél, és ennek az időnek mekkora legyen a szórása (deviation).

A jogosítványigénylés (driving licence) szolgáltatás esetén az előző példában is alkalmazott módon beállítjuk, hogy harminc percenként érkezzen egy ügyfél, öt perces szórással. A szoftverben az időértékeket másodpercben kell megadni, tehát a harminc perc ezernyolcszáz másodpercnek felel meg, míg az öt perc háromszáz másodpercnek. A megfelelő időértékek megadása után a save (mentés) gomb lenyomásával véglegesítjük a műveletet. Az adatok megadása után a felület jobb oldalán található grafikonon is azonnal nyomon követhető a függvény görbéjének és elhelyezkedésének a megváltozása (40. és 41. ábra).



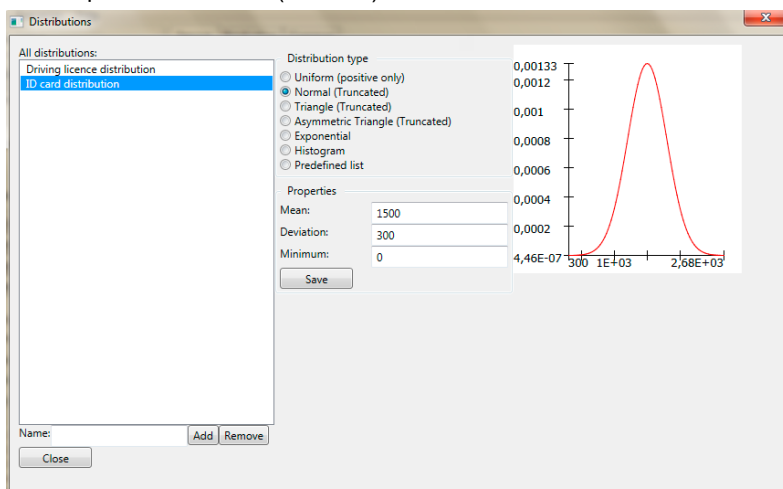
40. sz. ábra: egyedi eloszlás definiálása a driving licence (jogosítvány igénylése) szolgáltatást generáló jegykiadó automata (ticket machine) számára (2)



41. sz. ábra: egyedi eloszlás definiálása a driving licence (jogosítvány igénylése) szolgáltatást generáló jegykiadó automata (ticket machine) számára (3)

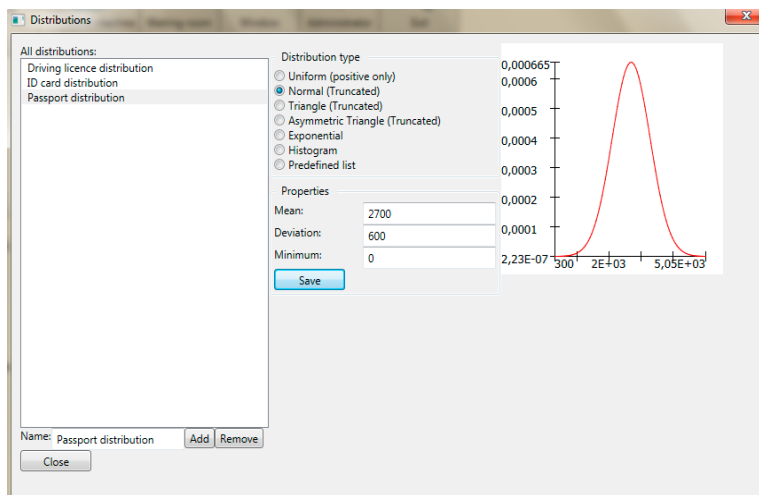
Az egyes eloszlástípusok főbb jellemzőiről és a beállítások további módjairól részletes leírás található a szimulációs szoftver felhasználói útmutatójában.

A jogosítvány igénylés szolgáltatás eloszlásának definiálása után második lépésként definiáljuk a személyi igazolvány (ID card) igénylése szolgáltatáshoz kapcsolódó érkező ügyfelek eloszlását a második jegykiadó automata számára. Létrehozuk az ID card distribution eloszlást, majd pedig ehhez az eloszláshoz beállítjuk, hogy átlagosan huszonöt percnként érkezen egy ügyfél öt perces szórással. Ez másodpercekben kifejezve azt fogja jelenteni, hogy ezeröt száz másodperces átlagot kell beállítanunk a normális eloszlás értékének, háromszáz másodperces szórással (42. ábra).



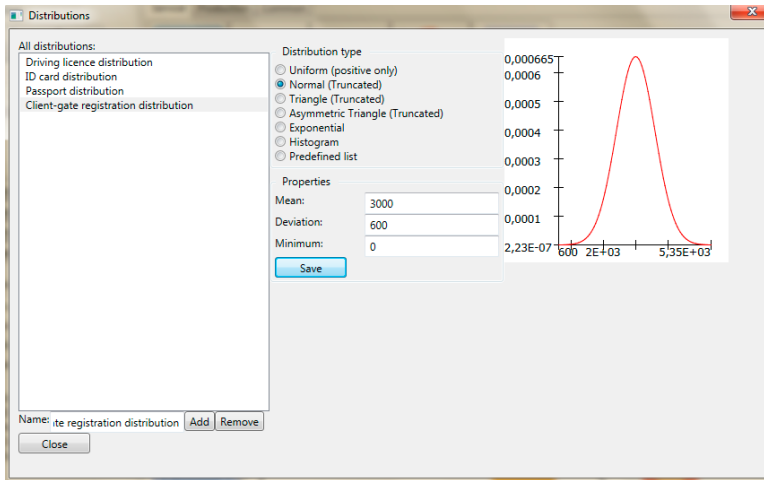
42. sz. ábra: egyedi eloszlás definiálása az ID card (személyi igazolvány igénylése) szolgáltatást generáló jegykiadó automata (ticket machine) számára

A személyi igazolvány igénylés szolgáltatás eloszlásának definiálása után harmadik lépésként definiáljuk az útlevel igénylése (passport) szolgáltatáshoz kapcsolódó érkező ügyfelek eloszlását a harmadik jegykiadó automata számára. Létrehozunk a passport distribution eloszlást, majd pedig ehhez az eloszláshoz beállítjuk, hogy átlagosan negyvenöt percenként érkezzen egy ügyfél, tíz perces szórással. Ez másodpercekben kifejezve azt fogja jelenteni, hogy kétezer hétszáz másodperces átlagot kell beállítanunk a normális eloszlás értékének, hatszáz másodperces szórással (43. ábra).



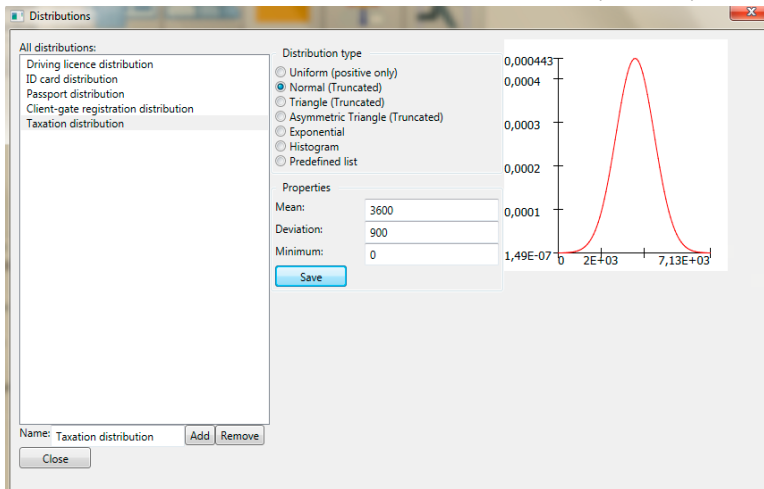
43. sz. ábra: egyedi eloszlás definiálása a passport (útlevel igénylése) szolgáltatást generáló jegykiadó automata (ticket machine) számára

Az útlevel igénylés szolgáltatás eloszlásának definiálása után negyedik lépésként definiáljuk az ügyfélkapu regisztráció (client-gate registration) szolgáltatáshoz kapcsolódó érkező ügyfelek eloszlását a negyedik jegykiadó automata számára. Létrehozunk a client-gate registration distribution eloszlást, majd pedig ehhez az eloszláshoz beállítjuk, hogy átlagosan ötven percenként érkezzen egy ügyfél, tíz perces szórással. Ez másodpercekben kifejezve azt fogja jelenteni, hogy háromezer másodperces átlagot kell beállítanunk a normális eloszlás értékének, hatszáz másodperces szórással (44. ábra).



44. sz. ábra: egyedi eloszlás definiálása a client-gate registration (ügyfélkapu regisztráció) szolgáltatást generáló jegykiadó automata (ticket machine) számára

Az ügyfélkapu regisztráció szolgáltatás eloszlásának definiálása után ötödik lépésként definiáljuk az adóügyek intézése (taxation) szolgáltatáshoz kapcsolódó érkező ügyfelek eloszlását az ötödik jegykiadó automata számára. Létrehozuk a taxation distribution eloszlást, majd pedig ehhez az eloszláshoz beállítjuk, hogy átlagosan hatvan percnként, azaz óránként érkezzen egy ügyfél tizenöt perces szórással. Ez másodpercekben kifejezve azt fogja jelenteni, hogy háromezer hatszáz másodperces átlagot kell beállítanunk a normális eloszlás értékének, hatszáz kilencszáz másodperces szórással (45. ábra).



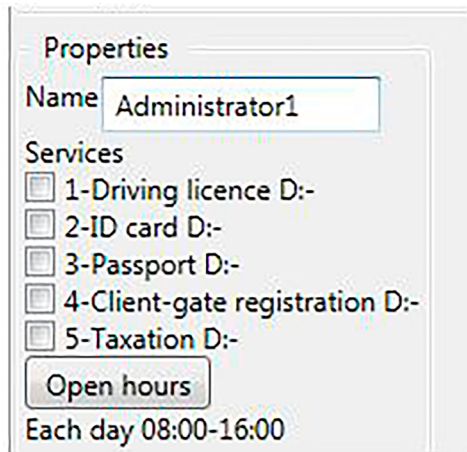
45. sz. ábra: egyedi eloszlás definiálása a taxation (adóügyek intézése) szolgáltatást generáló jegykiadó automata (ticket machine) számára

Az öt különböző szolgáltatás eloszlásainak beállítása és jegykiadó automatákhoz való hozzárendelése után ahhoz, hogy a szimulációt futtathassuk, szükséges, hogy definiáljuk az ügyintézők számára, hogy az egyes kormányablakokban milyen szolgáltatások esetén tudnak segítséget nyújtani, és hogy ezeket a szolgáltatásokat milyen eloszlással jellemezhető módon tudják végrehajtani.

Ebben a példában azt feltételezzük, hogy az egyes típusú szolgáltatások kezelését mindegyik ügyintéző hasonló értékekkel jellemezhető módon tudja elvégezni, azaz személytől független az ügyintézés hossza. Összetettebb példák esetén természetesen ezeken is változtathatunk, hogy még tovább finomítsuk a szimulációs modellt, azonban jelen esetben a szolgáltatásokra típusonként csak egyféle ügyintézési eloszlást fogunk definiálni.

A szimulációs szoftver futtatásához az egyes ügyintézők definiált ügyintézési eloszláson túl az is elengedhetetlenül szükséges, hogy kijelöljük, hogy az adott ügyintézők milyen típusú szolgáltatást vagy szolgáltatásokat tudnak kezelni a kormányablakokban, hiszen ebből tudják majd a váróteremben várakozó ügyfelek eldönteni, hogy melyik ablakhoz menjenek tovább.

Az ügyintéző objektumok alapbeállításainak első lépéseként tehát az első példához hasonló módon jelöljük ki az első ügyintéző objektumot a szimulációs felületen. A kijelölést követően láthatóvá válnak a bal oldali kezelőfelület properties (tulajdonságok) szakaszán az adott objektum egyedi értékei, ezek között pedig felsorolva láthatjuk az összes olyan szolgáltatás elnevezését, amelyet definiáltunk már korábban. Jelen esetben tehát látható az általunk a rendszerbe bevitt öt különböző szolgáltatás, amelyek egyesével kiválaszthatók (46. ábra).



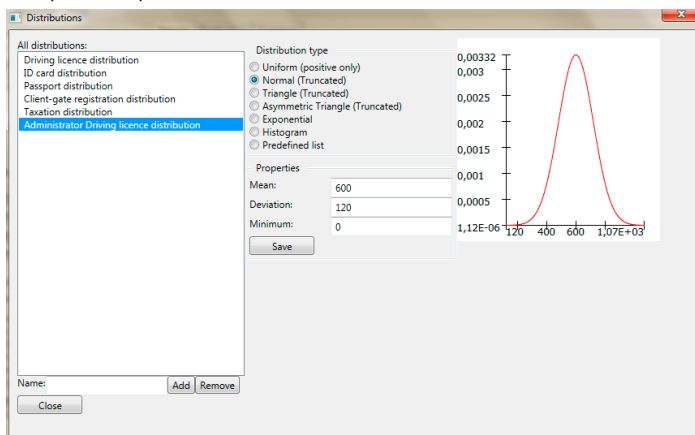
46. sz. ábra: az adminisztrátorok számára kiválasztható szolgáltatások listája a properties (tulajdonságok) menüpontban

Az első adminisztrátorhoz három szolgáltatást fogunk hozzárendelni, a jogosítvány igénylését (driving licence), a személyi igazolvány igénylését (ID card) és az útlevél igénylését (passport). Ehhez a szolgáltatások neve előtti checkboxokba pipát kell tennünk. Ekkor azonnal felugrik az eloszlások definiálására szolgáló ismerős ablak, ahol láthatjuk a már korábban általunk definiált öt szolgáltatás eloszlásait.

Az ügyintézők esetében tehát külön-külön is definiálható, hogy az egyes szolgáltatásokat milyen típusú eloszlásfüggvénnyel jellemezhető módon képesek kiszolgálni. Ez alapulhat valós mért adatokon, vagy ismételten használhatunk előre definiált formulákat is. Ebben a példában az ügyintézők számára is egy normális típusú eloszlást fogunk generálni, azonban ennek az eloszlásnak mások lesznek az átlag és szórás értékei.

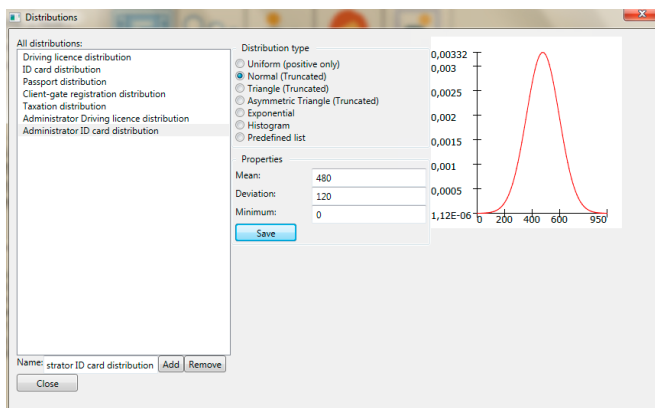
Elsőként a felugró ablak alsó részén ismét megadunk egy általunk kiválasztott elnevezést, ami jelen esetben administrator driving licence distribution (ügyintéző jogosítvánnyal kapcsolatos ügyintézési idejének eloszlása) nevet fogja kapni. A driving licence szolgáltatás ügyintézési idejének eloszlása normális típusú eloszlás lesz, az átlagnak (mean) tíz percet fogunk megadni, míg a szórása (deviation) kettő perc lesz, tehát az ügyintéző átlagosan tíz perc alatt tudja majd ezt a szolgáltatást végrehajtani, kettő perces szórással (47. ábra).

Nem szabad elfelejtenünk, hogy az időadatokat ismételten másodpercekben kell definiálnunk, tehát a tíz perc hatszáz másodpercnek felel meg, a két perc pedig százhusz másodpercnek felel meg. Az adatok megadása után pedig a felület jobb oldalán található grafikonon is ismét azonnal nyomon követhető a függvény görbéjének és elhelyezkedésének a megváltozása (47. ábra).



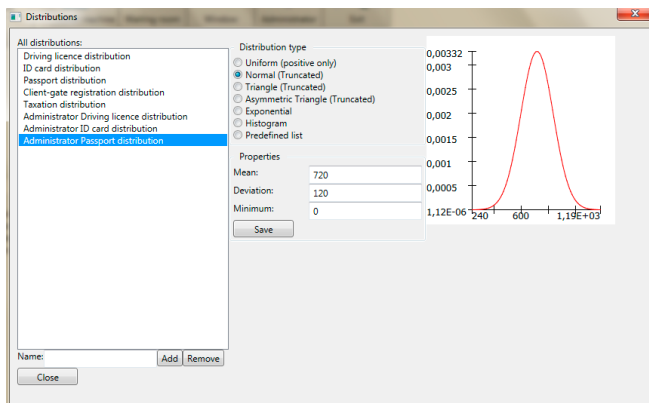
47. sz. ábra: egyedi eloszlás definiálása a jogosítvány igénylésével kapcsolatos ügyintézési időre (administrator driving licence distribution) az ügyintézők számára

Ezután az előzőekben leírt módon a személyi igazolvány igénylés szolgáltatás ügyintézését is hozzárendeljük az első ügyintézőhöz, majd az eloszlások megadására szolgáló ablak felugrása után definiáljuk az administrator ID card distribution eloszlást (ügyintéző személyi igazolvánnyal kapcsolatos ügyintézési idejének eloszlása). Ez az eloszlás is normális típusú eloszlás lesz, az átlagnak (mean) tíz nyolc percet fogunk megadni, míg a szórása (deviation) kettő perc lesz, tehát az ügyintéző átlagosan nyolc perc alatt tudja majd ezt a szolgáltatást végrehajtani, kettő perces szórással. Másodpercekben kifejezve ez négyszáznyolcvan másodperces átlagot és százhusz másodperces szórást jelent (48. ábra).



48. sz. ábra: egyedi eloszlás definiálása a személyi igazolvány igénylésével kapcsolatos ügyintézési időre (administrator ID card distribution) az ügyintézők számára

Következő lépésként az útlevél igénylés szolgáltatás ügyintézését is hozzárendeljük az első ügyintézőhöz, majd az eloszlások megadására szolgáló ablak felugrása után definiáljuk az administrator passport distribution eloszlást (ügyintéző útlevéllel kapcsolatos ügyintézési idejének eloszlása). Ez az eloszlás is normális típusú eloszlás lesz, az átlagnak (mean) tizenkét nyolc percet fogunk megadni, míg a szórása (deviation) kettő perc lesz, tehát az ügyintéző átlagosan tizenkét perc alatt tudja majd ezt a szolgáltatást végrehajtani kettő perces szórással. Másodpercekben kifejezve ez hétszázhusz másodperces átlagot és százhusz másodperces szórást jelent (49. ábra).



49. sz. ábra: egyedi eloszlás definiálása az útlevél igénylésével kapcsolatos ügyintézési időre (administrator passport distribution) az ügyintézők számára

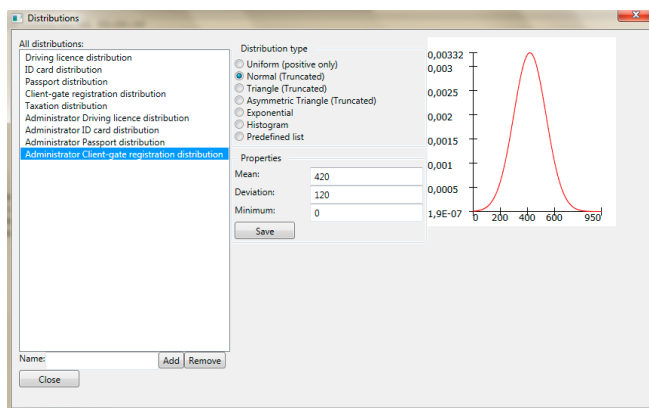
Az első ügyintéző tehát ezt a három szolgáltatást tudja majd intézni, így a váróteremből csak ezeket a szolgáltatásokat igénybe venni szándékozó ügyfelek mehetnek majd az általa birtokolt kormányablakhoz.

A második ügyintéző számára ugyanezt a három szolgáltatást fogjuk beállítani, a korábban leírt lépéseket követve, tehát hozzárendeljük a driving licence (jogosítvány igénylése), az ID card (személyi igazolvány igénylése) és a passport (útlevél igénylése) szolgáltatásokat. Mivel ezekhez a szolgáltatásokhoz már fentebb definiáltuk az ügyintézési időkre vonatkozó eloszlásokat, így a második ügyintéző esetében nincs más dolgunk, mint újból kijelölni ezt a három szolgáltatást a properties (tulajdonságok) menüpont alatt és egyszerű kijelöléssel hozzájuk rendelni a definiált eloszlások közül a megfelelőket az eloszlási ablak felugrása után.

A harmadik ügyintéző az első kettőtől abban fog eltérni, hogy az ő esetében az objektumhoz hozzárendeljük mind az öt igénybe vehető szolgáltatást. Tehát a harmadik ügyintéző a driving licence (jogosítvány igénylése), az ID card (személyi igazolvány igénylése), a passport (útlevél igénylése) szolgáltatásokon túl még a client-gate registration (ügyfélkapu regisztráció) és a taxation (adóügyek intézése) szolgáltatások igénybevétele esetén is tud segíteni a kormányablakba betérő állampolgároknak.

Először, akárcsak a második ügyintéző esetében, a harmadik ügyintéző számára is hozzárendeljük az első három szolgáltatást és a felugró ablakban kiválasztjuk a hozzájuk tartozó ügyintézési idők már létező eloszlásait.

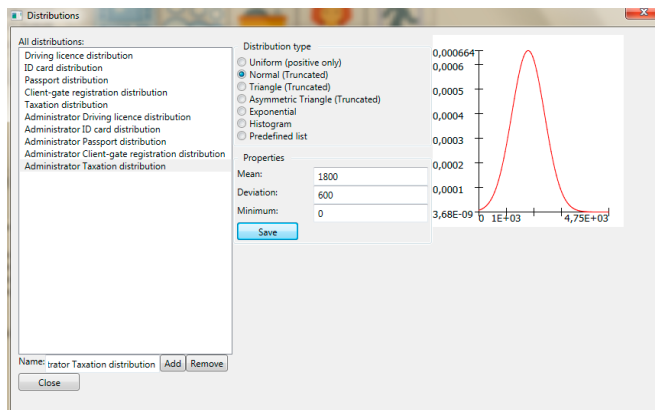
Következő lépésként az ügyfélkapu-regisztráció szolgáltatás ügyintézését is hozzárendeljük a harmadik ügyintézőhöz, majd az eloszlások megadására szolgáló ablak felugrása után definiáljuk az administrator client-gate registration distribution eloszlást (ügyintéző ügyfélkapu-regisztrációval kapcsolatos ügyintézési idejének eloszlása). Ez az eloszlás is normális típusú eloszlás lesz, az átlagnak (mean) hét percet fogunk megadni, míg a szórása (deviation) kettő perc lesz, tehát az ügyintéző átlagosan hét perc alatt tudja majd ezt a szolgáltatást végrehajtani kettő perces szórással. Másodpercekben kifejezve ez négyszázhusz másodperces átlagot és százhusz másodperces szórást jelent (50. ábra).



50. sz. ábra: egyedi eloszlás definiálása az ügyfélkapu regisztrációval kapcsolatos ügyintézési időre (administrator client-gate registration distribution) az ügyintézők számára

Végül az utolsó lépésként az adóügyek intézése szolgáltatás ügyintézését is hozzárendeljük a harmadik ügyintézőhöz, majd az eloszlások megadására szolgáló ablak felugrása után definiáljuk az administrator taxation distribution eloszlást (adóügyek intézésével kapcsolatos ügyintézési idejének eloszlása). Ez az eloszlás is normális típusú eloszlás lesz, az átlagnak

(mean) harminc percet fogunk megadni, míg a szórása (deviation) tíz perc lesz, tehát az ügyintéző átlagosan harminc perc alatt tudja majd ezt a szolgáltatást végrehajtani tíz perces szórással. Másodpercekben kifejezve ez ezernyolcszáz másodperces átlagot és hatszáz másodperces szórást jelent (51. ábra).



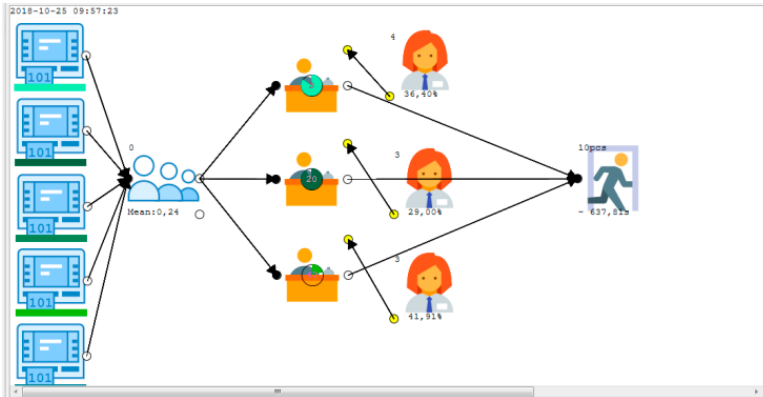
51. sz. ábra: egyedi eloszlás definiálása az adóügyek ügyintézésével kapcsolatos ügyintézési időre (administrator taxation distribution) az ügyintézők számára

Az ügyintézők alapbeállításainak megfelelő elvégzése után nincs más dolgunk, mint elindítani a szimulációs modell futtatását a bal oldali kezelőfelület felső részén található start (indítás) gomb segítségével. Amennyiben minden beállítást az előbb leírtaknak megfelelően végeztünk el mind a jegykiadó automaták, mind pedig az ügyintézők objektumok esetén, úgy a szimulációs modell futásának már gond nélkül el kell indulnia.

Amint az a gomb megnyomása után látható a folyamatszimuláció gond nélkül elindul, a jegykiadó automaták elkezdik generálni az egyes szolgáltatásokat igénybe vevő ügyfeleket, akik ezt követően bekerülnek a váróterembe, ahol mindaddig várakoznak, amíg sorra nem tudnak kerülni a kormányablakok valamelyikében, ahol az ügyintéző segítségével a szolgáltatás lebonyolítása végbe megy, majd a szolgáltatás befejezését követően a kijáraton át távoznak.

12.2.4. *Eredmények értékelése*

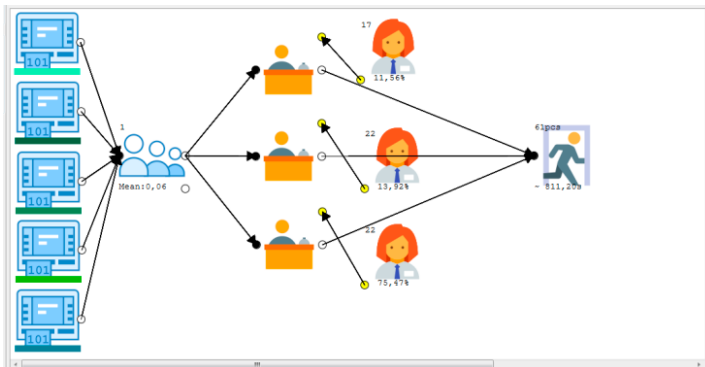
Amint az a szimulációs felületen elhelyezett objektumokon is jól látható, a szimulációs folyamat elindítása után az egyes objektumok állapotában változás áll be. Amint az alábbi ábrán is látható, az adott pillanatban mind a három ügyintéző elfoglalt, az első már négy ügyet intézett el és az ötödik van folyamatban, a második három ügyet intézett el és a negyedik van folyamatban, míg a harmadik szintén három ügyet intézett el és a negyedik van folyamatban, összesen pedig már tíz állampolgár felmerülő igényét elégítették ki. Amint az jól látható a több hozzárendelt szolgáltatástípus esetén az egyes típusok eltérő színnel jelennek meg, ami vizuálisan is megkönnyíti a modellen látható információk leolvasását. Ezzel kapcsolatban a szoftver felhasználói útmutatójában részletesebb tájékoztatás is található (52. ábra).



52. sz. ábra: a szimuláció futtatása közben leolvasható információk az egyes objektumok esetén

Az ügyintéző objektumáról szintén leolvashatók a valós idejű kihasználtság adatok, az objektum alsó részén látható a százalékos érték, amely az adott ügyintéző kihasználtságáról nyújt tájékoztatást. Az előző objektumokhoz hasonlóan a kijárat objektumról is olvashatunk le információt, a bal felső sarokban láthatjuk a távozott ügyfelek számát, míg az objektum alján látható számláló az átlagos átfutási idő mértékéről nyújt valós idejű tájékoztatást (52. ábra).

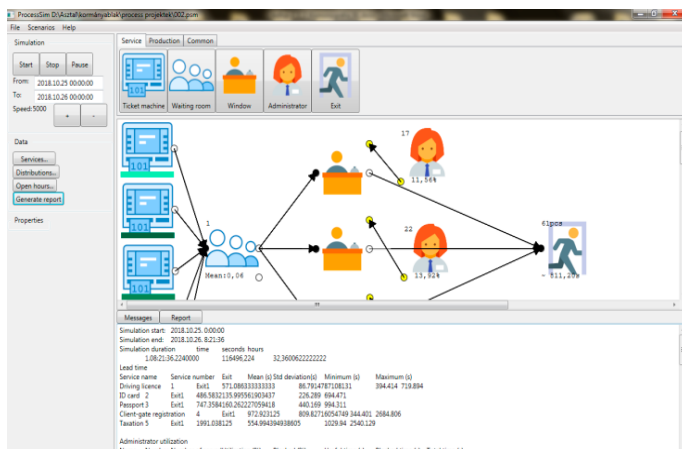
A szimulációs modell hosszabb ideig történő futtatása esetén jól látható, hogy miként alakultak az egyes objektumok értékei, a szimuláció leállítása után pedig elemezhetjük az addig végbement folyamatok adatait részletesen. Amint az ábrán is jól látható, a szimuláció leállításának pillanatáig az első ügyintéző tizenhét beérkezett ügyfelet szolgált ki, a második ügyintéző huszonkettő ügyfelet szolgált ki, a harmadik ügyintéző pedig szintén huszonkettő ügyfelet szolgált ki, összesen tehát hatvanegy ügyfelet szolgáltak ki a nyitvatartási idő alatt, míg a váróteremben egy ügyfél maradt várakozóként, akire a nyitvatartási idő lejártáig már nem került sor. A szimulációs felületről az is leolvasható, hogy az átlagos átfutási idő nyolcszázötvennégy másodperc volt, ami tizenhárom és fél percnél felel meg attól a pillanattól fogva, hogy az állampolgár belépett az ajtón egészen addig a pillanatig, amíg a szolgáltatás befejeztével távozott (53. ábra).



53. sz. ábra: a szimuláció futtatása után leolvasható információk az egyes objektumok esetén

A leállítás történhet kézzel a megfelelő gombok segítségével a kezelőfelületen, vagy ugyanitt megadhatunk egy előre definiált időintervallumot is, amin belül futtatni szeretnénk az adott szimulációs modellt. Ebben az esetben, ahogyan az első példa esetében is, az előre definiált időintervallumot választottuk, a szimuláció egy egész szimulált nap folyamatait vette végig a kormányablak kinyitásától egészen a definiált munkaidő hosszának lejártáig.

A végrehajtott szimulációs folyamat eredményeinek részletes megtekintéséhez az adatokról egy jelentést kell generálnunk, amit úgy tehetünk meg, ha a bal oldali kezelőfelületen rákattintunk a generate report (jelentés generálása) gombra. Ezt követően a szimulációs felület alatt elhelyezkedő üzenőpanel ismét megjelenik és láthatóvá válik rajta a report fül alatt a részletes jelentés minden adata (54. és 55. ábra).



54. sz. ábra: jelentés generálása az elvégzett szimuláció részletes adatainak áttekintéséhez és további elemzéséhez (1)

Service name	Service number	Exit	Mean (s)	Std deviation (s)	Minimum (s)	Maximum (s)
Driving licence	1	Exit1	571,088333333333	86,7944787108131	394,454	719,894
ID card	2	Exit1	488,5832125,995561903437	226,289	694,471	
Passport	3	Exit1	747,3584160,262277959418	440,169	994,311	
Client-gate registration	4	Exit1	972,902125	908,82716054749	344,401	2684,806
Taxation	5	Exit1	1991,038125	554,994394938605	1029,94	2540,129

55. sz. ábra: jelentés generálása az elvégzett szimuláció részletes adatainak áttekintéséhez és további elemzéséhez (2)

Ezeket az adatokat a könnyebb áttekinthetőség és a további elemzhetőség érdekében közvetlenül átmásolhatjuk egy táblázatkezelő program felületére úgy, hogy az adatok kijelölése után azokat másoljuk, majd pedig a táblázatkezelő program megfelelő gombja segítségével beillesztjük. Ezt követően az adott példa esetén az alábbi táblázatba rendezett eredményeket kapjuk (56. ábra).

Lead time						
Service name	Service number	Exit	Mean (s)	Std deviation(s)	Minimum (s)	Maximum (s)
Driving licence	1	Exit1	571.0863333333333	86.7914787108131	394.414	719.894
ID card	2	Exit1	486.5832	135.995561903437	226.289	694.471
Passport	3	Exit1	747.3584	160.262227059418	440.169	994.311
Client-gate registration	4	Exit1	972.923125	809.82716054749	344.401	2684.806
Taxation	5	Exit1	1991.038125	554.994394938605	1029.94	2540.129

56. sz. ábra: a szimuláció futásának részletes eredményei a táblázatkezelőben (1). Az egyes szolgáltatásokra vonatkozó összesített adatok (átfutási idő átlagos hossza és szórása, minimum és maximum értékek)

Az így kapott jelentés alkalmas arra, hogy kiolvassuk belőle az egyes szolgáltatásokra vonatkozó átfutási időket, átlagokat, szórásokat, az ügyintézők kihasználtságát, az egyes szolgáltatások igénybe vevőinek számát, a sorban állási idők átlagát és szórását és még sok minden mást (55. ábra). A jelentés tartalmáról és lehetséges további felhasználásáról az olvasó bővebb információkat találhat a szoftver működését részletesen tartalmazó felhasználói útmutatóban.

Az így kapott jelentésből az adott példa esetén megállapítható, hogy összesen hatvanegy állampolgár vette igénybe a kormányablak által kínált szolgáltatásokat az adott napon, a jelentés további adatai pedig részletes információval szolgálnak arról, hogy az egyes ügyintézők a hozzájuk rendelt szolgáltatástípusok esetén külön-külön hány ügyfelet szolgált ki a futási idő során, mekkora volt a kihasználtságuk, és hogy milyen átfutási idővel dolgoztak (57. ábra).

Administrator utilization						
Name	Number of served	Utilization (%)	Blocked (%)	Useful time (s)	Blocked time (s)	Total time (s)
Administrator1	17	11.5555351619244	0	10133.768	0	87696.224
Administrator2	22	13.9196768608874	0	12207.031	0	87696.224
Administrator3	22	75.46686111111111	0	21734.456	0	28800

57. sz. ábra: a szimuláció futásának részletes eredményei a táblázatkezelőben (2). Az adminisztrátorokra vonatkozó összesített adatok (kiszolgált ügyek száma, ügyintézési idő átlagos hossza és szórása, minimum és maximum értékek)

Amint az a táblázat adataiból kiolvasható, az első ügyintéző, aki három szolgáltatás (driving licence, ID card, passport) esetén volt illetékes, összesen tizenhét ügyfelet szolgált ki a szimuláció során, a kihasználtsága pedig 11,56 százalékos volt. A második ügyintéző, aki szintén három szolgáltatás esetén volt illetékes (driving licence, ID card, passport) összesen huszonkettő ügyfelet szolgált ki a szimuláció során, a kihasználtsága pedig 13,92 százalékos volt.

A harmadik ügyintéző, aki mind az öt szolgáltatás esetén illetékes volt (driving licence, ID card, passport, client-gate registration, taxation) szintén huszonkettő ügyfelet szolgált ki a szimuláció során, a kihasználtsága pedig 75,47 százalékos volt.

Az ügyintézők kihasználtságadataiból jól látszik, hogy a harmadik ügyintéző volt a legfoglaltabb, hiszen ő a szimulációs során hetvenöt százalékában dolgozott, holott ugyanúgy csak huszonkettő ügyfele volt, mint a második ügyintézőnek, akinek csupán 13,92 százalékos volt a kihasználtsága ennek dacára. Ennek okaira az ügyintézők részletes ügyintézési adatai között találunk magyarázatot, amelyet a kimutatás további részei tárnak fel előttünk (58. ábra).

Name	Service name	Service number	Total count	Mean processing time (s)	Std deviation (s)	Minimum (s)	Maximum(s)
Administrator1	Driving licence	1	8	598.71225	77.1177171134216	467.138	719.894
Administrator2	Driving licence	1	5	555.0872	104.069645174277	394.414	673.308
Administrator3	Driving licence	1	2	500.5805	56.3839876250341	460.711	540.45
Administrator1	ID card	2	6	554.4958333333333	84.2768950885512	424.229	650.619
Administrator2	ID card	2	10	425.9222	139.530231310319	226.289	643.959
Administrator3	ID card	2	4	504.201	138.04626247989	366.714	694.471
Administrator1	Passport	3	3	672.365	201.88036294053	440.169	806.336
Administrator2	Passport	3	7	738.910428571429	110.051802112849	616.063	938.68
Administrator3	Client-gate registration	4	8	431.802375	108.296077529465	301.9	665.51
Administrator3	Taxation	5	8	1907.759	500.202779737235	1029.94	2444.307

58. sz. ábra: a szimuláció futásának részletes eredményei a táblázatkezelőben (3). Az adminisztrátorokra vonatkozó részletes adatok (kiszolgált ügyek száma szolgáltatásonként, ügyintézési idő átlagos hossza és szórása szolgáltatásonként, minimum és maximum értékek)

Ahogy az adatokból leolvasható, driving licence szolgáltatásból a legtöbbet (nyolc ügy) az első ügyintéző intézett, ID card szolgáltatásból pedig a legtöbbet (tíz ügy) a második adminisztrátor intézett. A harmadik adminisztrátor mind a driving licence, mind az ID card mind pedig a passport szolgáltatások esetén a legkevesebb ügyet kapta, viszont ő volt egyedül képesítve arra, hogy client-gate registration és taxation szolgáltatásokat igénybe vevő ügyfelekkel is foglalkozzon, és noha a client-gate registration a legrövidebb átfutási idővel igénybe vehető szolgáltatás, azonban a taxation szolgáltatás ezzel szemben pedig a leghosszabb átfutási idővel rendelkező szolgáltatás a szimulációban definiált szolgáltatások közül. Noha a harmadik ügyintéző ugyanannyi ügyet intézett el szám szerint, mint a második ügyintéző, azonban ezek az ügyek nagyságrendileg sokkal több időt vettek igénybe, kiváltképp a taxation szolgáltatást igénybe vevő ügyfelek miatt (59. ábra).

Name	Driving licence	ID card	Passport	Client-gate registration	Taxation	Total
Administrator1	8	6	3	0	0	17
Administrator2	5	10	7	0	0	22
Administrator3	2	4	0	8	8	22
						61

59. sz. ábra: a szimuláció futásának részletes eredményei a táblázatkezelőben (4). Az Adminisztrátorokra vonatkozó részletes adatok szolgáltatástípusonként csoportosítva

Ahogy az a részletes adatok további elemzéséből is jól látszik, a harmadik ügyintéző esetén a taxation típusú szolgáltatások ügyintézési idejének hossza a várakozási időkre is befolyással volt. Az első három szolgáltatás, tehát a driving licence, az ID card és a passport esetén az ügyfelek várakozási idői rövidek voltak, hiszen ők az első két ügyintéző valamelyikéhez is kerülhettek, viszont az az állampolgár, aki client-gate registration vagy taxation szolgáltatás igénybevételének ügyében tért be a kormányablakba, hosszabb várakozási idővel kellett számolnia (60. ábra).

Name	Service name	Service number	Mean length	Mean waiting time (s)	Std dev waiting time (s)	Minimum waiting time (s)	Maximum waiting time (s)
WaitingRoom1	Driving licence	1	0	0	0	0	0
WaitingRoom1	ID card	2	0.00148723739980324	6.43315	28.7699214189055	0	128.663
WaitingRoom1	Passport	3	0.00334359291345561	28.4116	89.8453679696399	0	284.116
WaitingRoom1	Client-gate registration	4	0.0519035124230396	541.12075	786.139699992992	0	2269.259
WaitingRoom1	Taxation	5	0.00779381775676445	83.279125	130.731266726209	0	348.446

60. sz. ábra: a szimuláció futásának részletes eredményei a táblázatkezelőben (4). Az egyes szolgáltatásokat igénybe vevők várakozási ideje a Váróteremben kategóriánként (várakozási idő átlagos hossza és szórása, minimum és maximum értékek)

Így fordulhatott elő például, hogy az alapvetően a legrövidebb idő alatt elintézendő ügyfélkapu-regisztrációt igénybe venni szándékozó embereknek volt kimagaslóan a leghosszabb a várószobában töltött átlagos várakozási idejük, mivel ha már volt előttük egy taxation típusú szolgáltatást igénybe vevő állampolgár, akinek az ügyét éppen intézték, az jóval tovább tartott bármely más szolgáltatás ügyintézési idejénél, és a client-gate regisztrációra váró ügyfeleknek nem volt lehetőségük rá, hogy egy másik ablakhoz kerüljenek át (60. ábra).

13. ÖSSZEFOGLALÁS

Amint az látható, ez a szoftver bár egyszerűsége törekszik, mégis rendkívül sokoldalúan hasznosítható a kormányablakokban lezajló folyamatok szimulációjának terén. A bemutató példák csupán az alapokat kívánják felvázolni az olvasó számára, hogy kellő részletességgel megismerkedhessen a felhasználói funkciók tárházával, az alapvető működési elvvel, és az egyes elemek egymáshoz való viszonyának fontosságával, valamint, hogy átláthatóvá váljon a kapott eredmények alapvető értékelésének módja.

Amint az a szimuláció elvének rövid leírásából is kiderül, ezzel a módszerrel lényegesen egyszerűbben, gyorsabban és kevesebb erőforrás felhasználásával vizsgálhatjuk egy rendszer működését virtuális környezetben, anélkül, hogy a valódi rendszer működésében zavart okoznánk. A kötetben leírt két példa bemutató céllal készült, hogy megismertesse a felhasználókat a szoftver kezelésének módjával és alapvető funkciókkal, rávilágítva a lehetőségekre, amelyek a használatát rendkívül előnyössé teszik. A szoftver szolgáltatási folyamatok és gyártási folyamatok szimulálására is egyaránt alkalmas, melyek közül a szolgáltatási oldal került kiemelésre a kormányablakok működésével összefüggésben. A két bemutató példa egy egyszerű és egy összetettebb szimulációs modell felépítését mutatja be lépésről lépésre, hogy a felhasználó kellő mértékű bepillantást nyerhessen a szoftver kezelésének mibenlétébe, és ezáltal a példák tanulmányozása közben elsajátíthassa annak használatát.

A szerzők reményei szerint a szoftver működési leírása és a bemutató példák magyarázata kellő részletességű, és könnyen követhető, ezáltal a szoftver felhasználása az alapvető irányítási mechanizmusok elsajátítása után már nem fog gondot okozni senkinek sem, aki egy adott szolgáltatási folyamatot, vagy esetleg egy gyártósor működését szeretné modellezni a szimulációs szoftver segítségével.

IRODALOMJEGYZÉK

BANKS, Jerry – CARLSON, John S. – NELSON, Barry L. – NICOL, David M. (2013): *Discrete-Event System Simulation*. Pearson New International Edition (5th Edition), Pearson, USA.

BOHÁCS Gábor (2012): *Folyamatelemzés és szimuláció*. Járműipari Felsőoktatási és Kutatási Együttműködés, Európai Unió, Európai Szociális Alap.

GARRIDO, José M. (2009): *Object Oriented Simulation: A Modeling and Programming Perspective*. pp. 1–449. doi: 10.1007/978-1-4419-0516-1.

KELTON, W. David – SADOWSKI, Randall P. – ZUPICK, Nancy B. (2015): *Simulation with Arena*. McGrawHill, New York.

KOVÁCS Éva (szerk.) (2014): *Közigazgatási egyedi ügyek és gyakran előforduló élethelyzetek – Esettanulmány gyűjtemény a kormányablak ügyintéző képzés 4. egyedi ügyek és élethelyzetek című moduljához*. Budapest.

KOVÁCS Éva (2016): *Kormányzati koordináció Magyarországon: Esettanulmány az egyablakos ügyfélszolgálati reformról*.

KOVÁCS Zoltán (2010): *Szimulációs eszközök és megoldások műszaki és gazdasági rendszerekben*. „Innováció az egyetemi képzésben és kutatásban” Jubileumi tudományos konferencia, Balatonvilágos, 2010. augusztus 27–29.

KOVÁCS Zoltán – KOVÁCS Viktor (2015a): *Számítógépes karbantartási játék*. Veszprém, In: Élenjáró karbantartás, mint a vállalkozásirányítás sikerének előfeltétele, nemzetközi konferencia kiadványa, (szerk.) Szentes Balázs. 2015. április 16–17. Pannon Egyetem, p. 181–188.

KOVÁCS Zoltán – KOVÁCS Viktor (2015b): *Egy univerzális játékplatform*. „Mérleg és Kihívások” Nemzetközi Tudományos Konferencia. 2015. október 15–16. Miskolc–Lillafüred.

MARTIJN R. K. Mes (2017): *Simulation Modelling using Practical Examples: A Plant Simulation Tutorial*. University of Twente.

NAVRACSICS Tibor (2013): *A tisztességes ügyintézéshez való alapjog – ügyintézés a járási hivatalokban*. ACTA HUMANA • 2013/1. 7–33.

PRATEEK Sharma (2015): *Discrete-Event Simulation*. International Journal Of Scientific & Technology Research, Volume 4., Issue 04.

POKORÁDI László (2008): *Rendszerek és folyamatok modellezése*. Campus Kiadó, Debrecen.

TEMESI István – LINDER Viktória (2015): *Közigazgatási szakvizsga jegyzet*. Nemzeti Közszolgálati Egyetem, Budapest.

VIRÁG Rudolf (2012): *Az államigazgatási feladat- és hatáskör-telepítés új rendszere – a járási rendszer kialakítása*. Magyar Közigazgatás, 2. Új folyam, 1. sz.

VIRÁG Rudolf (2014): *A területi államigazgatás integrációja*. Magyar Napok 2014: Megújítás és megújulás. NKE Szolgáltató Kft., Budapest, p. 30.

ZEIGLER, Bernard P. – PRAEHOFER, Herbert – TAG Gon Kim (2000): *Theory of Modeling and Simulation: Integrating Discrete Event and Continuous complex dynamic systems*. 2nd edition, Academic Press.

JOGSZABÁLYTÁR

A járások kialakításáról, valamint egyes ezzel összefüggő törvények módosításáról szóló 2012. évi XCIII. törvény

A területi államigazgatási szervezetrendszer átalakítását megalapozó intézkedésekről szóló 1191/2010. (IX. 14.) Kormányhatározat

2010. évi CXXVI. törvény a fővárosi és megyei kormányhivatalokról, valamint a fővárosi és megyei kormányhivatalok kialakításával és a területi integrációval összefüggő törvénymódosításokról

2004. évi CXL törvény a közigazgatási hatósági eljárás és szolgáltatás általános szabályairól

515/2013. (XII. 30.) kormányrendelet a kormányablakokról

<http://www.jaras.info.hu/jarasi-ugyintezes> Letöltés ideje: 2017.08.20.

<http://www.jaras.info.hu/jarasi-hivatalok/jarasi-hivatalok-feladatkore> Letöltés ideje: 2017.08.20.

<http://www.jaras.info.hu/jarasi-hivatalok/jarasi-szakigazgatasi-szervek> Letöltés ideje: 2017.08.20.

<http://www.jarasikormanyablak.hu/jarasi-hivatalok/> Letöltés ideje: 2018.10.12.

<http://www.jarasikormanyablak.hu/kormanyablakok/> Letöltés ideje: 2018.10.12.

<http://www.kormanyablak.hu> Letöltés ideje: 2018.10.12.

<http://www.kormany.hu/hu/kozigazgatasi-es-igazsagugyi-miniszterium/hirek/magyary-program-a-koz-szolgaltatara> Letöltés ideje: 2017.08.20.

<http://www.kormany.hu/hu/kozigazgatasi-es-igazsagugyi-miniszterium/teruleti-kozigazgatasesert-es-valasztasokert-felelos-allamtitkarsag/hirek/kialakultak-a-jarasi-hivatalok-feladatkorei> Letöltés ideje: 2017.08.20.

<http://magyaryprogram.kormany.hu/hatasvizsgalatok-es-strategiai-dokumentumok-novelik-a-kozigazgatashatekonysagat> Letöltés ideje: 2017.08.20.

A Nemzeti Közszoigálati Egyetem kiadványa



Kiadó:

Nemzeti Közszoigálati Egyetem;
Közigazgatási Továbbképzési Intézet
www.uni-nke.hu

Felelős kiadó:

Prof. Dr. Kis Norbert rektorhelyettes
Címe: 1083 Budapest, Üllői út 82.

Kiadói szerkesztő:

Kelemen Dóra

Tördelőszerkesztő:

Friebert Máté

ISBN 978-963-498-228-9 (elektronikus)