

FAKULTA RIADENIA A INFORMATIKY  
ŽILINSKÁ UNIVERZITA V ŽILINE

## AGENTOVO ORIENTO VANÁ SIMULÁCIA

SEMESTRÁLNA PRÁCA Č. 3 Z PREDMETU DISKRÉTNÁ SIMULÁCIA

Autor: **Bc. Matej Poljak**

Cvičiaci: **Ing. Andrea Galadíková, PhD.**

Akademický rok: **2024/2025**

## Obsah

Architektúra riešenia .....	4
Entity .....	4
Model agentov .....	5
UML diagramy .....	10
Logika.....	10
Interakcia GUI so simuláciou .....	10
Animácia .....	11
Popis GUI .....	13
Experimentálna časť .....	16
Vyhodnotenie experimentov.....	17

## Zoznam obrázkov

Obrázok 1 – model hierarchie agentov a ich správ .....	5
Obrázok 2 - typy správ pre komunikáciu medzi agentmi .....	5
Obrázok 3 - asistenti agenta okolia .....	6
Obrázok 4 - asistenti agenta pre stolárov skupiny A .....	7
Obrázok 5 - asistenti agenta pre stolárov skupiny B .....	8
Obrázok 6 - asistenti agenta pre stolárov skupiny C .....	8
Obrázok 7 - asistenti agenta simulujúceho pohyb stolára .....	9
Obrázok 8 - UML diagram agentov a ich entít.....	10
Obrázok 9 - interakcia GUI so simuláciou.....	11
Obrázok 10 - objekty pre prácu s animáciou.....	12
Obrázok 11 - animácia montážnych miest .....	12
Obrázok 12 - animácia skladu.....	13
Obrázok 13 - konfiguračný panel GUI.....	13
Obrázok 14 - graf ustaľovania .....	14
Obrázok 15 - záložka so štatistikami .....	14
Obrázok 16 - zobrazenie aktuálneho stavu simulácie .....	15
Obrázok 17 - záložka s animátorom .....	15
Obrázok 18 -výsledky experimentov (časť A) .....	16
Obrázok 19 - výsledky experimentov (časť B) .....	17
Obrázok 20 - najvhodnejšia konfigurácia .....	17

# Architektúra riešenia

V tejto kapitole si popíšeme entity, ktoré budú tvoriť simuláciu, hierarchiu agentov a spôsob spolupráce medzi nimi.

## Entity

Najprv si popíšeme najdôležitejšie entity, s ktorými budú agenti v našej architektúre spolupracovať

### Objednávka

Je v systéme reprezentovaná triedou *Order.class* a disponuje jedinečným identifikátorom v rámci systému. Navyše, je obalovou triedou pre všetky kusy nábytku, ktoré sú vo vzťahu kompozície k tejto objednávke. Uchováva dátum vytvorenia a vyhotovenia objednávky.

### Kus nábytku

Je reprezentovaný triedou *Furniture.class*. Popisuje typ nábytku, jeho jedinečný identifikátor, začiatok jeho spracovania a vyhotovenia. Taktiež obsahuje referenciu na svoju objednávku, ktorej je súčasťou. Počas simulácie eviduje aktuálny technologický krok, ktorý sa vykonáva a aj montážne miesto kde sa tento nábytok vyrába.

### Typ nábytku

Pokrýva všetky možné druhy nábytku a je reprezentovaný enumeráciou *Furniture.Type.class*.

### Technologický krok

Pokrýva všetky technologické kroky, ktoré vedú stolári vykonávať a je reprezentovaný enumeráciou *Furniture.TechStep.class*.

### Stolár

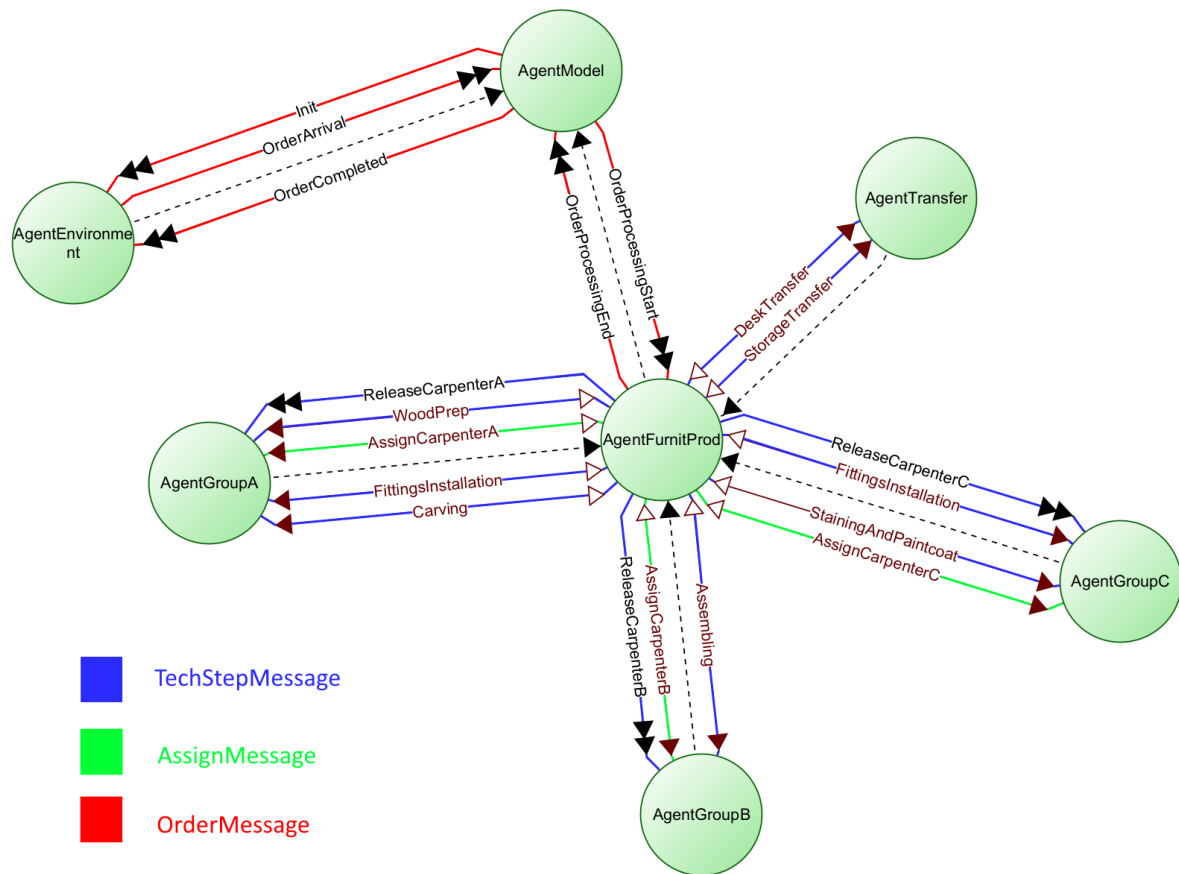
Entita, ktorá jednoznačne identifikuje stolára, miesto, kde sa nachádza a jemu pridelenú objednávku. Je reprezentovaný triedou *Carpenter.class*. Taktiež obsahuje informáciu zaradenia do jednej z pracovných skupín identifikovanej enumeráciou *Carpenter.Group.class*.

### Prideľovač montážnych miest

Jeho úlohou je udržiavať informáciu o všetkých montážnych miestach či je každý montážny stôl pridelený nejakej objednávke alebo nie. Má teda na starosť prideľovanie a manažovanie voľných pracovných stolov. Je reprezentovaný triedou *DeskAllocation.class*.

## Model agentov

V nástroji AbaBuilder sme si vytvorili model simulácie pre výrobu nábytku. (obrázok 1)



Obrázok 1 – model hierarchie agentov a ich správ

Popíšeme si bližšie zodpovednosti každého agenta a interakciu medzi nimi, ale najprv si definujme správy (obrázok 2), ktoré budeme v rámci našej agentovej komunikácie používať.

- OrderMessage** – je potomkom triedy *MessageForm*. Obsahuje referenciu na entitu *Order.java* pre potreby doručenia objednávky agentovi, ktorý ju bude vedieť náležite spracovať.
- AssignMessage** - je potomkom triedy *MessageForm*. Slúži na získanie stolára (trieda *Carpenter*) pre spracovanie prideleného kusu nábytku, ktorý zastrešuje trieda *Furniture*.
- TechStepMessage** - je potomkom triedy *MessageForm*. Obsahuje referenciu na entitu *Carpenter.java*, ktorá predstavuje prideleného stolára. Tu nie je potrebné udržiavať referenciu na pridelený kus nábytku, pretože tú si udržiava samotný stolár, čím signalizuje, že pracuje.

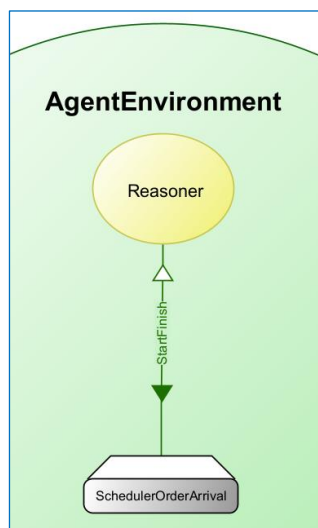


Obrázok 2 - typy správ pre komunikáciu medzi agentmi

Keď už máme definované správy, ktoré si agenti môžu posilať, popíšeme si jednotlivých agentov.

### AgentEnvironment

Je agentom okolia. Po prijatí správy *Init* od agenta modelu *AgentModel* generuje cez kontinuálneho asistenta Plánovač *SchedulerOrderArrival* (obrázok 3) nové objednávky, ktoré potom posila agentovi modelu správou *OrderArrival*. Agent okolia obsahuje štatistiky o priemernom čase vyhotovenia objednávky a o počte dokončených objednávok. Všetky správy na komunikáciu s agentom modelu sú typu *OrderMessage.java*.



Obrázok 3 - asistenti agenta okolia

### AgentModel

Je koreňom agentovej hierarchie. Slúži na preposielanie správ medzi agentom okolia a agentom *AgentFurnitProd*, ktorý spracováva vytvorené objednávky. Agent modelu po prijatí správy *OrderArrival* pošle správu *OrderProcessingStart* typu *notice* agentovi *AgentFurnitProd*, na ktorú po získaní správy *OrderProcessingEnd* notifikuje agenta okolia o vyhotovení objednávky správou *OrderCompleted*. Všetky spomenuté správy sú typu *OrderMessage.class*. Agent modelu nedisponuje žiadnymi asistentmi.

### AgentFurnitProd

Je agentom, ktorý zabezpečuje hlavnú logiku riadenia celého procesu vytvorenia všetkých kusov nábytku pre objednávku. Vo svojej podstate nemá žiadnych asistentov. Obsahuje prioritné fronty správ typu *TechStepMessage.class* pre iniciáciu začatia jednotlivých technologických krokov alebo presunov.

- Najprv si manažér agenta *AgentFurnitProd* **v závislosti od nasledujúceho technologického kroku** pre vybraný kus nábytku **vypýta stolára** od *AgentGroupA* pre skupinu A, *AgentGroupB* pre skupinu B, *AgentGroupC* pre skupinu C správou *AssignCarpenterA*, *AssignCarpenterB*, *AssignCarpenterC* typu *request*.
  - Ak v správe typu *response* priradenia nezíska manažér referenciu na stolára, **zaradí** správu typu *TechStepMessage.class* **do príslušného frontu** nespracovaných požiadaviek.
  - Ak je v správe typu *response* referenciu na stolára získa, **začne proces presunu stolára** jednou zo správ pre komunikáciu s agentom *AgentTransfer* **alebo** začne **proces**

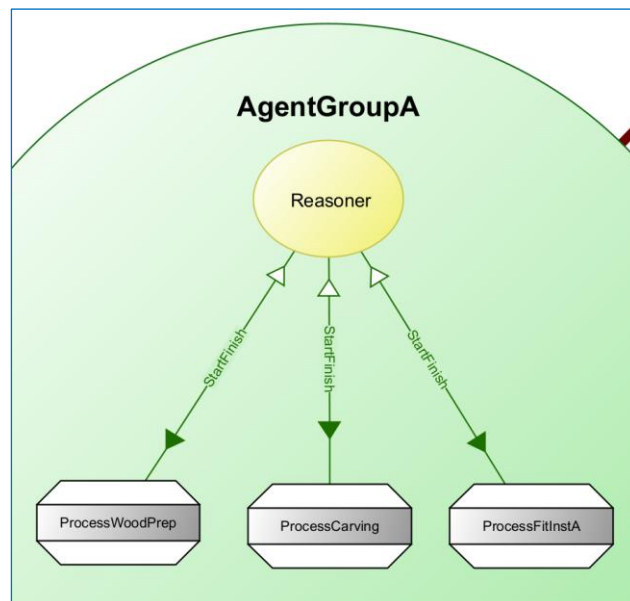
**technologického kroku** komunikáciou jedným z agentov *AgentGroupA*, *AgentGroupB*, *AgentGroupC*.

- Po vyhotovení posledného kusu nábytku jednej objednávky vráti správu typu *response* s kódom *OrderProcessing* agentovi modelu *AgentModel*.

Agent *AgentFurnitProd* obsahuje štatistiky priemerné vážené dĺžky všetkých frontov a priemerné časy čakania kusov nábytku vo frontoch.

#### *AgentGroupA*

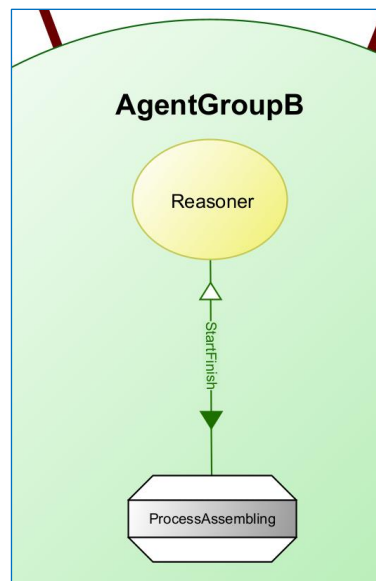
Predstavuje agenta, ktorý obsahuje referencie na všetkých stolárov skupiny A. Na správy typu request od agenta *AgentFurnitProd* s kódom *AssignCarpenterA* pošle správu *response* s inštanciou voľného stolára, ak taký existuje. V opačnom prípade pošle v správe hodnotu *null*. Zastrešuje pomocou kontinuálnych agentov (obrázok 4) typu Proces činnosti prípravy materiálu (správa *WoodPrep*), rezania (správa *Carving*) a montáže kovaní (správa *FittingsInstallation*). Po skončení činnosti manažér agenta *AgentGroupA* uvoľní stolára a v príslušnej správe typu *response* už neuvádza referenciu stolára. Agent *AgentGroupA* obsahuje štatistiku vyťaženia pracovnej skupiny A. Na komunikáciu sa využíva správa typu *TechStepMessage.class*.



Obrázok 4 - asistenti agenta pre stolárov skupiny A

#### *AgentGroupB*

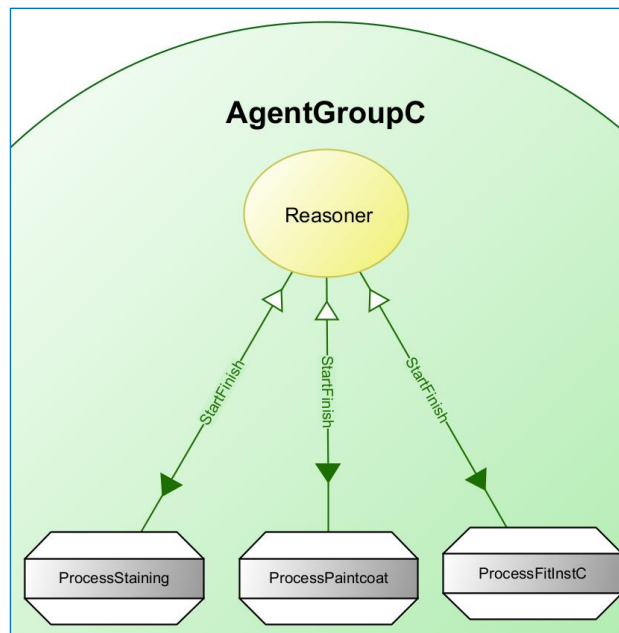
Predstavuje agenta, ktorý obsahuje referencie na všetkých stolárov skupiny B. Na správy typu request od agenta *AgentFurnitProd* s kódom *AssignCarpenterB* pošle správu *response* s inštanciou voľného stolára, ak taký existuje. V opačnom prípade pošle v správe hodnotu *null*. Zastrešuje pomocou kontinuálneho agenta (obrázok 5) typu Proces činnosti skladania nábytku. (správa *Assembling*) Po skončení činnosti manažér agenta *AgentGroupB* uvoľní stolára a v príslušnej správe typu *response* už neuvádza referenciu stolára. Agent *AgentGroupB* obsahuje štatistiku vyťaženia pracovnej skupiny B. Na komunikáciu sa využíva správa typu *TechStepMessage.class*.



Obrázok 5 - asistenti agenta pre stolárov skupiny B

### AgentGroupC

Predstavuje agenta, ktorý obsahuje referencie na všetkých stolárov skupiny C. Na správy typu request od agenta *AgentFurnitProd* s kódom *AssignCarpenterC* pošle správu *response* s inštanciou voľného stolára, ak taký existuje. V opačnom prípade pošle v správe hodnotu *null*. Zastrešuje pomocou kontinuálnych agentov (obrázok 6) typu Proces činnosti morenia materiálu s prípadným lakovaním (správa *StainingAndPaintcoat*), a montáže kovaní (správa *FittingsInstallation*). Po skončení činnosti manažér agenta *AgentGroupC* uvoľní stolára a v príslušnej správe typu *response* už neuvádza referenciu stolára. Agent *AgentGroupC* obsahuje štatistiku vyťaženia pracovnej skupiny C. Na komunikáciu sa využíva správa typu *TechStepMessage.class*.



Obrázok 6 - asistenti agenta pre stolárov skupiny C

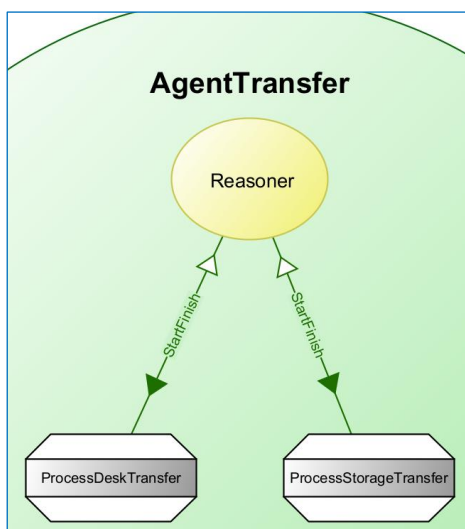


## AgentTransfer

Je agent slúžiaci na simuláciu procesu pohybu stolára. Agent *AgentFurnitProd* môže poslať 2 správy typu request.

- Správa *DeskTransfer* – využitie kontinuálneho asistenta typu *Proces* (obrázok 7) pre vygenerovanie času potrebného pre presun medzi montážnymi pracoviskami
- Správa *StorageTransfer* - využitie kontinuálneho asistenta typu *Proces* (obrázok 7) pre vygenerovanie času potrebného pre presun medzi skladoom a montážnym pracoviskom

Na komunikáciu sa využíva správa typu *TechStepMessage.class*.



Obrázok 7 - asistenti agenta simulujúceho pohyb stolára

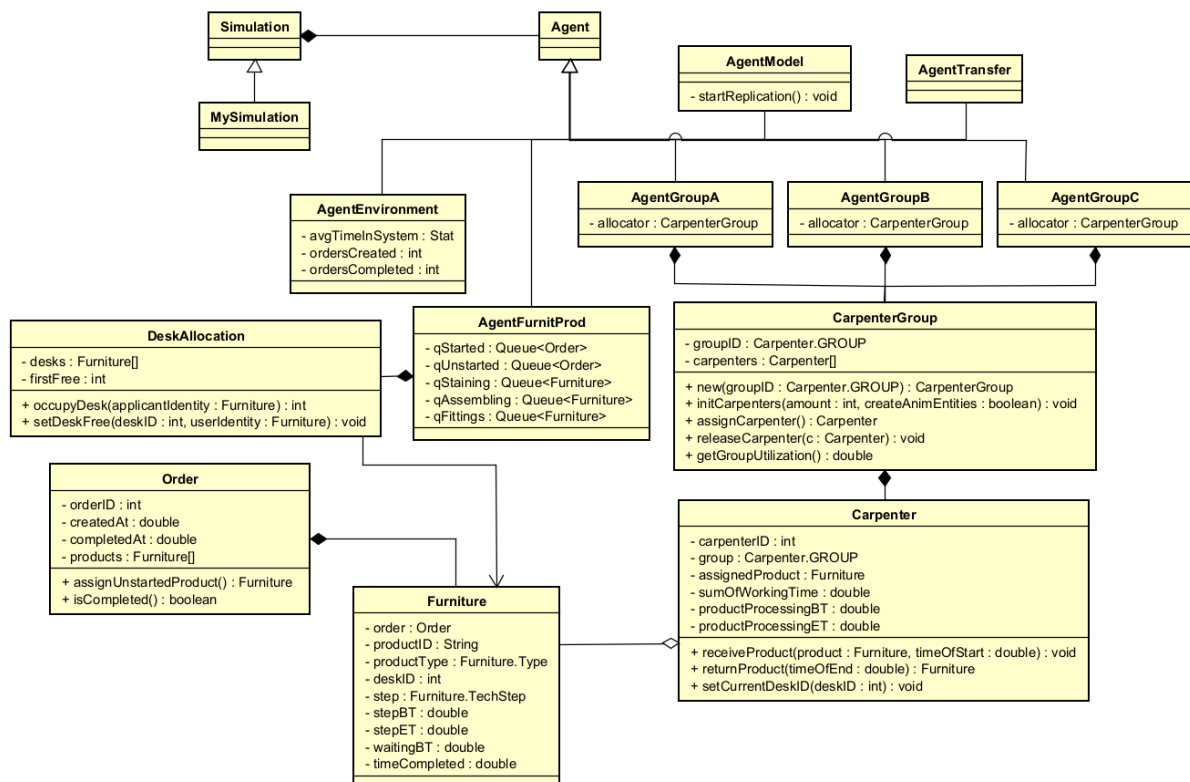
## UML diagramy

V tejto časti si popíšeme implementáciu systému v podobe UML diagramov tried pre lepšie pochopenie návrhu.

### Logika

V časti popisu logiky chceme zdôrazniť vzťahy medzi použitím agentov a entít. Obrázok 8 znázorňuje tie najdôležitejšie časti.

- Agenti *AgentGroupA*, *AgentGroupB*, *AgentGroupC* spravujú priradovanie zdrojov inštancií stolárov skrze triedu *CarpenterGroup*, ktorá poskytuje štatistiku vyťaženia skupiny stolárov
- Trieda *DeskAllocation* spravuje pridelenie montážnych miest na základe referencií nábytkov, ktorým majú byť pridelené. Je súčasťou agenta *AgentFurnitProd*.
- Agent *AgentFurnitProd* obsahuje všetky fronty pre čakajúce kusy nábytku na spracovanie. Sú to fronty pre: čiastočne začaté objednávky - *qStarted* (aspoň jeden kus nábytku sa začal spracovávať), nezačaté objednávky - *qUnstarted*, čakanie na morenie - *qStaining*, čakanie na skladanie - *qAssembling* a čakanie na montáž kovaní - *qFittings*.
- Trieda objednávky *Order* spravuje postupné pridelenie ešte nespracovaných inštancií nábytku a ich uchovanie
- Agent okolia *AgentEnvironment* vytvára objednávky a zároveň eviduje štatistiku o priemernom čase ich spracovania

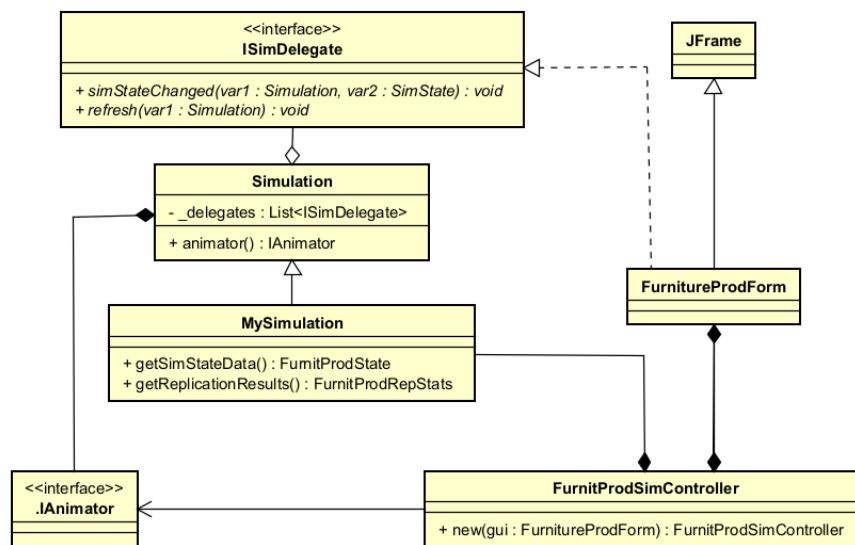


Obrázok 8 - UML diagram agentov a ich entít

### Interakcia GUI so simuláciou

V tejto časti ukážeme spôsob interakcie medzi GUI a simulačným jadrom, ktorý znázorňuje obrázok 9.

- GUI implementuje návrhový vzor *Observer*, cez ktorý dostáva oznámenia o zmene stavu simulácie
- Potomok *MySimulation* triedy *Simulation* poskytuje zmenu stavu simulácie cez metódu *getSimStateData()* a štatistiky po skončení replikácie cez metódu *getReplicationResults()*
- Na spracovanie žiadostí z GUI na komunikáciu so simuláciou slúži kontrolér *FurnitProdSimController* tak, aby GUI nebolo závislé od triedy simulácie a nebol v nej kód, ktorý logicky do GUI nepatrí. Tento kontrolér sa stará aj o vytvorenie a zrušenie animátora

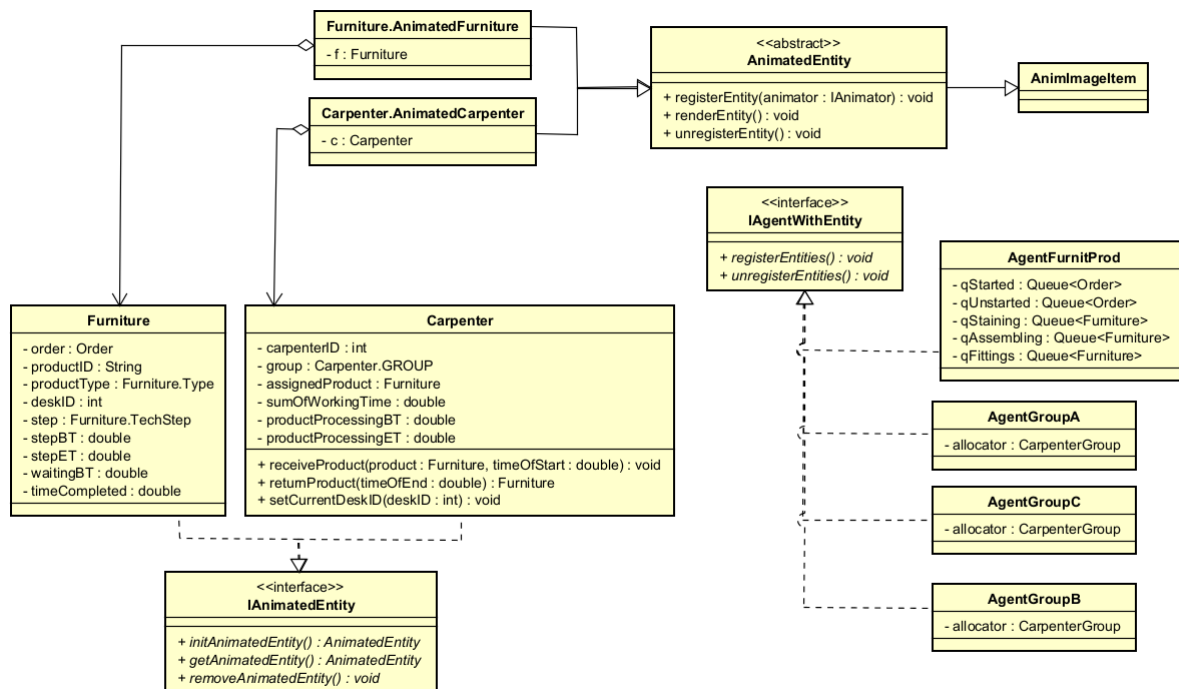


Obrázok 9 - interakcia GUI so simuláciou

## Animácia

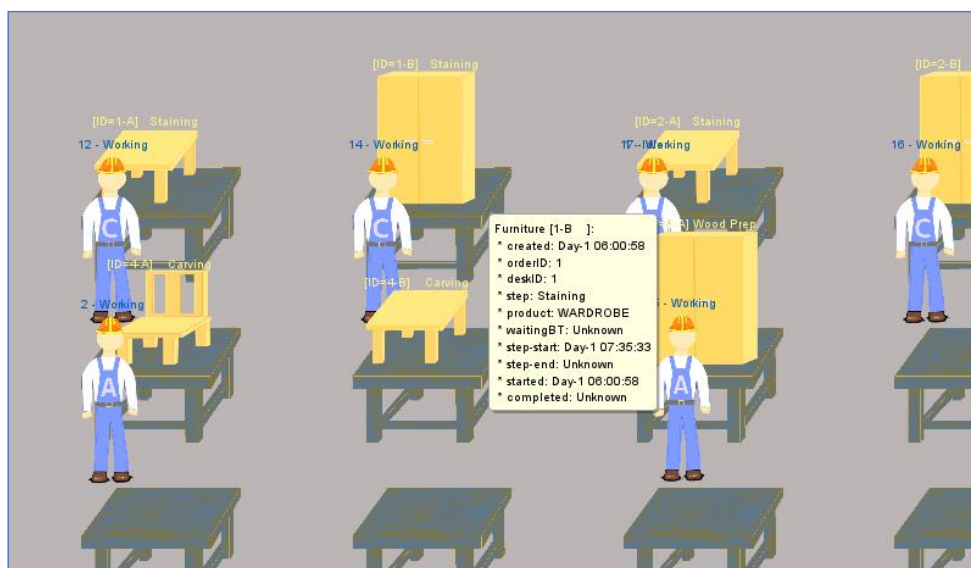
Pre účely animácie sme si vytvorili vnorené triedy *AnimatedCarpenter* (entity *Carpenter*) a *AnimatedFurniture* (entity *Furniture*). Tieto sú potomkom nami vytvorenej abstraktnej triedy *AnimatedEntity*, ktorá je zase potomkom animačného objektu *AnimImageItem*. (obrázok 10)

- Potomkovia *AnimatedEntity* si uchovávajú inštanciu svojej prislúchajúcej entity, ktorú získajú cez konštruktor. Prislúchajúca entita si vytvorí inštanciu svojho animačného objektu a po zmene stavu volá metódu *renderEntity()*, ktorá aktualizuje animačný objekt na základe svojej entity.
- Agenti, ktorí disponujú nejakou entitou, implementujú rozhranie *IAgentWithEntity* pre centrálnu registráciu a odregistrovanie animovaných entít z animátora, ktoré sa vyvolajú v triede *MySimulation* po vytvorení a pred vymazaním animátora

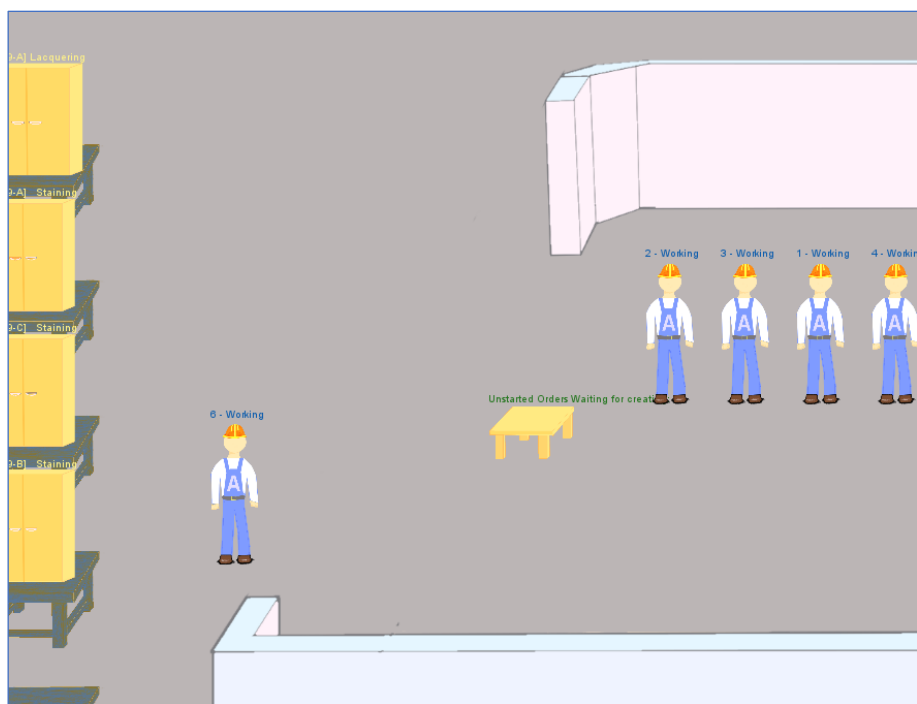


Obrázok 10 - objekty pre prácu s animáciou

Animácia našej stolárskej dielne animuje jednotlivé kusy nábytku, stolárov, montážne miesta a sklad. Animáciu môžeme vidieť na obrázku 11 a 12.



Obrázok 11 - animácia montážnych miest

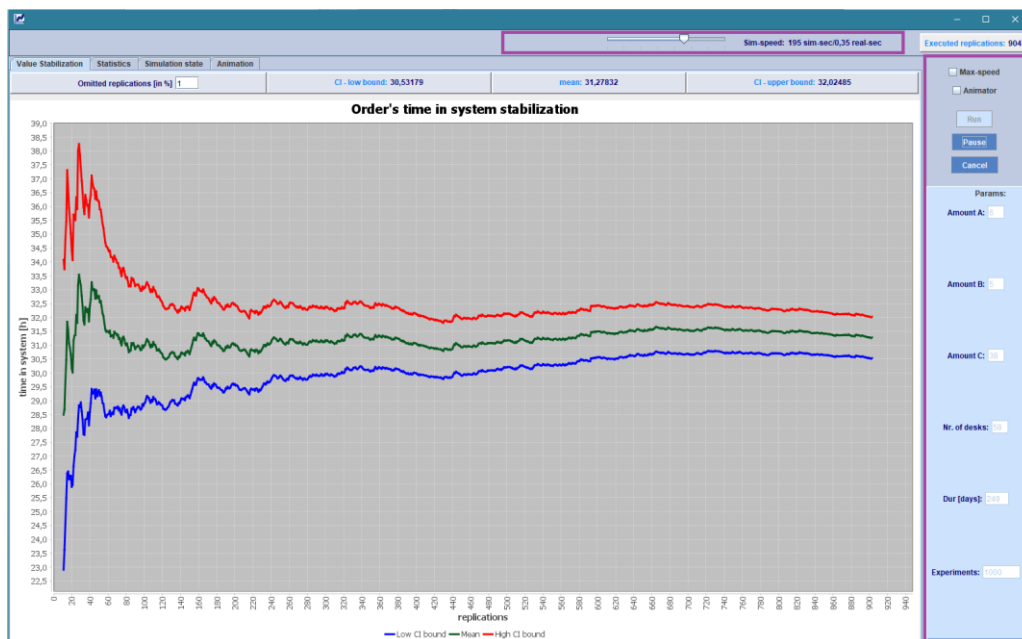


Obrázok 12 - animácia skladu

## Popis GUI

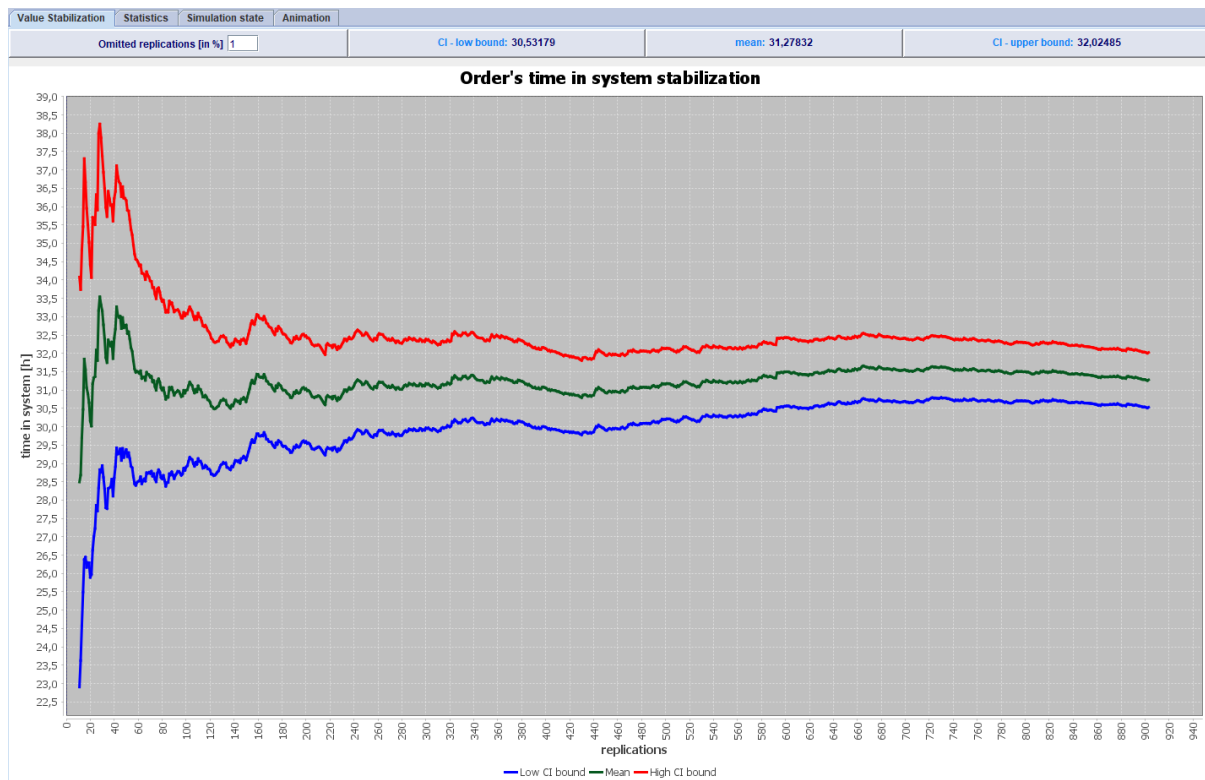
Po popísaní architektúry riešenia si môžeme ukázať zloženie grafického užívateľského rozhrania aplikácie. Tá sa skladá z piatich častí:

1. **Konfiguračný panel** (obrázok 13) pre zadanie vstupných parametrov pre simuláciu, ktorými sú počty pracovníkov pre skupiny, počet montážnych miest, počet simulovaných dní a počet replikácií. Taktiež obsahuje checkbox na maximálne rýchly beh a zobrazenie animátora. Okrem toho obsahuje v režime bez maximálnej rýchlosti aj slider pre zmenu rýchlosti plynutia simulačného času.



Obrázok 13 - konfiguračný panel GUI

2. **Graf ustaľovania hodnoty** (obrázok 14) priemerného času objednávky v systéme s príslušným 95% intervalom spoľahlivosti naprieč vykonanými replikáciami. Graf umožňuje vykresľovanie hodnôt až po vykonaní definovaného počtu % zo všetkých replikácií.



Obrázok 14 - graf ustaľovania

3. **Záložka so štatistikami** (obrázok 15) naprieč replikáciami (globálnymi – ľavá časť) a v rámci jednej replikácie (lokálnymi – pravá časť)

Value Stabilization   Statistics   Simulation state   Animation			Stats within one experiment (local)		
Overall Stats			Time in experiment: Day-70 08:18:10		
Stat name	95% Confidence Interval: <left-bound   MEAN   right-bound>	Unit	Stat name	Value	Unit
Amount of waiting for 'order processing'	< 38.16   41.21 >	[qy]	Amount of waiting for 'order processing'	7.73	[qy]
Amount of waiting for 'furniture processing'	< 115.74   124.93 >	[qy]	Amount of waiting for 'furniture processing'	24.72	[qy]
Amount of waiting for 'staining'	< 3.94   3.98 >	[qy]	Amount of waiting for 'assembling'	3.40	[qy]
Amount of waiting for 'assembling'	< 4.18   4.21 >	[qy]	Amount of waiting for 'staining'	3.01	[qy]
Amount of waiting for 'fittings inst'	< 0.06   0.06 >	[qy]	Amount of waiting for 'fittings inst'	0.06	[qy]
Waiting time for 'order processing'	< 19.01   20.50 >	[h]	Waiting time for 'order processing'	3.94	[h]
Waiting time for 'furniture processing'	< 19.19   20.68 >	[h]	Waiting time for 'furniture processing'	4.18	[h]
Waiting time for 'staining'	< 0.67   0.67 >	[h]	Waiting time for 'staining'	0.53	[h]
Waiting time for 'assembling'	< 0.71   0.72 >	[h]	Waiting time for 'assembling'	0.60	[h]
Waiting time for 'fittings inst'	< 0.03   0.03 >	[h]	Waiting time for 'fittings inst'	0.03	[h]
Utilization of group A:	< 89.29   89.42 >	%	Utilization of group A:	86.48	%
Utilization of group B:	< 95.67   95.79 >	%	Utilization of group B:	92.32	%
Utilization of group C:	< 97.67   97.78 >	%	Utilization of group C:	93.43	%
Order's time in system	< 30.53   32.02 >	[h]	Order's time in system	15.23	[h]
			Count of used desks	52.07	[qy]

Obrázok 15 - záložka so štatistikami

4. **Záložka so zobrazením aktuálneho stavu simulácie** (obrázok 16) so simulačným časom, frontami čakajúcich kusov nábytku na spracovanie a stolármi s detailami ich práce

Value Stabilization

Statistics

Simulation state

Animation

Orders Created: 1093

Used Desks: 58

Simulation Time: Day-71 08:58:10

Orders Completed: 1046

Queue 'Unstarted' | Length: 69

OrderID	ProductID	DeskID	Product	Step	WaitingBT	StepBT	StepET	Started	Order cre...
1072	1072-A	-1	WARDRO...		Day-69 1...	Unknown	Unknown	Unknown	Day-69 1...
1072	1072-B	-1	WARDRO...		Day-69 1...	Unknown	Unknown	Unknown	Day-69 1...
1073	1073-A	-1	TABLE		Day-69 1...	Unknown	Unknown	Unknown	Day-69 1...
1073	1073-B	-1	TABLE		Day-69 1...	Unknown	Unknown	Unknown	Day-69 1...
1074	1074-A	-1	WARDRO...		Day-69 1...	Unknown	Unknown	Unknown	Day-69 1...
1074	1074-B	-1	CHAIR		Day-69 1...	Unknown	Unknown	Unknown	Day-69 1...
1074	1074-C	-1	TABLE		Day-69 1...	Unknown	Unknown	Unknown	Day-69 1...
1074	1074-D	-1	TABLE		Day-69 1...	Unknown	Unknown	Unknown	Day-69 1...
1074	1074-E	-1	CHAIR		Day-69 1...	Unknown	Unknown	Unknown	Day-69 1...
1077	1077-A	-1	WARDRO...		Day-70 0...	Unknown	Unknown	Unknown	Day-70 0...
1077	1077-B	-1	TABLE		Day-70 0...	Unknown	Unknown	Unknown	Day-70 0...

Queue 'Started' | Length: 3

OrderID	ProductID	DeskID	Product	Step	WaitingBT	StepBT	StepET	Started	Order cre...
1071	1071-C	-1	TABLE		Day-69 11...	Unknown	Unknown	Unknown	Day-69 11...
1071	1071-D	-1	WARDRO...		Day-69 11...	Unknown	Unknown	Unknown	Day-69 11...
1071	1071-E	-1	TABLE		Day-69 11...	Unknown	Unknown	Unknown	Day-69 11...

Queue 'Staining' | Length: 7

OrderID	ProductID	DeskID	Product	Step	WaitingBT	StepBT	StepET	Started	Order cre...
1069	1069-D	7	TABLE	Staining	Day-71 07...	Day-71 07...	Day-71 07...	Day-71 07...	Day-69 10...
1069	1069-C	41	TABLE	Staining	Day-71 08...	Day-71 07...	Day-71 08...	Day-71 07...	Day-69 10...
1069	1069-E	2	TABLE	Staining	Day-71 08...	Day-71 07...	Day-71 08...	Day-71 07...	Day-69 10...
1070	1070-C	23	TABLE	Staining	Day-71 08...	Day-71 08...	Day-71 08...	Day-71 07...	Day-69 10...
1070	1070-B	53	TABLE	Staining	Day-71 08...	Day-71 08...	Day-71 08...	Day-71 07...	Day-69 10...
1070	1070-D	43	CHAIR	Staining	Day-71 08...	Day-71 08...	Day-71 08...	Day-71 08...	Day-69 10...
1070	1070-E	56	TABLE	Staining	Day-71 08...	Day-71 08...	Day-71 08...	Day-71 08...	Day-69 10...

Queue 'Assembling' | Length: 2

OrderID	ProductID	DeskID	Product	Step	WaitingBT	StepBT	StepET	Started	Order cre...
1059	1059-B	22	CHAIR	Assembling	Day-71 07...	Day-71 06...	Day-71 07...	Day-70 10...	Day-69 07...
1064	1064-A	10	TABLE	Assembling	Day-71 08...	Day-71 06...	Day-71 08...	Day-70 12...	Day-69 08...

Queue 'Fittings inst.' | Length: 0

OrderID	ProductID	DeskID	Product	Step	WaitingBT	StepBT	StepET	Started	Order cre...
---------	-----------	--------	---------	------	-----------	--------	--------	---------	--------------

Carpenter's group A | Working: 6 / 6

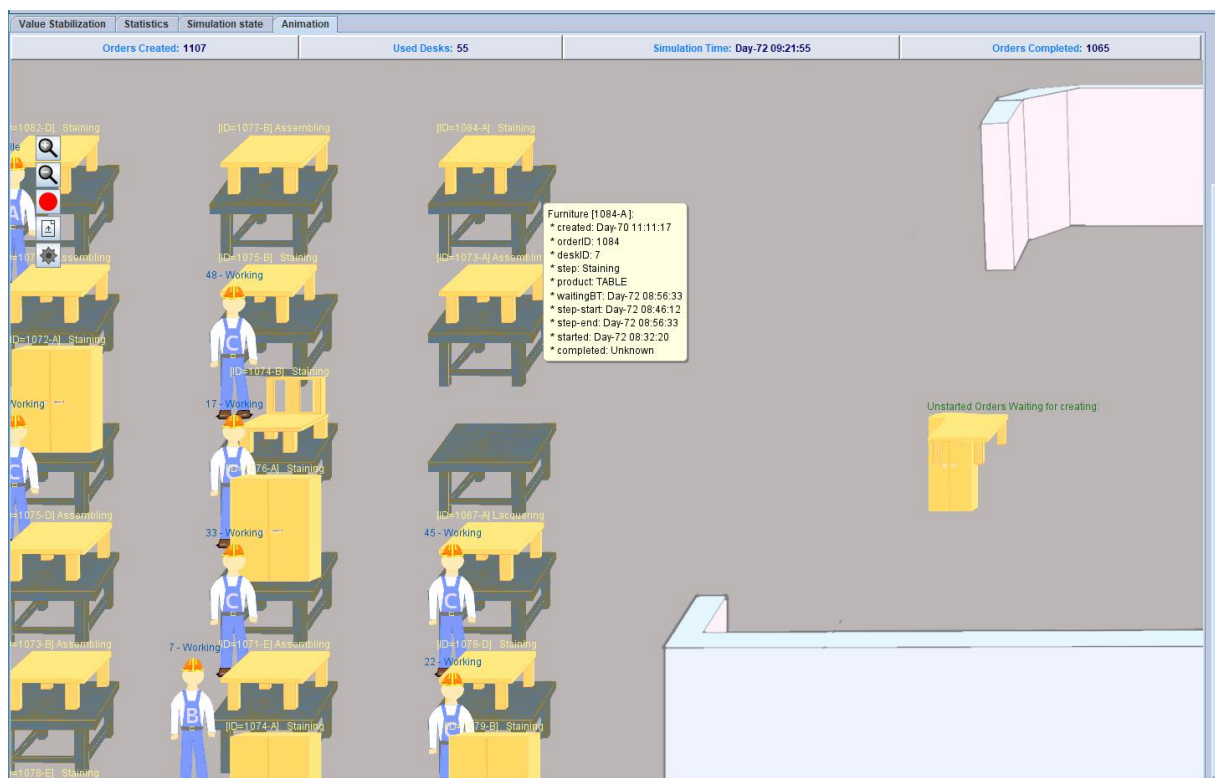
ID	Group	Working	DeskID	ProductID	ProcBT	ProcET	Product's details		
1	A	✓	51	1071-A	Day-71 08:17:33	Unknown	Product (productID= 1071-A, orderID= 1071, desk= 51; step=Carving ;TABLE ;waitingBT=Unk...		
2	A	✓	44	1040-A	Day-71 08:39:33	Unknown	Product (productID= 1040-A, orderID= 1040, desk= 44; step=Fit Inst. ;WARDROBE ;waitingBT=U...		
3	A	✓	40	1051-A	Day-71 08:41:05	Unknown	Product (productID= 1051-A, orderID= 1051, desk= 40; step=Fit Inst. ;WARDROBE ;waitingBT=U...		
4	A	✓	4	1071-B	Day-71 08:36:02	Unknown	Product (productID= 1071-B, orderID= 1071, desk= 4; step=Carving ;CHAIR ;waitingBT=Unk...		
5	A	✓	1035-A	Desks-Tra	Day-71 08:56:20	Unknown	Product (productID= 1035-A, orderID= 1035, desk= 38; step=Fit Inst. ;WARDROBE ;waitingBT=U...		
6	A	✓	52	1070-A	Day-71 07:41:11	Unknown	Product (productID= 1070-A, orderID= 1070, desk= 52; step=Carving ;WARDROBE ;waitingBT=...		

Carpenter's group B | Working: 5 / 5

ID	Group	Working	DeskID	ProductID	ProcBT	ProcET	Product's details		
7	B	✓	46	1054-C	Day-71 08:39:33	Unknown	Product (productID= 1054-C, orderID= 1054, desk= 46; step=Assembling ;WARDROBE ;waiting...		
8	B	✓	37	1057-E	Day-71 08:50:19	Unknown	Product (productID= 1057-E, orderID= 1057, desk= 37; step=Assembling ;TABLE ;waitingBT=...		
9	B	✓	15	1053-A	Day-71 08:35:07	Unknown	Product (productID= 1053-A, orderID= 1053, desk= 15; step=Assembling ;WARDROBE ;waiting...		
10	B	✓	51	1067-C	Day-71 08:44:06	Unknown	Product (productID= 1067-C, orderID= 1067, desk= 51; step=Assembling ;TABLE ;waitingBT=...		

Obrázok 16 - zobrazenie aktuálneho stavu simulácie

## 5. Záložka s animátorom (obrázok 17) pre grafickú vizualizáciu aktuálneho stavu v simulácii.



Obrázok 17 - záložka s animátorom



## Experimentálna časť

Na záver sa budeme venovať najdôležitejšej časti – experimentom. Experimentov bude potrebné urobiť viac, pretože musíme zohľadniť dva faktory, ktoré ovplyvňujú náklady zdrojov. Sú to celkový počet pracovníkov a počet montážnych miest. Budeme predpokladať, že náklady na jedného stolára typu A, B, C predstavuje rovnaké náklady.

Vstupné úvahy:

- Prvou základnou úvahou je, že proces morenia nábytku trvá veľmi dlho v porovnaní s ostatnými činnosťami. To znamená, že stolári skupiny C budú dlho nedostupní pre spracovanie ďalších kusov objednávok. Preto ich budeme potrebovať oveľa viac ako pre skupiny A a B.
- Druhou skutočnosťou je, že pracovníci zo skupiny A budú okrem prípravy a rezania materiálu aj vykonávať montáž kovaní. Keďže skončia vždy svoju prácu oveľa skôr ako stolári zo skupiny C, prioritne budeme prideliť spracovanie požiadavky montáže kovaní stolárom zo skupiny A. Preto bude pravdepodobne vhodnejšie ich počet navýšiť tak, aby počet stolárov zo skupiny A bol aspoň o jedného viac ako zo skupiny B.

Na obrázkoch 18 a 19 môžeme vidieť výsledky pre všetkých 52 simulovaných konfigurácií. Každá bola vykonaná pre 249 simulačných dní, pričom každý deň mal pracovnú dobu 8 hodín (do času spracovania objednávky sa zahrňal iba celkový pracovný čas).

Exp. Nr.	Reps;A;B;C;Desks	CI-Order'sTimeInSystem	CI-UtilizationA	CI-UtilizationB	CI-UtilizationC
1	31;2;2;18;70	<595,58   605,35>	<99,93   99,96>	<98,93   99,19>	<86,68   86,97>
2	31;2;2;18;80	<594,96   604,35>	<99,93   99,96>	<98,95   99,17>	<86,61   86,96>
3	34;2;2;18;90	<587,21   598,66>	<99,93   99,95>	<98,94   99,16>	<86,57   86,92>
4	95;4;4;25;60	<340,78   347,59>	<90,08   90,23>	<80,18   80,33>	<99,59   99,62>
5	91;4;4;25;70	<343,41   350,28>	<90,20   90,36>	<80,21   80,37>	<99,57   99,60>
6	105;4;4;25;80	<346,93   353,84>	<90,29   90,43>	<80,19   80,34>	<99,57   99,60>
7	271;6;6;30;60	<211,98   216,24>	<74,06   74,14>	<64,40   64,47>	<99,55   99,58>
8	266;6;6;30;70	<213,44   217,73>	<74,11   74,20>	<64,38   64,45>	<99,56   99,59>
9	266;6;6;30;80	<212,20   216,47>	<74,19   74,27>	<64,41   64,48>	<99,56   99,58>
10	1000;7;6;35;50	<86,93   89,19>	<73,60   73,64>	<74,80   74,85>	<99,10   99,13>
11	1000;7;6;35;60	<83,86   86,17>	<74,17   74,22>	<75,07   75,12>	<99,41   99,44>
12	1000;7;6;35;70	<83,71   85,97>	<74,20   74,25>	<75,06   75,11>	<99,40   99,43>
13	1000;7;6;35;80	<83,85   86,16>	<74,25   74,29>	<75,06   75,11>	<99,41   99,43>
14	1000;10;10;40;50	<23,61   24,74>	<55,65   55,74>	<48,14   48,22>	<92,87   93,02>
15	637;10;10;40;60	<14,06   14,35>	<56,01   56,16>	<48,49   48,61>	<93,57   93,81>
16	687;10;10;40;70	<14,17   14,46>	<56,01   56,16>	<48,52   48,64>	<93,62   93,86>
17	1000;15;10;35;50	<83,96   86,30>	<34,96   34,98>	<45,02   45,05>	<99,18   99,21>
18	1000;15;10;35;60	<82,13   84,42>	<35,07   35,09>	<45,13   45,15>	<99,41   99,44>
19	1000;15;10;35;70	<82,75   85,06>	<35,09   35,11>	<45,12   45,14>	<99,41   99,44>
20	1000;7;6;36;50	<67,74   69,98>	<74,91   74,96>	<76,52   76,57>	<98,60   98,64>
21	1000;7;6;36;60	<60,98   63,14>	<76,03   76,09>	<77,12   77,17>	<99,28   99,32>
22	1000;7;6;36;70	<59,85   62,04>	<76,04   76,09>	<77,09   77,15>	<99,26   99,31>
23	1000;8;7;36;50	<64,38   66,57>	<66,28   66,32>	<65,76   65,80>	<98,75   98,79>
24	1000;8;7;36;60	<59,27   61,42>	<67,00   67,05>	<66,14   66,18>	<99,26   99,30>
25	1000;8;7;36;70	<59,41   61,58>	<67,04   67,09>	<66,14   66,18>	<99,27   99,32>
26	1000;6;6;37;50	<56,54   58,68>	<85,71   85,77>	<77,46   77,51>	<97,57   97,62>

Obrázok 18 -výsledky experimentov (časť A)



Exp. Nr.	Reps;A;B;C;Desks	CI-Order'sTimeInSystem	CI-UtilizationA	CI-UtilizationB	CI-UtilizationC
27	1000;6;6;37;60	<41,92   43,86>	<88,50   88,59>	<78,75   78,82>	<98,97   99,04>
28	1000;6;6;37;70	<41,75   43,62>	<88,50   88,59>	<78,74   78,81>	<98,99   99,05>
29	1000;6;5;38;50	<63,25   65,40>	<85,30   85,34>	<92,24   92,29>	<94,25   94,29>
30	1000;6;5;38;60	<29,41   30,80>	<89,53   89,67>	<95,83   95,95>	<97,79   97,92>
31	1000;6;5;38;70	<26,67   27,88>	<89,79   89,95>	<96,04   96,17>	<98,01   98,14>
32	1000;5;5;39;50	<73,75   75,96>	<94,99   95,03>	<91,05   91,09>	<91,42   91,46>
33	1000;5;5;39;60	<35,40   37,09>	<98,80   98,90>	<95,13   95,23>	<95,57   95,67>
34	1000;5;5;39;70	<34,74   36,46>	<98,96   99,06>	<95,28   95,38>	<95,73   95,84>
35	1000;6;5;38;50	<63,18   65,33>	<85,27   85,32>	<92,22   92,27>	<94,24   94,29>
36	1000;6;5;38;55	<35,41   37,13>	<88,45   88,55>	<95,08   95,18>	<97,08   97,18>
37	1000;6;5;38;58	<30,74   32,21>	<89,20   89,33>	<95,62   95,74>	<97,59   97,70>
38	1000;6;5;38;59	<29,09   30,56>	<89,30   89,44>	<95,65   95,77>	<97,63   97,76>
39	1000;6;5;38;60	<28,80   30,20>	<89,51   89,65>	<95,84   95,97>	<97,80   97,92>
40	1000;6;5;38;70	<27,30   28,59>	<89,83   90,00>	<96,05   96,19>	<98,05   98,18>
41	1000;7;5;38;50	<59,60   61,70>	<75,64   75,69>	<92,58   92,63>	<94,17   94,22>
42	1000;7;5;38;55	<34,23   35,96>	<78,08   78,18>	<95,35   95,45>	<96,99   97,09>
43	1000;7;5;38;58	<28,58   29,91>	<78,50   78,62>	<95,77   95,89>	<97,42   97,54>
44	1000;7;5;39;60	<22,02   22,94>	<78,64   78,80>	<96,41   96,57>	<95,60   95,76>
45	1000;6;5;39;50	<60,00   62,16>	<85,29   85,34>	<92,59   92,64>	<92,23   92,28>
46	1000;6;5;39;55	<30,02   31,47>	<88,28   88,40>	<95,64   95,75>	<95,26   95,38>
47	1000;6;5;39;60	<23,38   24,47>	<89,17   89,35>	<96,35   96,51>	<95,95   96,11>
48	1000;8;6;40;55	<15,51   15,88>	<69,27   69,44>	<80,79   80,97>	<93,67   93,87>
49	679;8;6;40;60	<14,34   14,63>	<69,38   69,58>	<80,86   81,07>	<93,76   94,00>
50	559;8;6;40;65	<14,33   14,62>	<69,25   69,49>	<80,80   81,04>	<93,69   93,97>
51	1000;10;8;40;55	<14,88   15,21>	<56,10   56,23>	<60,70   60,83>	<93,71   93,91>
52	651;10;8;40;60	<14,13   14,41>	<56,04   56,20>	<60,61   60,77>	<93,59   93,83>

Obrázok 19 - výsledky experimentov (časť B)

## Vyhodnotenie experimentov

Z výsledkov sledovania simulácie vieme, že nemá zmysel do konfigurácie zadať do vstupného parametra počtu pracovných miest menej ako 50 miest, pretože stolári by síce mohli pracovať, ale nemajú kde, keďže ich je viac a teda vždy sú nejakí nevyužití.

Pre náš sledovaný záujem sa snažíme minimalizovať počet pracovníkov a montážnych miest, pričom celkový čas v systéme nesmie presiahnuť 32 pracovných hodín. Tomuto cieľu najviac vyhovuje konfigurácia **38** spolu so 49 stolármi a 59 montážnymi miestami. Môžeme ale vidieť, že konfigurácia **37** obsahuje ten istý počet pracovníkov, ale o 1 montážne miesto menej. Horná hranica intervalu spoľahlivosti však presahuje hornú hranicu akceptovateľného času. Skúsme teraz spustiť konfiguráciu **37** nie na 1000 replikácií, ale 5000 a uvidíme či sa interval spoľahlivosti ustáli tak, že bude celý pod prahom 32 hodín. Výsledok konfigurácie **53** (viď obrázok 20) nám ukázal, že sme dosiahli **lepší výsledok a zachovali kritériá** a teda našli optimálne riešenie pre náš problém.

Exp. Nr.	Reps;A;B;C;Desks	CI-Order'sTimeInSystem	CI-UtilizationA	CI-UtilizationB	CI-UtilizationC
53	5000;6;5;38;58	<30,93   31,57>	<89,30   89,36>	<95,68   95,73>	<97,65   97,71>

Obrázok 20 - najvhodnejšia konfigurácia