FAKULTA RIADENIA A INFORMATIKY

ŽILINSKÁ UNIVERZITA V ŽILINE

AGENTOVO ORIENTOVANÁ SIMULÁCIA

SEMESTRÁLNA PRÁCA Č. 3 Z PREDMETU DISKRÉTNA SIMULÁCIA

Autor: **Bc. Matej Poljak**

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

Akademický rok: **2024/2025**

Obsah

[Architektúra riešenia 5](#_Toc197591687)

[Entity 5](#_Toc197591688)

[Model agentov 6](#_Toc197591689)

[Experimentálna časť 11](#_Toc197591690)

[Vyhodnotenie experimentov 12](#_Toc197591691)

Zoznam obrázkov

[Obrázok 1 – model hierarchie agentov a ich správ 6](#_Toc197591692)

[Obrázok 2 - typy správ pre komunikáciu medzi agentmi 6](#_Toc197591693)

[Obrázok 3 - asistenti agenta okolia 7](#_Toc197591694)

[Obrázok 4 - asistenti agenta pre stolárov skupiny A 8](#_Toc197591695)

[Obrázok 5 - asistenti agenta pre stolárov skupiny B 9](#_Toc197591696)

[Obrázok 6 - asistenti agenta pre stolárov skupiny C 9](#_Toc197591697)

[Obrázok 7 - asistenti agenta simulujúceho pohyb stolára 10](#_Toc197591698)

[Obrázok 8 -výsledky experimentov (A) 11](#_Toc197591699)

[Obrázok 9 - výsledky experimentov (B) 12](#_Toc197591700)

[Obrázok 10 - najvhodnejšia konfigurácia 12](#_Toc197591701)

Zoznam tabuliek

No table of figures entries found.

# 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 nábytky, 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 vedia stolári vykonávať a je reprezentovaný enumeráciou *Furniture.TechStep.class*.

Stolár

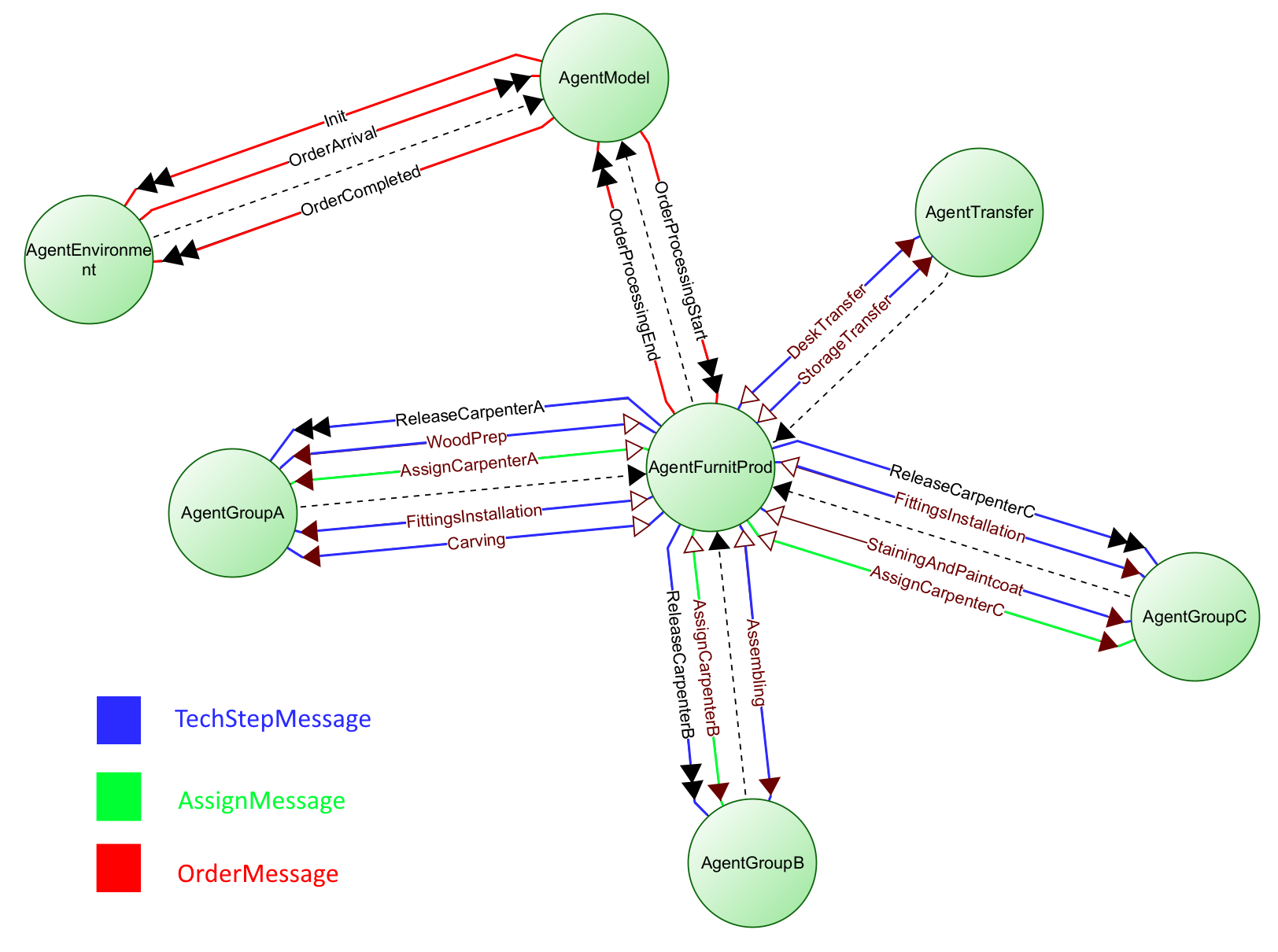
Entita, ktorá jednoznačne identifikuje stolára, miesto, kde sa nachádza a jemu pridelenú objednávku. Je reprezentovaný triedou *Carpenter.class*. Taktiež obsahuje informáciu zaradenia do jednej z pracovných skupín identifikovanej enumeráciou *Carpenter.Group.class*.

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 výrobňu 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.

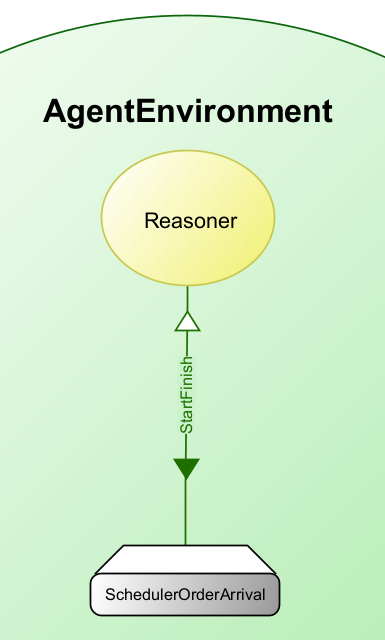


Obrázok 2 - typy správ pre komunikáciu medzi agentmi

Keď už máme definované správy, ktoré si agenti môžu posielať, 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 posiela 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 žiadnými asistentmi.

AgentFurnitProd

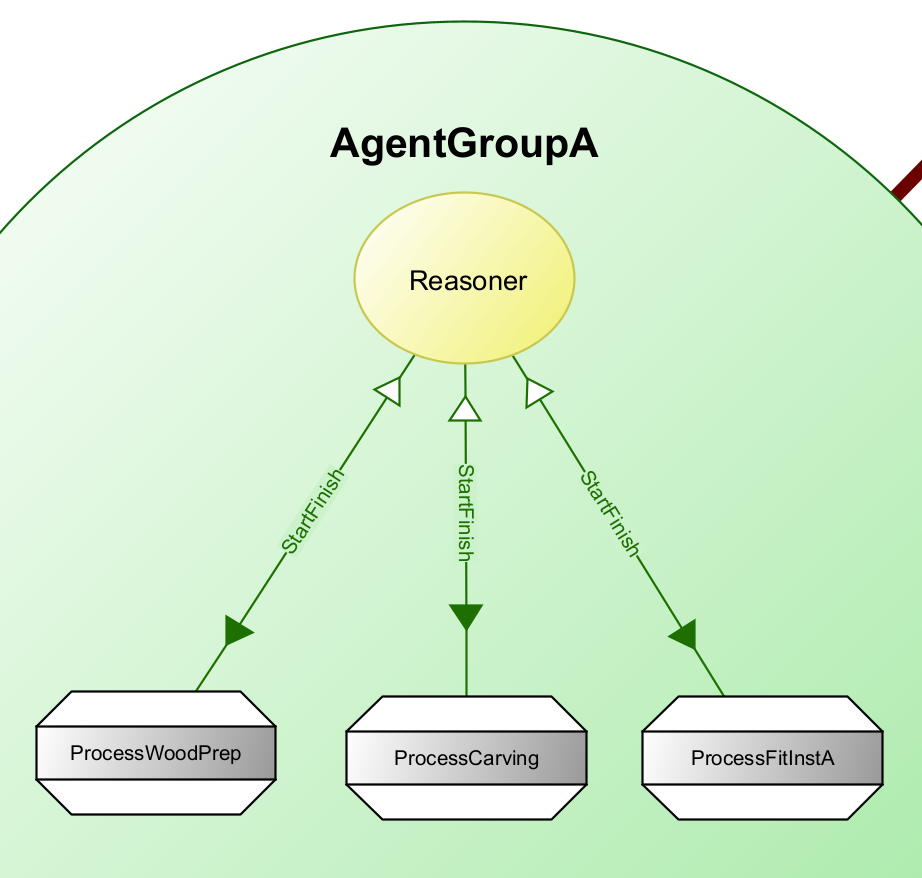
Je agentom, ktorý zabezpečuje hlavnú logiku riadenia celého procesu vytvorenia všetkých kusov nábytku pre objednávku. Vo svojej podstate nemá žiadnych asistentov. Obsahuje prioritné fronty správ typu *TechStepMessage.class* pre iniciáciu začatia jednotlivých technologických krokok alebo presunov.

* 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 agentovm *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

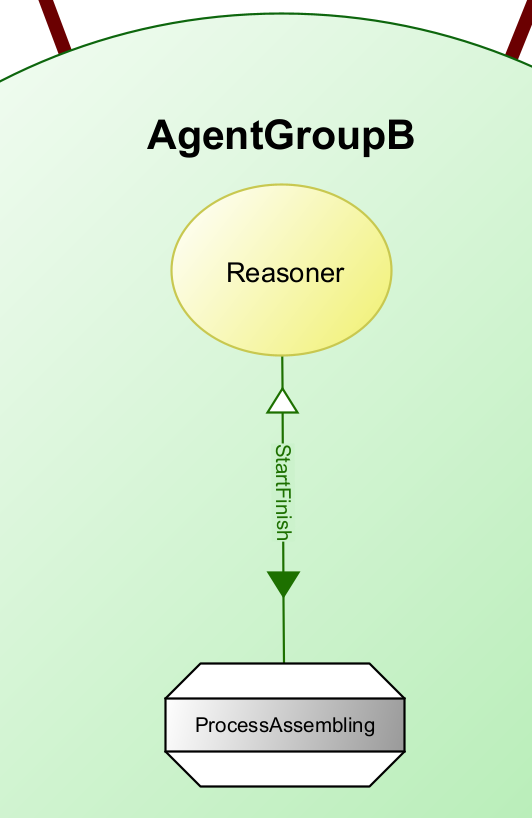
Predstavuje agenta, ktorý obsahuje referencie na všetkých stolárov skupiny A. Na správy typu request od agenta *AgentFurnitProd* s kódom *AssignCarpenterA* pošle správu *response* s inštanciou voľného stolára, ak taký existuje. V opačnom prípade pošle v správe hodnotu *null*. Zastrešuje pomocou kontinuálnych 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 vyťaženosti pracovnej skupiny A. Na komunikáciu sa využíva správa typu *TechStepMessage.class*.



Obrázok 4 - asistenti agenta pre stolárov skupiny A

AgentGroupB

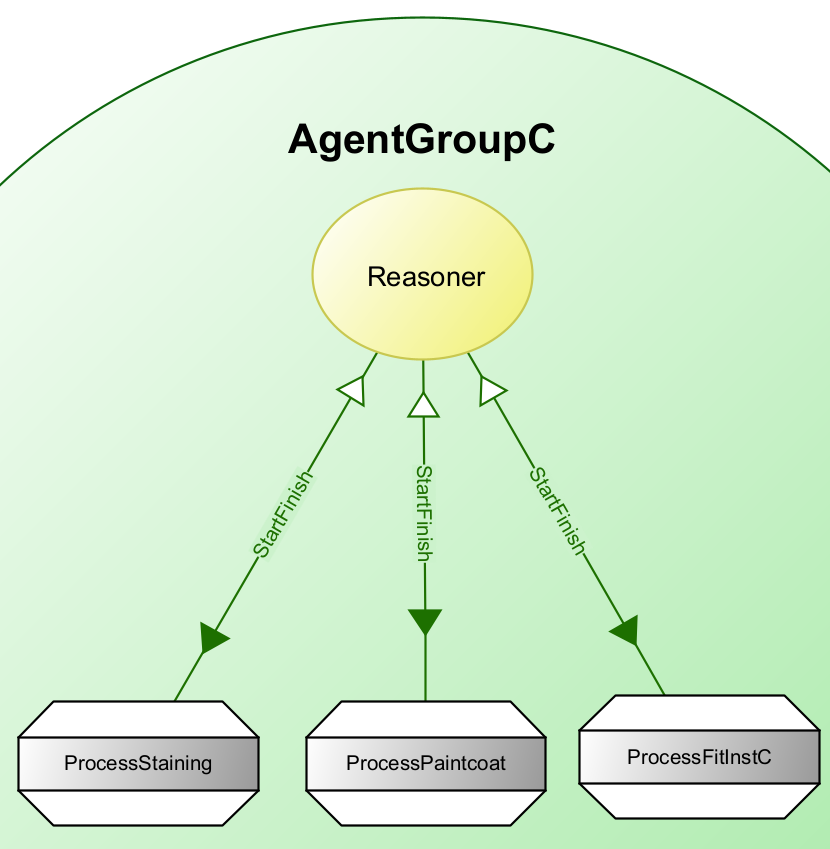
Predstavuje agenta, ktorý obsahuje referencie na všetkých stolárov skupiny B. Na správy typu request od agenta *AgentFurnitProd* s kódom *AssignCarpenterB* pošle správu *response* s inštanciou voľného stolára, ak taký existuje. V opačnom prípade pošle v správe hodnotu *null*. Zastrešuje pomocou kontinuálneho 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 vyťaženosti 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 vyťaženosti pracovnej skupiny C. Na komunikáciu sa využíva správa typu *TechStepMessage.class*.



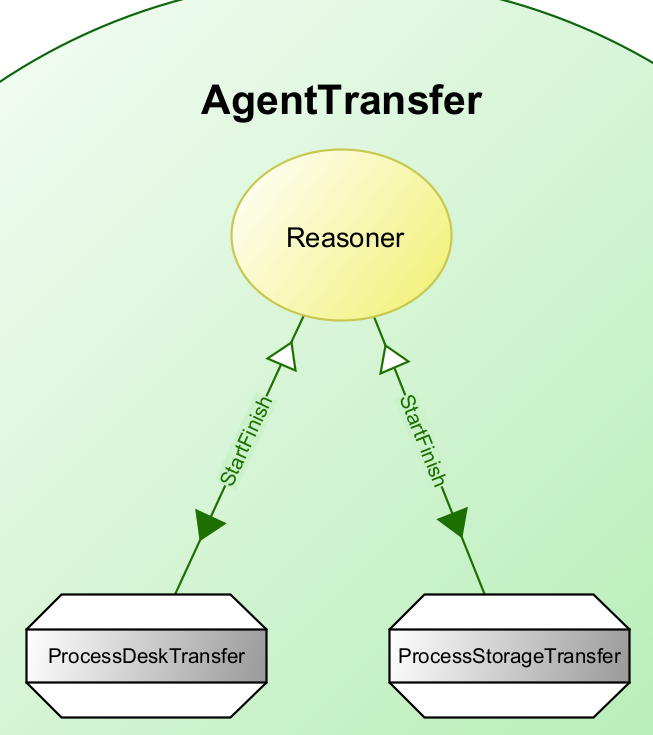
Obrázok 6 - asistenti agenta pre stolárov skupiny C

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 skladom a montážnym pracoviskom

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



Obrázok 7 - asistenti agenta simulujúceho pohyb stolára

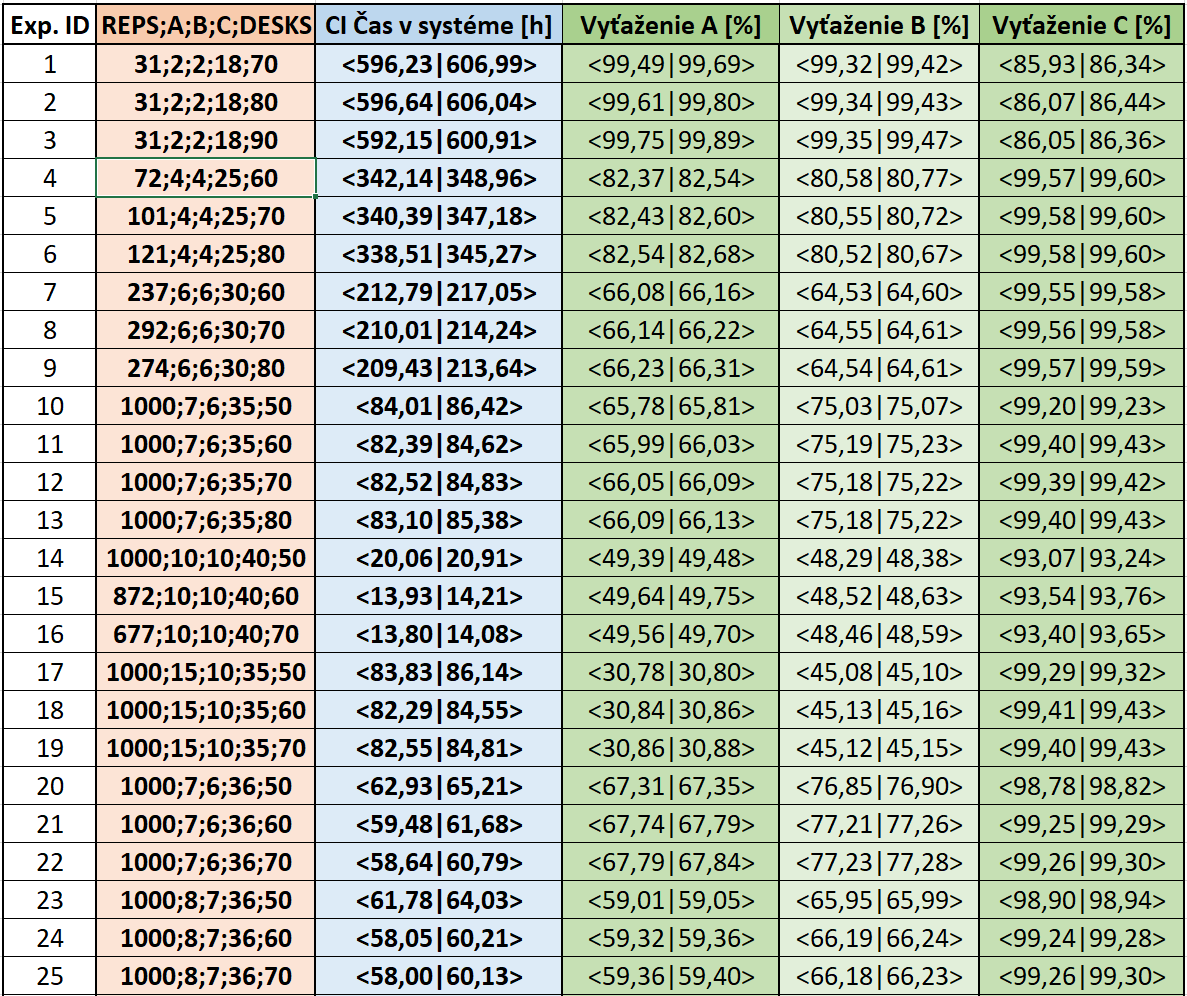
# Experimentálna časť

Experimentov bude potrebné urobiť viacej, 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.

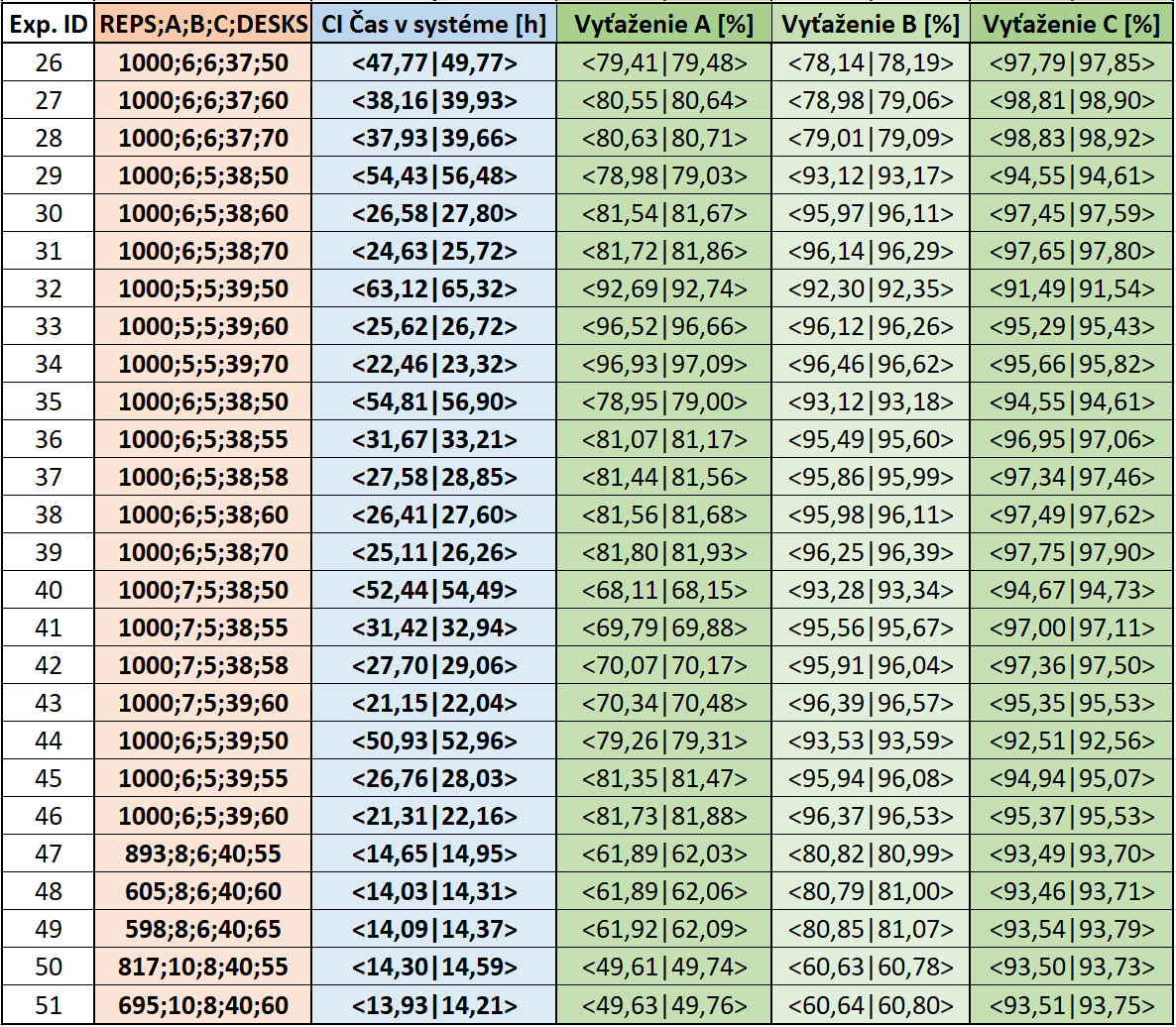
Vstupné úvahy:

* 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 prideľovať 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 8 a 9 môžeme vidieť výsledky pre všetkých 51 simulovaných konfigurácií. Každá bola vykonaná pre 249 simulačných dní, pričom každý deň mal pracovnú dobu 8 hodín (do času spracovania objednávky sa zahŕňal iba celkový pracovný čas).



Obrázok -výsledky experimentov (A)



Obrázok - výsledky experimentov (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 presiahnúť 32 pracovných hodín. Tomuto cieľu najviac vyhovujú konfigurácie **30** a **33** spolu so 49 stolármi a 60 montážnymi miestami. Skúsme teraz ešte znížiť počet montážnych miest na 58 (**obrázok 10**). Vidíme, že sme dosiahli ešte **lepší výsledok** a **zachovali kritériá**.



Obrázok - najvhodnejšia konfigurácia