

Enterprise Dynamics Tutorial

Enterprise Dynamics 10

Written by: InControl Simulation Solutions

Jaap Bijzerweg 21A

3446 CR Woerden

The Netherlands

www.incontrolsim.com

Date: 19/09/2022 15:07

Version: 10



Voorwoord

Voor het mogen besturen van een auto is het hebben van een rijbewijs een wettelijk vereiste. Het is het aantoonbare bewijs van een met succes afgelegde instructie waarbij behalve de beheersing van het voertuig ook de regels voor veilig gebruik in de omgang met andere weggebruikers getoetst zijn. Weinigen zullen de illusie hebben dat het doorlezen van het meegeleverde boekje in het handschoenenvak van het dashboard volstaat voor gebruik van een auto.

Voor simulatiesoftware met zijn uitgebreide mogelijkheden bestaat geen rijbewijs, maar ook hiervoor is training, voor zowel de beheersing van de software alsook het verantwoord toepassen bij simulatieproblemen, een vereiste. En naar analogie van meegeleverde boekwerk bij een nieuwe auto: alleen de handleiding doorploegen is zelden voldoende.

Enterprise Dynamics® (ED), van InControl Simulation Solutions in Woerden is zo'n softwareprogramma voor (discrete) simulatie. Het wordt geleverd met een Engelstalige handleiding in een aantal delen. Echter vanuit beginnende Nederlandse gebruikers -en in het bijzonder vanuit het onderwijs- kwam de vraag naar een Nederlandstalige, meer didactisch opgezette instructie waarmee deze groep gebruikers snel en zelfstandig de eerste beginselen van ED zou kunnen leren.

Dit heeft geleid tot deze tutorial en vormt een eerste aanzet om aan deze vraag te voldoen. De tutorial vormt een belangrijk onderdeel van de Educational Suite: het geheel van software en modern onderwijsmateriaal met als doel studenten uit te dagen en docenten een flexibel en gevarieerd onderwijsprogramma aan te bieden. Andere onderdelen van deze Educational Suite zijn onder meer een verzameling casussen (probleem, uitwerking volgens een simulatiemethodiek, bijbehorende ED-modellen) en een studentenversie ED. Over een leerboek simulatie wordt nog nagedacht.

In de toekomst zal deze tutorial herhaaldelijk aangepast en uitgebreid worden, wat alleen al voor nieuwe ED-versies noodzakelijk is. Een woord van dank gaat uit naar alle mensen die hebben bijgedragen aan de ontwikkeling van deze tutorial, in het bijzonder naar onze gebruikers die suggesties hebben gedaan hoe we de ED Tutorial en het Help-systeem verder kunnen verbeteren. Wij nodigen u uit om al uw bevindingen met deze tutorial of de software, suggesties voor verbetering, enzovoorts te sturen aan Support@incontrolsim.com.

Het Enterprise Dynamics Development en Support team en het Enterprise Dynamics Training & Education team.

1 Inhoudsopgave

1.	Inhoudsopgave	1-2
2.	De opzet van de Tutorial	3-4
2.1.	De didactische invalshoek	3-4
2.2.	Notatie	3-4
2.3.	De opbouw van de tutorial	3-4
2.4.	Het leren van simuleren	3-4
3.	De achtergrond van ED	5
4.	De eerste kennismaking met ED	6-8
4.1.	Enterprise Dynamics opstarten	6-8
4.2.	Uitleg van de schermonderdelen	6-8
4.3.	De menustructuur	6-8
4.4.	De structuur van de bibliotheek en het model	6-8
5.	De basis van het bouwen van een model	9
5.1.	Atomen in het model slepen	9-12
5.2.	De kanalen	13-17
6.	Het bekijken van resultaten	18-20
6.1.	Manieren om de resultaten te meten	20
6.2.	Het meten van de resultaten	20-31
7.	Spelen met strategieën	32-33
7.1.	Het aanpassen van de inputstrategie	33
7.2.	Het aanpassen van de Queue Discipline	33-34
7.3.	Het aanpassen van het Send to Statement	34-35
8.	Meer atomen van Assembler tot Unpack	36-40
9.	Enterprise Dynamics en Excel	41
9.1.	De bank	41-42
9.2.	De koppeling met Excel	42-44
9.3.	Wegschrijven van gegevens naar Excel	44
9.4.	Inlezen van gegevens uit Excel	44-45
9.5.	To pool or not to pool?	45-46
10.	De Human Resource	47
10.1.	Aan het werk met de Human Resource	47-50

10.2. Verplaatsen in de ruimte	50-51
10.3. Lopen (vrije beweging)	51-53
10.4. Netwerken in het algemeen	53-54
10.5. Lopen via een netwerk	54-55
10.6. Prioriteiten toekennen	55-56
10.7. Meer mogelijkheden van de Human Resource	56
11. De Transporter	57
11.1. Functionaliteiten van de Transporter(s)	57
11.2. Werken met de Transporter	57-59
11.3. De (Advanced) Transporter in het netwerk	59-61
11.4. De samenhang van Dispatcher, Advanced Transporter en Destinator	61-64
11.5. Laad- en losstrategieën van de Advanced Transporter	64-65
11.6. Het gebruik van meerdere transporters	65-67

2 De opzet van de Tutorial

2.1 De didactische invalshoek

Deze tutorial leert een beginnende gebruiker de weg in het simulatiepakket Enterprise Dynamics, verder afgekort als ED.

Het beginpunt in hoofdstuk 4 is simpelweg het openingsscherm van ED. Van daaruit leert men via een stapsgewijze aanpak steeds meer van ED kennen. Dit gebeurt meestal aan de hand van (beknopte) casussen: kleine problemen in een bepaalde context. Door het stap voor stap bouwen van of juist experimenteren met een bijbehorend model worden de functionaliteiten van ED duidelijk tegen de achtergrond van een toepassing.

Deze aanpak heeft een aantal voordelen:

1. De gebruiker leert ED kennen via het modelleren van een simulatieprobleem wat overeenkomt met datgene waarvoor men in de praktijk het pakket benut.
2. Functionaliteiten van een programma onthoudt men beter wanneer deze gekoppeld zijn aan toepassingen.
3. De gebruiker kan terugvallen op een aantal oplossingen in voorbeelden wanneer hij of zij geconfronteerd wordt met een soortgelijk probleem.

Kortom, deze tutorial tracht niet alleen te vertellen *wat* een onderdeel van de software doet, maar ook *hoe* en *wanneer* het gebruikt wordt. Sommige onderdelen, met name de overzichten in de bijlagen hebben daarnaast een functie als naslagwerk.


Om misverstanden te voorkomen: het is beslist niet zo dat deze instructie de (Engelstalige) handleiding -die onder de helpfunctie te vinden is- overbodig maakt. Gebruikers die verder willen komen in het bouwen van simulatiemodellen zullen juist veelvuldig de handleiding raadplegen omdat deze een complete weergave van alle functionaliteiten in de software vormt.

2.2 Notatie

Vanwege de Nederlandstalige opzet zijn Engelse termen soms vertaald, maar vaak is gekozen om termen niet te vertalen vanwege enerzijds het ontbreken van een geschikt Nederlands equivalent en anderzijds omdat dit het aanleren van ED zou bemoeilijken (vooral in gevallen van Engelse menuopties).

Wanneer naar een submenu wordt verwezen, gebeurt dit door het symbool |. Zo verwijst File | Preferences naar het submenu Preferences (hieronder vallen een aantal standaardinstellingen) dat onder het File menu te vinden is. Benadrukken van belangrijke zaken gebeurt door het **vet** of *cursief* schrijven van woorden. De casussen en de bijbehorende vragen worden in dit lettertype weergegeven. Wanneer er in een tekst 4DScript code staat opgenomen, wordt dit lettertype gebruikt.

Een waarschuwing wordt voorafgegaan door: *Let op!* Een nuttige tip wordt in veel gevallen weergegeven in een Tip box:

 Nuttige tips zijn te vinden in Tip boxes.

2.3 De opbouw van de tutorial

Na een korte algemene inleiding in hoofdstuk 3 over het programma ED, begint in hoofdstuk 4 de kennismaking met het pakket zelf in de vorm van een overzicht van de menustructuur en de onderdelen van het openingsscherm van ED. De basisbeginselen van het pakket worden gedemonstreerd in hoofdstuk 5 aan de hand van een eenvoudig wachtrijprobleem met één loket. Hieraan worden later twee loketten toegevoegd om het concept van de zogenaamde kanalen te tonen. De gebruikte atomen zijn de Source, Queue, Server en Sink.

In hoofdstuk 6 staan de verschillende manieren om resultaten van een studie te bekijken centraal. Dit gebeurt aan de hand van een casus in een timmerfabriek, waarin onder andere het werken met batches duidelijk wordt. De gebruiker leert daarbij als nieuwe onderdelen de Monitor, het Summary Report, diverse grafieken, en de Experiment Wizard kennen. Daarnaast oefenen we op elementair niveau de stappen in een simulatiestudie van modelbouw, validatie, experimentopzet en analyse van resultaten.

In hoofdstuk 7 ligt de nadruk op het bekijken van de voorgeprogrammeerde manieren waarmee producten toegang tot een volgend atoom verkrijgen (inputstrategie), geplaatst worden in een wachtrij (rijdiscipline) of na bediening doorverwezen worden (send to).

Hoofdstuk 8 introduceert acht nieuwe atomen waaronder de Assembler en de Conveyor, terwijl hoofdstuk 9 gewijd is aan de koppeling tussen ED en Excel. Dit hoofdstuk is tevens een start in 4DScript o.a. via de belangrijke

labels.

Hoofdstuk 10 introduceert de Human Resource atomen en Hoofdstuk 11 de Transporter.

Naast deze tutorial wordt aangeraden ook de Engelstalige tutorial *A first start in 4DScript* door te nemen. Deze tutorial vormt het vertrekpunt voor het leren van onderliggende programmeertaal in ED.

2.4 Het leren van simuleren

Simulatie is geen eenvoudige techniek: een verstandig gebruik van een simulatieprogramma vraagt kennis van de theoretische achtergronden van (discrete) simulatie zoals kennis van kansverdelingen en het proces van modelbouw zoals validatietechnieken of het opzetten van experimenten. Hoewel de tutorial ook op dit terrein kennis aanbrengt, is het naar zijn aard geen leerboek in simulatie. Hiervoor wordt naar bestaande boeken of dictaten over discrete simulatie verwezen.

Het ontwikkelen van gevoel voor simulatie en het bouwen van (simulatie)modellen in het bijzonder is vooral een kwestie van veel oefenen.

Kijk regelmatig op de website voor vernieuwingen. Hierop zal in de nabije toekomst onder andere het eerste deel van deze tutorial in diverse talen te downloaden zijn.

De voorbeeldmodellen in deze tutorial zijn meegeleverd bij de studentversie. Gebruik ze bij twijfel over de juistheid van de eigen modellen!

3 De achtergrond van ED

Enterprise Dynamics is een objectgeoriënteerd softwareprogramma voor het modelleren, simuleren en visualiseren van dynamische processen. Vanuit standaard bibliotheken kunnen gebruikers onderdelen -atomen genaamd- halen voor het bouwen van hun model. ED is gebaseerd op dit idee van atomen als bouwsteen van ieder model.

Een atoom kan een machine, loket of product voorstellen, maar ook een niet-fysiek karakter hebben zoals een grafiek. Omwille van het onderscheid in het soort atomen zullen we spreken van *basisatomen* (een vijftal veelgebruikte atomen: product, source, sink, server en queue), *transport* atomen (gerelateerd aan transport), *results* atomen enz.

Door de open structuur van ED kan de (gevorderde) gebruiker met de Atom Editor zijn eigen atomen bouwen. Denk bijvoorbeeld aan een machine met zeer specifieke eigenschappen. Op dit moment bevat ED zo'n 100 standaardatomen, maar dit aantal groeit nog steeds. De beginnende gebruiker zal voor zijn of haar toepassingen met zo'n 30 veelgebruikte atomen kunnen volstaan.

De atomen zijn dus voorgedefinieerde bouwstenen om snel modellen te kunnen maken en een studie uit te voeren. Daarnaast bestaat binnen ED een eigen programmeertaal 4DScript geheten, waarmee specifieke zaken uit de werkelijkheid in het model verwerkt kunnen worden. In deze tutorial wordt hier slechts summier op ingegaan.

Door de open structuur van ED kan de (gevorderde) gebruiker ook zelf bibliotheken en applicaties met een eigen layout ontwikkelen.

In Enterprise Dynamics kan een probleemeigenaar allerlei problemen modelleren. Door middel van experimenteren kan een oplossing voor een probleem gevonden worden of een specifieke vraag beantwoord worden. Voordelen van het gebruik van Enterprise Dynamics in de industrie en de logistiek zijn:

- De mogelijkheid om nog niet bestaande systemen te testen tijdens het vroege stadium van het ontwerp.
- Testen en verbeteren van voorgestelde wijzigingen die bijvoorbeeld voortgekomen zijn uit Lean en of Six Sigma studies, zonder dat het impact heeft op de operationele omgeving.
- Modelleren en analyse van scenarios, zoals Rough-Cut capaciteitsplanning.
- Optimalisatie en bescherming van investeringsplannen voor productie en transportmiddelen.
- Inschatten van de invloed van onzekerheden en variatie, zoals uitvallen van machines en variabele procestijden op het presteren van het systeem.
- Analyse en visualisatie van operationele systemen in 2D en 3D.

ED Educational is het pakket voor het onderwijs en is een combinatie van de software Enterprise Dynamics met een aantal uitgebreide case studies die de leraar kan gebruiken als opdracht voor de studenten.

Deze tutorial is bruikbaar om de basis vaardigheden te leren van elke Enterprise Dynamics applicatie, maar gaat ook dieper in op een aantal atomen die geleverd worden als onderdeel van Enterprise Dynamics. Daarnaast worden onderwerpen aangesneden en vragen gesteld die een belangrijk onderdeel vormen bij elke simulatie studie. Deze tutorial vormt daarmee ook een belangrijk onderdeel van ED Educational.

InControl Simulations Solutions is een simulation solution provider: naast bouw, ontwikkeling en verkoop van de Enterprise Dynamics software levert het ook uitgebreide diensten op het vlak van simulatie zoals trainingen en consultancy. De consultancy betreft zowel het uitvoeren van simulatiestudies als applicatiebouw.

4 De eerste kennismaking met ED

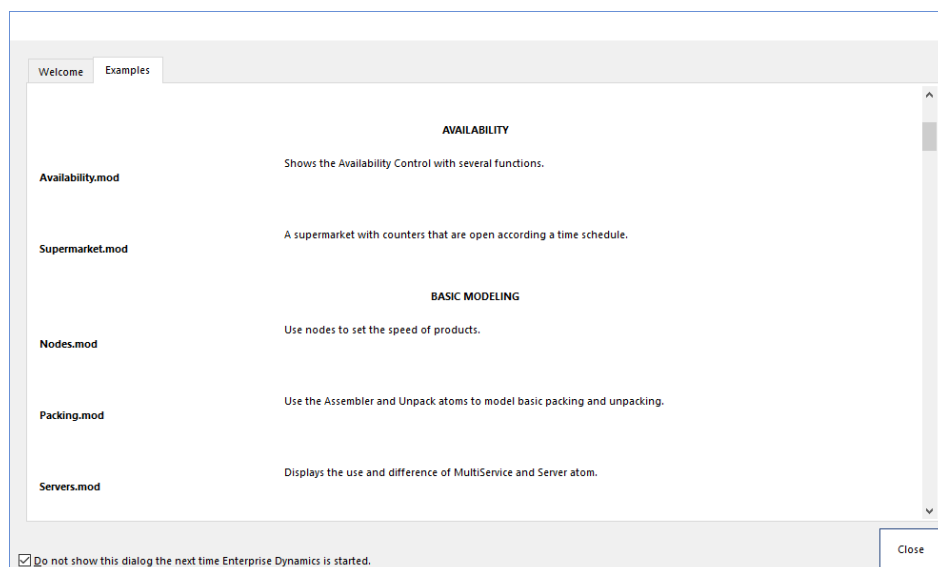
4.1 Enterprise Dynamics opstarten

Enterprise Dynamics kan opgestart worden via het Start Menu. Bij het opstarten wordt eerst een opstartscherm getoond, zie Figuur 4-1.



Figuur 4-1: Splash screen

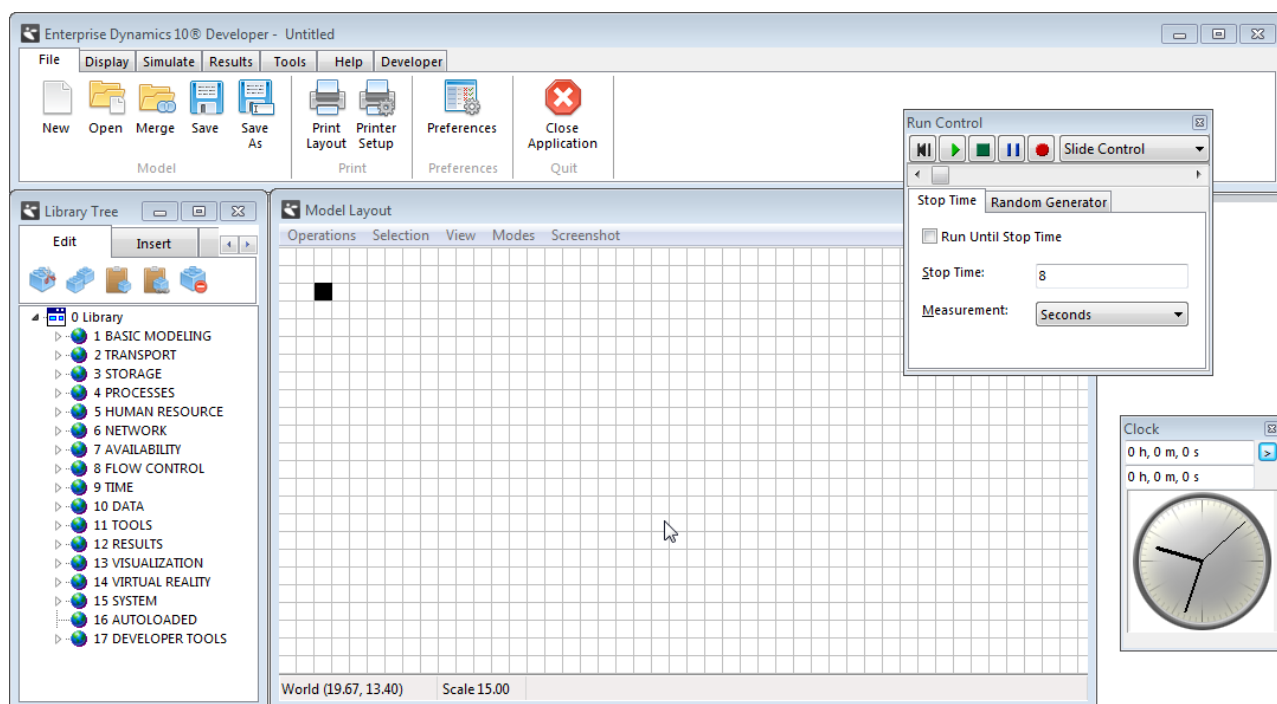
De eerste keer dat je Enterprise Dynamics start krijg je de Example Wizard getoond, zie Figuur 4-2. Deze wizard biedt enkele van de beschikbare voorbeeldmodellen aan die geleverd worden bij Enterprise Dynamics. Klik op een voorbeeldmodel om deze direct in te laden. Deze modellen zijn ook te vinden vanuit het Help menu.



Figuur 4-2: Example pagina van de Example Wizard

4.2 Uitleg van de schermonderdelen

Zodra Enterprise Dynamics helemaal is gestart, zou het openingsscherm er ongeveer uit moeten zien als in Figuur 4-3.



Figuur 4-3: Opbouw openingsscherm Enterprise Dynamics

Dit scherm is opgebouwd uit de volgende onderdelen:

- Een *menubalk*: hiermee kunnen onder andere bestanden worden geopend en opgeslagen. In paragraaf 4.3 wordt er verder op de menustructuur ingegaan.
- De bibliotheek (*Library Tree*): hier staan alle atomen opgenomen die een gebruiker in het model kan plaatsen. Elk atoom heeft een bepaalde functie en door de juiste atomen te combineren, is het mogelijk een bedrijfsproces in Enterprise Dynamics na te bouwen ('modellieren'). In hoofdstuk 5 komt het bouwen van de modellen aan de orde.
- Het model scherm (*model layout window*): in dit deel van het scherm worden de modellen gebouwd.
- De *run control*: hiermee wordt de simulatie gestart en kan de snelheid van de simulatie worden ingesteld.
- De *clock*: Toont de gesimuleerde tijd.

4.3 De menustructuur

De werking en het uiterlijk van de menu's is gelijk aan de menu's in andere Windows applicaties, zoals Word en Excel. In de onderstaande tabel wordt bij de meest gebruikte menuopties een korte uitleg gegeven.

In de menubalk staat het hoofdmenu. De onderdelen hiervan zijn:

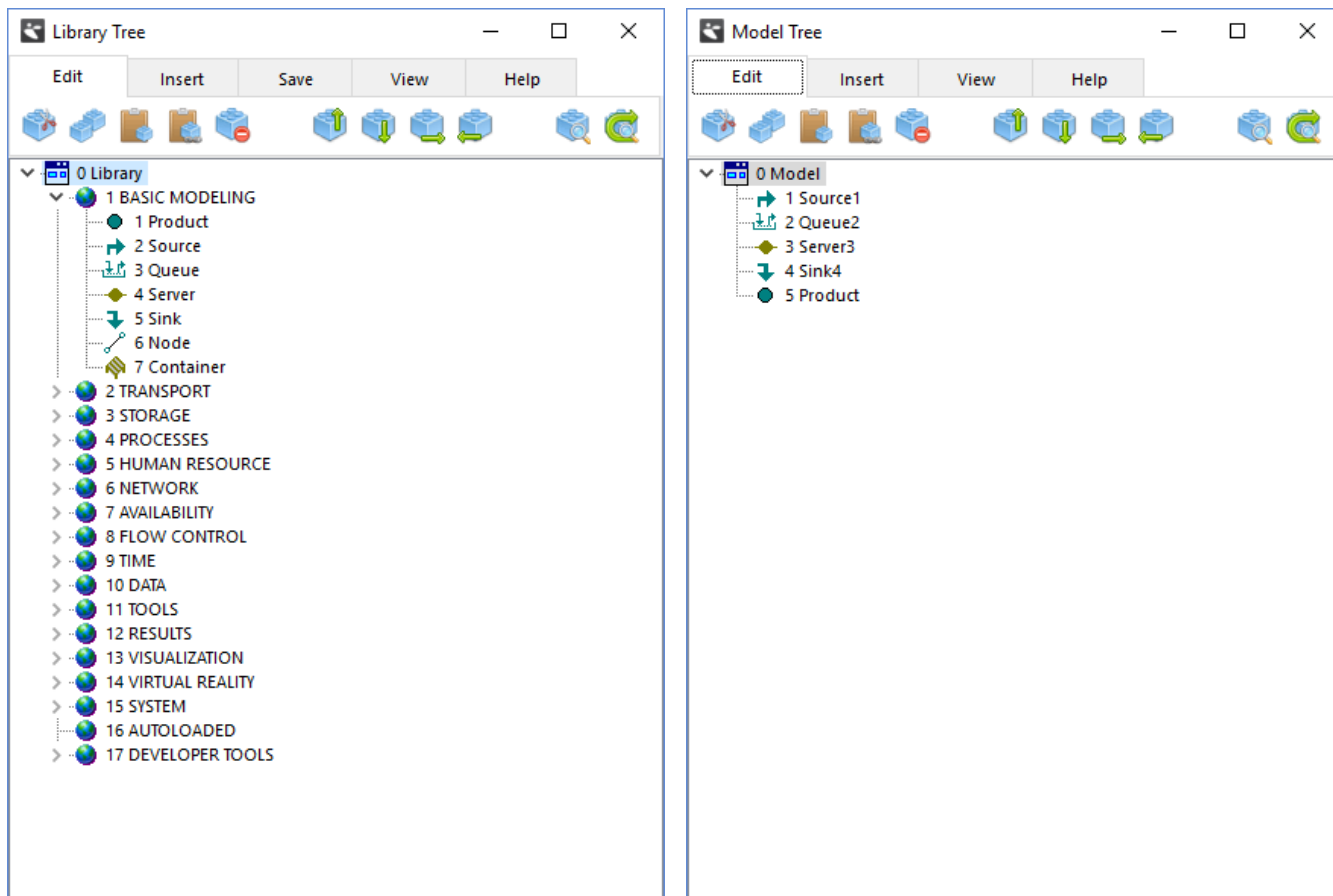
File	Voor het maken, openen of bewaren van bestanden of het instellen van voorkeursinstellingen van ED en van standaardfunctionaliteiten zoals afdrucken of locaties van bestanden.
Display	Voor het openen van de 2D en 3D modelweergave <i>Builder</i> voor het bouwen en <i>Viewer</i> voor afspelen.
Simulate	Voor het openen van een Run Control en Clock Window en het ontwerpen en uitvoeren van experimenten.
Results	Voor rapportages en grafieken over een enkele simulatierun of het analyseren van resultaten van een experiment.
Tools	Bevat onder andere het 4DScript interact Window om 4DScript code uit te testen en de Autofit tool om een passende verdeling te vinden bij data.
Help	Open de handleiding, de Example Wizard met alle voorbeeld modellen en Open the documentation, the Example Wizard alsook om bedrijfs- en product versie-informatie te vinden.
Developer	Handige tools voor ontwikkelaars om bijvoorbeeld bibliotheken te creëren.

Achter ieder hoofdmenu-item gaan submenu's schuil, zie de documentatie voor een volledige beschrijving.

4.4 De structuur van de bibliotheek en het model

In ED gebruiken we een boomstructuur voor het visualiseren van de ordening van atomen. Hiermee geven we aan welke atomen welke andere atomen bevatten. Zo geeft de 'main tree' een compleet overzicht van de applicatie, de bibliotheek en het geopende model. Twee andere belangrijke boomstructuren zijn:

- De Library Tree (bibliotheek) zie Figuur 4-4, hierin staan alle atomen die de gebruiker in het model kan plaatsen. De atomen zijn ingedeeld in groepen. Zo is er onder meer een groep *Transport* en een groep met *Human resources*. Door een atoom te selecteren en naar het model scherm (*Model Layout* of *2DBuilder*) te slepen, wordt het atoom in het model geplaatst.
- De Model Tree, hierin staan alle atomen die in het model geplaatst zijn. Tijdens de simulatie run wordt de Model Tree niet gerefreshed. Druk op de functie toets F5 om de Model Tree te refreshen.



Figuur 4-4: De Library Tree (bibliotheek) & Model Tree

De 2DBuilder ook wel Model Layout genoemd en de Library Tree kunnen geopend worden via de Display pagina van het hoofdmenu de snelkoppellingsknoppen worden getoond in Figuur 4-5.



Figuur 4-5: Knoppen voor het openen van de Library Tree en de 2D Builder/Model Layout scherm

5 De basis van het bouwen van een model

Nadat in het vorige hoofdstuk alleen theorie over Enterprise Dynamics is besproken, wordt er in dit hoofdstuk begonnen met het bouwen van een simpel model in Enterprise Dynamics. Overigens ligt hierbij de nadruk op het leren werken met ED en niet op het volledig verantwoord uitvoeren van een simulatiestudie. Aan het einde van dit hoofdstuk moet de gebruiker in staat zijn een model te ontwikkelen waarin meerdere machines parallel gebruikt worden.

5.1 Atomen in het model slepen

De eerste stap bij het bouwen van een model, is het plaatsen van de juiste atomen in het model. In deze paragraaf beginnen we met het bouwen van een eenvoudig model dat bestaat uit de volgende vier onderdelen, zie ook Figuur 5-1.

Source: dit atoom zorgt ervoor dat producten in het model binnenkomen.

Queue: dit atoom is een wachtrij voor klanten of producten.

Server: dit atoom dient als machine of loket. Atomen ondergaan hier een verandering en verblijven een bepaalde tijd (de bewerkingstijd) in dit atoom.

Sink: dit atoom zorgt ervoor dat klanten of producten het model weer kunnen verlaten.

Casus 1

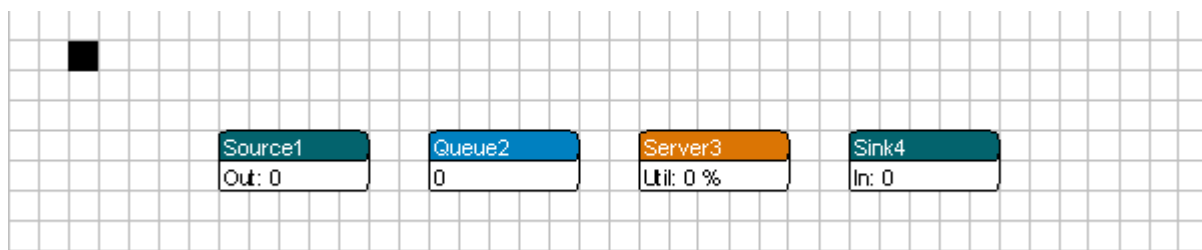
Bij een postkantoor in Amsterdam komen gemiddeld 20 klanten per uur binnen en de medewerkster van het postkantoor heeft gemiddeld twee minuten nodig om een klant te helpen. Uiteraard komen er het ene uur wat meer klanten binnen en het andere uur wat minder. Ook de twee minuten die de medewerkster van het postkantoor nodig heeft, zal variëren. Een klant die alleen postzegels wil kopen heeft immers minder tijd nodig, dan een klant die een nieuwe rekening wil openen. De klanten worden op volgorde van binnenkomst geholpen.

Een paar klanten hebben geklaagd over de wachtrijen en omdat de manager van het postkantoor de service aan zijn klanten erg belangrijk vindt, wil hij dit laten onderzoeken.

Vragen en opdrachten

1. Hoe hoog is de bezettingsgraad van de medewerkster? Wat betekent dit?
2. Valt er op basis van deze informatie een schatting van de gemiddelde wachtrij te maken?
3. In welke resultaten van een studie zal de manager interesse hebben?

Simulatie is een manier om inzicht te krijgen in de gemiddelde wachtrij. Omdat dit het eerste model is dat we gaan bouwen, wordt het model stap voor stap besproken. We gebruiken daarvoor de 4 atomen die in het begin van deze paragraaf zijn besproken en plaatsen deze in de hierboven besproken volgorde in ons model. Als eerste wordt dus de **Source** geplaatst, daarna de **Queue**, de **Server** en als laatste de **Sink**. Het plaatsen van de atomen gebeurt het slepen van de genoemde atomen uit de Library. Deze atomen zijn in de library allemaal te vinden onder de categorie "BASIC MODELING". Klik de "+" behorende bij de eerste categorie "BASIC MODELING" van de bibliotheek om deze te openen. Wanneer dit gedaan is, ontstaat Figuur 5-1.

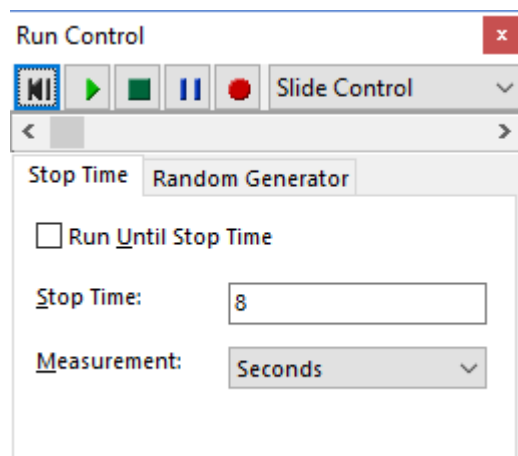


Figuur 5-1: Het eerste model

Bij vergissingen volstaat om het atoom te selecteren door erop te klikken en om vervolgens met de Deletetoets het atoom te verwijderen.

Om te controleren of de klanten in de goede volgorde door het model heen lopen, gaan we de simulatie starten. Dit doen we met behulp van het Run Control scherm. Als dit scherm nog niet is geopend, ga dan naar de Simulate pagina in het hoofdmenu en druk op de Run Control knop. Zorg ervoor dat in het Run Control scherm het onderdeel Slide Control is geselecteerd, zie Figuur 5-2. Hierdoor is het mogelijk om met de schuif de snelheid van

de simulatie aan te passen.



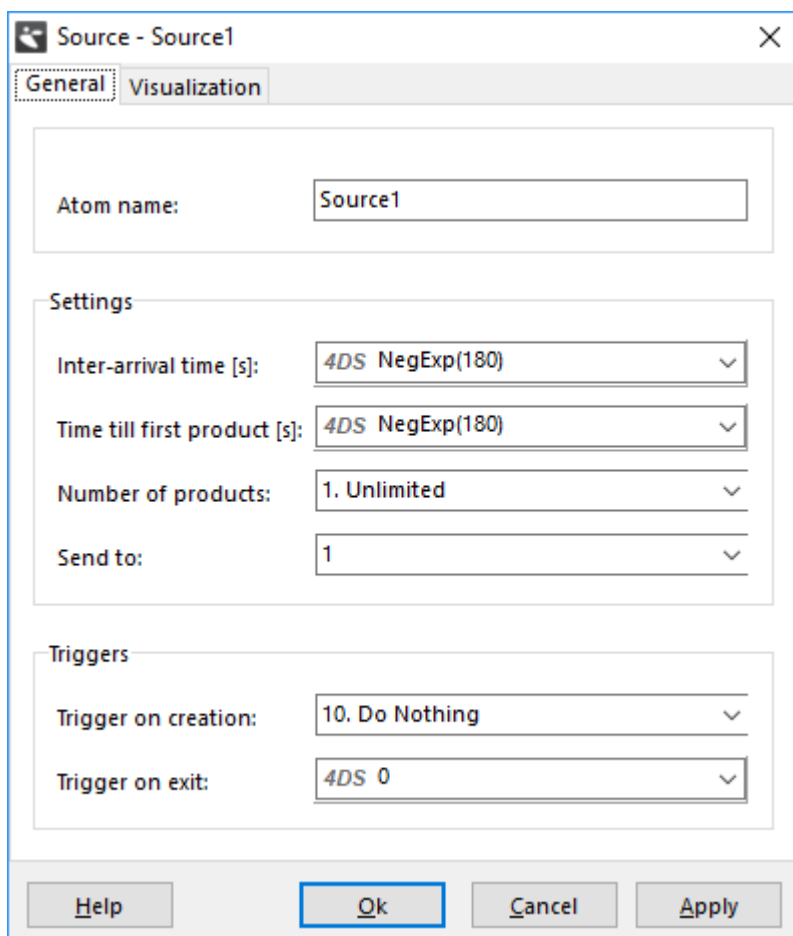
Figuur 5-2: Het Run Control scherm met de optie *Slide Control* geselecteerd

Voordat de simulatie gestart kan worden, moet er eerst op de Reset knop gedrukt worden. Dit is de meest linkse knop in het Run Control scherm. Nadat er op de Reset knop is gedrukt, verschijnt er een blauw bolletje naast het Source atoom. Dit is een product atoom. Door nu op de startknop te drukken (groene driehoek) zullen producten, in dit geval klanten, zich door de andere atomen, hier het postkantoor, verplaatsen. Pas zo nodig de snelheid aan, door het blokje te verslepen dat onderaan in het Run Control window te vinden is.

In de regel zal er in dit voorbeeld geen wachtrij voor het loket (Server) verschijnen. Het blauwe bolletje is dus alleen zichtbaar in het loket (Server). Daarnaast is er in de Server een percentage zichtbaar. Dit percentage geeft de bezettingsgraad van de Server weer. Wanneer het percentage niet zichtbaar is, kan het nodig zijn om het scherm in of uit te zoomen. Dit is mogelijk door gelijktijdig op de linker en rechter muisknop te drukken en de muis naar voren of naar achteren te bewegen of door het scrollwiel te gebruiken.

We hebben nu de atomen in de juiste volgorde geplaatst, maar hebben nog niet ingevuld hoeveel klanten per uur binnenkomen en hoeveel tijd de medewerkster nodig heeft om een klant te helpen. Voordat we deze tijden in Enterprise Dynamics kunnen invullen moeten we er rekening mee houden dat alle tijden in Enterprise Dynamics worden genoteerd in seconden. Wanneer de medewerkster twee minuten nodig heeft om een klant te helpen, noteren we dit in Enterprise Dynamics als 120 seconden. Daarnaast is het alleen mogelijk om op te geven hoeveel tijd tussen twee aankomsten van klanten zit, de zogenaamde tussenaankomsttijden (inter-arrival time).

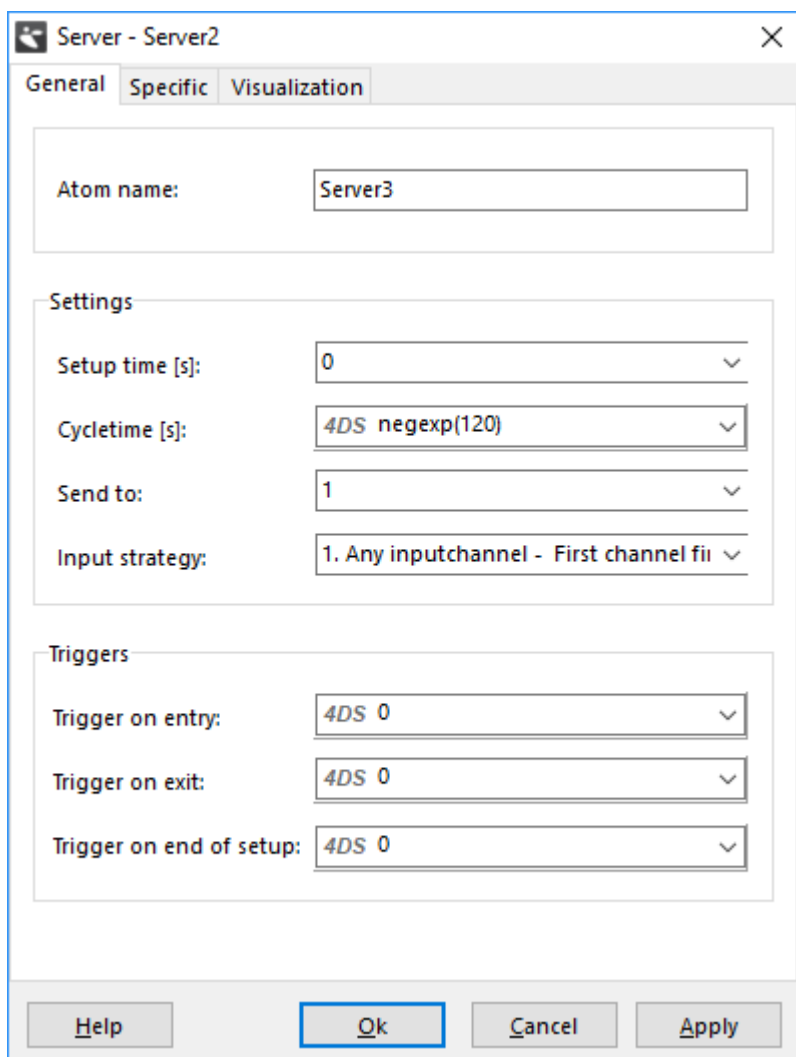
Als eerste veranderen we de instellingen van de Source, zodat er per uur gemiddeld twintig nieuwe klanten het postkantoor binnen komen. Door op het Source atoom te dubbelklikken opent er een invoerscherm, zie Figuur 5-3. Bij het veld Inter-arrival time kan de tussenaankomsttijd worden opgegeven. De onzekerheid in het aankomst proces kan worden opgevangen door gebruik te maken van een kansverdeling. In dit voorbeeld gebruiken we de exponentiële verdeling, een kansverdeling die veel gebruikt wordt voor grillige aankomst- of bedieningsprocessen. In Enterprise Dynamics moet daar de volgende 4DScript code voor worden gebruikt: `NegExp (e1)`. Hierbij staat de parameter e1 voor de gemiddelde waarde of verwachting van de exponentiële distributie. Vul nu als tussenaankomsttijd 180 in en bevestig met Ok:



Figuur 5-3: Invoerscherm van de Source

Belangrijk: Rechts van de invoervelden staat vaak een naar beneden wijzende driehoek. Enkel- of Dubbelklikken geeft een lijst met beschikbare en voorgedefinieerde opties. Kijk eerst hierin of er voor jouw probleem een bruikbare optie tussenstaat! Voor een uitleg van deze opties zie de Help documentatie. Hierin staan de belangrijkste atomen en hun invoervelden volledig beschreven. De grijze tekst “4DS” betekent dat hier 4DScriptcommando’s, de programmeertaal van ED, ingevuld kunnen worden.

Vervolgens vullen we ook de tijd in die een medewerker nodig heeft om 1 klant te helpen. Ook hier gebruiken we weer de exponentiële verdeling. Dubbelklik nu op het Server atoom, zodat de bedieningstijd (Cycle time) kan worden aangepast, zie Figuur 5-4. Pas op het General tabblad de cycle time nu zo aan dat de medewerkster gemiddeld twee minuten nodig heeft voor een klant. Selecteer hiervoor “NegExp(10)” en klik op het 4DScript veld. Er verschijnt een 4DScript Editor scherm. Pas hierin het gemiddelde aan.



Figuur 5-4: Invoerscherm van de Server

Reset en start met Run Control de simulatie opnieuw. Doordat we de tijd tussen twee aankomsten hebben vergroot is het handig om de snelheid van de simulatie te vergroten. Door voldoende in te zoomen is het zichtbaar hoeveel klanten er in de rij staan te wachten.

Het is mogelijk om het blauwe bolletje dat nu een klant symboliseert te vervangen door een andere afbeelding. Dubbelklik hiervoor op het blauwe bolletje links van de source. Selecteer het tabblad Visualization. Bij het invoerveld 2D Icon dubbelklik je nogmaals op het blauwe bolletje. De Resource Manager verschijnt met daarin de iconen die de gebruiker kan selecteren. Klik op het gewenste nieuwe icoon en op OK in de Resource Manager: het blauwe bolletje in het Product scherm is vervangen door het nieuwe icoon.

Ook het icoon dat Enterprise Dynamics in de 3D weergave gebruikt kan worden aangepast. Gebruik hiervoor de optie 3D icon en kies uit de lijst voor Person. Bekijk het model nu in 2D en in 3D. Voor het openen van de 3D weergave: kies voor de hoofdmenu-optie Display en daarna voor één van de mogelijkheden van 3D-weergave. Door met de muis te bewegen en op de muisknoppen te klikken kan er door het 3D model worden genavigeerd.



Er zijn veel meer iconen beschikbaar. Klik op de Resource Manager button op de Tools pagina van het hoofdmenu om de manager te openen. Klik op Add om een bestaand icoon aan de standaardlijst (de Resource Manager) toe te voegen! Als een icoon is toegevoegd aan de Resource Manager dan kan je het icoon ook aan een product atoom toekennen.

In deze simulatiestudie zorgt de keuze voor een ander icoon slechts voor een mooiere simulatie. Wanneer er echter onderscheid gemaakt zou moeten worden tussen verschillende soorten klanten, kan het gebruik van meerdere iconen zorgen voor meer duidelijkheid. De ene groep klanten kan dan bijvoorbeeld een groen icoon krijgen en de andere een blauw icoon.

Het model kan nu opgeslagen worden. Ter controle is het model van dit postkantoor te vinden bij de tutorial-modellen onder de naam Postoffice1.mod.

5.2 De kanalen

Casus 1 (vervolg)

In de omliggende wijken worden 2 postkantoren gesloten. De directeur hoopt dat zijn postkantoor veel nieuwe klanten kan gaan verwelkomen en rekt zelfs op een verdubbeling van de klanten. Er zullen dus gemiddeld 40 klanten per uur binnen gaan komen in zijn postkantoor. Om er achter te komen of zijn postkantoor met het ene bestaande loket deze toestroom kan verwerken, besluit hij opnieuw een simulatie studie uit te voeren.

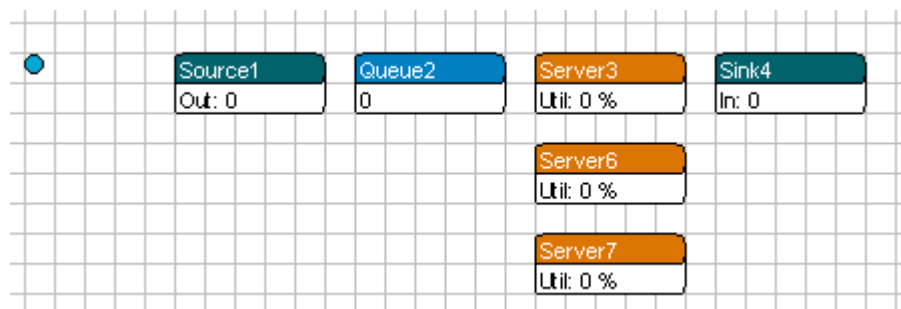
Vragen en opdrachten

4. Is het zinvol om in deze situatie te gaan simuleren? Maak voor het beantwoorden van deze vraag gebruik van de te verwachten bezettingsgraad en controleer het eigen vermoeden door het model postoffice1.mod aan te passen aan de nieuwe situatie en een simulatierun uit te voeren.


Casus 1 (vervolg)

Het blijkt dat één loket te weinig zal zijn om de toestroom aan te kunnen. De directeur wil voor de zekerheid zelfs drie loketten openen. Opnieuw wil hij graag inzicht in de gevolgen hiervan op de wachtrij.

Om te beginnen plaatsen we de twee loketten onder het reeds bestaande loket (kies het juiste atoom om een loket mee te modelleren). Vervolgens moeten de instellingen voor de cycle time van de loketten worden veranderd, zodat ze overeenkomen met het eerste loket. Maak hierbij eventueel gebruik van paragraaf 5.1. Het model zou er nu uit moeten zien zoals Figuur 5-5.

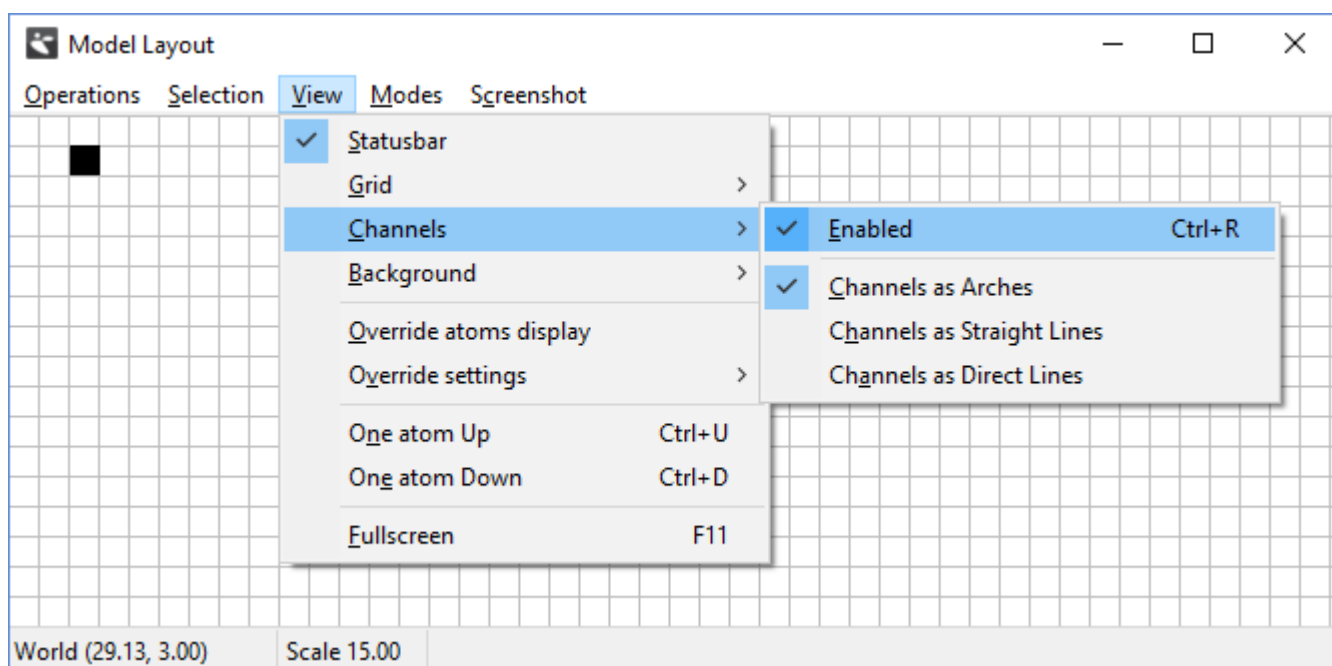


Figuur 5-5: Postkantoor met drie loketten

 Atomen kunnen simpel - met alle ingevulde velden en instellingen - gekopieerd worden! Klik daartoe op het te kopiëren atoom en vervolgens op F6, of kopieer en plak met Ctrl+C Ctrl+V. Selecteer voor het kopiëren van meerdere atomen met rechtsklikken op de muis een gebied, en vervolgens weer F6 of Ctrl+C Ctrl+V. Bovenstaande procedures in combinatie van de pijltjestoetsen zijn ook handig voor het verschuiven van meerdere atomen tegelijk.

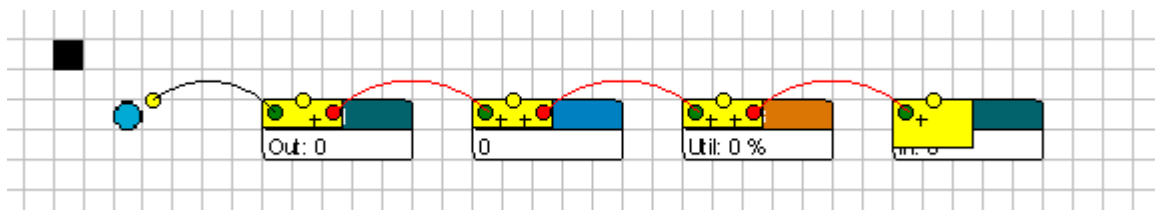
Wanneer de simulatie nu gestart wordt, gaan de klanten echter nog steeds slechts naar het bovenste loket. Om dit aan te passen moet gekeken gaan worden naar de manier waarop Enterprise Dynamics de klanten doorstuurt.

Kies in het menu View van het scherm Model Layout voor de optie Channels | Enabled, zie Figuur 5-6. Hierdoor worden de zogenaamde kanalen (Channels) van de atomen zichtbaar. Deze kanalen dienen in ED voor de routing van producten en de doorgave van informatiestromen en worden daarom hier uitgebreid besproken.



Figuur 5-6: Kanalen aanzetten

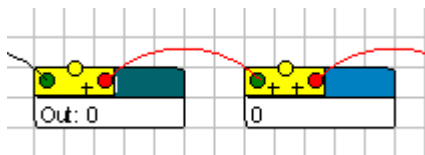
Als de enabled optie van de channels aanstaat, dan zie je de kanalen tussen de atomen zoals weergegeven in Figuur 5-7.



Figuur 5-7: het postkantoor model met de kanalen aanzet

Wanneer de lijnen tussen twee kanalen niet als bogen worden weergegeven, maar als hoekige lijnen, is het handig om de optie *Channels as Arches* in het menu View | Channels aan te zetten, zie opnieuw Figuur 5-6 en het resultaat in 5-7.

Zodra de kanalen zijn aanzet, wordt er aan alle atomen een blokje toegevoegd waarop de in- en uitgangskanalen van het betreffende atoom zijn weergegeven, zie Figuur 5-8.



Figuur 5-8: Verbonden kanalen

In het blokje zijn een aantal bolletjes te zien. Het linker bolletje is een ingangskanaal en het rechter is een uitgangskanaal. Een kanaal kan geopend (weergegeven in groen) of gesloten (weergegeven in rood) zijn. Wanneer zowel in- als uitgangskanaal open zijn, is de verbinding tussen het in- en uitgangskanaal gereed (weergegeven in groen), anders is de verbinding niet gereed (weergegeven in rood).

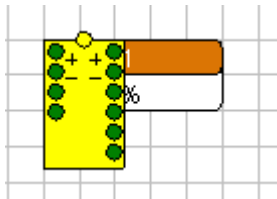
Het middelste bolletje (boven op het gele blokje) is bedoeld om informatie te versturen en te ontvangen, in deze tutorial zullen we dit kanaal als het centrale kanaal ('central channel') aanduiden. Dit centrale kanaal wordt gebruikt om informatie over een atoom te registreren. Er is altijd maar één centraal kanaal aanwezig op een atoom, maar het is wel mogelijk om meerdere atomen met hetzelfde centrale kanaal te verbinden.

Producten komen een atoom binnen via een ingangskanaal en verlaten een atoom via een uitgangskanaal. Een ingangskanaal behoort altijd verbonden te zijn met een uitgangskanaal of een centraal kanaal. Een uitgangskanaal behoort altijd verbonden te zijn met een ingangskanaal of een centraal kanaal. Ieder ingangs- of uitgangskanaal kan slechts met één ander kanaal verbonden zijn.


Door op het “+” teken naast een bolletje te klikken, wordt het aantal in- of uitgangskanalen vergroot en door rechts-klikken op een kanaal verkrijgt men een overzicht van alle verbindingen van het atoom in kwestie. Door te klikken op het “-“ teken dat verschijnt zodra er twee of meer in- of uitgangskanalen zijn, kunnen deze weer verwijderd worden.

Belangrijk: hoewel met kanalen enorm veel geregeld kan worden, is de essentie van kanalen het aangeven op een atoom (lees: wachtrij, loket) wat alle mogelijke volgende locaties zijn waar producten die zich in dat atoom bevinden, naartoe gestuurd kunnen worden. Daarmee verzorgen de kanalen een (ruwe) routing van producten door het model!

In Figuur 5-9 staat een atoom weergegeven met 4 ingangskanalen, 6 uitgangskanalen en het centrale kanaal.

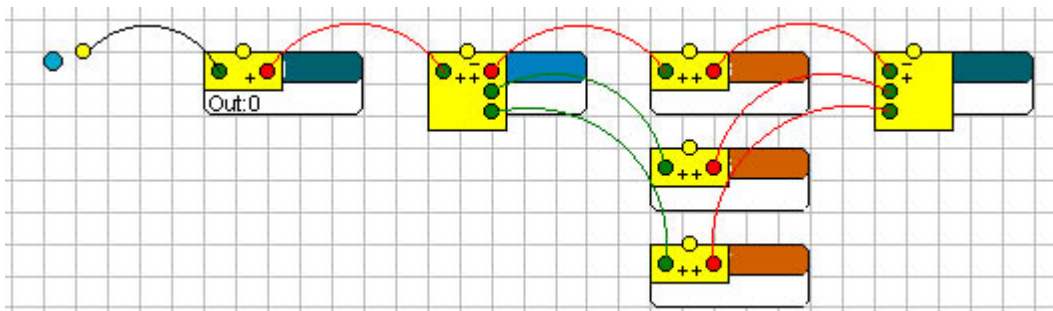


Figuur 5-9: Atoom met 4 ingangskanalen en 6 uitgangskanalen

 Voor mensen met kleurenblindheid kan het een probleem zijn om het verschil tussen een open en een gesloten kanaal te herkennen. Alle kleuren kunnen echter onder File | Preferences... | Appearances aangepast worden. Verander de kleur van “2D -> Open channel” en “2D -> Closed channel” en kies zelf twee voldoende contrasterende kleuren.

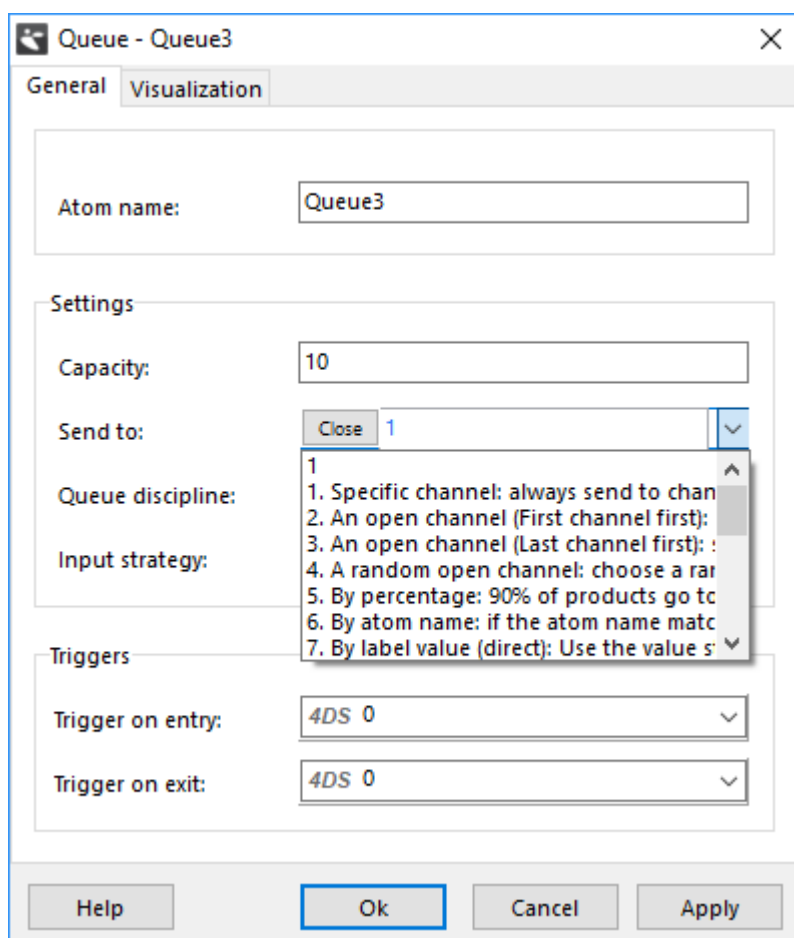
Door een lijn te trekken met de muiscursor tussen een uitgangskanaal en een ingangskanaal worden twee atomen met elkaar verbonden. Door een lijn te trekken tussen een in- of uitgangskanaal en het centrale kanaal van hetzelfde atoom, wordt een verbinding verbroken.

Verbindt de kanalen in het model in Enterprise Dynamics nu zodanig dat de klanten naar één van de drie loketten gaan en vervolgens vanuit een loket naar de uitgang gaan. Het resultaat zou er uit moeten zien als Figuur 5-10. Start de simulatie weer (niet vergeten te resetten!) en controleer of de klanten nu wel naar de juiste loketten gaan.



Figuur 5-10: Postkantoor met verbonden kanalen

De reden waarom de klanten nog steeds alleen voor het bovenste loket kiezen is dat in Enterprise Dynamics nog is opgegeven dat klanten (atomen) standaard via uitgangskanaal 1 moeten worden verstuurd. In het invoerscherm, dat zichtbaar wordt bij dubbelklikken op een atoom, is in te stellen via welk kanaal de producten de wachtrij moeten verlaten. Door nu op de Queue te dubbelklikken kunnen we de instellingen voor ons model in orde maken. We moeten hiervoor de waarde in het veld Send To aanpassen, zie Figuur 5-11: wanneer we willen dat ieder open kanaal gekozen kan worden zijn de opties 2, 3 en 4 relevant.



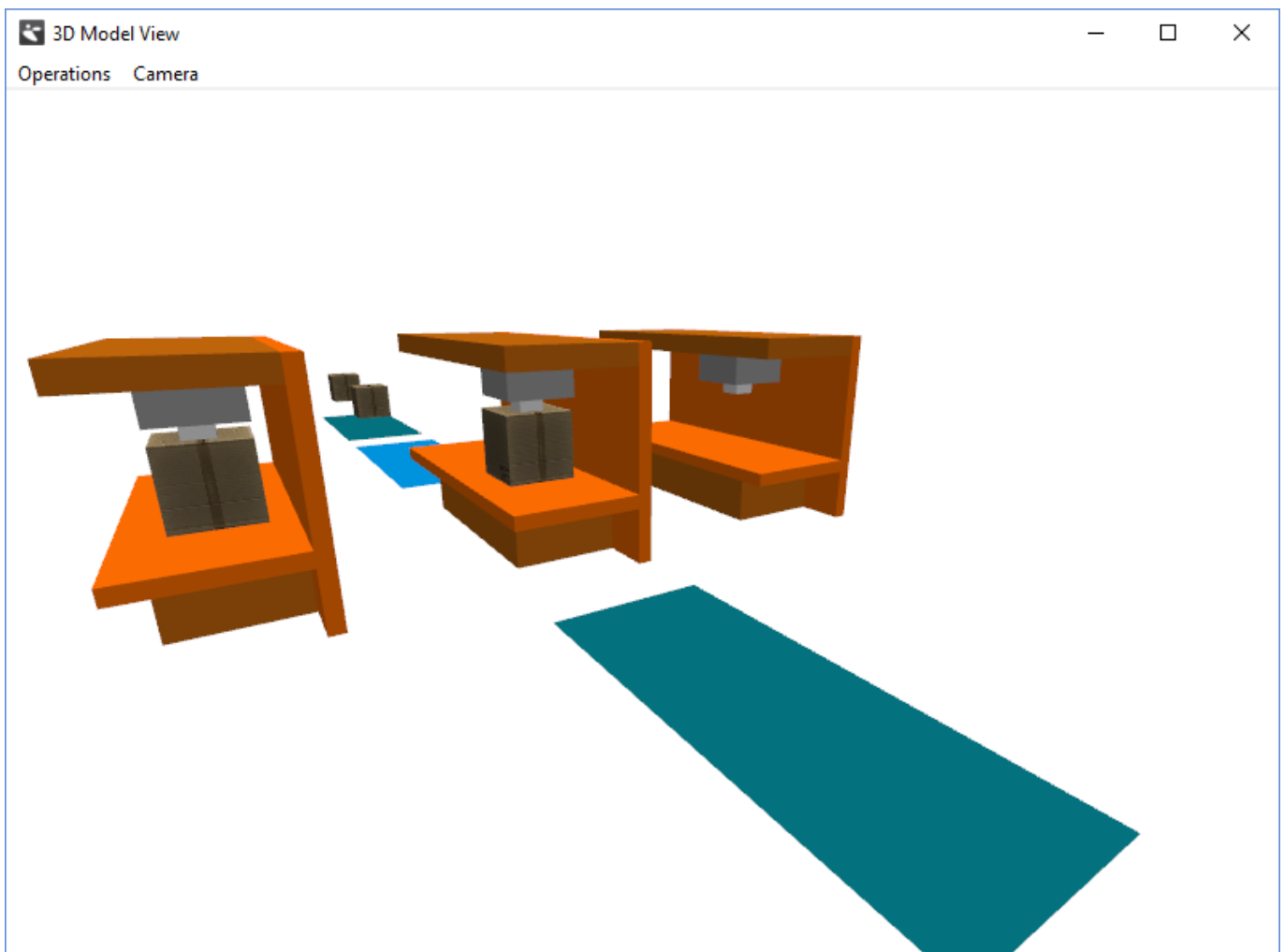
Figuur 5-11: Send to van de Queue

Bereken dat bij optie 2 resp. optie 3 de eerste resp. laatste server de hoogste bezettingsgraad zal krijgen, terwijl de bezettingsgraden van de servers bij optie 4 allen min of meer gelijk zullen zijn!

Bij het Send To invoerveld kun je een getal invullen, zelf met 4DScript een stukje programmacode schrijven wat het uitgangnummer oplevert of gebruikmaken van voorgedefinieerde code. Deze laatste mogelijkheid is zeker voor beginnende gebruikers de gemakkelijkste methode. Door in het invoerveld op het driehoekje met de pijl naar beneden te drukken, verschijnt een lijst met mogelijkheden waaruit de gebruiker zijn keuze kan maken. Nadat de keuze is gemaakt voor een bepaalde regel kan de tekst die in het blauw staat weergegeven worden aangepast. Uiteraard is die mogelijkheid alleen aanwezig indien er iets aangepast kan worden.

Kies nu zelf een strategie waarbij de bezettingsgraad van de drie loketten ongeveer even groot is. Wanneer de bezettingsgraden van de loketten niet direct zichtbaar zijn moet er worden in- of uitgezoomd (gebruik de beide muisknoppen).

Je kunt het model ook in 3D bekijken door in het hoofdmenu Display | 3D Viewer te klikken, zie Figuur 5-12.



Figuur 5-12: 3D model view met drie loketten gemodelleerd als machines

Vragen en opdrachten

5. Voldoet de opstelling met drie loketten?

Het model kan nu opgeslagen worden. Ter controle is het model van dit postkantoor te vinden bij de tutorial models onder de naam Postoffice2.mod.

6 Het bekijken van resultaten

In het vorige hoofdstuk is aan de hand van een uitgebreid stappenplan een postkantoor gesimuleerd. Daarbij is geleerd hoe atomen in het model worden geplaatst en hoe de kanalen worden verbonden. Daarnaast is er uitgelegd hoe ervoor gezorgd kan worden dat de atomen op de juiste momenten het model binnenkomen en de juiste tijd bij de machines (of loketten) verblijven. Alle basisprincipes voor het bouwen van een model zijn dus besproken.

In dit hoofdstuk bouwen we een complexer model en kijken vooral naar manieren om de resultaten van de simulatiestudie te volgen en te meten. Als eerste volgt weer een (eenvoudige) casus en (na het bouwen van een bijbehorend model) worden de verschillende manieren van resultaatmeting uitgelegd. Vervolgens oefenen we de toepassing hiervan in het model.

Casus 2: De timmerfabriek

Algemene context: Een timmerfabriek heeft meestal een eigen ontwerpafdeling en daaruit voortvloeiend een complexe besturing omdat iedere order in feite een apart project is waarvan de bewerkingstijden van de aparte productiestappen zo goed mogelijk geschat dienen te worden. Een dergelijke omgeving kent vaak grote tussenvoorraden, lange doorlooptijden en wisselende bottlenecks.

De directie van een timmerfabriek die voornamelijk ramen en kozijnen produceert wil meer inzicht krijgen in knelpunten, productieaantallen en doorlooptijden. Men besluit daartoe een simulatiestudie te laten verrichten. Vanwege de eenvoud wordt er in deze casus alleen gekeken naar de productie van ramen.

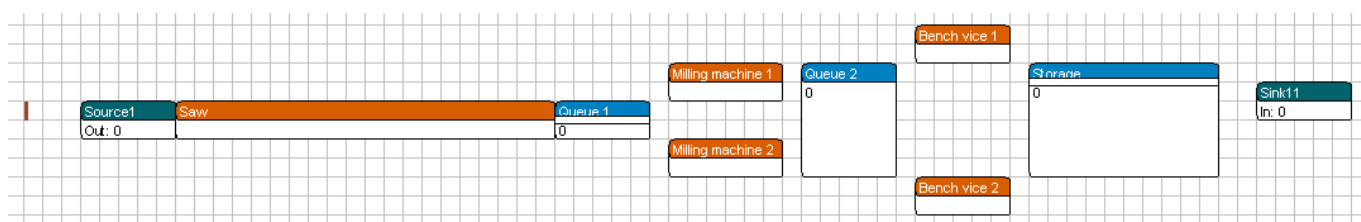
Het productieproces voor de ramen is op te delen in een aantal stappen. Als eerste komen er lange balken binnen, die op de kortzaagmachine gezaagd worden tot korte planken. Hierbij wordt uit één lange balk tien kleine planken gezaagd. Deze kleine planken worden vervolgens op twee freesbanken voorzien van het gewenste patroon. Wanneer de planken zijn gefreesd kunnen ze naar de opsluitbank. Bij deze opsluitbank worden steeds 4 planken in een frame geplaatst, waarna ze snel gelijmd worden. Zowel van de freesbank als van de opsluitbank zijn er twee identieke exemplaren aanwezig die parallel van elkaar in het productieproces zijn opgenomen. De planken worden dus slechts één keer gefreesd en gelijmd. Tussen alle productiestappen zijn steeds voorraadplaatsen met voldoende ruimte aanwezig. Er zijn voldoende balken aanwezig om de kortzaagmachine continu te laten produceren. Om de voorraad hout in het systeem te overzien, geldt dat de tussenvoorraden voor iedere bewerking niet meer dan 100 planken mogen bevatten.

Het verzagen van een balk op kortzaagmachine is uniform verdeeld tussen twee en drie minuten en de bewerkingstijd van een plank op de freesmachine is normaal verdeeld met een gemiddelde van 36 seconden en een standaarddeviatie van 2 seconden. De opsluitbank heeft precies twee minuten nodig om een raam in elkaar te lijmen. De productie gaat onafgebroken door van 9.00 tot 17.00. De producten die aan het einde van de dag nog niet klaar zijn, blijven liggen tot de volgende dag.

Vragen en opdrachten (analyse vooraf)

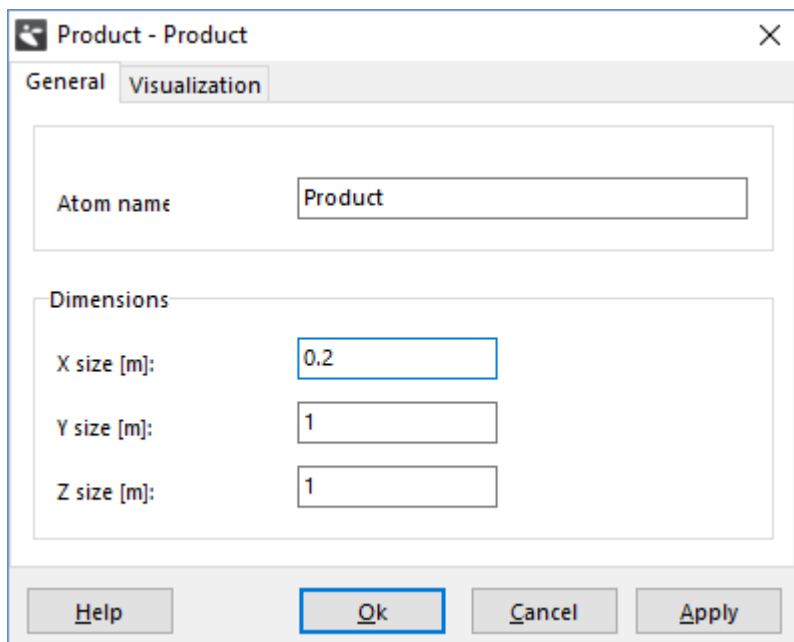
1. Maak een schets van het proces met bij ieder deelproces de capaciteit per uur in balken, planken of ramen.
2. Hoeveel ramen verlaten naar verwachting per dag de fabriek?

Bouw nu het model in Enterprise Dynamics. De kansverdelingen zijn te vinden in het pull-downmenu van de cycle times; let op dat voor de normale verdeling de uitdrukking $\max(0, \text{normal}(36, 2))$ gebruikt wordt om te verhinderen dat negatieve bedieningstijden ontstaan. Op het tabblad “Specific” van de server kun je instellingen wijzigen zodat met Batches van meerdere producten tegelijk gewerkt wordt. Zet de Batch Rule op “1 in, B out” met batch size 10 voor de 1:10 operatie van balken naar planken en op “B in, 1 out” met batch size 4 voor de 4:1 operatie van planken naar ramen. Verander de naam van de Servers, zodat het model eenvoudiger te lezen wordt. Als alles is goed gegaan, ontstaat een model zoals in Figuur 6-1.

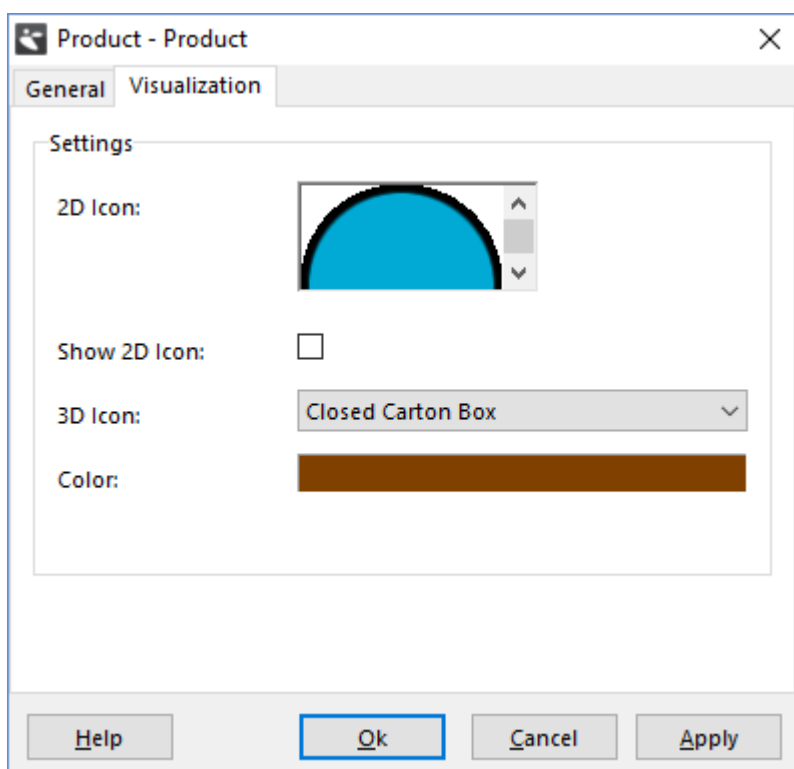


Figuur 6-1: Lay-out van de timmerfabriek

De atomen kunnen van formaat veranderd worden, zoals het Saw en Queue 2 atoom in Figuur 6-1, door het atoom eerst te selecteren en vervolgens het zwarte blokje in de linker bovenhoek of de rechter onderhoek te verslepen. Het productatoom (het blauwe bolletje) is te veranderen in een bruine `plank': dubbelklik op het product en wijzig het invoerscherm zoals weergegeven in Figuur 6-2 en 6-3.



Figuur 6-2: Algemene instellingen van het product aanpassen



Figuur 6-3: Visualisatie-instellingen van het product aanpassen

Bevestig de wijzigingen t.o.v. het oorspronkelijke invoerscherm. Wat is er nu gebeurd?

Het 2D Icon met het blauwe bolletje bestaat nog steeds, maar door het uitvinken van Show 2D Icon is alleen de bruine achtergrondkleur zichtbaar. Deze heeft een lengte van 0.2 m en een breedte van 1 meter, alsof het een staande balk betreft. In 3D speelt de hoogte (size Z) van 0.2 meter nog een rol. Merk op dat het producticoon in 2D en 3D kunnen verschillen. Kies in 3D bijvoorbeeld voor de Pallet of een ander icoon!

Het raster dat steeds achter de atomen zichtbaar is, fungeert als het coördinaatvlak waarbij ieder vak een

afmeting van 1 bij 1 meter heeft. Het zichtbare zwarte vak correspondeert met de coördinaten (0,0). Voor dit abstracte model van een timmerfabriek speelt de fysieke ruimte geen rol, maar in andere toepassingen zoals magazijnen met loopafstanden of heftrucks die een bepaalde afstand moeten afleggen, speelt dit duidelijk wel!

Sla het eigen model op. Bij onzekerheid over het eigen model kan voor de volgende paragrafen ook gebruik gemaakt worden van het meegeleverde model onder de naam timber1.mod.

6.1 Manieren om de resultaten te meten

Nu er een werkend model is, kunnen we gaan beginnen met het meten van de resultaten. Er zijn binnen Enterprise Dynamics een aantal manieren om deze resultaten te meten.

1. Informatie die op de atomen wordt afgebeeld.
Zo staat er op elke Queue afgebeeld hoeveel producten er in de rij staan, op een server staat de bezettingsgraad en op een Source of Sink staat hoeveel producten het model zijn binnengekomen c.q. hebben verlaten. Deze informatie is vooral bruikbaar om gedurende een simulatie te bekijken of het model logisch werkt.
2. De Result Atomen.
De result atomen kunnen gevonden worden in de groep Results van de Library Tree. Hier zijn verschillende atomen te vinden waar resultaten mee weergegeven of bewaard kunnen worden. Voorbeelden zijn de StatusIndicator en de StatusMonitor die respectievelijk de huidige toestand (bijvoorbeeld 'busy', 'blocked' of 'idle') van een atoom of de fractie van de tijd dat een atoom in de verschillende toestanden geweest is grafisch weergeven. Een Generic Monitor atoom kan grafische verschillende informatie over een bepaald atoom weergeven. De gebruiker kan instellen welke informatie moet worden getoond. Deze informatie is vooral bruikbaar om gedurende een simulatie te bekijken of het model logisch werkt.
3. Het Overzichtsrapport (Summary Report) en de grafieken vanuit het Results menu.
Hierin verschijnen (tussentijdse) resultaten van een lopende simulatierun. Het is vooral een handige methode om een snel overzicht te krijgen van de toestand van het systeem en een snelle controle van de parameterinstellingen.
4. Een experiment opstellen met de Experiment Wizard.
Deze manier verschilt wezenlijk van de vorige drie en wordt gebruikt voor de eigenlijke studie. Vooraf wordt ingesteld hoe lang een meetperiode duurt en hoe vaak die periode gesimuleerd moet worden bv. 10 keer een half jaar. Ook moet bij een experiment worden opgegeven wat er gemeten moet gaan worden. Het resultaat van een experiment verschijnt in de vorm van een betrouwbaarheidsinterval.

6.2 Het meten van de resultaten

In paragraaf 6.1 zijn een viertal manieren genoemd waarmee de resultaten van een simulatiestudie getoond kunnen worden. In deze paragraaf gaan we deze vier methoden toepassen op de timmerfabriek. Zorg ervoor dat het model van de timmerfabriek in Enterprise Dynamics geopend is.

1. Informatie die op het atoom wordt afgebeeld

Zorg ervoor dat de klok van Enterprise Dynamics zichtbaar is op het scherm (submenu van Simulate). Zet vervolgens de simulatie aan en kijk hoeveel producten er in acht uur tijd de fabriek verlaten. In het Settings menu op de pagina "Stop Time" van de Run Control is de optie "Run until stop time" te vinden. Vink deze optie aan, zet de stop tijd op 8 en selecteer bij "measurement" uren (hours). Als er nu een simulatie aangezet wordt zal deze precies 8 uur duren.

Vragen en opdrachten

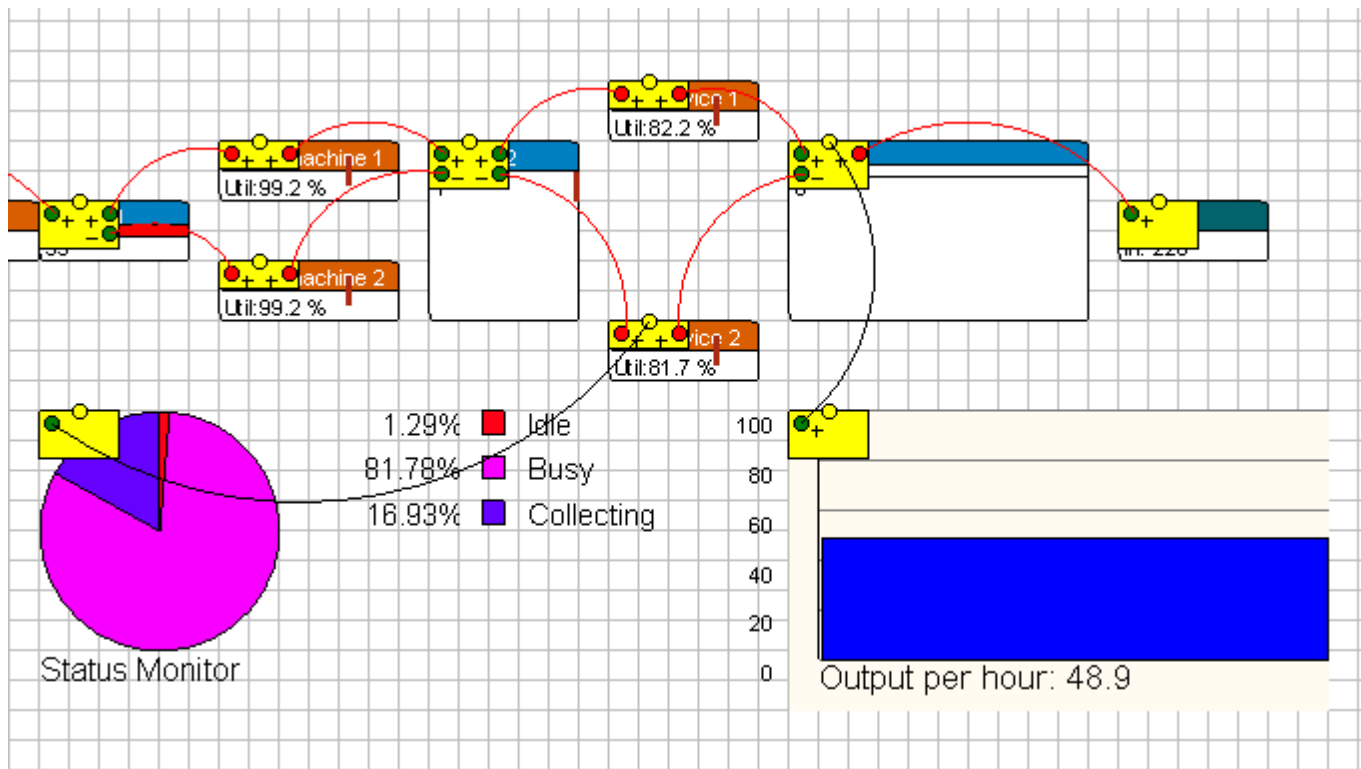
3. Voer dit een aantal malen uit met Simulate| Reset+ Run until stop time en noteer het aantal geproduceerde ramen. Komt dit overeen met de uitgevoerde analyse vooraf? Wat is het knelpunt (bottle neck) in dit proces?

2. Result atomen

Meerdere van de Result atomen zijn eenvoudig in hun gebruik. Zoek in de Library Tree de StatusMonitor en Generic Monitor atoom, die in de Results groep te vinden zijn. Merk op dat de StatusMonitor in de subgroep Status te vinden is. Sleep beide atomen in het model.

Om gebruik te maken van de Status Monitor, is het voldoende om zijn input kanaal te verbinden met een centraal kanaal van het atoom dat je wilt onderzoeken. Als de simulatie is gereset en gestart wordt dan zal de

StatusMonitor de toestand statistieken van het atoom weergeven. Verbind de StatusMonitor aan een van de opsluitbanken. Reset en start het model. Onderzoek welk fractie van de tijd de opsluitbanken in de toestand wachten op arriveren van planken, het verzamelen van planken als de eerste gearriveerd is en op het echte bewerken van de planken. Wanneer de kanalen zichtbaar zijn gemaakt zou het atoom er uit moeten zien als in figuur 6-4.

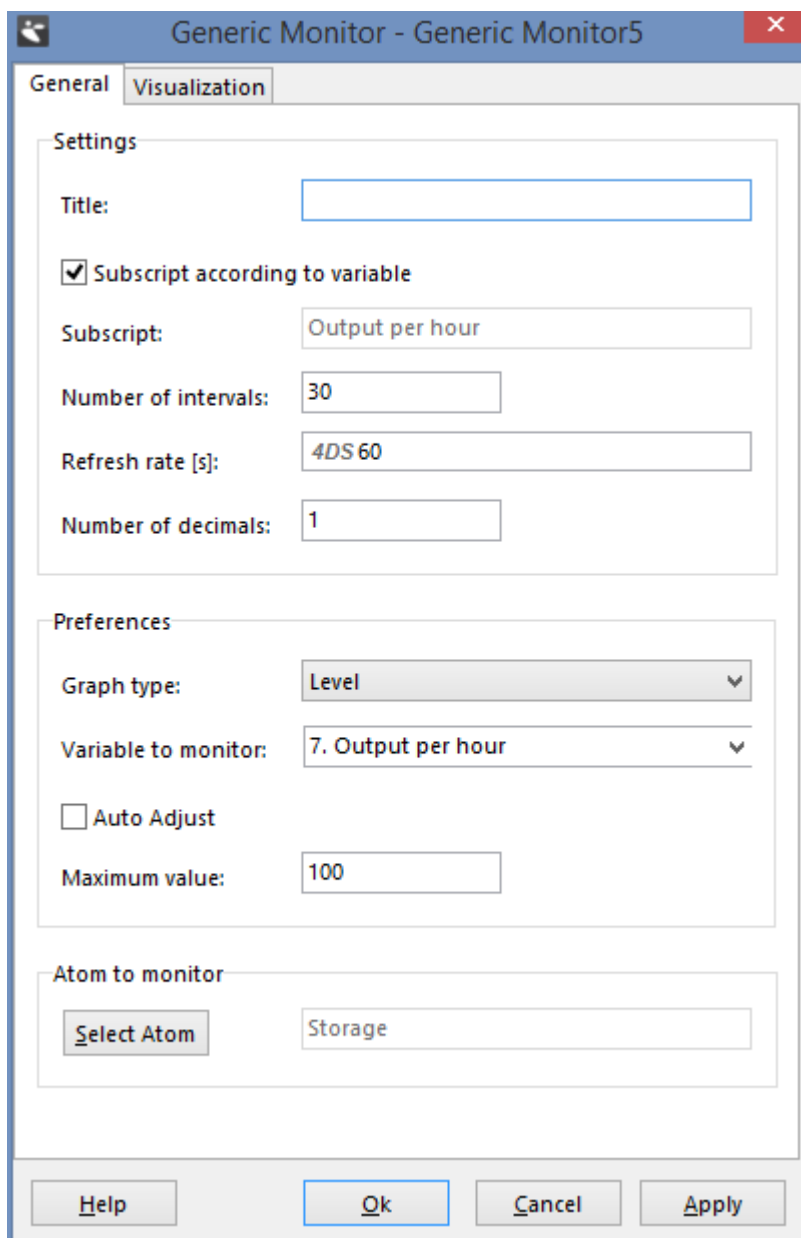


Figuur 6-4: Modelgedeelte met een (reeds verbonden) Result atomen; links een StatusMonitor en rechts een Generic Monitor

Het Generic Monitor atoom vergt iets meer werkt om te gebruiken. Sleep het monitor atoom vanuit de Resultgroep in de Library Tree in het model. De eerste stap is weer het verbinden van de monitor met het atoom dat bekeken gaat worden. In deze casus willen we weten hoeveel ramen per uur de fabriek verlaten. Verbind het ingangskanaal van de monitor met het centraal kanaal (het informatiekanal!) van de laatste Queue. Het is ook mogelijk om op de monitor te dubbelklikken en in het invoerscherm dat verschijnt op de knop 'Select Atom' te klikken en vervolgens het atoom te selecteren dat bekeken gaat worden.


Als tweede wordt opgegeven wat er bijgehouden moet gaan worden. Dubbelklik hiervoor weer op het monitor atoom, er verschijnt dan een invoerscherm, zie Figuur 6-5. Kies de juiste optie in het Variable to Monitor veld. In deze casus hebben we de optie "Output per hour" nodig. Wanneer de gevraagde meetvariabele niet in het lijstje voorkomt, is het noodzakelijk zelf een stukje 4DScript code te programmeren. Selecteer dan de optie 4DScript expression en klik op het vierkantje zodat een 4DScript editor verschijnt. Voor beginnende gebruikers is dit echter niet aan te raden

Nu wordt duidelijk waarom we een Queue voor de Sink geplaatst hebben: hierdoor kunnen we gebruik maken van de voorgedefinieerde logica "Output per hour", de Sink heeft namelijk geen output!



Figuur 6-5: InvoerScherm van de Generic Monitor

Als derde stap kunnen nog een aantal instellingen voor de monitor aangepast worden. Zet de *sample rate* op 60. Elke 60 seconden zal er nu een meting gedaan worden. Omdat het gemiddelde aantal geproduceerde ramen niet een geheel getal hoeft te zijn, zetten we de optie Number of decimals op 1. Probeer zelf de opties Auto-adjust, Maximum value en Show Y-axis uit. Zet de instellingen uiteindelijk zoals in Figuur 6-5 is aangegeven. Start nu de simulatie.

 Het gebruik van de resultaat atomen zoals de Generic Monitor kunnen de snelheid van de simulatie vertragen. Zeker als op de Generid Monitor een lage sample rate gekozen wordt van bijvoorbeeld 5 seconden. Hierdoor zal elke 5 **simulatie** seconde een berekening gedaan worden en wordt het resultaat opnieuw getekend, dit kost veel processortijd. Dit fenomeen treedt nog steeds op als de Model layout gesloten wordt. Vaak is het voldoende een sample rate van 60 of misschien wel 3600 seconde in te stellen.

Vragen en opdrachten

4. Hoeveel ramen verlaten er gemiddeld per uur de fabriek?

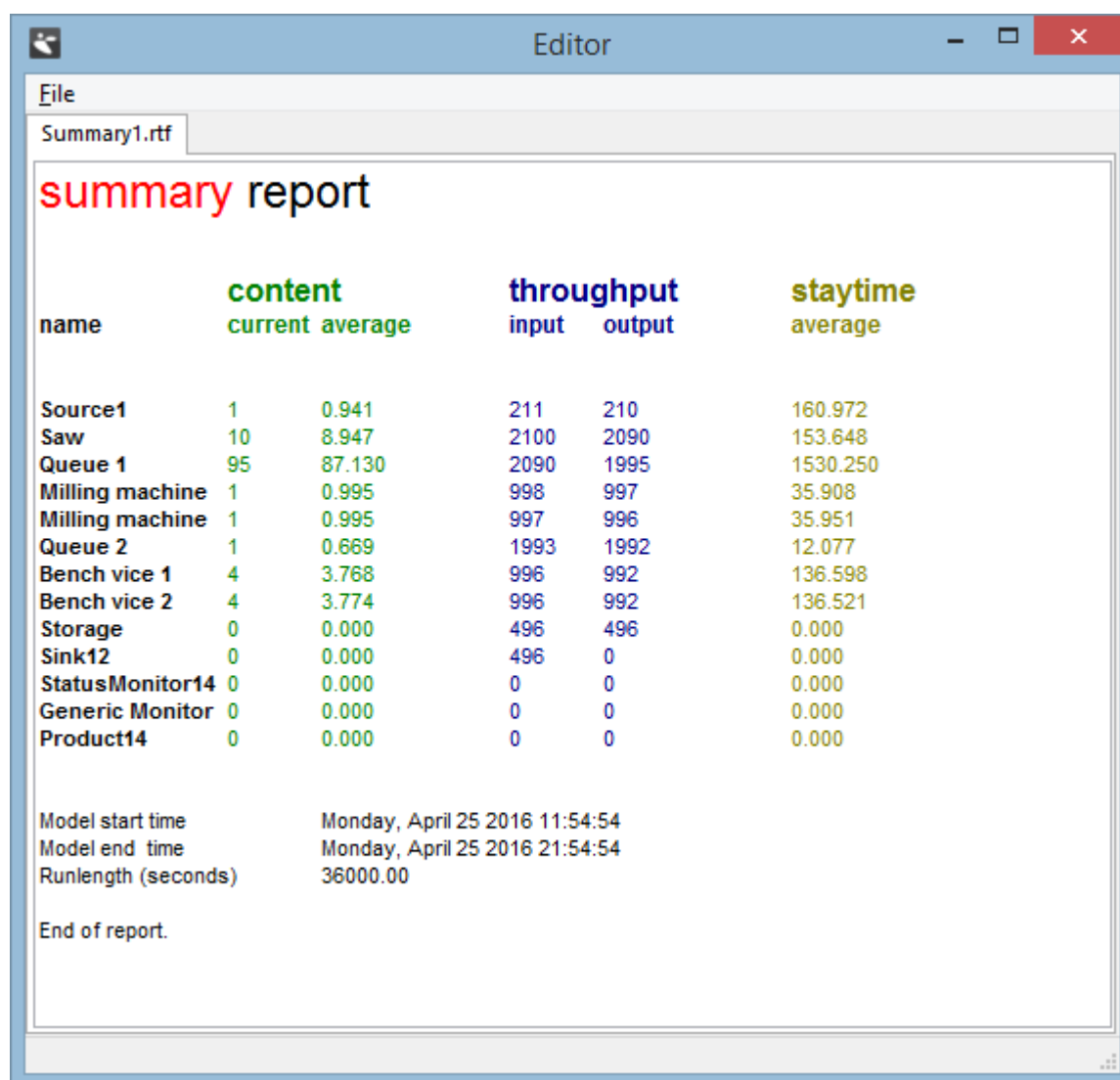
3. Het overzichtsrapport en grafieken

Een derde methode om resultaten te meten is door gebruik te maken van de Summary Report en Graphs die te vinden zijn op het tabblad 'Results'.

Als eerste zal de Summary Report (overzichtsrapport) worden besproken. Hiermee is een overzicht te krijgen van de toestand van het model, waarbij een opsplitsing wordt gemaakt naar de volgende onderdelen:

- Current Content:** het aantal producten dat in een atoom aanwezig is op het moment dat het rapport wordt aangemaakt.
- Average Content:** het gemiddeld aantal producten dat er in het atoom aanwezig was.
- Throughput Input:** het aantal producten dat het atoom is binnengekomen.
- Throughput Output:** het aantal producten dat het atoom verlaten heeft.
- Staytime average:** de gemiddelde verblijftijd van de producten in het atoom.

Door in het tabblad Results te klikken op Summary Report verschijnt er een rapport in het scherm zoals in Figuur 6-6.



Figuur 6-6: Summary report

Zo is in Figuur 6-6 af te lezen dat een product afgerond gemiddeld 36 seconden in Milling machine 1 aanwezig is, dat er 992 producten geproduceerd zijn in Bench vice 2 en er nog 4 aanwezig zijn.

Zoals je kunt aflezen uit de regel achter de Saw en de twee Bench vices, worden alle metingen op de servers waar in batches gewerkt wordt, gedaan in de kleine eenheid, dat wil zeggen de eenheid waaruit elke batch bestaat. Kijk bijvoorbeeld naar de Output van Source 1, dit zijn 210 producten. Bij de Saw is er een input van 2100. De metingen van de Saw zijn namelijk in kleine planken. Ook de metingen op de Bench vices gaan in kleine planken en niet in raamkozijnen.

Het is overigens mogelijk dat de output (niet de input) van de Saw niet een veelvoud van 10 is (bijvoorbeeld 2098

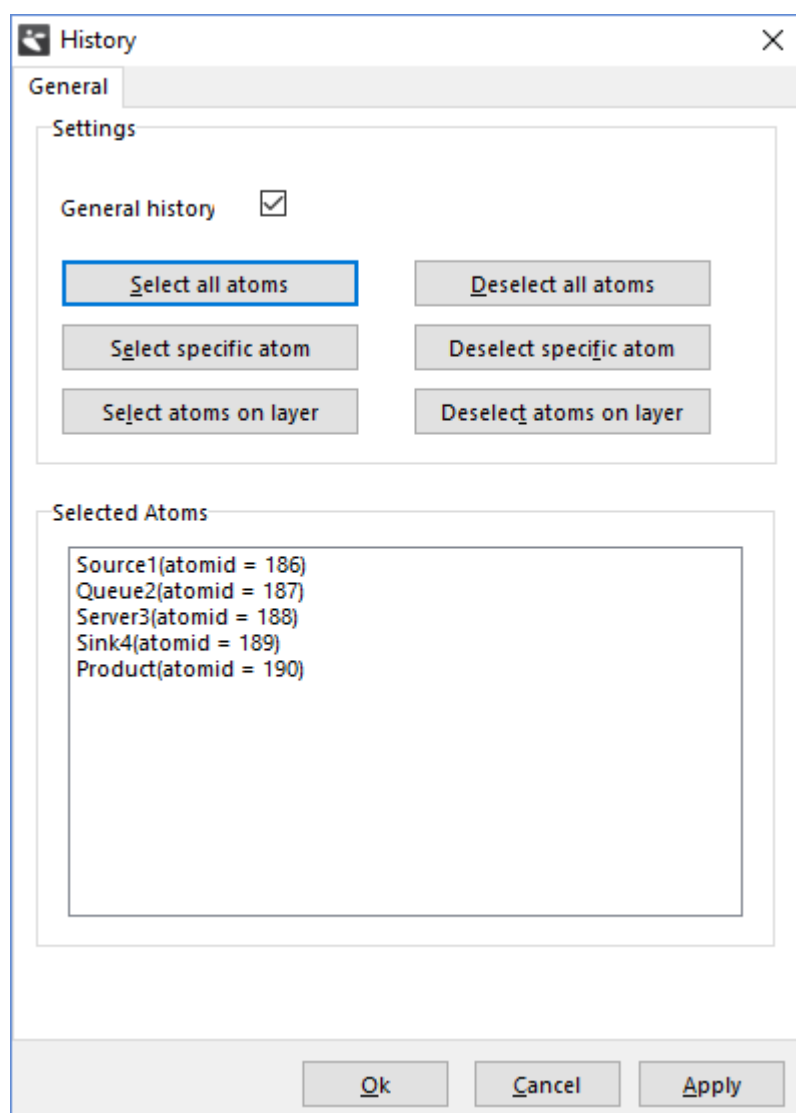
in plaats van 2100), omdat als de queue erachter vol is en dus een blokkade vormt, de planken nog maar gedeeltelijk de zaag hebben kunnen verlaten. De planken verlaten namelijk één voor één Queue1.

Bereken vanuit dit rapport dat gemiddeld 49,6 ramen per uur geproduceerd zijn!

Met de tweede optie Graphs is het mogelijk om een grafische weergave te geven van een variabele. Voor deze methode is het nodig de History voor de atomen aan te zetten voor het atoom waar de grafiek van gemaakt moet worden. Dit kan door te klikken op 'History' op het tabblad Simulation.

Waarschuwing! Zowel de optie 'General history' moet aangevinkt worden als de afzonderlijke atomen dienen geselecteerd te worden, zie ook Figuur 6-7. Het selecteren van alle atomen door te klikken op 'Select All atoms' lijkt in eerste instantie een handige mogelijkheid, omdat dan altijd van alle atomen een geschiedenis wordt bewaard. Echter, vooral bij grotere modellen wordt de simulatie onnodig traag.

Op dit moment zal alleen ingegaan worden op de mogelijkheid tot het maken van grafieken met behulp van de knop Graphs, omdat met het Graph atoom al snel geprogrammeerd moet worden. Voor gevorderde gebruikers kan het Graph atoom handig zijn omdat met behulp van 4DScript alle mogelijke variabelen afgebeeld kunnen worden.



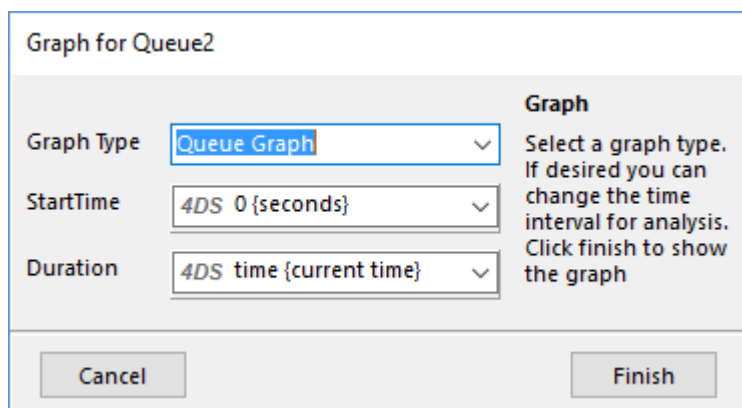
Figuur 6-7: History invoerscherm

Als de optie 'General history' is aangevinkt en er zijn enkele atomen geselecteerd via het History invoerscherm dan zal elke simulatie run data verzameld worden van de in het History scherm geselecteerde atomen. Deze data kan in een grafiek weergegeven worden met behulp van de Graphs knop die te vinden is op het Results tabblad.

Door te klikken op Graphs verschijnt er een scherm waarmee het atoom kan worden geselecteerd waarvan de grafiek moet worden getoond. Het Selectorscherm verschijnt alléén als er geen atomen geselecteerd zijn in de Model Layout. Als er wel een atoom is geselecteerd, verschijnt het scherm dat in Figuur 6-8 is afgebeeld.

Let op: heb je echter een atoom geselecteerd waarvan géén history is opgeslagen, dan krijg je de melding

“History not on”. Het makkelijkste is, om eerst het atoom te selecteren waarvan je de history bewaard hebt, en daarna pas te klikken op Graphs.



Figuur 6-8: Invoerscherm Graph

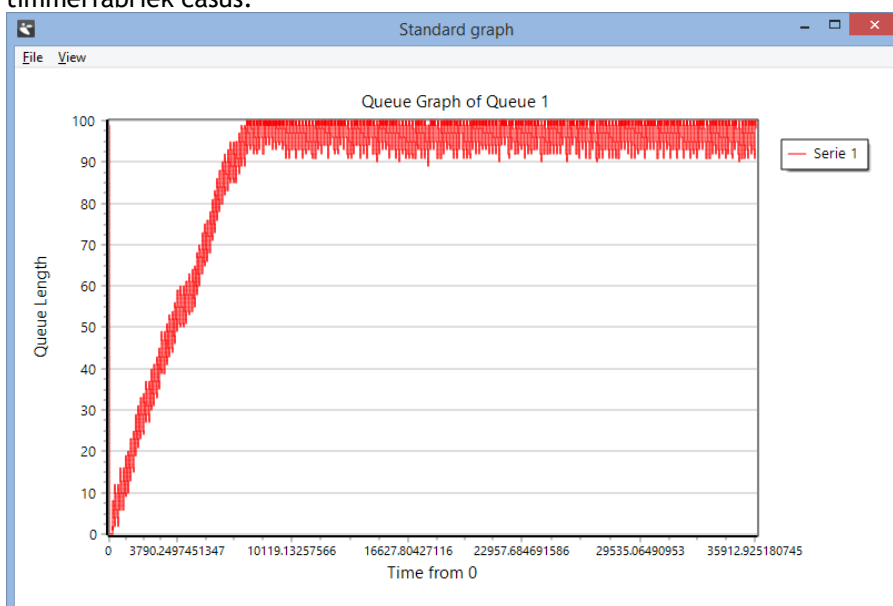
De laatste twee invoervelden bespreken we eerst:

- **Start time**
Hier wordt het tijdstip opgegeven waarop de grafiek begint. Indien er hier een 0 staat, begint de grafiek bij de eerste meting. Als er een waarde 10 staat, worden de metingen van de eerste 10 seconden niet meegenomen in de grafiek.
- **Duration**
Hier wordt ingevuld hoelang de tijdschaal van de grafiek is. Als hier de waarde 3600 staat, wordt het eerste uur na de starttijd van de grafiek weergegeven. De standaardoptie is “Time”, dit is het aantal gesimuleerde seconden.

Bij het eerste invoerveld **Graph Type** wordt het type grafiek geselecteerd. Er zijn de volgende 5 mogelijkheden:

1. Queue Graph

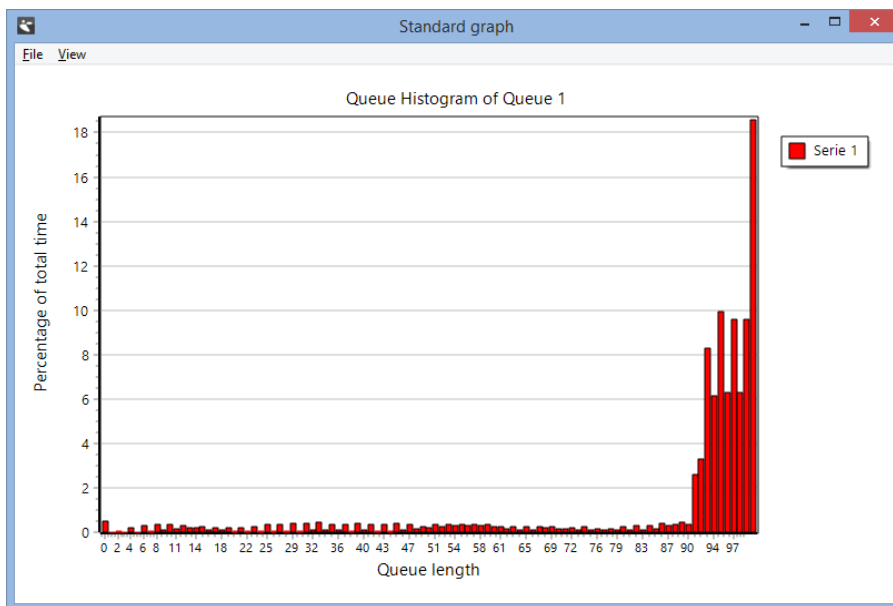
Hiermee wordt een grafiek gemaakt van het aantal producten in een atoom. Bij een server atoom zal het aantal producten in de regel nooit groter zijn dan 1, maar bij een queue atoom wordt de lengte van de wachtrij worden weergegeven in de grafiek. Figuur 6-9 geeft de grafiek van de eerste buffer uit de timmerfabriek casus.



Figuur 6-9: Queue graph

2. Queue Histogram

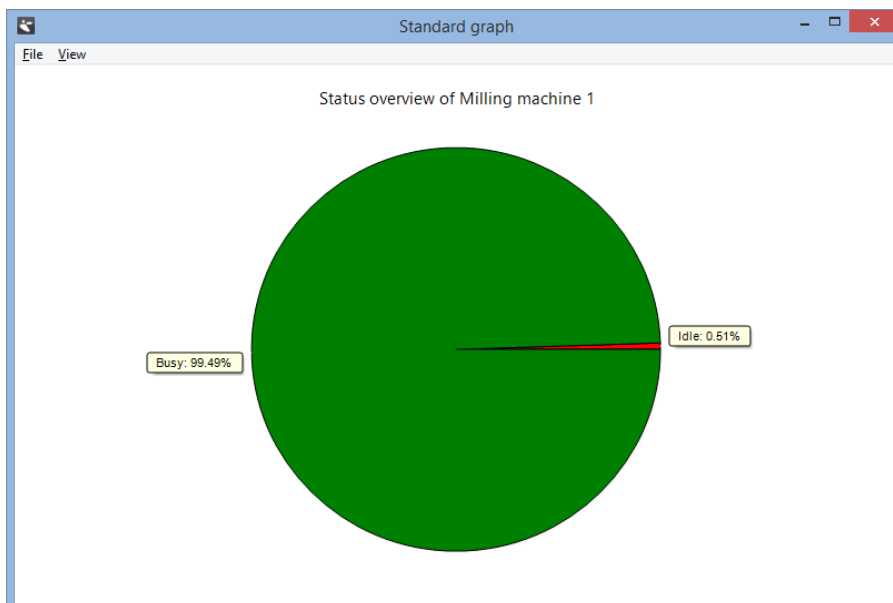
Hierbij wordt steeds het percentage weergegeven van elke `voorraad hoogte`, zie Figuur 6-10, die de distributie van de rijlengte weergeeft.



Figuur 6-10: Queue histogram

3. Status Pie

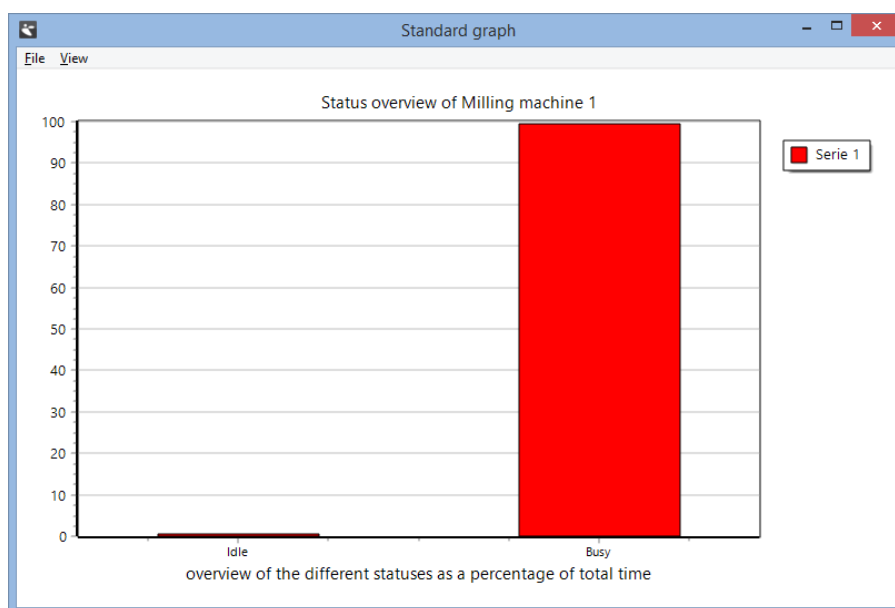
Hierbij wordt de status van het atoom in een schijfdiagram weergegeven, zie Figuur 6-11, waarin een Status Pie is opgesteld voor de eerste freesbank van de Timmerfabriek case.



Figuur 6-11: Status Pie

4. Status Bar

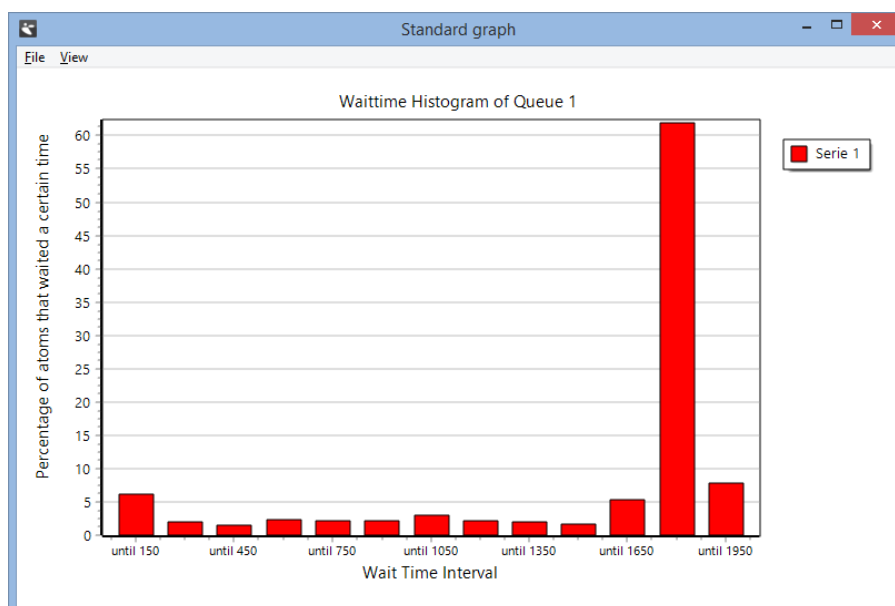
Met de optie Status Bar wordt ook de status van het atoom weergegeven, maar dan in een staafgrafiek, zie Figuur 6-12.



Figuur 6-12: Status Bar

5. Wait Histogram

Hierbij wordt er een histogram gemaakt van de verblijftijd van een product in het atoom, zie Figuur 6-13 voor een verdeling van de wachttijden van de kleine planken in Queue 1.



Figuur 6-13: Wait Histogram

4. Het opstellen van een experiment met de Experiment Wizard

De drie methoden van resultaatmeting die hierboven zijn besproken zijn vooral bruikbaar om gedurende de simulatie direct resultaten te zien. Voor conclusies over het systeem zijn de bovenstaande methoden minder geschikt. Het experiment is dan de aangewezen methode. Voor het opstellen van een experiment gebruiken we de Experiment Wizard die te vinden is op het tabblad Simulate. Een uitgebreide beschrijving is te vinden in het document Experimentation Environment (PDF) via het tabblad Help.

Belangrijk is het verschil tussen het bekijken van resultaten van een enkele run met behulp van bijvoorbeeld het Summary Report, de result atomen en de Graphs aan de ene kant en experimenten aan de andere kant: de eerste opties gebruiken we in het proces van bouwen en testen en voor eerste resultaten over een enkele run, terwijl we experimenten later in het proces benutten: we vertrouwen min of meer de geldigheid van ons model en wensen nu resultaten over meerdere runs, omdat we geen conclusies op de uitkomst van een enkele run willen baseren.

Casus De Timmerfabriek (vervolg)

Vragen en opdrachten

5. Ontwerp een experiment met een opwarmperiode van tien uur en een meetperiode van 100 uur, waarmee met 95% betrouwbaarheid de gemiddelde raamproductie per tijdseenheid (uur of dag of week) bepaald kan worden. Klopt dit met de vooraf gemaakte eigen berekeningen en de uitkomsten van de andere manieren van resultaatmeting?

We tonen nu stap voor stap de oplossing van vraag 5.

Er zijn vier stappen in experimentatie:

Step 1 Definitie van de Experiment Settings

Hier definiëren we experiment instellingen zoals het aantal runs, de lengte van de meetperiode en de opwarmperiode, et cetera.

Step 2 Definitie van de Performance Measures (prestatie indicatoren)

Hier definiëren we de uitvoer variabelen van een atoom of groep van atomen.

Step 3 Experimentation

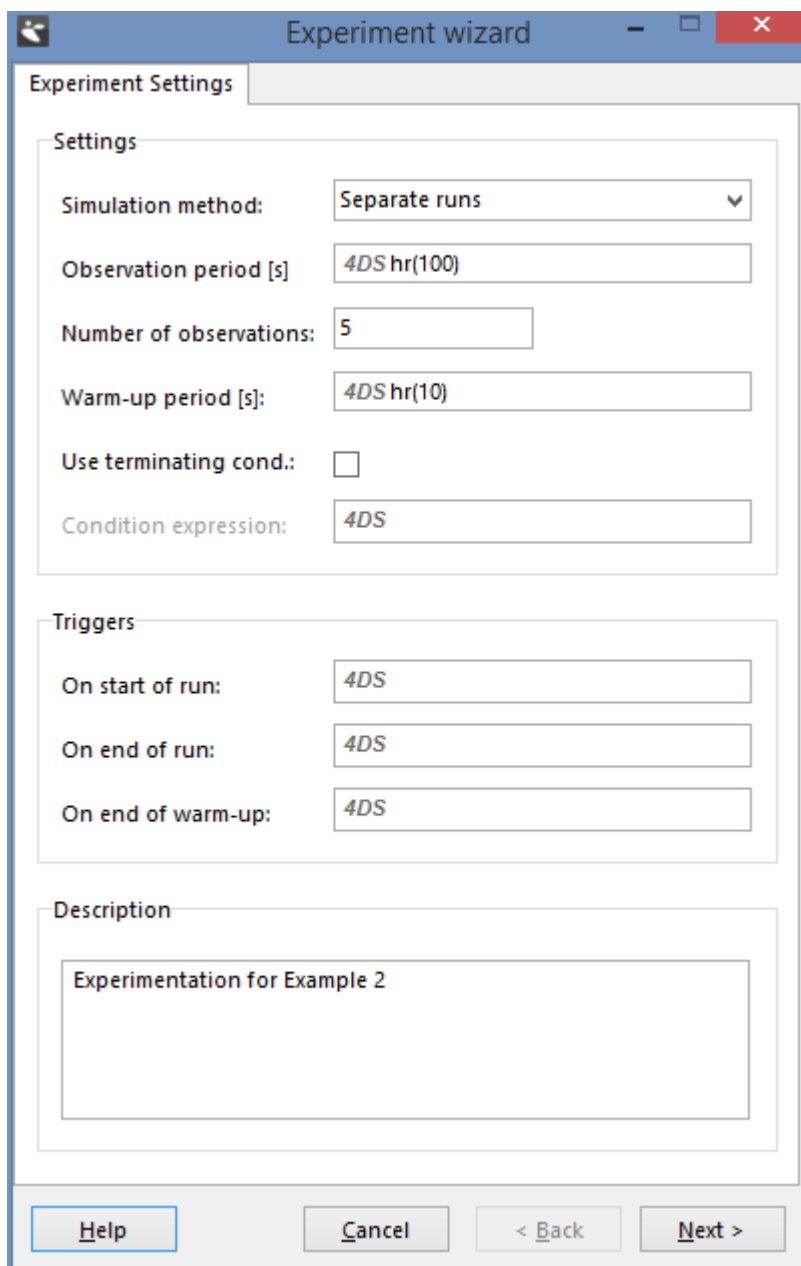
De daadwerkelijke simulatie volgens de experiment instellingen. Deze stap vraagt geen actie van de gebruiker!

Step 4 Rapport definitie en uitvoeranalyse

Resultatomen zoals een Generic Monitor atoom zijn niet nodig bij een experiment en kunnen zelfs het experiment onnodig vertragen. Mocht u last hebben van buitensporig trage experimenten, dan kunt u overwegen om bijvoorbeeld de Sample rate van het Generic Monitor atoom zeer hoog te zetten, bijvoorbeeld 36000, of deze atomen volledig te verwijderen in een experiment versie van het model.

Klik Experiment Wizard op het Simulate tabblad, het scherm van de Experiment Wizard verschijnt, zie Figuur 6-14. Stel vervolgens de gevraagde duur van de meetperiode (Observation period), het aantal runs en de opwarmperiode. Deze instellingen zijn allemaal te vinden op de Experiment Settings pagina van de Experiment Wizard, zie Figuur 6-14. Als u dat gedaan heeft kunt u op 'Next' drukken.

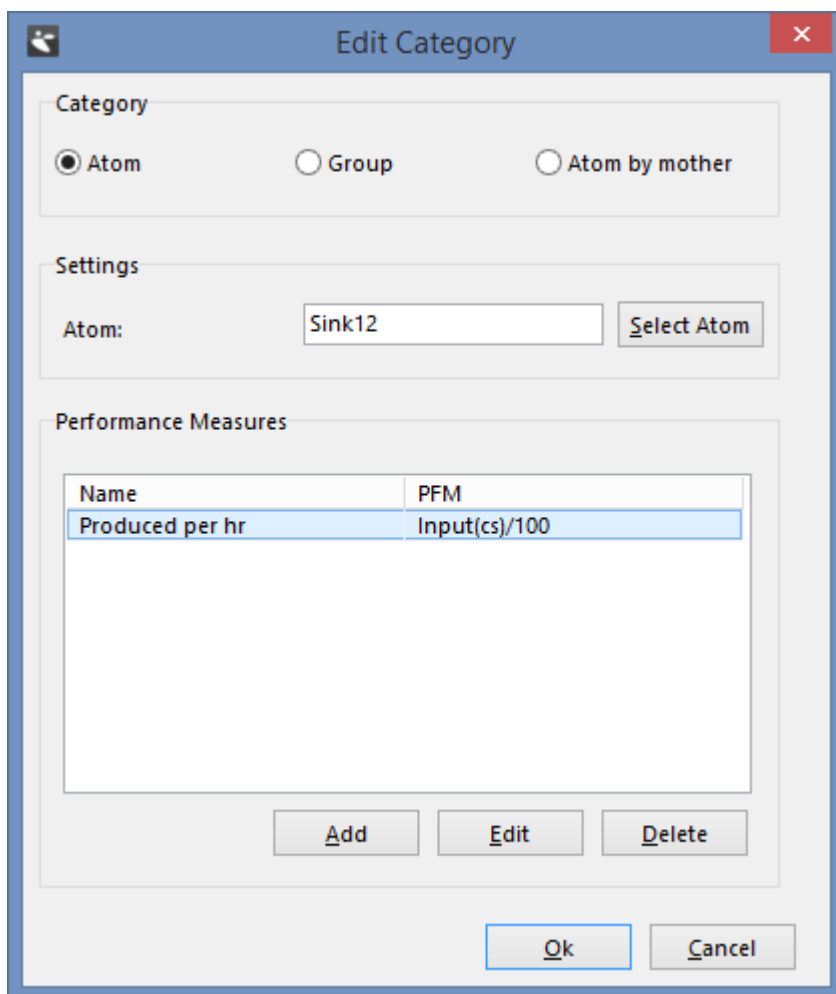
We zijn nu toe aan het invoeren van de uitvoervariabelen oftewel de prestatie indicatoren (Performance Measures (PFMs)). Klik op 'Add' om een nieuwe uitvoervariabele te definiëren. Nu verschijnt het 'Edit Category Window', zie Figuur 6-15. We zijn in dit geval geïnteresseerd in de capaciteit per uur van het productie systeem. We kunnen dit meten door te tellen hoeveel producten de Sink binnen gaan. Om de productie capaciteit te bepalen kunnen we de invoer bij de Sink meten. Om een PFM op de Sink te definiëren klikken we eerst 'Select Atom' en selecteren daarna de Sink. Nu moeten we nog definiëren wat we willen meten van de Sink, daarvoor klikken we op 'Add' in de Performance Metric sectie. We kunnen een naar invoeren voor onze uitvoervariabele en een Performance metric uit de lijst selecteren.



Figuur 6-14: Experiment Settings pagina van de Experiment Wizard

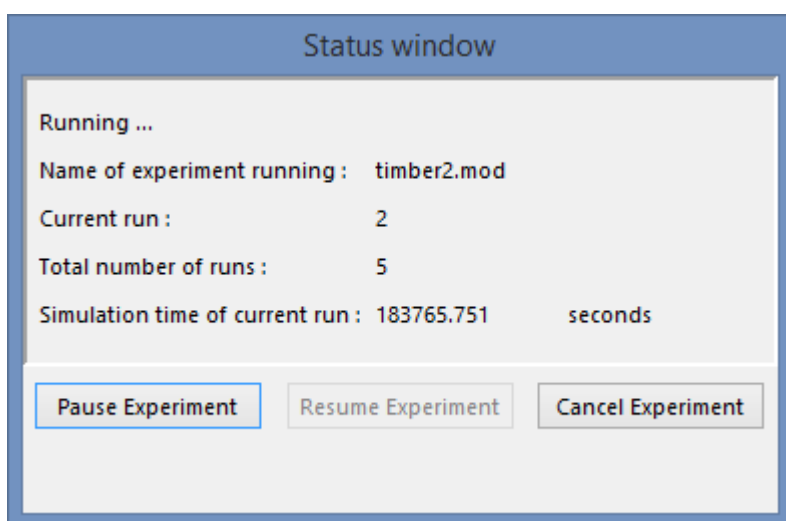
Definieer op de Sink een uitvoervariabele genaamd `Produced per hr` , zie Figuur 6-15. De input over 100 uur delen we door 100 om het aantal ramen per uur te bepalen. Selecteer hiervoor de optie “[USER DEFINED]” als PFM en pas deze tekst aan en type `Input(cs)/100`. Het woord `cs` is een 4DScript code voor “current selection”, dat ook voor een groep atomen kan staan.

Als zowel de naam als de PFM ingevoerd zijn kan op ‘Ok’ geklikt worden. Nu zouden we opnieuw op ‘Add’ kunnen klikken om een tweede PFM te definiëren op de Sink. In dit geval hebben we aan een PFM voldoende en kunnen we ook hier op ‘Ok’ klikken. We zien nu onze PFM op de Sink in de lijst van PFMs. We zouden opnieuw op ‘Add’ kunnen klikken om een PFM op een ander atoom te definiëren. Ook dat is nu niet nodig dus klikken we op ‘Next’.



Figuur 6-15: De uitvoervariabele gedefinieerd op het Sink12 atoom

De daadwerkelijke simulatie gebeurt - na de start van het experiment - in stap 3, zie Figuur 6-16 en kan enige tijd in beslag nemen: in ieder run van 100 uur moeten 500 ramen gemaakt worden van 2000 planken en 200 balken!



Figuur 6-16: De status van het experiment gedurende de simulatie

Na de uitvoering van het experiment kan op eenvoudige wijze een rapport gemaakt worden, zie Figuur 6-17 voor een rapport in een tabel. Vanuit deze tabel kunnen resultaten geëxporteerd worden naar Excel ('save as .csv' aanklikken en daarna inlezen in Excel) of op een nettere manier gepresenteerd worden via QuickReport klik 'Preview'.

Results Table							
Observation period :	1800000						
Warmup period :	36000						
Number of observations :	5						
Simulation method :	Separate runs						
Description :	Experimentation for Example						
Atom :	Sink12						
		Average	St.Deviation	Lower bound	Upper bound	Minimum	Maximum
Produced per hr		50.01	0.02	49.99	50.04	49.99	50.04

Buttons: Help, Close, Save as .csv, Preview

Figuur 6-17: Een eerste resultaat in Results Table

We lezen af dat met 95% zekerheid de onbekende gemiddelde productie per uur tussen 49,97 en 50,03 ramen zal liggen, een antwoord dat we inmiddels via meerdere manieren tegengekomen zijn.

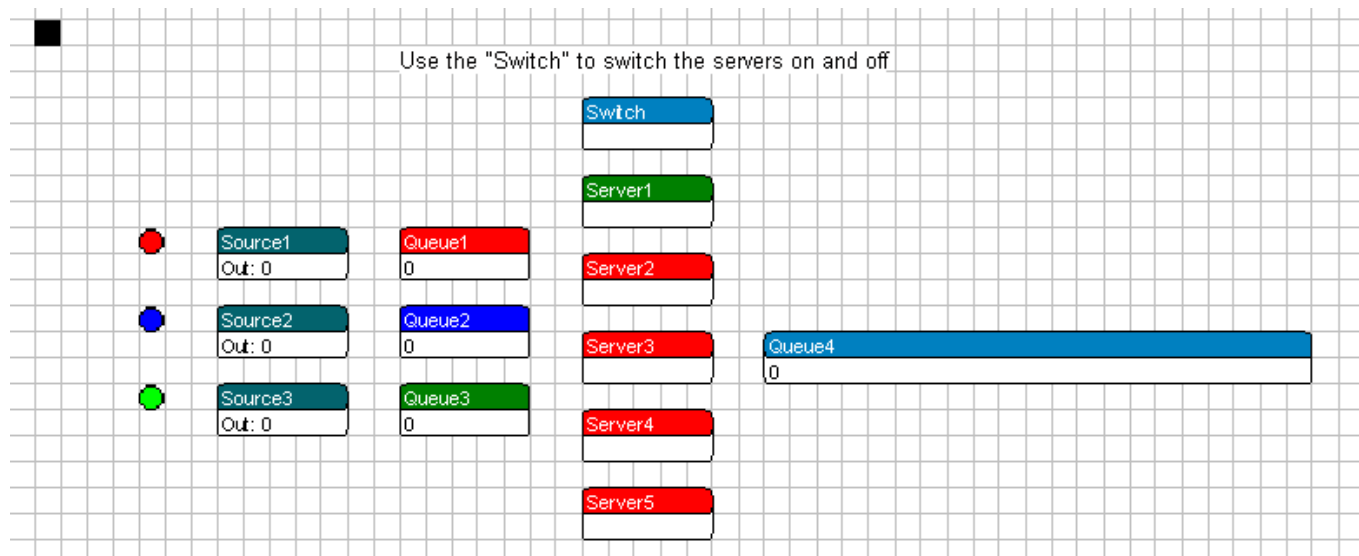
Lees voor meer informatie over experimenteren, met name over de terminologie en de precieze definitie van onze standaard uitvoervariabelen, onze aparte tutorial Experimentation Environment (PDF).

Dit model met het bijbehorende experiment is meegeleverd onder de naam timber2.mod.

7 Spelen met strategieën

In dit hoofdstuk staan de (voorgedefinieerde) strategieën waarmee producten toegang tot een volgend atoom verkrijgen (inputstrategie), geselecteerd worden uit een wachtrij (queuediscipline) en na bediening doorverwezen worden (send to) centraal.

Om dit te demonstreren is een simpel model gebouwd bestaande uit drie Sources, vijf Servers, vier Queue's en een schakelaar (een zelfgebouwd atoom: Enterprise Dynamics biedt de flexibiliteit hiertoe) om de servers aan en uit te zetten. Er is geen Sink, dus alle producten blijven in de laatste queue, zie ook Figuur 7-1. Het model is meegeleverd onder de naam *strategy.mod*.



Figuur 7-1: De lay-out van *strategy.mod*

Waarschuwing: omdat er een aanpassing is gedaan in de broncode van het laatste rij-atom en gebruik gemaakt is van een nieuw ontworpen atoom, is het voor beginnende gebruikers niet mogelijk dit model zelf te bouwen. Zij kunnen het meegeleverde *Strategy.mod* model gebruiken. Gevorderde gebruikers kunnen het model wel bouwen. Zij maken gebruik van de standaard atomen en passen de laatste queue met behulp van de Atom Editor aan. De Atom Editor is te vinden op het Display tabblad. Met de editor kan de code van de On 2D Draw Eventhandler van de Queue zo aan worden gepast dat er 100 producten zichtbaar gemaakt worden in de queue. Omdat de schakelaar die in deze casus gebruikt wordt ook geen standaard atoom is, moeten gevorderde gebruikers de servers handmatig aan en uitzetten. Dit kan onder meer met de Availability Control atomen.

Vanuit de drie sources stromen drie type klanten naar de achterliggende rijen. De toestroom is zo ingesteld dat de drie linker rijen vrij snel vol zullen lopen en dat niet meer dan 10 klanten per Source gegenereerd worden. Vanuit de rij worden de klanten bediend door een Server om vervolgens in de laatste rij te belanden. De schakelaar (Switch) dient voor het aan- en uitzetten van servers.

Vanuit de drie wachtrijen kunnen in principe alle servers bereikt worden, de eerste server kan steeds via kanaal 1 bereikt worden, de tweede via kanaal 2 etc. Echter, in de oorspronkelijke instelling van *strategy.mod* is (via de schakelaar) alleen server 1 aangezet. Dit is te zien door het groen oplichten van de eerste server bij een simulatie en het rood oplichten van de anderen. Verder zijn vooralsnog de meeste standaardinstellingen gehandhaafd d.w.z. dat:

1. alle input strategieën nog op *Any inputchannel* staan,
2. alle queue disciplines nog op *First in first out* staan,
3. de *send to* statements op de Queues op optie 4 staan: *A random open channel: choose a random channel from all the open output channels*. Dit betekent dat onder de ingeschakelde, groene Servers een onbezette Server gezocht wordt.

De producten zijn met hun icoon zichtbaar in de laatste queue. Het is dus precies zichtbaar in welke volgorde de producten de queue zijn binnengekomen (mits de queue discipline op FIFO staat ingesteld).

Doordat de producten een zogenaamd *label* meekrijgen op het moment dat ze de source verlaten, kunnen ze ook op basis van dit label worden gesorteerd in de laatste queue. Ter zijde, labels hebben meestal de rol van etiketten die aan producten hangen. Ze kunnen een kleur, een gewicht, een grootte o.i.d. voorstellen. Ze worden gedefinieerd in 4DScript, de programmeertaal van Enterprise Dynamics, en spelen een grote rol in de meeste

modellen. In hoofdstuk 9 wordt dieper op de toepassing van labels ingegaan.

Bekijk zelf de overige instellingen van `strategy.mod` en vergroot het begrip door het uitvoeren van een paar simulatie runs. Dan zijn we nu klaar voor het daadwerkelijk experimenteren met de strategieën...

7.1 Het aanpassen van de inputstrategie

Met de input strategie wordt de toegang tot een atoom vanuit voorafgaande atomen (met uitgangskanalen naar dit bewuste atoom) geregeld. Door de inputstrategie worden zowel één of meerdere ingangen vrijgegeven en ook via een bepaalde volgorde bekeken via welk kanaal als eerste producten mogen worden ontvangen.

De inputstrategie is het beste te vergelijken met de regelstrategie van een verkeersregelininstallatie waarbij onafhankelijk van het verkeersaanbod vanuit de toestand Rood overgeschakeld wordt naar Groen voor een of meerdere toevoerwegen en waarbij bovendien de voorrang van afwikkeling van deze toevoerwegen is bepaald.

De eerste drie inputstrategieën openen alle ingangskanalen, de laatste twee openen steeds één ingangskanaal.

Als oefening wordt nu (vanuit het ongewijzigde beginmodel) gekeken naar de gevolgen van het veranderen van de inputstrategie op server 1. Verander steeds de inputstrategie zoals hieronder beschreven en kijk wat er gebeurt door het (herhaaldelijk) uitvoeren van een simulatierun.

1. *Any input channel*

Wanneer geactiveerd opent deze strategie alle ingangskanalen van een atoom. Indien meer dan één van de -via het invoerkanaal verbonden- atomen wil en kan zenden, dan heeft het atoom met het laagste nummer als ingangskanaal voorrang. Zolang er dus steeds producten via het eerste kanaal binnenkomen, komen de andere kanalen niet aan bod!

2. *Largest queue*


Wanneer geactiveerd opent deze strategie alle ingangskanalen van een atoom. Indien meer dan één van de -via het invoerkanaal verbonden- atomen wil en kan zenden, dan heeft het atoom met de grootste rij of inhoud voorrang. Merk op dat in het geval van meerdere even lange wachtrijen steeds wordt gekozen voor het ingangskanaal met het laagste nummer.

3. *Longest waiting*

Wanneer geactiveerd opent deze strategie alle ingangskanalen van een atoom. Indien meer dan één van de -via het invoerkanaal verbonden- atomen wil en kan zenden, dan heeft het atoom met de hoogste gemiddelde verblijftijd voorrang. In het geval van meerdere atomen met een zelfde gemiddelde verblijftijd steeds wordt gekozen voor het ingangskanaal met het laagste nummer. Merk op dat dit dus niet per definitie betekent dat de wachtrijen in aantallen ongeveer even lang worden, zoals bij de vorige optie wel het geval was.


4. *Round robin*

Opent eerst het eerste ingangskanaal en wacht op toezending van een product via dit ingangskanaal. Bij de volgende cyclus komt het tweede ingangskanaal aan de beurt enz. Wanneer het laatste ingangskanaal geweest is, wordt de procedure met het eerste ingangskanaal hervat.

 Bij het eerste product dat binnenkomt geldt deze regel nog niet en dus zal productvolgorde in ons voorbeeld $x,2,3, 1,2,3, \dots$ zijn, waarbij x ieder mogelijk product kan zijn!

5. *Round robin with content check*

Deze strategie is vergelijkbaar met de standard Round Robin, met dat verschil dat wanneer het atoom dat met het inkomen kanaal verbonden is, geen content heeft, deze overgeslagen wordt. Wanneer geen van de atomen die via de inkomende kanalen verbonden zijn een content heeft, dan wordt gewacht op het eerste atoom dat iets aanbied en wordt de strategy vanaf dat moment vervolgt.

 Deze strategie begint vanaf het moment dat het eerste product binnenkomt. In het geval van drie inkomende kanalen zal het dus als volgt gaan: $x,2,3, 1,2,3, 1,2,3$ waar x 1,2 of 3 kan zijn!

6. *Channel 1*

Hierbij wordt een vast ingangskanaal opgegeven dat gebruikt moet worden. In dit geval mogen dus alleen producten via ingangskanaal **1** naar binnen. Merk op dat bij het eerste product dat binnenkomt deze regel nog niet geldt en alle kanalen openstaan.

7.2 Het aanpassen van de Queue Discipline

In dit deel is het de bedoeling de Queue Discipline van de laatste queue aan te passen en te kijken wat er gebeurt.

Stel de inputstrategie van de eerste server in op Largest Queue, omdat de producten hierdoor zo veel mogelijk door elkaar de server binnenkomen (en dus ook weer verlaten). Laat verder de overige vier servers uitgeschakeld, omdat bij vijf ingeschakelde servers minder duidelijk te zien is hoe de atomen zich verplaatsen. Hieronder staan de zes mogelijke instellingen weergegeven:

1. *First in first out*
Hierdoor worden de atomen in de volgorde van binnenkomst in de queue geplaatst.
2. *Last in first out*
Hierdoor worden de binnenkomende producten vooraan in de wachtrij geplaatst. De producten verlaten de queue dus in omgekeerde volgorde van binnenkomst.
3. *Random*
Deze rijdiscipline plaatst de producten een willekeurige plaats in de wachtrij.
4. *Sort by Label Ascending*
Hierbij worden de producten met de laagste waarde voor een bepaald label vooraan in de wachtrij geplaatst. Test dit uit door te sorteren op het label 'product' dat op alle product atomen in deze casus aanwezig is. *Let op:* wanneer de producten niet goed gesorteerd worden heeft dit waarschijnlijk te maken met een spatie voor of achter de naam van het label.
5. *Sort by Label Descending*
Hierbij worden de producten met een hoogste waarde voor een bepaald label vooraan in de wachtrij geplaatst.
6. *User defined*
Hierbij worden de producten op een door de gebruiker opgegeven positie in de wachtrij geplaatst. Probeer dit uit voor de waarde 5.

7.3 Het aanpassen van het Send to Statement

In dit derde deel van de casus worden een aantal 'Send to' statements voor de drie wachtrijen naar de servers besproken. Met het 'Send to' statement geven we het uitgangskanaal op waarheen we de producten zenden. Er kan door de gebruiker een getal worden ingevuld, of gekozen worden uit één van de 21 (!) voorgedefinieerde opties.

Als eerste schakelen we de vijf servers in, anders valt er weinig te verwijzen. Hierdoor zullen ze de kleur groen krijgen en producten toelaten.

Experimenteer nu op de wachtrijen met de volgende verwijzingen:

1. *Specific channel: always send to channel 1.*
Hierbij zal het Product Atoom altijd naar een vast uitgangskanaal worden gestuurd. Kies bijvoorbeeld kanaal 3.
2. *An open channel (First channel first): search, starting from the first channel, and send to the first open channel found.*
Hierbij wordt het Product Atoom naar het eerste open kanaal gestuurd dat Enterprise Dynamics tegenkomt. Er wordt gezocht vanaf het eerste uitgangskanaal (kanaal nummer 1).
3. *An open channel (Last channel first): search, starting from the last channel, and send to the first open channel found.*
Er wordt gezocht vanaf het laatste kanaal en het product wordt naar het eerste open kanaal gestuurd dat Enterprise Dynamics tegenkomt.
4. *A random open channel: choose a random channel from all the open output channels.*
Er wordt door Enterprise Dynamics een willekeurig open kanaal gekozen. Bij lange simulatieruns leidt dit tot gelijke bezettingsgraden van bijvoorbeeld een groep servers waar naar verwezen wordt.
5. *By percentage: 90% of products go to channel 1, the remaining percentage go to channel 2.*
Hierbij wordt een bepaald percentage van de producten naar het ene kanaal gestuurd en de rest naar een ander kanaal. De gebruiker kan de kanalen en het percentage opgeven. Probeer eens 75% naar server 1 en de rest naar server 5 te sturen.
6. *By user: enter your own 4DScript expression resulting in a value between 1 and the number of channels:1.You can press the small button for the 4DScript editor.*
Een door de gebruiker geschreven 4DScript code die het uitgangskanaal oplevert. Door op het kleine vierkante knopje te drukken naast de tekst, verschijnt de 4DScript editor. Vul bijvoorbeeld 2 in.

Dit rondt het voorbeeld over de strategieën af. Uiteraard zijn er veel meer mogelijkheden van verwijzing, echter deze zullen door de modelbouwer in 4DScript, de onderliggende programmeertaal van ED, geprogrammeerd moeten worden

8 Meer atomen van Assembler tot Unpack

In de vorige hoofdstukken zijn de basis atomen besproken. Vervolgens is geoeft in het bouwen en experimenteren met eenvoudige modellen. In dit hoofdstuk worden acht nieuwe atomen geïntroduceerd. Met deze nieuwe atomen kunnen ook complexere situaties gemodelleerd worden. Tevens schenken we meer aandacht aan de visualisatiemogelijkheden in ED.

De atomen die we in dit hoofdstuk bespreken, zijn:

1. *Assembler*
Dit atoom wordt gebruikt wanneer meerdere atomen opgaan in een nieuw atoom. Hierbij kunnen de oude atomen bewaard blijven (verpakken) of worden verwijderd (assemblage).
2. *Unpack*
Dit atoom kan samengevoegde atomen weer opsplitsen. Wanneer met behulp van een Assembler bijvoorbeeld een doos is gevuld, kan de Unpack deze weer uitpakken.
3. *Container*
De container is een atoom om producten op te plaatsen of in te verpakken. Voorbeelden zijn een doos of pallet.
4. *Accumulating conveyor*
De Accumulating conveyor is een transportband die tevens als buffer functioneert. Wanneer de producten de transportband niet meer kunnen verlaten, blijft de band doordraaien. Hierdoor worden de producten op de transportband tegen elkaar geschoven. De Accumulating conveyor wordt gebruikt om rollenbanen mee te modelleren.
5. *Non-Accumulating conveyor*
De Non-Accumulating conveyor lijkt sterk op de Accumulating conveyor, maar draait niet door op het moment dat de producten de transportband niet meer kunnen verlaten. De afstand tussen producten blijft dus altijd gelijk. De Non-Accumulating conveyor wordt gebruikt om kettingbanen mee te modelleren.
6. *MultiService*
Het MultiService atoom fungeert als een groep parallelle Servers: het bevat de basisfunctionaliteiten van het Server atoom en kan daarnaast meerdere producten gelijktijdig en onafhankelijk van elkaar verwerken.
7. *Lock*
Het Lock atoom laat slechts een vooraf ingesteld aantal producten door. Alle volgende producten worden geblokkeerd.
8. *Unlock*
Zodra een product het Unlock atoom verlaat, mag het Lock atoom weer een nieuw product doorlaten. Hierdoor is het mogelijk de voorraad onderhanden werk of hoeveelheid transportmiddelen in het model te beheersen.

Al deze atomen komen in de hierna volgende casus aan bod.

Casus 3: Stapelen en wikkelen

Op de expeditie afdeling van een fabriek worden producten op een pallet gestapeld. De producten en de pallets arriveren via twee afzonderlijke transportbanden bij deze stapelaar of palletiseerrobot. Deze robot kan de producten, afhankelijk van hun afmeting, zowel op als naast elkaar stapelen.

Vanaf de robot gaat de gestapelde pallet met producten via een volgende transportband naar een zogenaamde wikkelaar, waar plastic om de producten wordt gewikkeld. Op de wikkelaar kunnen de producten van meerdere pallets gelijktijdig en onafhankelijk van elkaar gewikkeld worden.

Het systeem bestaat uit de volgende onderdelen:

1. Elke 5 seconden komt via een rollenbaan een product aan bij de stapelaar.
2. Via een tweede rollenbaan arriveert gemiddeld om de 40 seconden volgens negatief exponentiele verdeling een pallet bij de stapelaar.
3. De stapelrobot plaatst steeds 8 producten op de pallet en heeft daar per pallet, indien alle producten aanwezig zijn, precies 20 seconden voor nodig.
4. De volle pallets gaan via een kettingbaan naar de wikkelaar.
5. De wikkelaar kan maximaal 4 pallets tegelijk van een plasticfolie voorzien. Per pallet is gemiddeld 120 seconden (negatief exponentieel verdeeld) nodig om de producten van de pallet in plastic te verpakken.
6. De lengte van elke transportband doet hier niet echt ter zake: kies deze in de orde van 10 meter. De snelheid is steeds 1 m/s. De producten zijn 50 cm in lengte, breedte en hoogte en de pallets zijn 1 meter

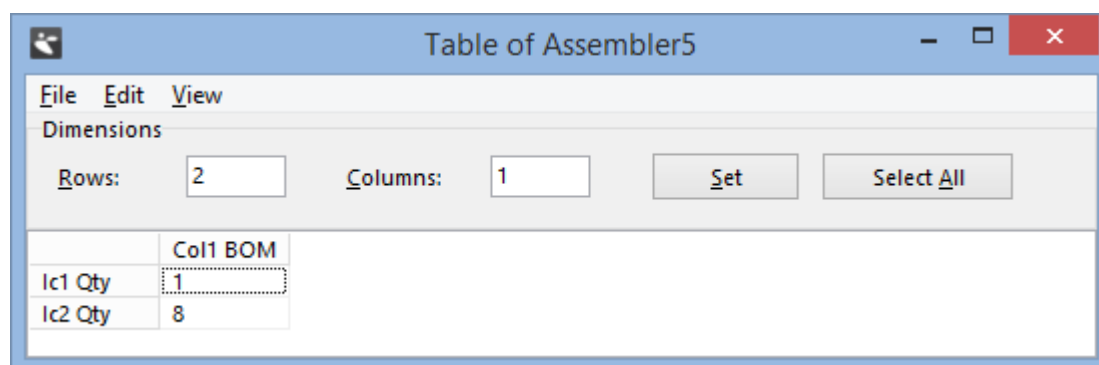
lang en breed.

We willen dit systeem modelleren in Enterprise Dynamics: voor de rollenbaan gebruiken we de Accumulating Conveyor, voor de kettingbaan een Non-Accumulating Conveyor. Verder voor de stapelaar een Assembler, voor de wikkelaar een Multiservice atoom en voor de pallet de Container.

Maar...bereken vooraf nog even of dit systeem de toestroom zal kunnen verwerken!

De meeste atomen kunnen zonder problemen in het model geplaatst worden, omdat ze op de standaard manier met andere atomen verbonden zijn. Bij de Assembler moeten de pallets via het eerste kanaal (gevisualiseerd met een vierkantje) het atoom binnenkomen en de producten via het tweede kanaal. Dubbelklik op de Assembler om het aantal producten per pallet in te stellen. Klik hiervoor op het tabblad General de knop 'Edit B.O.M. Table' in. Er komt dan een tabel te voorschijn, zie Figuur 8-1 waar per ingangskanaal opgegeven wordt hoeveel atomen er samengevoegd moeten worden. In de bijgeleverde help bij de software bevindt zich een complete beschrijving van alle gebruikte atomen, waaronder de Assembler!

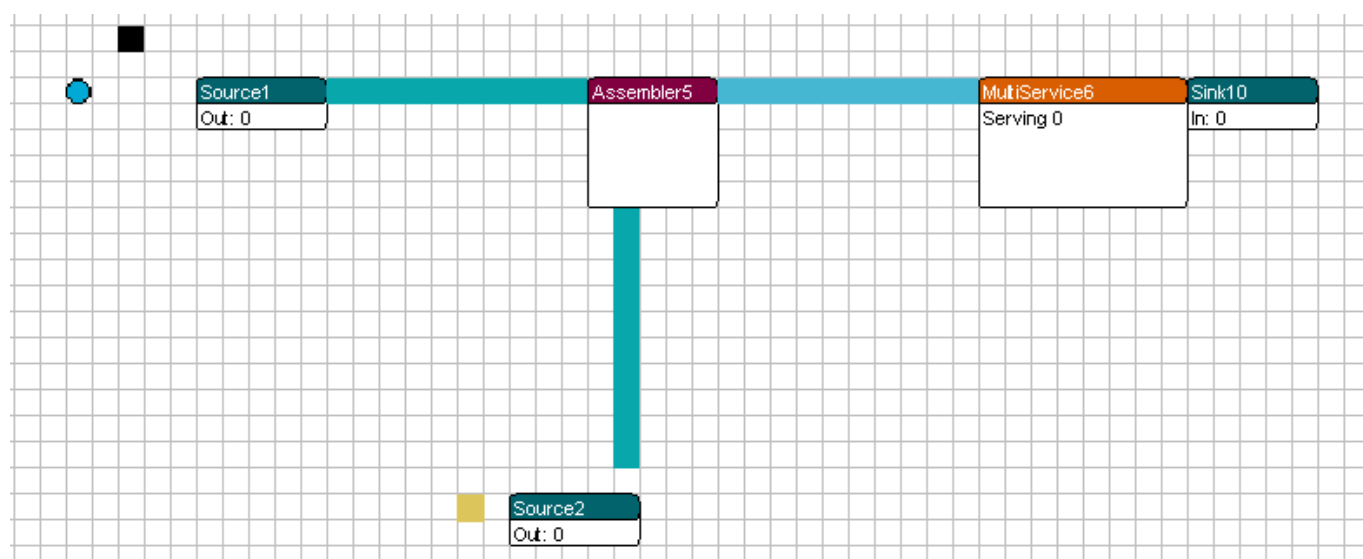
De producten worden door de Assembler automatisch op de pallet gestapeld. De ruimte die voor elk product gereserveerd wordt, staat ingesteld in het Container atoom. Om te voorkomen dat alle producten in één hoge kolom op elkaar gestapeld worden, moet de afmeting van het product op het productatoom aangepast worden. Zie hiervoor de uitwerking bij casus 2!



Col1 BOM	
Ic1 Qty	1
Ic2 Qty	8

Figuur 8-1: Bill of Material (BOM) van de Assembler

Wanneer alle atomen in het model staan, ziet de lay-out van het model er uit als in Figuur 8-2.



Figuur 8-2: Lay-out van het conveyorsysteem

Sla het eigen model op. Bij onzekerheid over het eigen model kan voor de volgende paragrafen ook gebruik gemaakt worden van het meegeleverde model onder de naam conveyor1.mod.

Laat het systeem enige tijd runnen en controleer de werking in 3D. Speel wat met afmetingen van de producten of de container en kijk wat er gebeurt! Doe dit eveneens met de snelheden van de transportbanden of bewerkingstijden: zo geeft een veel hogere bewerkingstijd op de MultiService opstoppingen voor de wikkelaar en verduidelijkt dit de werking van de Non-Accumulating Conveyor.

Vragen en opdrachten

1. Voor de beheersing van de werklast willen we niet meer dan vijf pallets tegelijk in dit systeem. Pas het model aan deze wens aan.

Dit wordt bereikt door een Lock en Unlock-atoom in het model te plaatsen. De Lock en Unlock atomen worden in het productieproces tussengevoegd: de Lock op de plaats waar de beperking begint, de Unlock waar de beperking eindigt. Het Lock atoom sluit zichzelf nadat er een vooraf ingesteld aantal atomen zijn gepasseerd. Het Unlock atoom heft deze blokkade weer op.

Door alleen een Lock te plaatsen kunnen er bijvoorbeeld niet meer dan vijf atomen het systeem binnenkomen. Door ook een Unlock te plaatsen, zorgen we er voor dat niet meer dan vijf atomen tegelijk aanwezig kunnen zijn in het systeem. Het Lock atoom wordt altijd ná de Source geplaatst, omdat de Source anders het Lock atoom zelf als product ziet en het systeem instuurt. Het Unlock atoom wordt altijd geplaatst op de plek waar de begrenzing van het aantal atomen weer moet eindigen, bijvoorbeeld na een bepaalde machine of vlak voor de Sink. In deze casus zal de Unlock dus direct voor de Sink geplaatst moeten worden.

Door het plaatsen van een Unlock atoom opent het Lock atoom zichzelf automatisch wanneer er een atoom het Unlock atoom verlaat. Het Lock atoom en Unlock atoom worden automatisch met elkaar verbonden. Mocht het toch nodig zijn ze handmatig te verbinden, dan kan dit door het tweede uitgangskanaal van het Lock atoom te verbinden met het tweede ingangskanaal van het Unlock atoom. Om de werking van het Lock en Unlock atoom duidelijker te zien, kan het Lock atoom tijdelijk zo ingesteld worden dat het na 1 of 2 pallets al sluit.

Pas het model nu aan of open conveyor2.mod.

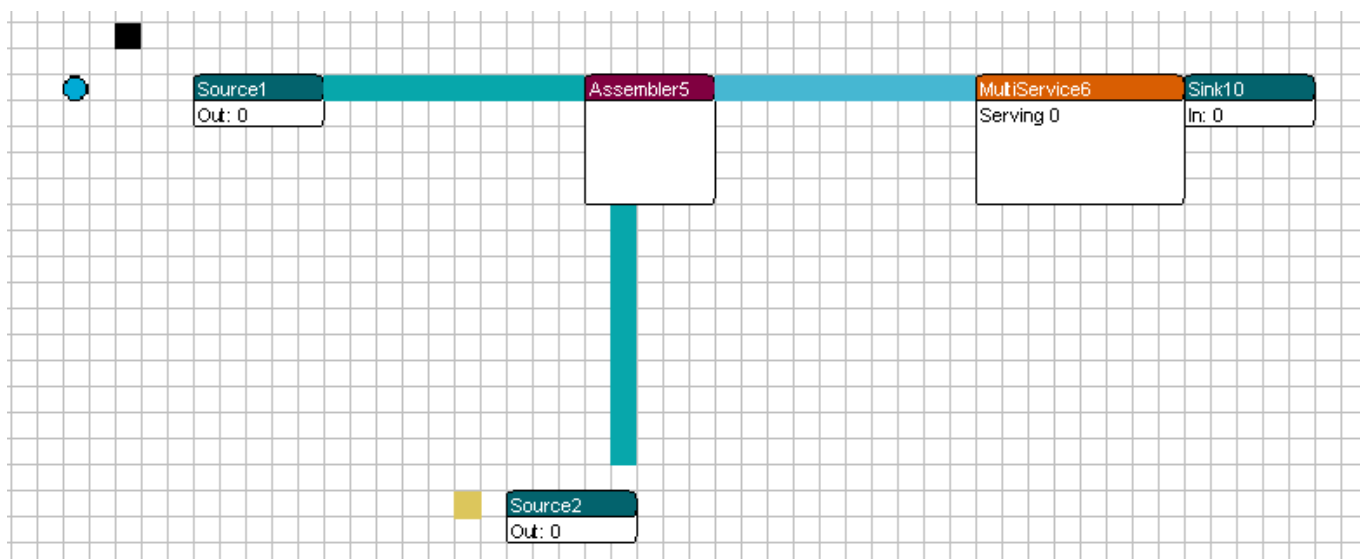
Vragen en opdrachten

2. Het bedrijf wil nu na de wikkelaar de pallets scheiden van de producten en via een extra transportsysteem terugvoeren naar de stapelaar. Het aantal pallets in het systeem blijft beperkt tot 5. Pas het model hiervoor aan.

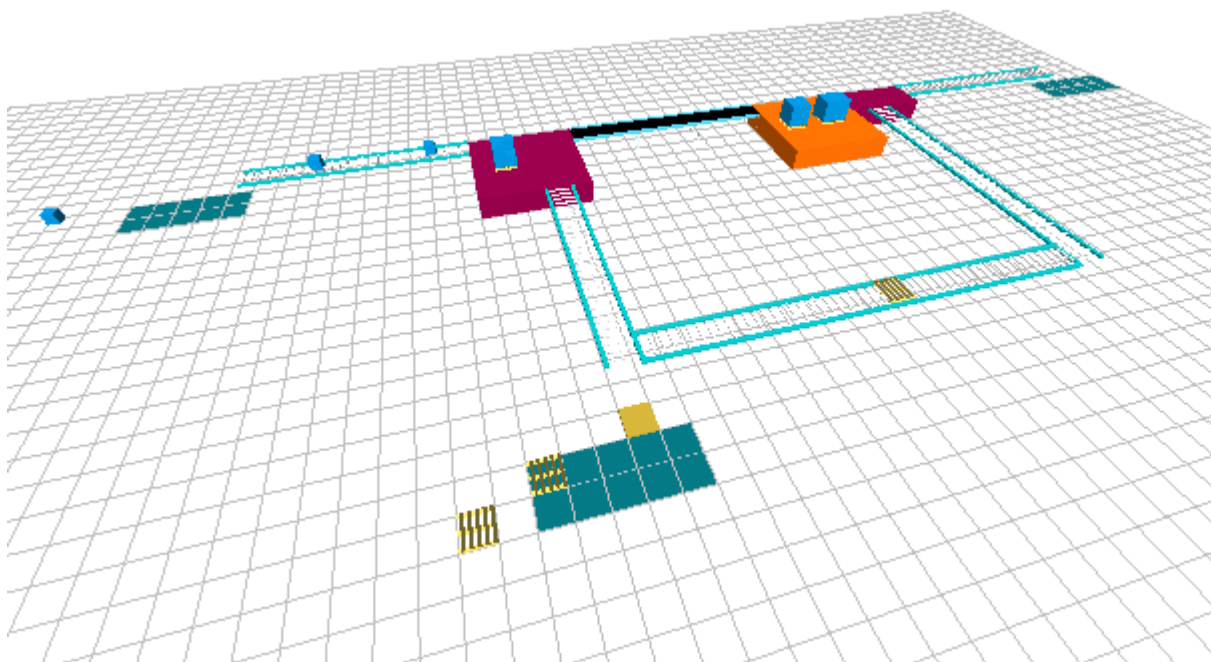
Om hergebruik mogelijk te maken zijn twee grote aanpassingen nodig. Ten eerste scheiding van product en pallet, ten tweede transport van de pallets naar de Assembler. Voor het scheiden van de pallet en de producten is het Unpack atoom nodig. Geplaatst na de MultiService, stuurt het de producten naar het eerste uitgangskanaal en de pallets naar het tweede. Door op dit tweede uitgangskanaal een rollenbaan aan te sluiten en deze te verbinden met de al bestaande rollenbaan, kunnen de pallets hergebruikt worden.

Doordat de pallets het systeem nu niet meer verlaten, vervalt het Unlock atoom. De Lock kan vervallen door in de Source bij Number of Products 5 te kiezen. Merk op dat de 40 seconden tussenaankomsttijd bij het genereren van de pallets geen rol spelen en ook anders ingesteld kan worden. Wanneer de extra rollenbanen in het model zijn opgenomen, zou het model eruit kunnen zien zoals in Figuur 8-3 en 8-4 (conveyor3.mod).

De lengte van de transportbanden is zo ingesteld dat de rondgang van de pallets direct visueel duidelijk is.



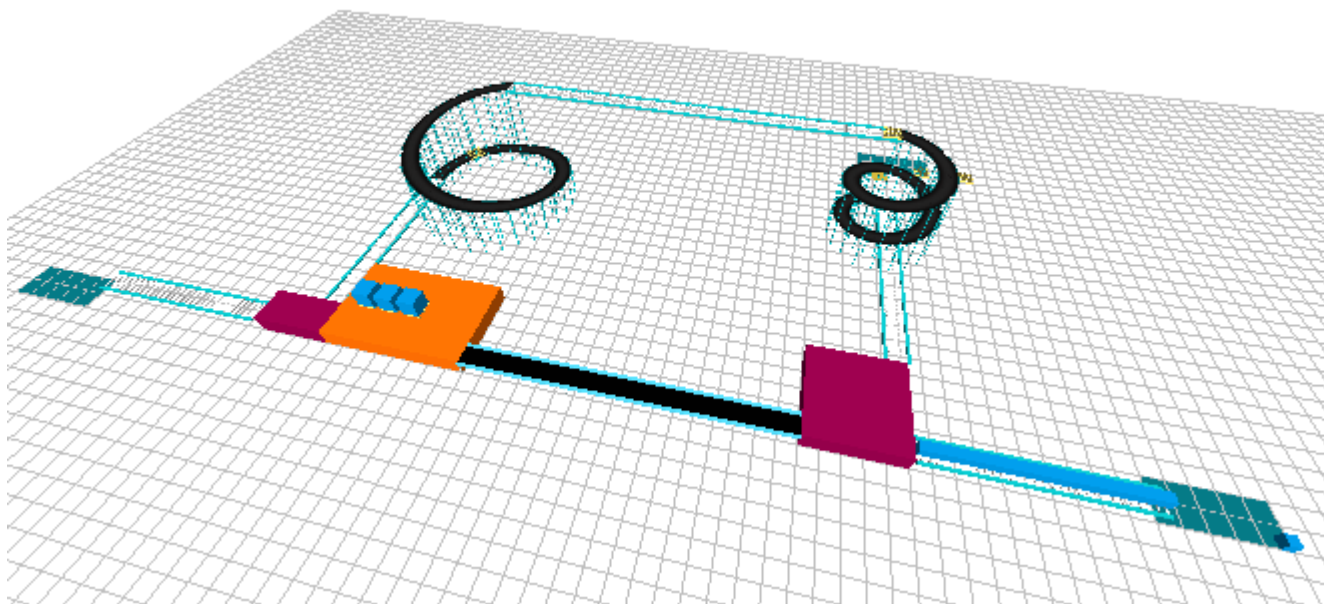
Figuur 8-3: Hergebruik van pallets in 2D



Figuur 8-4: Hergebruik van pallets in 3D

Uiteraard kunnen in plaats van de rechte Conveyors ook Conveyors met een bocht worden gebruikt. In Figuur 8-5 is de lay-out gespiegeld en gebruik gemaakt van gebogen conveyors, die ook nog in hoogte instelbaar zijn.

Beide modellen zijn te vinden als conveyor3.mod resp. conveyor4.mod.



Figuur 8-5: In hoogte instelbare Conveyors

We eindigen met een paar vragen rondom conveyor3.mod.

Vragen en opdrachten

3. Hoe lang duurt een gemiddelde ronde van een pallet (exclusief wachttijden)?
4. Te weinig pallets in het systeem beperkt de productie per uur, terwijl vanaf een zeker moment extra pallets geen of nauwelijks invloed hebben op deze uurproductie. Verklaar! Zoek het 'optimale' aantal pallets in het systeem door experimenteren (Bekijk via een Generic Monitor op Sink de gemiddelde

uurproductie en varieer de hoeveelheid toegelaten pallets met de Lock of via de Source.

9 Enterprise Dynamics en Excel

Vanuit Enterprise Dynamics kunnen we verbindingen naar andere softwareprogramma's zoals Excel, Word of Access leggen. Dit voor het inlezen of wegschrijven van gegevens. In dit hoofdstuk wordt een veelgebruikte koppeling tussen ED en Excel besproken. Dit gebeurt door het uitbouwen van de eerdere casus over een postkantoor.

Verder introduceren we een aantal commando's in 4DScript en het belangrijke concept van *labels* in ED.

Na afloop van dit hoofdstuk kan de gebruiker de koppeling met Excel leggen en beheerst bovendien de eerste beginselen van 4DScript.

9.1 De bank

Casus 4: De Bank

Een bank in Rotterdam fungeert als proefobject om nieuwe concepten voor afhandeling van klanten te testen. Er is sprake van meerdere klanttypen met verschillende bedieningstijden.

In deze casus kiezen we -ter illustratie- voor een reductie tot twee klanttypen. Zo arriveren nu gemiddeld 50 klanten per uur met een bedieningstijd van 1 minuut. Deze klanten duiden we aan met type A. Verder arriveren er ook nog gemiddeld 5 klanten van het type B met een bedieningstijd van 10 minuten.

Er is een gezamenlijke wachtrij waar beide klanttypen op volgorde van binnenkomst geholpen worden aan één van de twee loketten. Alle aankomstprocessen veronderstellen we exponentieel verdeeld. Alle bedieningstijden zijn constant.

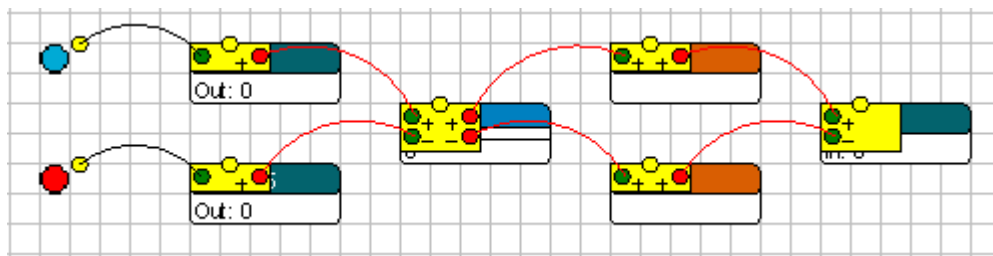
Vragen en opdrachten

1. Hoe hoog is de bezettingsgraad van dit systeem?
2. Bereken met behulp van de wachtrijtheorie de gemiddelde wachttijden en wachtrijen indien beide klanttypen een eigen wachtrij met bijbehorend loket hebben.
3. Kun je deze karakteristieken ook berekenen voor het systeem met de gezamenlijke wachtrij? Waarom wel of niet?
4. In welk systeem zal de gemiddelde wachttijd lager zijn? Beargumenteer!

Probeer deze vragen, die vanuit theoretisch standpunt interessant zijn, te beantwoorden, we komen er later op terug. Degenen die niet bekend zijn met de wachtrijtheorie over $M/D/1$ en $M/D/2$ -systemen kunnen de vragen 2 en 3 overslaan.

Voor dit moment bouwen we eerst het model met de gezamenlijke wachtrij. Daarna leggen we de verbinding naar Excel, zie Figuur 9-1 voor de lay-out van het model met de kanalen.

Voor het onderscheid worden de klanten van type A met een blauw bolletje weergegeven en die van type B met een rood bolletje.



Figuur 9-1: Lay-out van de bank

Op de Servers moet nu een bedieningstijd geloot worden. Echter, deze is afhankelijk van het type klant, dus is er sprake van een keuze uit meerdere formules!

Dit is opgelost met *labels*. Labels zijn etiketten die - meestal - aan producten gehangen worden. Ze kunnen een kleur, gewicht, barcode of bedieningstijd voorstellen. Een product kan meerdere labels met zich meedragen en het aantal labels per product kan ook verschillen.

Het idee is nu om de bedieningstijd van een klant in een vroeg stadium te loten en de uitkomst als label mee te geven en af te lezen wanneer de bediening begint. Het mooie van dit idee is dat op de Server niet naar klanttype gekeken hoeft te worden, maar alleen naar het label van de klant. Bovendien is het idee ook makkelijk toe te

passen bij drie of meer klanttypen!

We hangen nu aan ieder product op de 'Trigger on Exit' van de eerste resp. tweede Source een label, genaamd servicetime. Deze bevat de juiste duur van de bedieningstijd:

```
Label([servicetime], i) := mins(1)           voor klanten van type A
Label([servicetime], i) := mins(10)        voor klanten van type B
```

Belangrijk!

De noodzaak van een verdere verdieping in de onderliggende structuur van ED en de programmeertaal 4DScript is duidelijk. Voor meer inzicht in de syntax en opbouw van 4DScript is dit het moment om de Engelstalige tutorial *A first start in 4DScript* door te spitten!



Dubbelklikken in velden waar 4DScript ingevoerd kan worden, activeert de 4DScript-editor. De F2-toets geeft een lijst met 4DScript woorden en een beknopte uitleg per commando.

Bij de cycle time van de Servers wordt het gewenste label opgevraagd en gebruikt als bedieningstijd:

`Label([servicetime], First(c))`. In plaats van rechtstreeks een getal in te voeren, wordt dus een 4DScript commando gebruikt dat een getal oplevert.

Zie de tutorial *A first start in 4DScript* voor de syntax van het commando Label en een uitleg van de betekenis van `First(c)`.

Het volledige model is te vinden als bank1a.mod. Dit model kan eenvoudig omgevormd worden naar een systeem van gescheiden wachtrijen (bank1b.mod).

Vragen en opdrachten

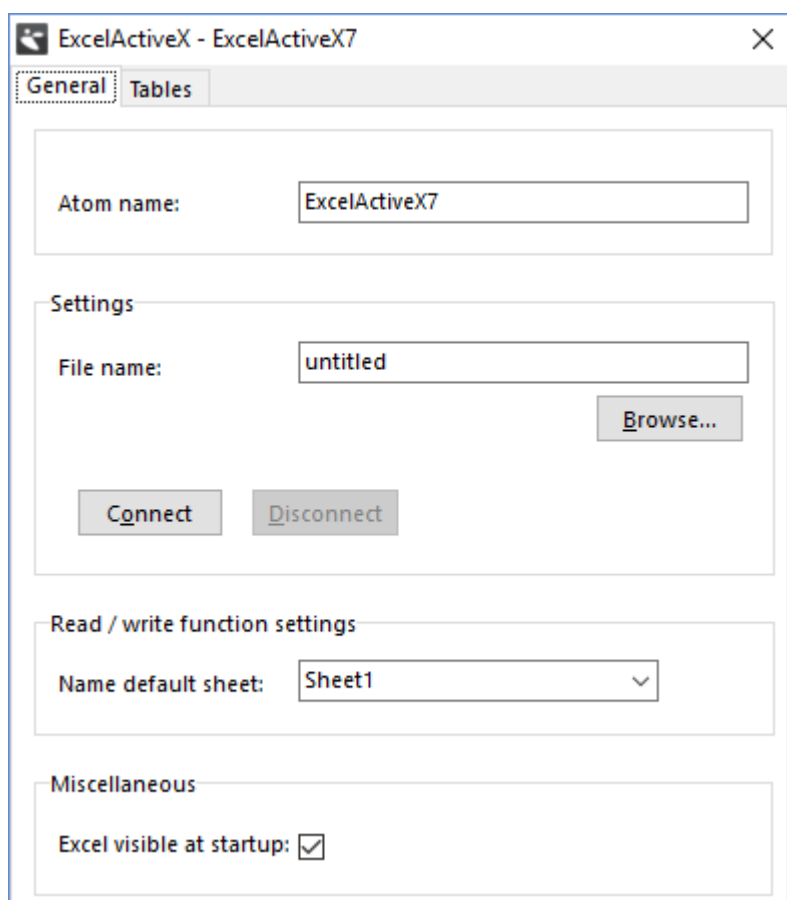
5. Schat de gemiddelde wachttijden in beide systemen op basis van een aantal runs van 100 uur. Kloppen de antwoorden met de eigen intuïtie?

9.2 De koppeling met Excel

In plaats van alleen te vertrouwen op de statistieken in ED, schrijven we de ruwe data zoals wachttijden en doorlooptijden weg in Excel. Daarna kunnen we de analysemogelijkheden binnen Excel gebruiken om bijvoorbeeld gemiddelden te bepalen.

Algemene procedure

1. Creëer een nieuw Excelbestand in dezelfde directory als het model. In de meeste gevallen zal dit in je documenten zijn: `...\Documents\Enterprise Dynamics\Models`. In dit voorbeeld is het bestand bank.xls genoemd.
2. Sleep het ExcelActiveX atoom in het model. Dit atoom is te vinden in de bibliotheek in de categorie 'Data'. Dit atoom 'organiseert' hierna de verbinding tussen ED en een Excel bestand. Een ExcelActiveX atoom hoeft niet verbonden te worden met andere atomen via kanalen.
3. Dubbelklik op het ExcelActiveX atoom. Het volgende scherm verschijnt:

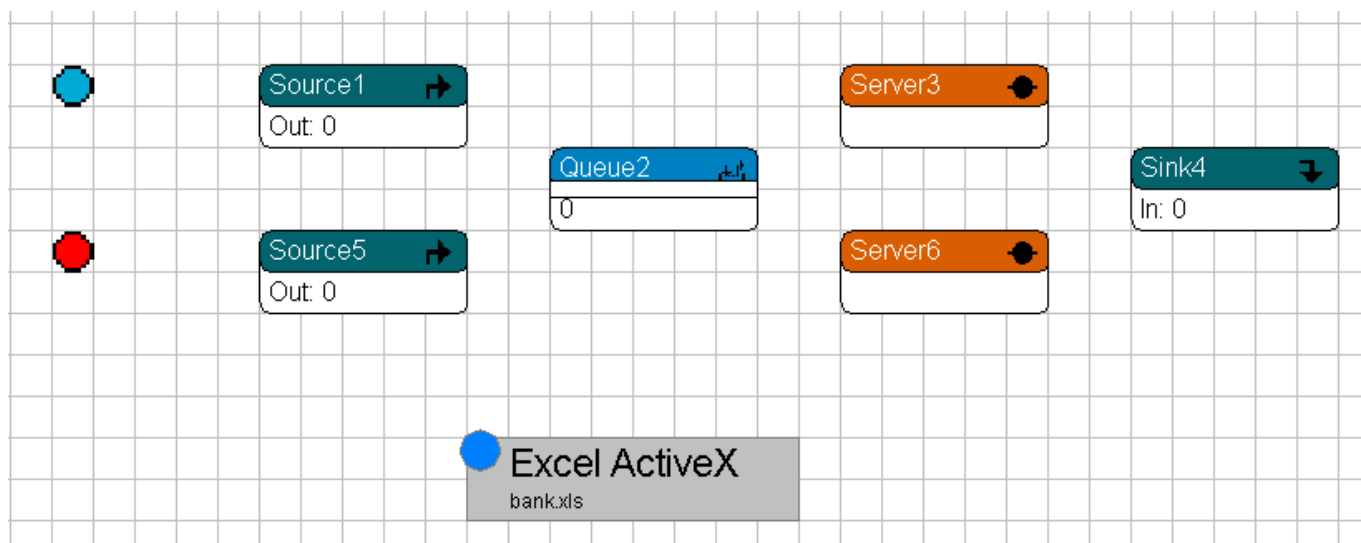


Figuur 9-2: Het invoerscherm van het ExcelActiveX atoom

Selecteer via Browse het zojuist gecreëerde Excelfile. Indien gereed, klik op OK en de verbinding met Excel wordt tot stand gebracht, waarbij het Excel icoon onder aan het scherm verschijnt. Vervolgens kan een werkblad van dit bestand als standaard opgegeven worden. *Let Op:* De naam van het hier opgegeven *default sheet* (werkblad) moet corresponderen met één van de werkbladen in het Excel bestand! De `ExcelActiveX_Read` en `ExcelActiveX_Write` functies zullen van dit werkblad lezen en schrijven behalve als een parameter met een ander werkblad wordt meegegeven. Als je Nederlandse instellingen hebt zul je dus de Name default sheet waarschijnlijk moeten veranderen in *Blad1*. Als de connectie al tot stand gebracht is kun je via de drop down lijst eenvoudig een werkblad van je bestand selecteren. Vink tevens 'Excel visible at startup' aan om Excel op de achtergrond van ED te zien werken.

De koppeling is nu tot stand gebracht en -bij voldoende uitzoomen- leesbaar op het ExcelActiveX atoom. Bij het opnieuw openen van dit model volgt de koppeling automatisch, de stappen 1 tot en met 3 hoeven dus niet elke keer handmatig uitgevoerd te worden!

Wanneer alles naar wens verlopen is, ontstaat de volgende lay-out:



Figuur 9-3: De bank met Excelverbinding

9.3 Wegschrijven van gegevens naar Excel

Om de koppeling in deze casus in zijn werk te zien, schrijven we van iedere klant zijn wachttijd weg naar Excel. Dit gebeurt door het volgende commando op de 'Trigger on Exit' van de Queue:

```
ExcelActiveX_Write(Output(c), 1, Age(i)).
```

`ExcelActiveX_Write(a,b,c)` schrijft de uitkomst van de expressie `c` weg op cel(a,b) van de default sheet in het Excelbestand. Het commando `Age` houdt de leeftijd van een product sinds de binnenkomst in het model bij. Deze wordt hier bepaald op het moment van het verlaten van de Queue en is daarom gelijk aan de wachttijd van het product! Het commando `Output` geeft het aantal producten dat op dat moment het betreffende atoom verlaten heeft en fungeert hier als teller. Samengevat: de wachttijd van klant `i` wordt weggeschreven in rij `a`, kolom `b` van het Excelbestand `bank.xls`.

Stel de snelheid van de simulatie nu laag in en reset de simulatie. Maak vervolgens een run en switch via de taakbalk van ED naar het Excelbestand. Als alles goed gegaan is, gebeurt er iets prachtigs: simultaan aan het proces in ED worden de wachttijden weggeschreven in de eerste kolom van `bank.xls`!

Sta even stil bij dit moment: wat met één gegeven kan, kan ook met veel meer gegevens. Het is dus op een simpele manier mogelijk om grote hoeveelheden ruwe data te bewaren en weg te schrijven om vervolgens het hele beschikbare arsenaal van Excel-bewerkingen hierop los te laten!

Voor de personen waarbij niets verschijnt of een grote hoeveelheid #####. Het probleem met een grote hoeveelheid ##### treedt soms op wanneer de kolombreedte onvoldoende is om alle gegevens in een cel te tonen. Door de betreffende kolom breder te maken, wordt dit probleem opgelost.

Voor het wegschrijven van doorlooptijden in de tweede kolom volstaat op de Trigger on Entry van de Sink:

```
ExcelActiveX_Write(Input(c), 2, Age(i)).
```

Let op: de wachttijd van klant 100 staat in rij 100, kolom 1, maar de doorlooptijd van diezelfde klant 100 hoeft niet op rij 100, kolom 2 te staan. Waarom?

Vragen en opdrachten

6. Bepaal in Excel de gemiddelde wachttijd en doorlooptijd van de eerste 1000 klanten. De modellen met de Excel koppeling staan in de directory met voorbeelden onder de namen `bank2a.mod` en `bank2b.mod`. Het Excel bestand heet `bank.xls`.

Vragen en opdrachten (Voor de liefhebbers)

7. Schrijf de wachttijden van klanten van type A in kolom 3 en de wachttijden van klanten van type B in kolom 4 en bepaal -bij voldoende klanten- de gemiddelde wachttijd. Stemmen de resultaten overeen met eerdere resultaten?

9.4 Inlezen van gegevens uit Excel

Het koppelen van ED en Excel is gelukt evenals het wegschrijven van uitvoerdata, maar...het inlezen van data uit Excel komt ook vaak voor. Dit tonen we nu als laatste.

In de context van ons voorbeeld zouden daadwerkelijk gemeten tussenaankomsttijden van klanten beschikbaar kunnen zijn. Dit aankomstpatroon willen we inlezen en gebruiken voor een simulatie in ED.

Genereer via een formule 100 (positieve!) getallen in de eerste en tweede kolom van bank.xls. Of gebruik natuurlijk de zojuist tot stand gekomen wachttijden... In beide gevallen is het handig om de eerdere `ExcelActiveX_Write` commando's even weg te halen.

Schrijf nu op de Source voor klanttype A bij de Interarrival-time: `ExcelActiveX_Read(Output(c)+1,1)` en bij Trigger on Exit: `Label([servicetime], i) := mins(1)`.

Schrijf op de bijbehorende Trigger on Enty van de Queue: `if(Output(c) = 100, CloseOutput(c))`.

Het commando `ExcelActiveX_Read(a,b)` leest de waarde in cel(a,b) van de default sheet van het Excel bestand, terwijl het commando `CloseOutput` er voor zorgt dat na 100 klanten de uitgang van het Source atoom sluit, waardoor ook het inlezen stopt. Wanneer dit niet gebeurt, blijft ED doorlezen en zal het lege cellen als een tussenaankomsttijd van 0 interpreteren.

Zoek nu zelf uit hoe de statements voor klanten van type B moeten luiden! Dit model is te vinden onder de naam bank3.mod.

Ter afsluiting: aankomstpatronen worden met een Source gegenereerd. Voor vaststaande productieschema's of dienstregelingen gebruiken we in ED het atoom Arrivallist. Dit atoom kan ook gebruikt worden om (grote) schema's met het juiste format vanuit Excel te knippen en plakken. Vanaf dat moment is de data in een interne tabel beschikbaar en is, althans voor data-invoer, een openstaande koppeling met Excel overbodig.

Inlezen via een tabel in ED zorgt voor een grote snelheidswinst en is voor de meeste situaties dan ook sterk aan te raden!

9.5 To pool or not to pool?

De resultaten van het vernieuwde postkantoor tonen iets opmerkelijks: het model met een *gemeenschappelijke* wachtrij (bank1.mod) heeft een hogere gemiddelde wachttijd dan het model met *gescheiden* wachtrijen (bank2.mod)!

Wij vonden de volgende uitkomsten:

gemiddelde wachttijd	gemeenschappelijke wachtrij (in minuten)	gescheiden wachtrijen (in minuten)
klanten van type A	6.18	2.45
klanten van type B	6.18	25.7
Gemiddelde over <u>alle</u> klanten	6.18	4.56

Figuur 9-4: Uitkomsten van een paar simulatie runs

Hoe is dit nu mogelijk? Leren we niet overal dat combineren tot verbetering leidt?

In het geval van gescheiden wachtrijen kan immers een rij van klanten van type A voor de eerste balie ontstaan, terwijl voor de balie bestemd voor klanten van type B niets te doen is. De verklaring luidt: *variatie!*

'Combineren' leidt tot vermindering van gemiddelde wachttijden indien van één type klant sprake is en niet van twee zoals in dit geval.

Zonder al te diep in de wachtrijtheorie te duiken: bij het systeem met gescheiden wachtrijen is sprake van twee onafhankelijke M/D/1-modellen. Hiervoor bestaan gesloten formules voor de belangrijkste wachtrij karakteristieken.

Deze geven voor de gemiddelde wachttijd W_q van klanten van type A resp. B: $W_q = 2.5$ minuten resp. $W_q = 25$ minuten.

De gemiddelde verwachte wachttijd over alle klanten bedraagt hierdoor $50/55 \times 2.5 + 5/55 \times 25 = 4.54$ minuten.

Bij het systeem met een gezamenlijke wachtrij ontstaat een M/G/k systeem met twee klanttypen. Voor het

kunnen toepassen van de formules uit de wachtrijtheorie 'mengen' we de klanten tot één klanttype waarvan er gemiddeld 55 per uur aankomen met een verwachte bedieningstijd van $E[S]=50/55 \times 1 + 5/55 \times 10 = 20/11$ oftewel 1.81 minuten.

De variantie $\text{Var}[S]$ van deze bedieningstijd is 6.69 en de variatiecoëfficiënt C , gedefinieerd als $\text{Var}[S]/(E[S])^2$, is 2.025.

Dit geeft bij benadering -formules zijn niet gegeven- dat $W_q = 6.25$ minuten.

De resultaten van onze runs komen aardig met deze theoretische waarden overeen: *De gemiddelde wachttijd bij gescheiden rijen loopt met 27% terug!*

De verklaring? De irreguliere klanten, waarvan de bedieningstijd 10 maal zo lang is als die van de reguliere klanten, verstoren bij gescheiden wachtrijen de wachttijd van de reguliere klanten niet meer. Hoewel dit tot gevolg heeft dat de verwachte wachttijd van de irreguliere klanten erg lang wordt, vermindert de gemiddelde wachttijd over alle klanten fors.

Vragen en opdrachten

8. Pas het model van de gescheiden wachtrijen zo aan dat eenzijdige overloop voor klanten van type A mogelijk is. Dit wil zeggen dat de eerste in de wachtrij van type A gebruik mag maken van het loket voor klanten van type B, mits er geen wachtenden van type B zijn. Je kunt natuurlijk ook experimenteren met tweezijdige overloop.

Wat betekent dit voor de gemiddelde wachttijd voor de klanten?

Hint: met toevoegen van kanalen en het juiste 'send to' statement is dit vrij eenvoudig te realiseren!

10 De Human Resource

In dit hoofdstuk staat de Human Resource centraal. De Human Resource is een capaciteitsbron die op meerdere bewerkingsplaatsen nodig is. Dit is meestal een persoon van vlees en bloed, maar de Human Resource kan ook een stuk gereedschap voorstellen. We beginnen met een casus waarin we de Human Resource nodig hebben, maar de verplaatsingstijd tussen machines te verwaarlozen is. Vervolgens gaan we in op de wijzen waarop Enterprise Dynamics afstanden en beweging nabootst. De casus wordt daarna uitgebreid met de vrije beweging van de Human Resource in het platte vlak én volgens vastgelegde looppaden.

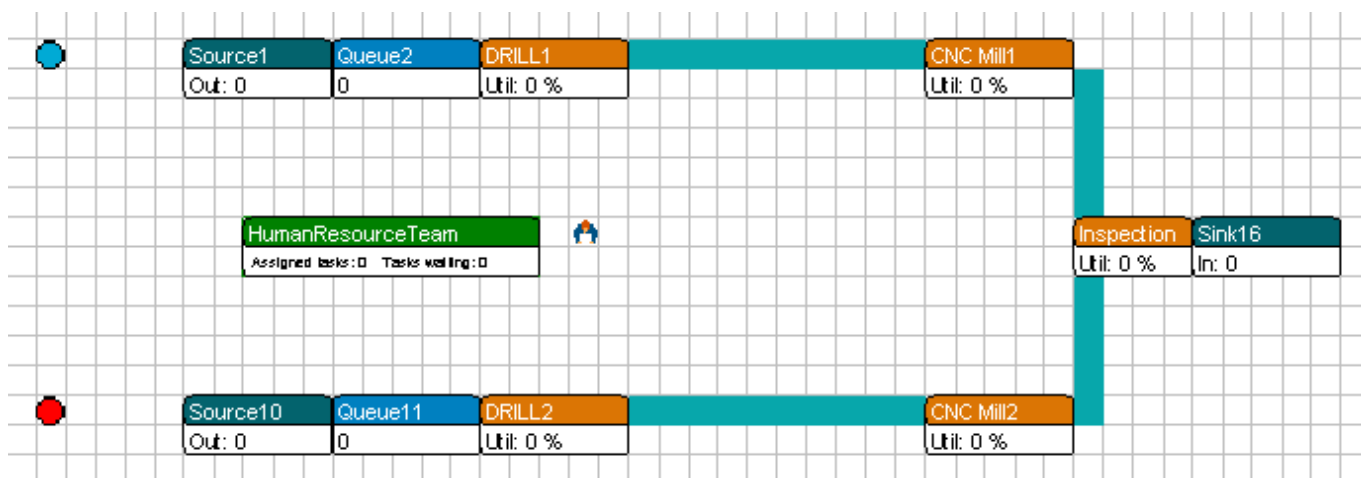
10.1 Aan het werk met de Human Resource

Casus 5: De human resource voor Red en Blue

Twee halffabricaten HalfBlue and HalfRed arriveren bij twee gescheiden, maar identieke productielijnen. Iedere productielijn bestaat uit een boormachine voorafgegaan door een opslagplaats. Na de boormachine vervoert een conveyor van tien meter lengte de producten naar een CNC machine. Vanaf de CNC machine gaan de producten via een vijf meter lange conveyor naar de eindcontrole. Hier worden zowel de rode als blauwe producten op volgorde van aankomst geïnspecteerd. Daarna verlaten de producten het systeem

De halffabricaten arriveren ieder onafhankelijk van elkaar met een gemiddelde tussenaankomsttijd van een uur. Het boren duurt gemiddeld 20 minuten en de inspectie gemiddeld 6 minuten. Alle tijden volgen een exponentiële verdeling. De CNC machines daarentegen zijn zeer constant en hebben een bedieningstijd van 10 minuten.

Een operator doet zowel de eindcontrole als het boren van beide producten en loopt dus voortdurend tussen de boormachines en eindinspectie heen en weer. Voorlopig veronderstellen we dat deze looptijden te verwaarlozen zijn.



Figuur 10-1: De productielijn voor de halffabricaten

De lay-out van dit productiesysteem in Enterprise Dynamics is te zien in Figuur 10-1. Bouw dit maar even na!

Vragen en opdrachten

1. Hoe hoog is de (theoretische) bezettingsgraad van de operator?
2. Ga na dat de bezettingsgraden van de boormachine(s), de CNC machine(s) en inspectie (als plaats) resp. 33%, 16% en 20% bedragen.

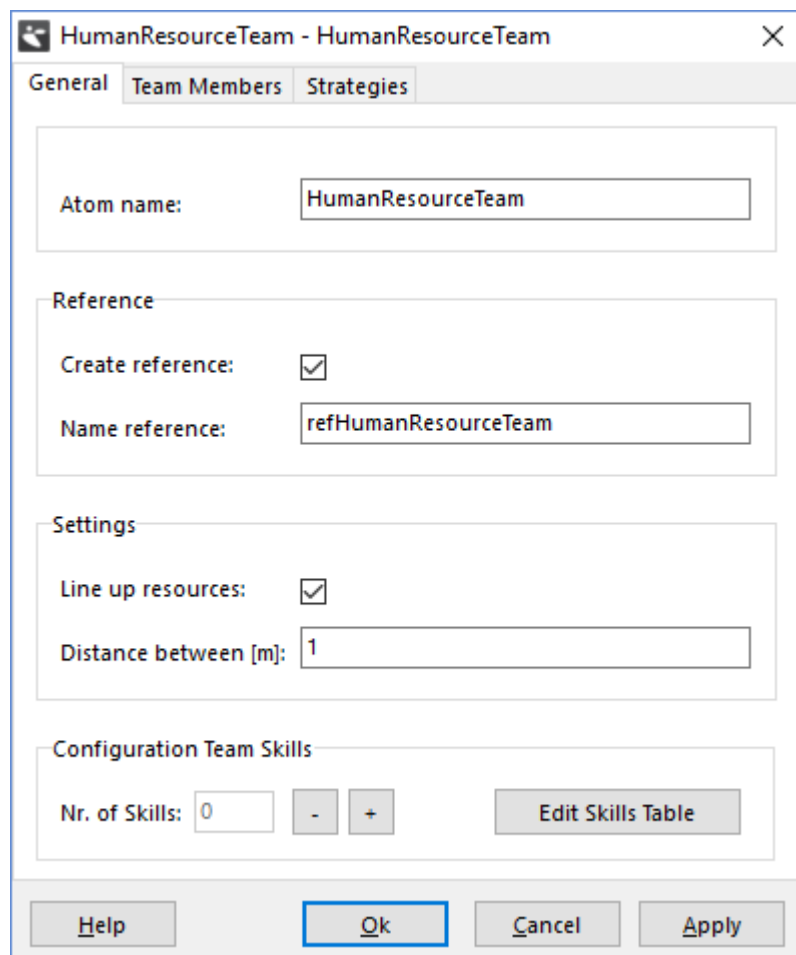
Controleer deze antwoorden door het model lange tijd te laten runnen: deze antwoorden moeten ongeveer terugkomen!

Zowel voor het boren als de inspectie zijn de bewerkingsplaatsen als Servers weergegeven. Hier geschieden immers handelingen die een bepaalde tijd kosten. Toch is dit onvoldoende: de lay-out klopt, maar er is een persoon die heen en weer wandelt tussen het twee boormachines en de inspectie.

Het idee is nu om die plaatsen waar de handeling fysiek geschiedt gewoon te modelleren, in dit geval als Server. Wanneer nu een product op de plaats van het boren of de inspectie arriveert, wordt de assistentie van een Human Resource (operator) gevraagd. Pas wanneer deze ter plekke is, kan de bediening starten. Wanneer de Human Resource (operator) niet meer nodig is, wordt hij weer vrijgegeven.

In Enterprise Dynamics is dit uitgewerkt via het Human Resource atoom en het HumanResourceTeam atoom. Iedere Human Resource is deel van een team. Een team kan uit één of meerdere Human Resources bestaan.

We slepen nu een Human Resource en een HumanResourceTeam atoom in het model. Beide atomen zijn te vinden in de bibliotheek in de categorie Human Resources. Door te klikken op de HumanResourceTeam verschijnt het invoerscherm weergegeven in Figuur 10-2.

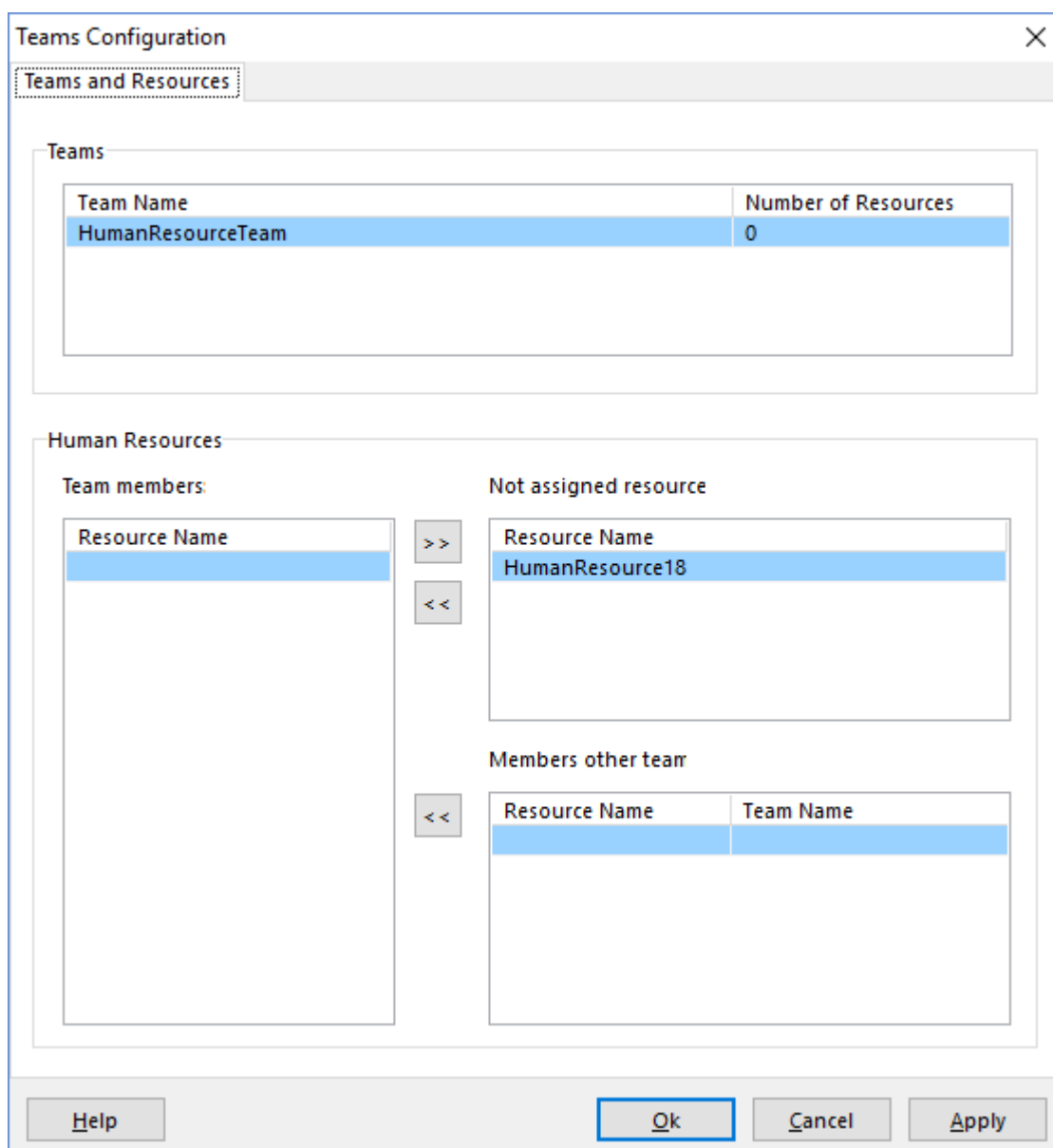


Figuur 10-2: Invoerscherm van het HumanResourceTeam atoom

Ga naar de *Team Members* pagina en klik *Team Configuration*. Een nieuw scherm opent, zie Figuur 10-3. De tabel boven aan het team configuratiescherm toont alle teams in het model. In dit geval hebben we slechts één in ons model en deze staat geselecteerd in de Teams tabel. Merk op dat de Human Resource die we in het model geslept hebben te vinden is in de tabel *Not Assigned Resources*. Klik op de knop << om de Human Resource toe te voegen aan de *Team Members* tabel. De Human Resource is nu toegevoegd aan het HumanResourceTeam.

Het is erg belangrijk om een atoom referentie naar het team te creëren, dit maakt het makkelijk om aan te geven van welk team je een resource nodig hebt. Ga naar de *General* pagina van het HumanResourceTeam en vink de optie *Create reference* aan. We gebruiken nu als *Name Reference* `refHumanResourceTeam`. Deze referentie naam moet uniek zijn.

De plaats van beide atomen in de model layout doet overigens niet ter zake...



Figuur 10-3: Team configuration Scherm

Nu moeten we nog regelen dat de Human Resource aangeroepen wordt bij de inspectie en het boren. Omdat dit nodig is bij het arriveren van een product, ligt het voor de hand om dit in de Triggers te regelen. Op ieder Server atoom staat in de standaardlijst van de triggers een voorbeeld hoe een Human Resource aan te roepen en vrij te geven. Selecteer de standaard logica en pas deze aan zodanig dat de volgende code op de triggers van het inspection en drilling atomen komt te staan:

```
Trigger on Entry:   HumanResourceTeam_CallResources(refHumanResourceTeam, c, i, 1)
Trigger on Exit:   HumanResourceTeam_FreeResources(refHumanResourceTeam, i)
```

De call en free functies hebben een referentie naar het team waar een resource van gevraagd wordt nodig. Eerder hebben we het team een referentie genaamd `refHumanResourceTeam` gegeven. Deze referentie gebruiken we nu als eerste parameter van de functies. De tweede parameter van de call functie vraagt om een atoom referentie naar de locatie waar de taak uit gevoerd moet worden. Meestal en ook in dit geval zal de taak locatie hetzelfde atoom zijn als van waaruit de aanroep gedaan wordt, daarom kunnen we gebruik maken van de atoom referentie `c`. De derde parameter is een referentie naar het atoom waar de handeling op uitgevoerd dient te worden, dit is het product dat net binnen is gekomen, dus kunnen we `i` gebruiken om naar dit product te verwijzen. De laatste parameter van de call functie geeft aan hoeveel Human Resources we nodig hebben voor deze taak.

Omdat beide commando's een grote rol spelen, volgt hier de volledige syntax:

HumanResourceTeam_CallResources (e1, e2, e3, e4, {e5, e6})

Dit commando heeft 6 variabelen; variabelen 5 en 6 zijn optioneel. Hierbij is:

- e1 een referentie naar het HumanResourceTeam
- e2 een referentie naar de task location
- e3 een referentie naar het served atoom
- e4 het aantal Human Resources dat nodig is voor de taak
- e5 een waarde waarmee een prioriteitsregel voor de taal gedefinieerd kan worden
- e6 de benodigde vaardigheid die een Human Resource moet hebben om de taak uit te kunnen voeren

Voorbeeld: `HumanResourceTeam_CallResources (refHumanResourceTeam, c, i, 2)` roept 2 Human Resources. De Human Resources worden toegekend aan het product van het atoom waarop dit callstatement aangeroepen is. Alle events op dit atoom worden uitgesteld tot de benodigde hoeveelheid HumanResources aanwezig is.

HumanResourceTeam_FreeResources (e1, e2)

- e1 een referentie naar het HumanResourceTeam
- e2 een referentie naar het served atoom

Voorbeeld: `HumanResourceTeam_FreeResources (refHumanResourceTeam, i)` zorgt dat alle Human Resources die werkten aan het served atoom en behoorde tot het team gerefereerd met `refHumanResourceTeam` worden vrijgelaten.

Open `human resource11.mod` in Tutorial Models of pas de Server atomen op juiste wijze in het eigen model aan. Run het model om te zien dat de Human Resources steeds verspringt van de ene boormachine naar de andere of de inspectieplaats. Bewerkingsplaatsen die willen aanvangen met een bewerking maar nog niet over de Human Resource kunnen beschikken, kleuren rood!

Elke keer als er een oproep gedaan wordt voor een Human Resource wordt er een taak aan het team toegekend. Deze worden standaard op volgorde van binnenkomst afgewikkeld (FIFO). Dit kan er toe leiden dat het model vastloopt in een zogeheten deadlock, omdat een boormachine zijn producten niet meer kwijt kan. Maak een lange run om dit te bestuderen. Later zullen we kijken hoe deze deadlock te vermijden is.

De bezettingsgraad van de Human Resource is te achterhalen met behulp van het StatusMonitor atoom of via de Experiment Wizard en zal meer dan 80% bedragen (zie status Busy). Klopt dit met de eigen berekening?

Kortom, het blijkt dat het model de toestroom goed aan kan, maar dat de Human Resource hard zal moeten werken terwijl het lopen (nog) geen tijd kost. En over dit 'lopen' gaat de volgende paragraaf...

10.2 Verplaatsen in de ruimte

Een verplaatsing van de Human Resource kan op 3 manieren gebeuren:

1. Verspringen zonder tijdsduur (standaard)
2. Lopen
3. Lopen via een netwerk

De eerste manier gebruiken we wanneer looptijden of transporttijden in het probleem te verwaarlozen zijn. Bij de tweede manier introduceren we transporttijd als gevolg van afstand tussen twee locaties zonder ons druk te maken hoe de route afgelegd wordt. Bij de laatste manier verplaatst de human resource zich over een netwerk van paden en knooppunten. Dit is bedoeld voor situaties zoals orderpickers in gangpaden van een magazijn, maar ook voor voertuigen als heftrucks en AGV's die in hoofdstuk 11 door Transporters nagebootst worden.

De complexiteit van verplaatsingsgedrag neemt via manier één tot drie duidelijk toe. Stel daarom als modelbouwer altijd de vraag of het toevoegen van transporttijden en transportnetwerken relevant is in het kader van het probleem!

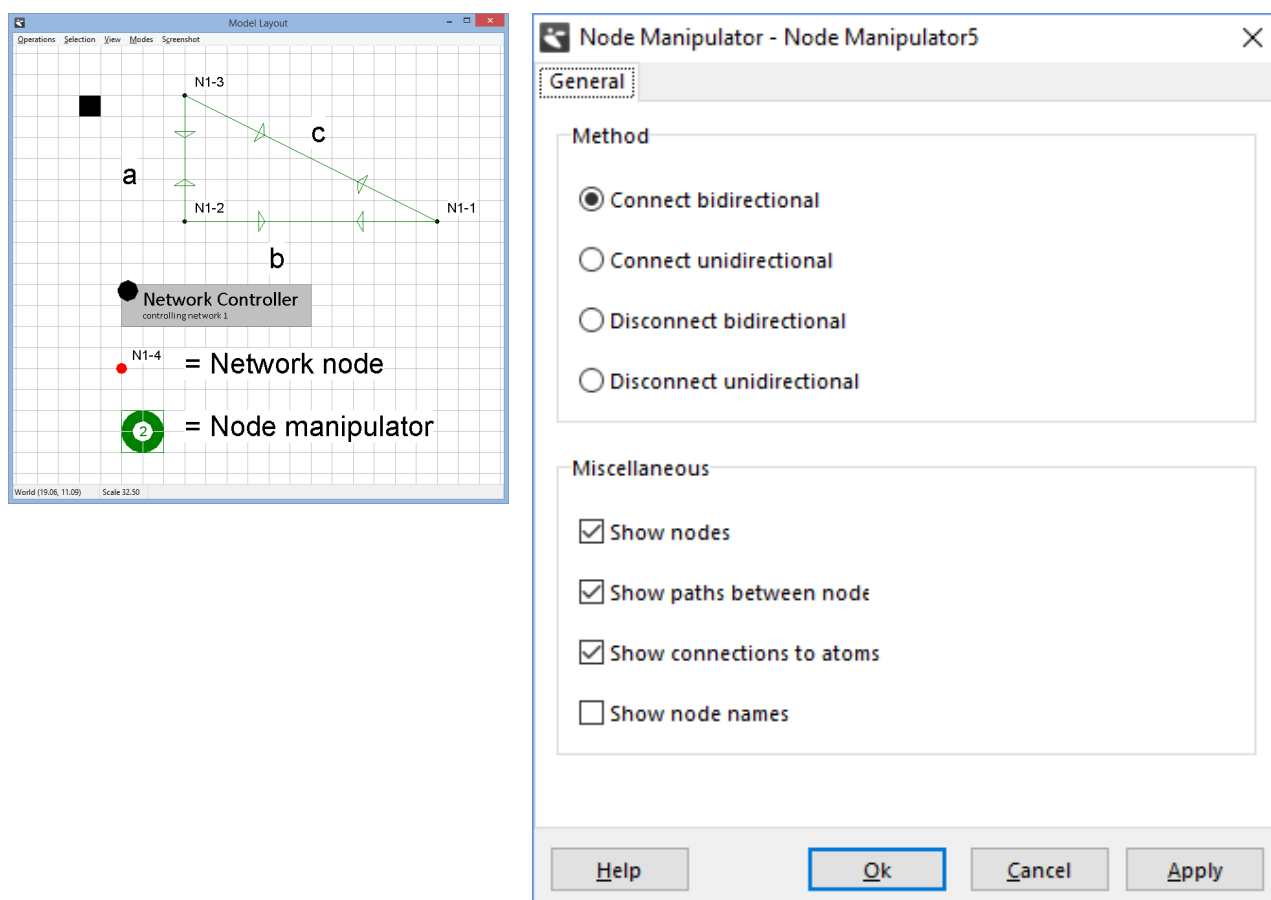
Alvorens het verplaatsen van de Human Resource in meer detail te bestuderen, eerst meer informatie over

afstanden en ruimtegebruik in Enterprise Dynamics. De voorbeeldmodellen in deze tutorial maakten tot nog toe, met uitzondering van de conveyor in casus 4, geen gebruik van de fysieke ruimte. Wat nu indien de locatie, de grootte en de onderlinge afstanden tussen plaatsen wel een rol spelen?

In Figuur 10-4 zien we het raster dat steeds achter de atomen zichtbaar is, naast een aantal andere atomen die later aan bod komen. Dit raster fungeert als het coördinaatvlak waarbij ieder vak een afmeting van 1 bij 1 meter heeft. Het zichtbare zwarte vak correspondeert linksboven met de coördinaten (0,0) en rechtsonder met (1,1). Dit betekent dat Enterprise Dynamics de zuidoosthoek als eerste kwadrant opvat, waar dit normaal gesproken op de noordoosthoek betrekking heeft. Linksonder in de Model Layout staan de coördinaten van de cursor weergegeven.

Beweeg zelf de cursor in Enterprise Dynamics om enig gevoel voor het coördinatenstelsel te krijgen!

In Figuur 10-4, zien we de drie punten; N1-1, N1-2 en N1-3, die een driehoek vormen.

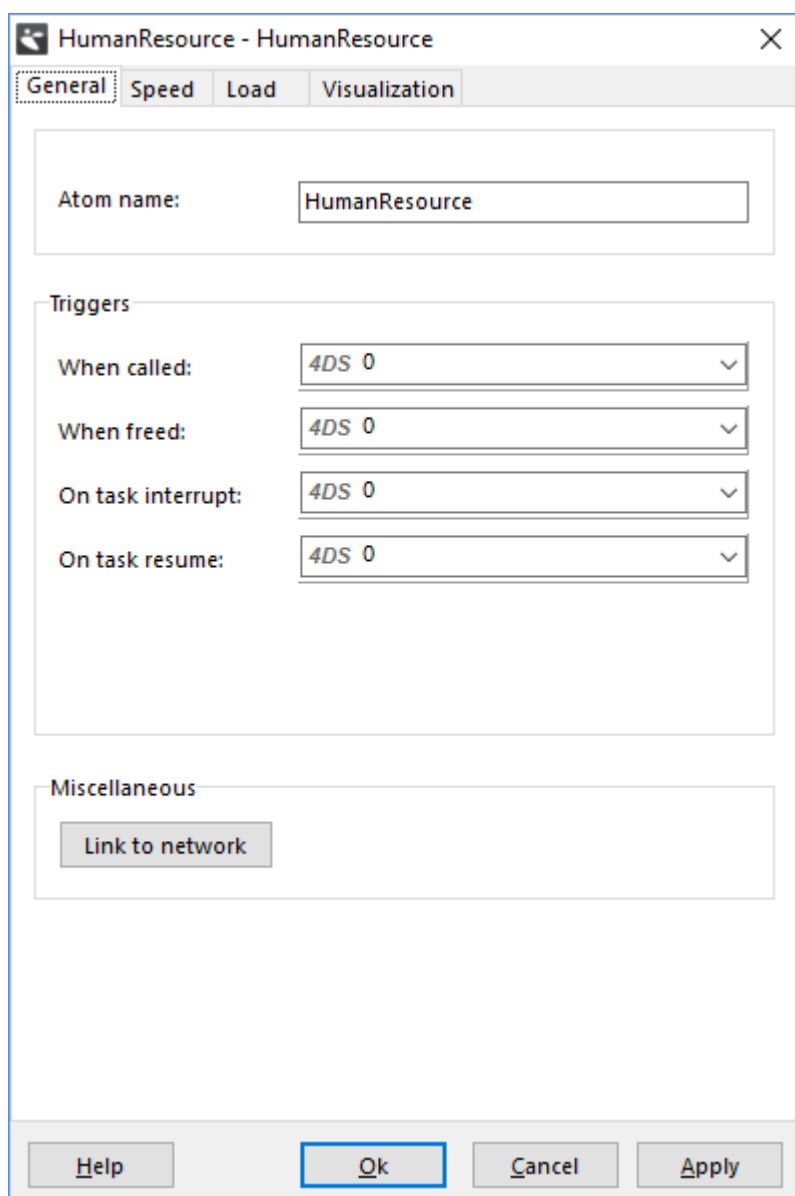


Figuur 10-4: Coördinaten, afstanden & een aantal netwerkatomen

De afstand van N1-1 tot N1-2 bedraagt 5 meter: indien een human resource met standaard snelheid van 1 m/s tussen deze punten wandelt, duurt dit dan ook 5 seconden. Voor de afstand c geldt de stelling van Pythagoras: $c^2 = a^2 + b^2$. Ga na dat $c = 13$!

10.3 Lopen (vrije beweging)

Wanneer voor de Human Resource afstanden een rol spelen, kunnen we de standaardinstelling 'verspringen zonder tijdsduur' omzetten in 'lopen'. Dubbelklik op de Human Resource om het invoerscherm te openen, zie Figuur 10-5. Vink de optie 'Walk' aan dat te vinden is op de Speed pagina van het HumanResource invoerscherm. De optie 'Walk' zorgt ervoor dat de Human Resource zich met een zelf in te stellen snelheid (standaard 1 m/s) tussen zijn huidige locatie en een nieuwe locatie verplaatst. De gehanteerde afstandsformule is de formule van Pythagoras. De afstand wordt dus hemelsbreed bepaald, los van enig tussenliggend object. We spreken van vrije verplaatsing ('free range').



Figuur 10-5: Invoerscherm van de Human Resource

Attentie!

Atomen verschillen van grootte. De maatgevende coördinaten zijn steeds die van het punt in de linker bovenhoek van een atoom!

De overige velden van het Human Resource atoom worden later uitgebreider behandeld.

Casus 5 (vervolg): De operator voor Red en Blue

Pas nu *human resource1.mod* aan zodat de Human Resource loopt zoals beschreven of open *human resource2.mod*.

Vragen en Opdrachten

Run het model en merk op dat:

3. De human resource zich inderdaad schuin over het scherm van een boormachine naar de inspectie en vice versa verplaatst.
4. Dat de animatie van de human resource bij de Server verschilt van de animatie in het vorige model.

Bereken en controleer in het model dat genoemde verplaatsing iets minder dan 21 seconden duurt ($\sqrt{436}$ seconden om precies te zijn).

De bezettingsgraad van de Human Resource gaat ook omhoog: op een uur zal gemiddeld 40 minuten aan boren, 12 minuten aan inspectie, maar ook 1 à 2 minuten aan wandelen besteed worden. Dit kan via de Experiment Wizard

gecontroleerd worden...

10.4 Netwerken in het algemeen

In de vorige paragraaf is vrije verplaatsing in een directe lijn bekeken. Dit is niet altijd mogelijk door muren of andere obstakels. Vaak geschiedt beweging via vastgelegde paden.

In Enterprise Dynamics heeft dit geleid tot het concept van een *netwerk*. Een netwerk is een verzameling van punten en lijnen. Human Resources of Transporters kunnen aan een dergelijk netwerk gekoppeld worden waarna voor hen alleen verplaatsing via dit netwerk mogelijk is. De punten vormen de kruispunten in het netwerk, de lijnen vormen de paden. Deze paden kunnen eenrichtingswegen zijn of in twee richtingen bewandeld worden.

Voor het creëren en inbedden van een netwerk, volgen we de volgende stappen:

1. Sleep een aantal **Network Nodes** tezamen met een **Node Manipulator** in de Model Layout. Beiden zijn te vinden in de categorie *Network* en afgebeeld in Figuur 10-4.
2. Verbind twee gekozen punten door
 - a. het plaatsen van de Node Manipulator over het eerste punt
 - b. daarna rechtsklikken (Node Manipulator wordt geheel groen)
 - c. de Node Manipulator over het tweede punt te plaatsen
 - d. opnieuw rechtsklikken

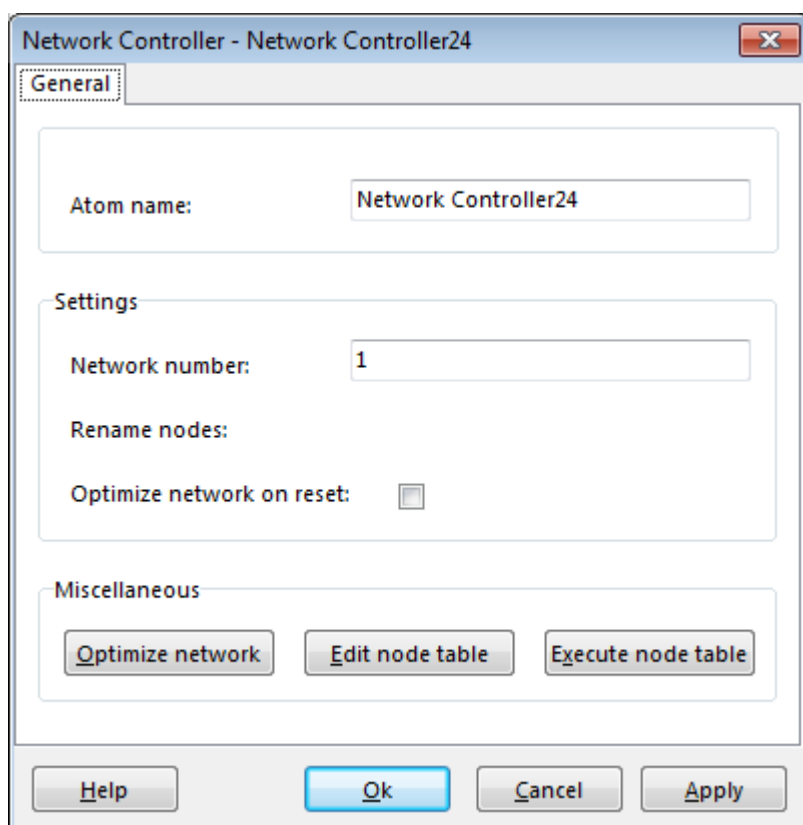
Tussen de 2 punten ontstaat een groene verbinding, die afhankelijk van de eigen keuze op de Node Manipulator, in een of twee richtingen doorlopen kan worden. Herhaal dit proces voor alle gewenste verbindingen.

3. Verbind op dezelfde wijze ieder van de atomen waar de Human Resource moet assisteren met een (dichtbij geplaatst) punt in het netwerk. *Let op*: rechtsklik bij zo'n atoom met de Node Manipulator op linkerbovenhoek van dit atoom, deze verbinding toont zich met een blauwe lijn.
4. Verbind de Human Resource met het netwerk door rechtsklikken met de cursor op de Human Resource en te klikken op *Link to network*, zie Figuur 10-5.
5. Sleep een **Network Controller** in de Model layout en kies na rechtsklikken voor de optie *Optimize Network*.

De vetgedrukte namen verwijzen naar nieuwe atomen, allen te vinden onder de categorie *Network*. De *Network Node* behoeft nauwelijks uitleg. In Figuur 10-6 zijn alle opties van de Node Manipulator te zien. Voor het weghalen van verbindingen wordt stap 2 opnieuw doorlopen, nu echter in de optie *Disconnect bidirectional* of *Disconnect unidirectional*.

Bij *Connect bidirectional* (tweerichtingsweg) resp. *Connect unidirectional* (eenrichtingsweg) wordt de Node Manipulator groen met een 2 resp.1 als opschrift. Bij *Disconnect bidirectional* resp. *Disconnect unidirectional* wordt de Node Manipulator rood met een 2 resp.1 als opschrift.

In Figuur 10-6 staan alle opties van de Network Controller. Maak opgave 5 om met deze atomen vertrouwd te raken, evenals bovengenoemde stappen en lees de Help file voor deze atomen!



Figuur 10-6: Invoerscherm van de Network Controller

Vragen en opdrachten:

5. Experimenteer met een nieuw model bestaande uit een Source, een Server, een tussenliggend netwerk, en een Conveyor gevolgd door een Sink. Probeer een paar eenrichtingswegen toe te voegen, verbindingen weer weg te halen, et cetera. Zorg dat de kortste route voor de heen- en terugweg verschilt. Verbind de Server en de Conveyor met het netwerk. Ontstaan er blauwe verbindinglijnen? Voeg een HumanResourceTeam met HumanResource toe. De HumanResource assisteert bij de Server en de Conveyor. Verbind de HumanResource met het netwerk zoals beschreven. Optimaliseer het netwerk tenslotte met de Network Controller.
Loopt de HumanResource netjes heen en weer over verschillende routes?

Important!

Optimize Network via de Network Controller maakt het mogelijk dat gebruikers van het netwerk steeds het kortste pad kiezen (intern wordt op het netwerk het algoritme van Dijkstra toegepast). *Het is daarom noodzakelijk dat bij iedere wijziging in het netwerk deze stap ook weer wordt uitgevoerd.*

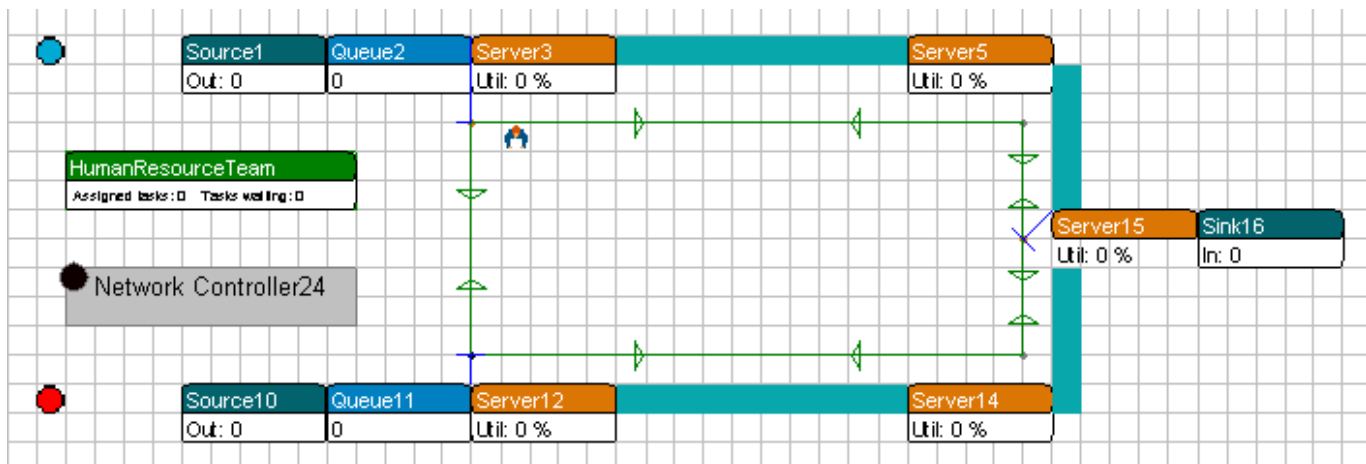
10.5 Lopen via een netwerk

Casus 5 (vervolg): De operator voor Red en Blue

Voeg een netwerk met 4 punten en 5 verbindingen zoals te zien is in Figuur 10-7 toe aan de eerdere modellen. De blauwe lijnen vormen de verbindingen tussen het netwerk en de atomen waar de Human Resource benodigd is. De snelheid van de human Resource blijft 1 m/s. Voer alle stappen beschreven in paragraaf 10-4 en toegepast in oefening 5 consequent uit.

Vragen en opdrachten

6. Ga na dat de Human Resource via het netwerk loopt en bovendien vanaf iedere locatie de kortste route naar de volgende locatie kiest. Merk op dat de Human Resource zich opstelt bij het knooppunt dat naar het atoom leidt waar de Human Resource gevraagd wordt.
7. Ga na dat het lopen van het inspectiepunt naar een boormachine 23 seconden duurt en tussen de boormachines 8 seconden. Dit betekent dat voor het afleggen van de blauwe verbindingen geen tijd wordt gerekend!



Figuur 10-7: De Human Resource in het netwerk
Dit model is te vinden onder *human resource3.mod*.

10.6 Prioriteiten toekennen

Wie lang genoeg één van deze drie modellen simuleert, zal waarnemen dat af en toe de conveyors vollopen tot de situatie ontstaat dat de human resource een product wil afgeven aan een conveyor, maar hiertoe niet in staat is omdat de conveyor vol is. Dit noemen we een *deadlock*: het systeem slaat vast omdat geen actie meer mogelijk is.

Dit is hier mogelijk doordat de human resource standaard alle gevraagde opdrachten FIFO afhandelt. Stel dat zowel voor de boormachines als voor de inspectie wachtrijen staan en dat de human resource op tijdstip 100 klaar is met assistentie bij inspectie. Boormachine 1 en 2 bevatten een product en schreeuwen al een tijdje om een human resource: dit is te zien doordat de atomen rood kleuren. Veronderstel verder dat de opdracht bij boormachine 1 op tijdstip 90 en bij boormachine 2 op tijdstip 95 binnenkwam.

Wat gebeurt er nu achtereenvolgens op tijdstip 100? Allereerst wordt de inspectie-opdracht beëindigd (loskoppeling Human Resource), daarna wordt een nieuw product bij inspectie geplaatst, vervolgens kijkt de human resource in zijn lijst met drie potentiële en kiest volgens het FIFO-principe de oudste. Let op dat deze lijst niet twee, maar drie potentiële opdrachten bevat. Dit zal straks cruciaal blijken bij de prioriteitsregels!

Dus wanneer de human resource op tijdstip 100 een nieuwe keus maakt, staat in zijn takenlijst:

Call	From	Time of call
1	Drilling machine 1	90
2	Drilling machine 2	95
3	Inspection	100

Figuur 10-8: Oproepvolgorde voor de human resource zonder prioriteiten

Op basis van het FIFO-principe gaat de human resource nu naar boormachine 1, vervolgens naar boormachine 2 en dan weer naar de inspectie. Het is nu duidelijk dat bij grote drukte de human resource rond blijft cirkelen en het systeem kan vollopen!

We willen nu aan de inspectietaak prioriteit verlenen. De syntax van het call-statement in paragraaf 10-1 biedt hulp. Verander op Inspectie de Trigger on Entry van

```
HumanResourceTeam_CallResources(refHumanResourceTeam, c, i, 1)
```

in

```
HumanResourceTeam_CallResources(refHumanResourceTeam, c, i, 1, 2)
```

De oproepen vanuit Inspectie hebben nu prioriteit 2, terwijl die van de boormachines (standaard)prioriteit 1 behouden. Een '1' toevoegen aan deze laatste statements mag, maar is niet nodig. Verder is het nodig dat de strategie van de taakafhandeling aangepast wordt van *FIFO* naar *By Priority descending*, dit is in te stellen door de *Sort task by* die te vinden is op de *strategie* pagina van het HumanResourceTeam invoerscherm in te stellen op

By *priority descending*. Wanneer de oproepen in Figuur 10-8 nu gesorteerd worden op prioriteit, en pas daarna op tijdstip van oproep, ontstaat een heel ander beeld:

Call	From	Time of call	Priority
1	Inspection	100	2
2	Drilling machine 1	90	1
3	Drilling machine 2	95	1

Figuur 10-9: Oproepvolgorde voor de operator met prioriteiten

Dit betekent in feite dat de rij inspectieopdrachten nu eerst wordt afgehandeld. Waarom?

Wie dit fenomeen in werking wil zien, kan het beste één van de drie human resource modellen zonder voorrang nemen, kijken wanneer het model dreigt vol te lopen, de run stopzetten en de prioriteit van de inspectie aanpassen, en vervolgens verder runnen.

De modellen met prioriteit zijn meegeleverd: *human resource1 with priority.mod*, enzovoorts.

10.7 Meer mogelijkheden van de Human Resource

Met Human Resources is veel meer mogelijk. Denk aan twee benodigde human resources bij eenzelfde taak, bijvoorbeeld bij het samen tillen van een zware plaat. Ook kan een human resource langere tijd bij een product blijven. Verder kan het zijn dat een taak alleen uitgevoerd kan worden door iemand met een bepaalde bekwaamheid (skill) dat niet door alle leden van het team beheerst wordt. Deze mogelijkheden zijn vervat in een opgave.

Vragen en opdrachten

8. Voeg in het model *human resource1.mod* een tweede human resource toe en realiseer het volgende:
 - a. Voor inspectie zijn 2 human resources gelijktijdig nodig.
 - b. Bij het boren blijft steeds een (willekeurige) human resource nodig, maar deze begeleidt dit product nu tot de CNC-machine.
 - c. Idem als twee, alleen verzorgt human resource Red nu de rode producten en human resource Blue de blauwe producten. Definieer hiervoor op het Team twee skills een voor het afhandelen van de rode en de andere de blauwe producten.

Deze drie veranderingen tezamen zijn te vinden in *human resource4.mod*.

Er zijn nog meer mogelijkheden met het Human Resource atoom dat we hier niet hebben beschreven. Het is bijvoorbeeld mogelijk dat een Human Resource een product verplaatst naar een andere locatie. Bekijk de *example models* die geleverd worden bij de installatie om meer demonstraties te zijn van de human resource atomen.

11 De Transporter

Het vervoer van producten over afstand gebeurt in de praktijk met vorkheftrucks, AGV's, Reach Trucks of andere voertuigen. In Enterprise Dynamics bootsen we deze voertuigen na met de Transporter of de Advanced Transporter

De Transporter kan vrij in de ruimte bewegen, maar ook gebonden zijn aan paden in een netwerk. Dit roept al direct associaties op met de Human Resource van het vorige hoofdstuk. Inderdaad gebruiken we voor de vrije beweging in de ruimte of aan gebondenheid aan paden voor de Transporter dezelfde concepten als behandeld voor de Human Resource. We raden daarom aan om eerst hoofdstuk 10 door te werken!

In de groep Floorbound van de categorie Transport zijn twee transporters te vinden zijn nl. de Transporter en de Advanced Transporter. Alleen de laatste kan over paden van een netwerk rijden In paragraaf één zetten we de mogelijkheden van beide transporters op een rij, waarna in paragraaf twee een casus rondom de Transporter volgt. In paragraaf drie herhalen we eerst deze casus voor de Advanced Transporter, waarna uitbreiding volgt door toevoeging van een netwerk. In de vierde paragraaf beschrijven we in meer detail de ideeën achter en samenhang van de Dispatcher, de Advanced Transporter en de Destinator. Paragraaf 5 gaat voor de Advanced Transporter in op laad- en losstrategieën, terwijl paragraaf 6 de gelijktijdige inzet van meerdere Advanced transporters beschrijft.

11.1 Functionaliteiten van de Transporter(s)

Met de Transporter is het volgende mogelijk:

1. Het verplaatsen van één product van ophaallocaties naar brenglocaties.
2. Het *automatisch* bepalen van rijtijden op basis van afstand en opgegeven snelheid tussen de coördinaten van haal- en brengpunten (vrije verplaatsing zonder netwerk).
3. Het toevoegen van versnelling en vertraging voor optrekken en afremmen, met behoud van de automatische bepaling van de juiste rijtijden.
4. Het toevoegen van laad- en lostijden bij het halen en brengen.

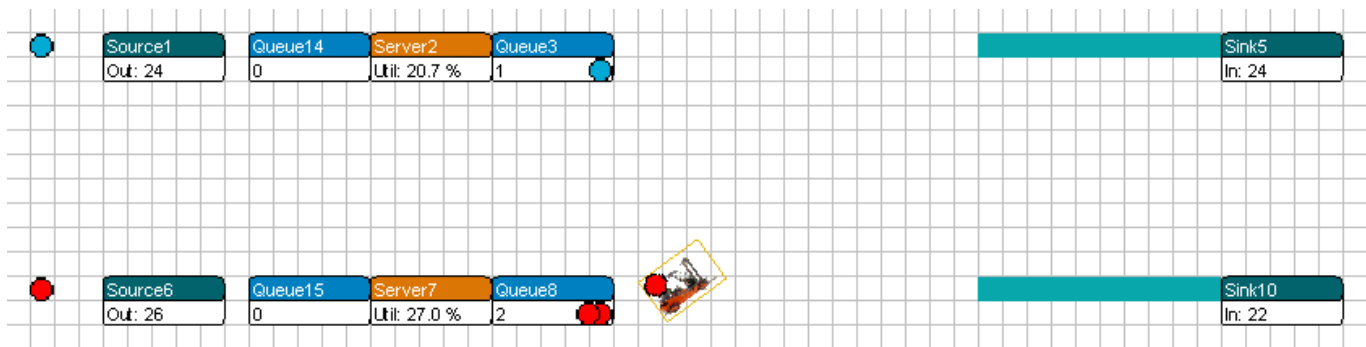
De Advanced Transporter heeft alle mogelijkheden van de Transporter plus:

5. Het gelijktijdig verplaatsen van *meerdere* producten van haal- naar brengpunten;
6. Het verplaatsen via de paden van een netwerk, met instelling van het maximum aantal voertuigen op een verbinding;
7. Het heffen van producten dwz. het plaatsen van goederen op een bepaalde hoogte zoals in magazijnstellingen gebruikelijk is;
8. Het instellen van (x,y,z)-coördinaten waar producten gehaald of geplaatst kunnen worden.

11.2 Werken met de Transporter

Casus 6: Halen en brengen

Identiek aan casus vijf zijn er twee parallelle productielijnen waar blauwe of rode producten na bewerking op een machine op de productvloer wachten op een vorkheftruck. De vorkheftruck transporteert het product naar een verder gelegen conveyor, waarna de producten het systeem verlaten. De afstand vanaf de machine naar de achterliggende conveyor bedraagt ongeveer 15 meter in directe lijn. De afstand tussen de beide machines bedraagt ongeveer 10 meter. Blauwe en rode producten arriveren bij de machines met een gemiddelde tussenaankomsttijd van 2 minuten. De machines hebben een gemiddelde productietijd van 20 seconden. Aankomst- en productietijden volgen een exponentiele verdeling. Alle buffers hebben een standaard capaciteit van 10, de conveyor een standaardsnelheid van 1 m/s. De layout van dit systeem is te zien in Figuur 11-1.



Figuur 11-1: Lay-out van de productielijn

De vorkheftruck heeft een snelheid van 1 m/s en kan vrij in de ruimte bewegen. We veronderstellen in eerste instantie dat producten met gelijke kans naar een van de beide conveyors gestuurd worden.

Vragen en opdrachten (vooraf)

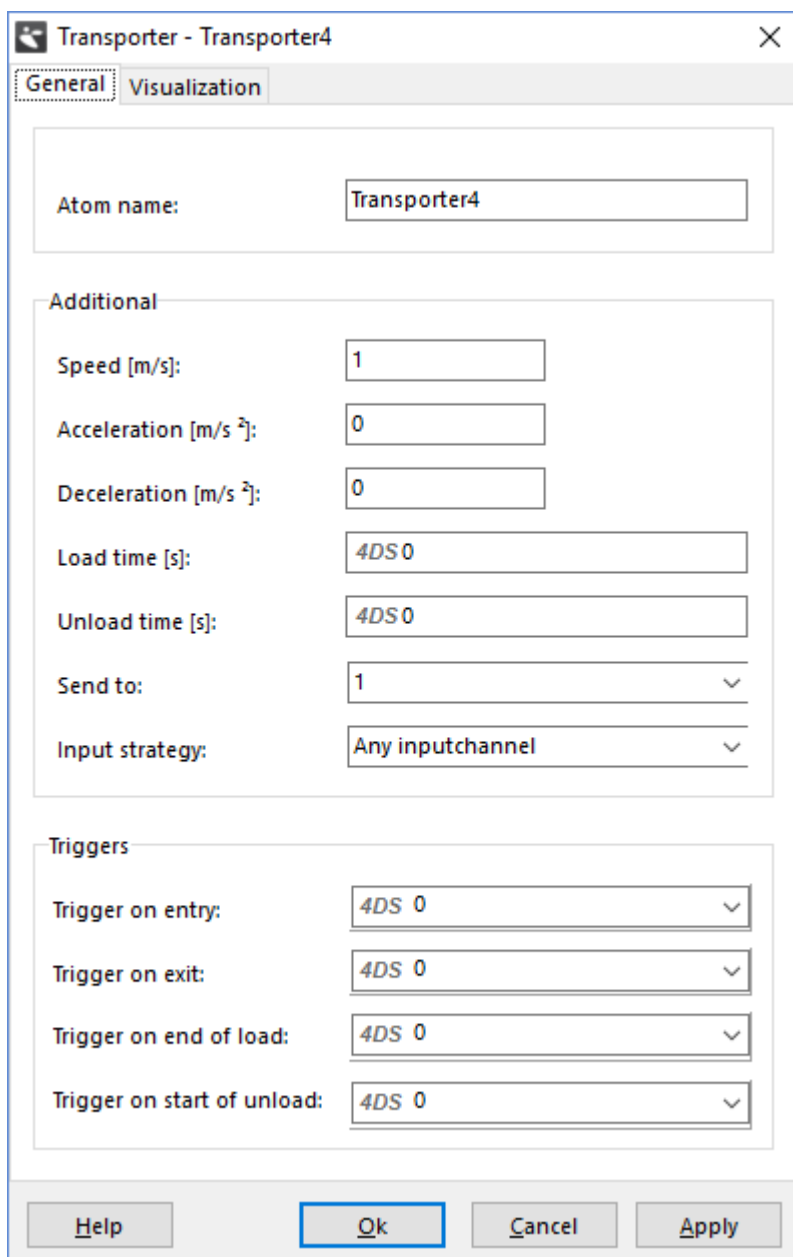
1. Hoe hoog zal de theoretische bezettingsgraad van de vorkheftruck bij benadering zijn?
2. Hoe hoog wordt deze bezettingsgraad wanneer het oppakken en het wegzetten beiden 5 seconden kosten?
3. Beantwoord beide vragen wanneer de producten alleen naar de dichtstbijzijnde conveyor gestuurd worden: blauwe producten naar de bovenliggende conveyor, rode producten naar de onderliggende conveyor.

De beantwoording van deze vragen kost hopelijk niet teveel moeite! Nu gaan we onze antwoorden controleren door het bouwen van dit model met een Transporter.

Sleep de Transporter (niet de Advanced Transporter!) uit de groep Floorbound van de categorie Transport in het model. Verbind invoerkanaal 1 resp. 2 van de Transporter met het uitvoerkanaal van de buffer na de machine van de eerste resp. tweede productielijn. Verbind uitvoerkanaal 1 resp. 2 van de Transporter met het invoerkanaal van de conveyor van de eerste resp. tweede productielijn. Dubbelklikken op de Transporter geeft het invoerscherm, zie Figuur 11-2.

De snelheid staat standaard al op 1 m/s ingesteld. Wijzig het Send to statement zodanig dat random uit de twee conveyors gekozen wordt. Kies als Input Strategy voor de Largest Queue (goed zichtbaar op scherm) of the Longest Waiting.

Input strategy en Send to statements zijn in hoofdstuk 7 uitgebreider besproken!



Figuur 11-2: Invoerscherm van de Transporter

Maak nu een aantal runs met het eigen model of met *transporter1.mod*. Definieer in de Experiment Wizard 5 runs van 8 uur met als uitvoervariabele Status op de Transporter. Kloppen de antwoorden voor de bezettingsgraad met de eigen berekeningen?

Voeg vervolgens 5 seconden voor laden en lossen toe op Load Time en Unload Time. Voer het experiment nogmaals uit. De bezettingsgraad van de vorkheftruck stijgt nu van 55% naar 71%.

Voor de beantwoording van vraag 3 moeten we het model aanpassen. Dit kunnen we handig doen via een label, die we 'destination' noemen. Dit label krijgt waarde 1 voor de blauwe producten en waarde 2 voor de rode producten. Op het Send to statement van de Transporter wordt dit label gebruikt voor de juiste conveyor.

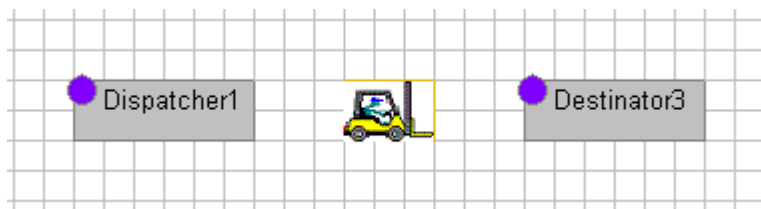
Maak weer een aantal runs met het eigen model of *transporter2.mod*. De bezettingsgraad zal ten opzichte van het vorige model met een paar procent gedaald zijn. Dit geldt zowel voor de variant met lostijd als de variant zonder lostijd. Verklaar waarom!

11.3 De (Advanced) Transporter in het netwerk

In de eerste paragraaf beweerden we dat de Advanced Transporter alles kan wat de Transporter ook kan. Daarom vervangen we, alvorens casus 6 uit te breiden, in model 1 eerst de Transporter door de Advanced Transporter en

vergelijken de resultaten. Deze resultaten moeten uiteraard hetzelfde zijn!

De Advanced Transporter zal vrijwel altijd twee extra atomen nodig hebben: de Dispatcher en de Destinator. Beiden zijn eveneens in Floorbound van de categorie Transport te vinden. De Dispatcher regelt het ophalen van producten (to dispatch = verzenden) door de Advanced Transporter, de Destinator regelt de bestemmingen van de producten. In Figuur 11-3 zijn de drie atomen in de model layout weergegeven. De Advanced Transporter en de Transporter zijn door middel van een verschillend icoon goed te onderscheiden!



Figuur 11-3: De Dispatcher, de Advanced Transporter en de Destinator

Van de Dispatcher worden de ingangskanalen verbonden met de uitgangskanalen van de ophaallocaties, en de uitgangskanalen met één of meerdere Advanced Transporters. De logica van het ophaalproces wordt op de Dispatcher geschreven.

Van de Destinator worden de ingangskanalen verbonden met de uitgangskanalen van één of meerdere Advanced Transporters, en de uitgangskanalen met ingangskanalen van de brenglocaties. Behalve het aanpassen van kanalen, gebeurt er verder niets op de Destinator.

De Advanced Transporter zit dus in het midden en bevat o.a. parameters die betrekking hebben op de verdere routing, snelheid, en laden en lossen.

Verwijder nu uit *transporter1.mod* de Transporter en sleep de drie bovenvermelde atomen in het model. Verbind vervolgens de kanalen op de juiste wijze. Om de Advanced Transporter te vergelijken met de Transporter, moet uit de wachtrij van rode en blauwe producten nog de kleinste wachtrij gekozen worden.

Kies daarom op de Dispatcher voor sort tasks by:

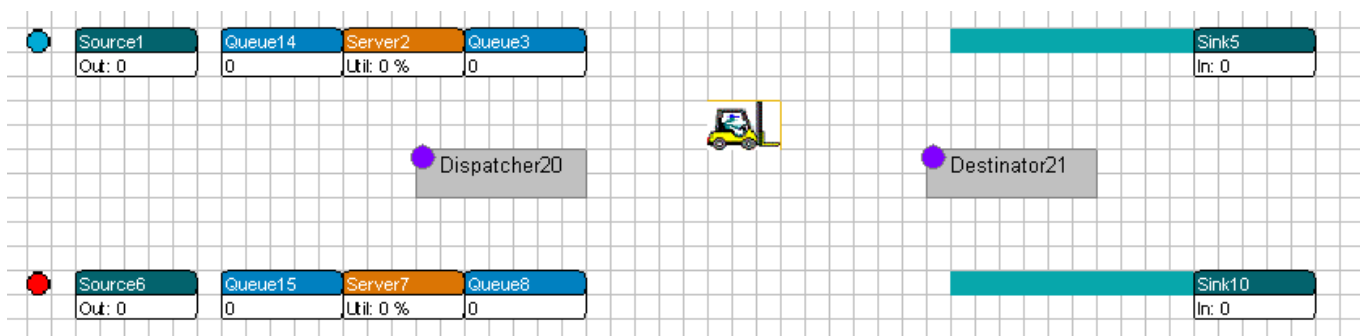
3. Content minimum --> atoms in the container with the smallest content are dispatched first.

Kies op de Advanced Transporter bij General parameters op Send To voor:

5. By percentage: 50% of the products go to channel 1, the remaining percentage goes to channel 2

of kies voor optie 4 waarin random een outputkanaal gekozen wordt.

Dit model is te vinden onder *transporter3.mod* en weergegeven in Figuur 11-4.



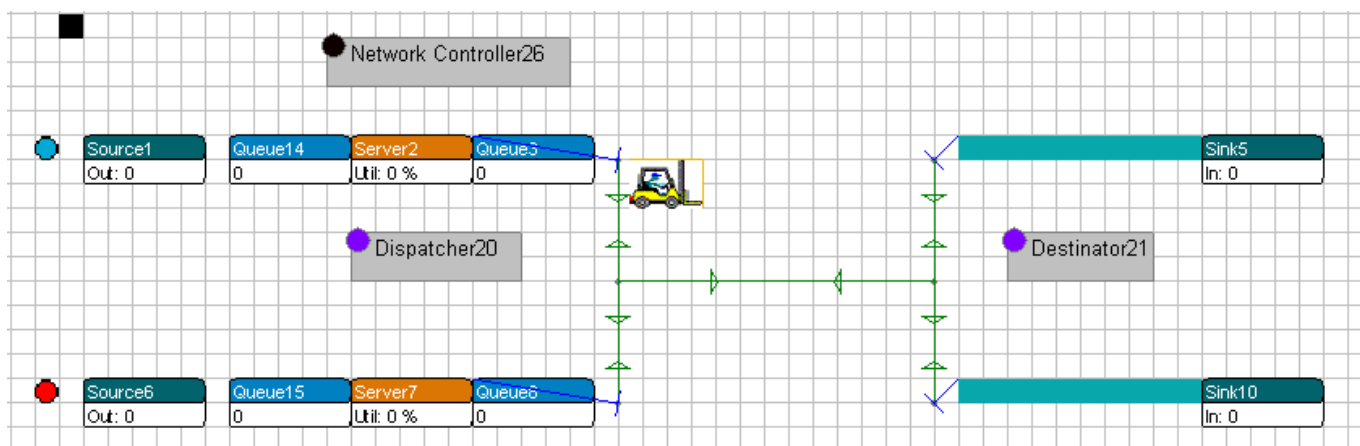
Figuur 11-4: De advanced transporter zonder netwerk

Wanneer we dit model runnen vinden we mbv de Experiment Wizard eveneens een bezettingsgraad van 55% op de Advanced Transporter.

De volgende onderdelen hebben allen betrekking op de Advanced Transporter. Omdat dit uit de context duidelijk is zullen we het voorvoegsel 'advanced' weglaten.

Casus 6 (eerste vervolg)

De productielijnen en de lopende banden blijken zich in twee verschillende hallen te bevinden. Beiden hallen zijn met een pad verbonden en ook de beweging in ieder van de hallen verloopt via vastgelegde paden zoals getoond in Figuur 11-5.



Figuur 11-5: De advanced transporter in een netwerk

Twee punten zijn verbonden met een ophaallocatie aan het einde van een productielijn, 2 punten zijn verbonden met een brenglocatie voor een lopende band, en twee punten vormen de eindpunten van het pad tussen de twee hallen. Alle verbindingen kunnen in beide richtingen doorlopen worden.

Vragen en opdrachten

4. Voeg, uitgaande van *transporter3.mod*, het bovenstaande netwerk met 6 punten toe.
5. Hoe hoog wordt de bezettingsgraad van de transporter nu?
6. Hoe hoog wordt de bezettingsgraad bij het toevoegen van 5 seconden voor het laden en lossen?

Belangrijk!

Gebruik steeds het stappenplan in hoofdstuk 10 voor het bouwen van een netwerk. Verbind eerst de transporter met het netwerk, alvorens het netwerk te optimaliseren!

Kijk eerst op het scherm of het model goed werkt. Rijdt de transporter via de verbindingen? Kloppen de transporttijden? Gebruik bij twijfel *transporter4.mod*!

Indien alles goed gegaan is dan, komt uit vraag 2 een bezettingsgraad van 77%, omdat een heen-en-weer cyclus 46 seconden duurt. Gemiddeld komt eens per minuut een product in een buffer: $46/60$ geeft afgerond 77%. Bij 5 seconden laden en lossen verhogen we de cyclustijd met 10 seconden: de bezettingsgraad stijgt naar $56/60$ ofwel 93%.

11.4 De samenhang van Dispatcher, Advanced Transporter en Destinator

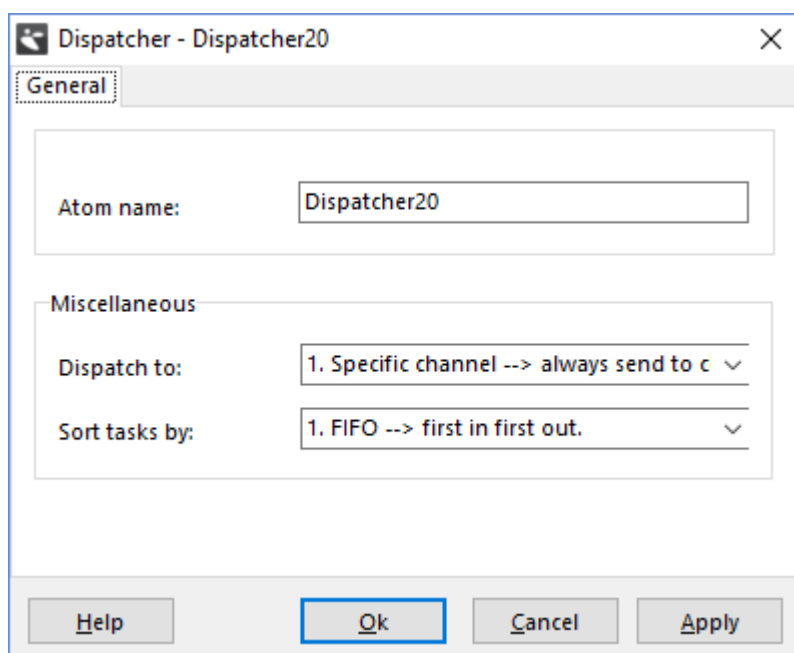
Via casus 6 zijn een aantal eigenschappen van beide Transporters gedemonstreerd. De gewone Transporter is min of meer volledig behandeld: alleen versnellen en vertragen is overgeslagen, maar dit valt bij beide Transporters makkelijk in te bouwen en op het scherm te bekijken. Voor de Advanced Transporter is bovendien het rijden via een netwerk getoond. Maar... er is veel meer mogelijk met de Advanced Transporter!

Eerst beschrijven we de Dispatcher en de Destinator in meer detail. Beiden kunnen alleen gebruikt worden in combinatie met de Advanced Transporter.

De Dispatcher

De Dispatcher genereert transporttaken (dus geen producten!) voor de atomen verbonden met de invoerkanalen van de Dispatcher. Deze taken worden toebedeeld aan de Advanced Transporters die aan de uitvoerkanalen van de Dispatcher hangen: dit kunnen er dus meer dan één zijn!

Er zijn meerdere voorgedefinieerde *Dispatch to* regels met als default: 1. Specific channel --> always send to channel 1. Dit is de transporter die aan het eerste uitgangskanaal van de Dispatcher hangt.

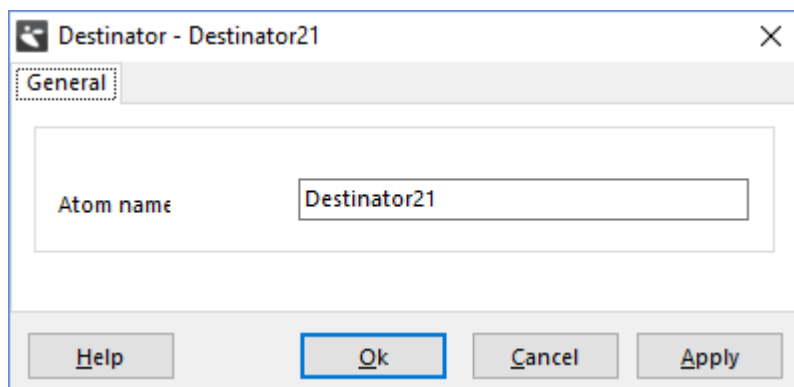


Figuur 11-6: Het invoerscherm van de Dispatcher

Deze transporttaken kunnen gesorteerd worden ('Sort tasks by') via verschillende voorgedefinieerde sorteerregels, waarbij de FIFO-regel de standaard vormt. Bestudeer alle voorgedefinieerde toedelings- en sorteerregels!

De Destinator

De ingangskanalen van de Destinator zijn verbonden met de uitgangskanalen van één of meerdere Advanced Transporters. De uitgangskanalen van de Destinator zijn verbonden met alle mogelijke brenglocaties. Omdat alle logica van het brengen op de Advanced Transporter geplaatst is, valt aan de Destinator verder weinig te zien, zie figuur 11-7.



Figuur 11-7: Het invoerscherm van de Destinator

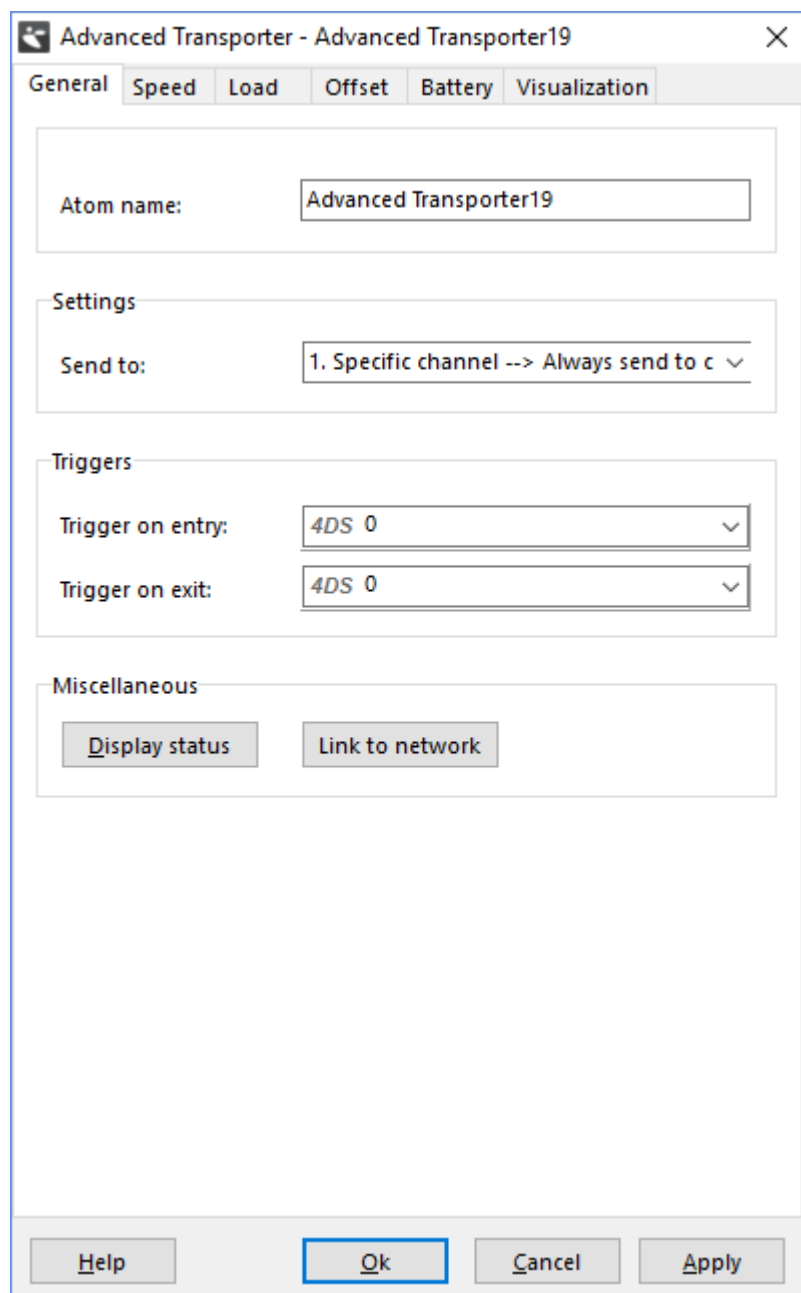
De Advanced Transporter

In Figuur 11-8 is te zien dat er zes tabbladen te onderscheiden zijn in het invoerscherm van de Advanced Transporter:

- General parameters
- Speed parameters
- Load parameters
- Offset parameters
- Battery parameters
- Visualization

We bespreken hier de General en de Load parameters. De Speed parameters spreken voor zich, de Offset parameters komen in een later hoofdstuk tezamen met het Warehouse atoom ter sprake.

Bij de algemene parameters, zie Figuur 11-8, zien we weinig nieuws. Het is wel belangrijk te weten dat het Send to statement uitgevoerd wordt op de uitgangskanalen van de Destinator.



Figuur 11-8: Het invoerscherm van de Advanced Transporter

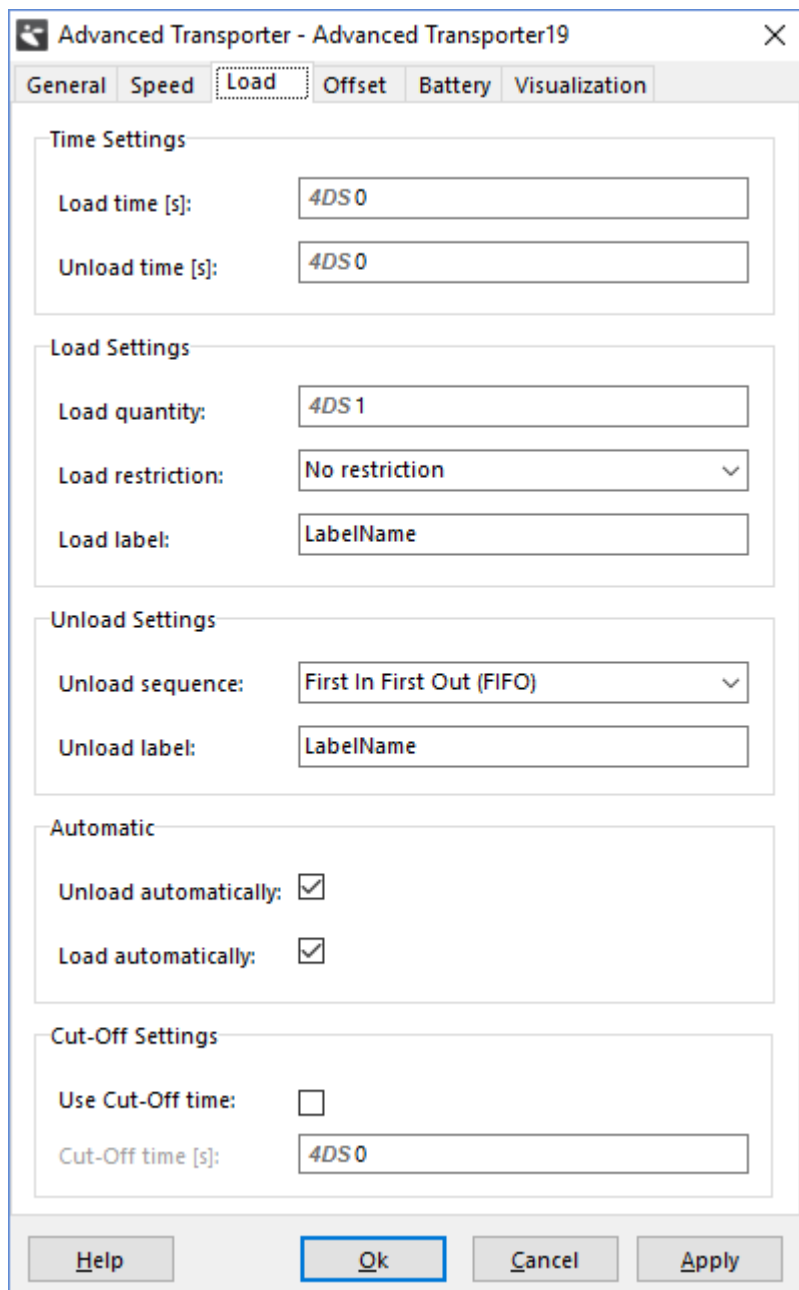
De Load parameters, zie Figuur 11-9 daarentegen bevatten wel een hoop nieuwe opties. Allereerst kan de laadhoeveelheid ingesteld worden. Pas wanneer deze hoeveelheid opgehaald is, zal de transporter tot uitladen over gaan!

Wanneer er geen laadrestrictie ingesteld is, zal het inladen op volgorde van de transporttaken geschieden, dus mogelijkterwijs holt de transporter heen en weer tussen laadlocaties. Er zijn in totaal 6 instellingen mogelijk:

- No restriction (default)
- Same name
- Same label value
- Same label text
- Same container
- Same mother

Een paar van deze opties lichten we toe.

De laadhoeveelheid met bijbehorende restrictie wordt pas actief nadat het eerste product geladen is. Stel bijvoorbeeld dat we een laadhoeveelheid van 5 hebben en dat een transporter een transport uitvoert die hem als eerste naar een product met de naam `radio` in Queue10 leidt.



The screenshot shows the 'Advanced Transporter - Advanced Transporter19' dialog box with the 'Load' tab selected. The settings are as follows:

- Time Settings:** Load time [s]: 4DS0, Unload time [s]: 4DS0
- Load Settings:** Load quantity: 4DS1, Load restriction: No restriction (dropdown), Load label: LabelName
- Unload Settings:** Unload sequence: First In First Out (FIFO) (dropdown), Unload label: LabelName
- Automatic:** Unload automatically: , Load automatically:
- Cut-Off Settings:** Use Cut-Off time: , Cut-Off time [s]: 4DS0

Buttons at the bottom: Help, Ok, Cancel, Apply.

Figuur 11-9: Load parameters van de Advanced Transporter

Dan leidt de *Load restrictie*:

- Same name tot het wachten en oppikken van nog 4 radio's uit alle haallocaties
- Same container tot het wachten en oppikken van nog 4 radio's uit Queue10

Deze radio's moeten in beide situaties nog wel kunnen verschijnen. Pas daarom goed op voor deadlocks!

11.5 Laad- en losstrategieën van de Advanced Transporter

We willen nu in onze casusprobleem de volgende opties inbouwen:

1. Twee producten ophalen en wegbrengen naar de conveyors;
2. Twee *dezelfde* producten ophalen en wegbrengen naar de conveyors;

3. Twee *dezelfde* producten ophalen en wegbrengen naar een specifieke conveyor: de blauwe producten naar de eerste conveyor, de rode naar de tweede conveyor.

Optie 1 is gemakkelijk te realiseren: verander de laadhoeveelheid in 2. Ga na dat de transporter nu zowel rood-blauw als elke andere combinatie vervoert, afhankelijk van de volgorde van aankomst van producten in de buffers.

Dit betekent bij een rode en daarna blauwe aankomst, dat de transporter pas gaat rijden naar rood bij aankomst van het rode product in de buffer. Vervolgens rijdt de transporter, eventueel na wachten, naar het blauwe product. Daarna gaat de reis naar de conveyors. Merk op dat het laatst opgehaalde product te zien is in het icoon van de Advanced Transporter.

Het lossen geeft ook veel varianten te zien: rood en blauw gelost op dezelfde conveyor en op verschillende conveyors. Dit was echter ook de bedoeling van optie 1, maar waarom gebeurt dit zo?

De verklaring zit in het Send to statement: met 50% kans gaat een product naar de eerste conveyor en met net zoveel kans naar de tweede. Kennelijk wordt in de buffer per product geloot welke conveyor 'uitverkoren' is. Optie 1 is te zien in *transporter5a.mod*.

Voor Optie 2 veranderen we vanuit Optie 1 alleen de *Load restriction* in *Same container*, it is te zien in *transporter5b.mod*. Maar... nog steeds gebeurt het afleveren ongestructureerd.

Voor Optie 3 grijpen we terug op het label destination, zoals we eerder in paragraaf 11.2 hebben gedaan: de blauwe producten krijgen een label 'destination' met waarde 1, de rode een label 'destination' met waarde 2. Pas vervolgens vanuit optie 2 het *Send to* statement op de Transporter aan: 7. By label value (direct): Use the value stored in the Label named Destination and send to the corresponding channel. Deze oplossing is te vinden in *transporter5c.mod*.

Bedenk zelf alternatieve vragen en tracht ze vervolgens op te lossen...

11.6 Het gebruik van meerdere transporters

Meerdere (advanced) transporters met verschillende eigenschappen op hetzelfde netwerk: hoe werkt dat? En wat als ze niet mogen passeren? Deze vraagstukken bekijken we weer via onze casus.

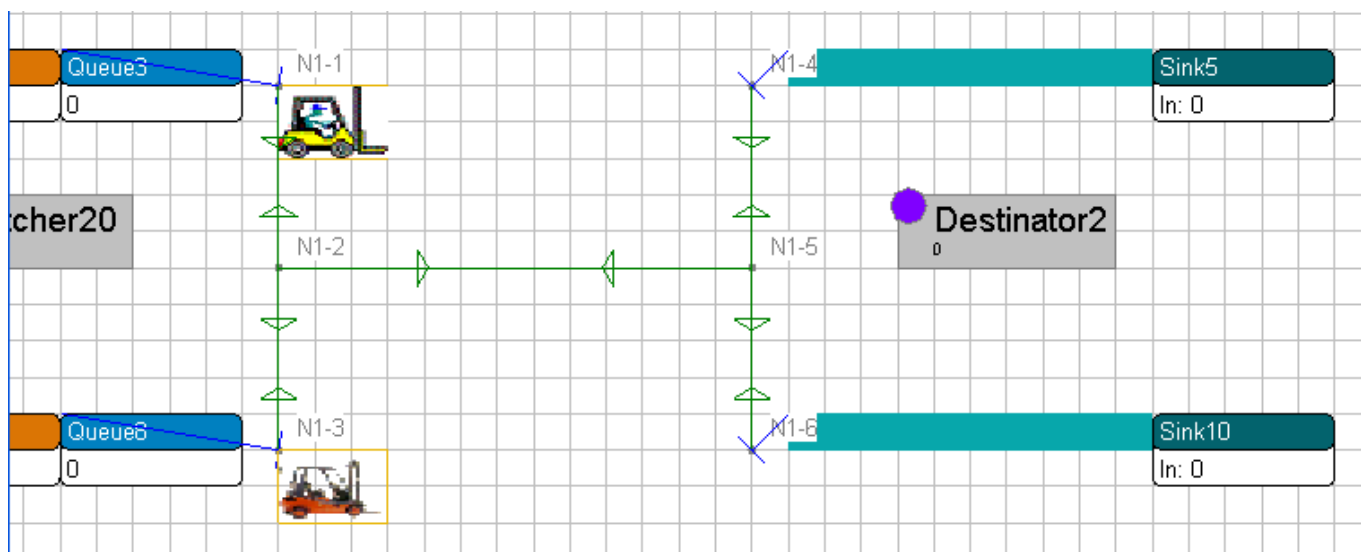
Casus 6 (tweede vervolg)

Vanwege een grote vraag bij de klanten verdubbelt de productiestroom van beide productlijnen. Onze ene transporter kan deze toestroom niet verwerken en een tweede transporter is aangeschaft. Beide transporters vervoeren nog steeds één product per keer naar een willekeurig gekozen afvoerlijn, echter de tweede transporter is aanmerkelijk sneller en heeft een snelheid van 2 m/s. Wanneer beide transporters vrij zijn om een product op te halen, wordt dit product door de snelle transporter opgehaald.

Vragen en opdrachten

7. Voeg, uitgaande van *transporter4.mod*, een tweede transporter met gevraagde eigenschappen toe aan het netwerk, en verander de toevoer van producten. Vergeet niet het netwerk te optimaliseren!
8. Bestudeer het gedrag van beide transporters in het eigen model of open in geval van twijfel *transporter6.mod*. Verloopt het inhalen van Transporter 1 door Transporter 2 probleemloos?
9. Beantwoord -ter controle- nogmaals de vragen over de bezettingsgraden van de beide transporters.

We hebben in *transporter6.mod* een tweede Advanced Transporter toegevoegd aan het netwerk. Ter onderscheid is het icoon van de tweede transporter veranderd in dat van de gewone Transporter. Dit verandert de functionaliteit van deze Advanced Transporter uiteraard niet! In Figuur 11-10 is een deel van de lay-out met beide Transporters te zien.



Figuur 11-10: Twee transporters in één netwerk

Met de Node Manipulator en de optie Show Nodes zijn de namen van de nodes zichtbaar gemaakt. Als thuisbasis voor de gele resp. rode transporter is node N1-1, resp. N1-3 gekozen. Dit is te zien bij het resetten van het model.

Het toewijzen van een product aan de snelle transporter wanneer beide Transporters vrij zijn, gebeurt op de Dispatcher met Dispatch to:

3. A free Transporter (LTF) --> look for a free Transporter. Check the Last Transporter connected to the dispatcher First.

Dit omdat de snelle transporter aan het tweede uitvoerkanaal van de Dispatcher hangt. Bestudeer de andere *Dispatch To*-regels om een indruk van de mogelijkheden te krijgen.

Met behulp van de Experiment Wizard beantwoorden we vraag 9. Met voorrang voor Transporter 2 vonden we voor de Transporters als groep een bezettingsgraad van 48% (Status Idle 0.52), terwijl dit met voorrang voor Transporter 1 stijgt tot 56% (Status Idle 0.44).

Theoretisch is dit bij een gelijke werklastverdeling 57.5%: 2 opdrachten per minuut, waarvan een uitgevoerd door transporter 1 (46 sec) en een uitgevoerd door transporter 2 (23 sec). Gemiddeld is dit 34,5 seconden werk op 60 seconden.

Zoek zelf een verklaring waarom deze 57.5% toch nog hoger ligt, dan de gemeten 56% bij voorrang voor de langzame transporter!

Casus 6 (derde vervolg)

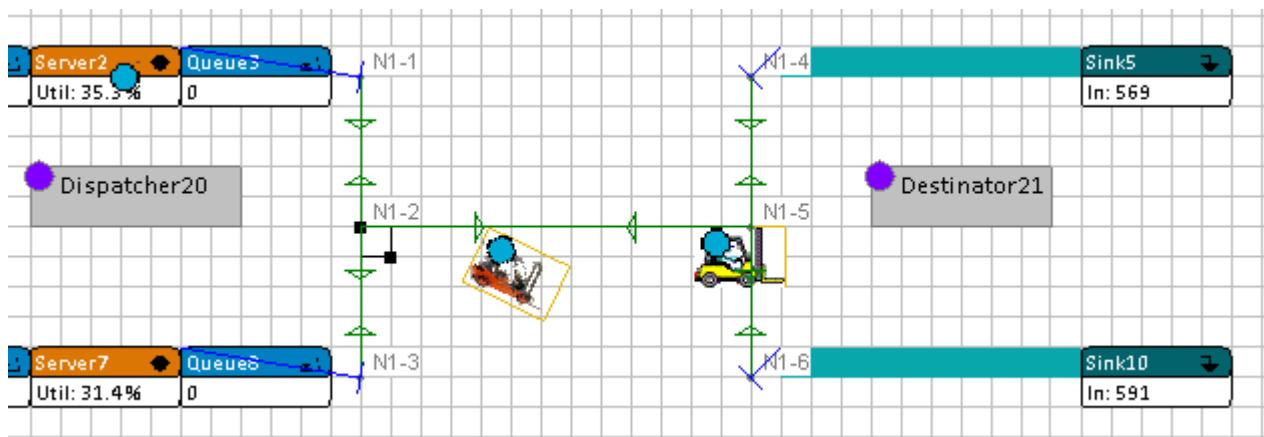
Vanwege de veiligheid mag zich op het middenpad in beide richtingen nog slechts een transporter bevinden. Een tweede transporter dient te wachten tot de desbetreffende rijrichting weer vrij is. Hiermee is inhalen op dit traject dus onmogelijk geworden.

Vragen en opdrachten

10. Realiseer dit in het netwerk met de twee transporters.

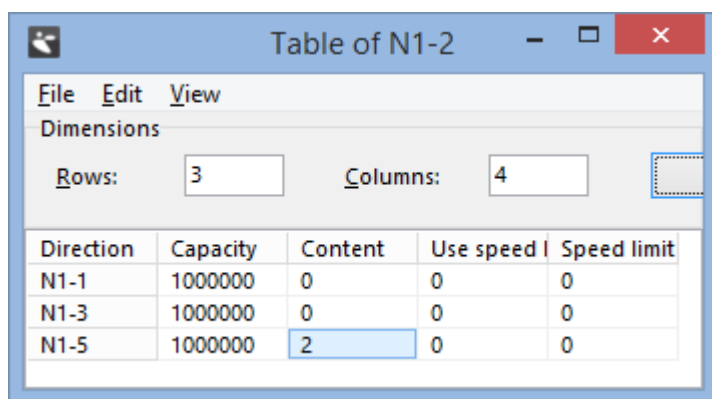
Om vraag 10 te beantwoorden, bekijken we hoe we capaciteitsrestricties op delen van een netwerk kunnen realiseren.

In Figuur 11-11 zien we dat 2 transporters op de verbinding van N1-2 naar N1-5 rijden, een situatie die dus uitgesloten dient te worden.



Figuur 11-11: Twee transporters op de netwerkverbinding N1-2 naar N1-5

Tevens is het punt N1-2 aangeklikt. Dit is te zien door het zwartomkaderde vierkant rechtsonder N1-2. Bij dubbelklikken op dit punt verschijnt de tabel van Figuur 11-12.



Direction	Capacity	Content	Use speed	Speed limit
N1-1	1000000	0	0	0
N1-3	1000000	0	0	0
N1-5	1000000	2	0	0

Figuur 11-12: De network table van N1-2

Hierin staan de drie verbindingen vanuit N1-2 naar N1-1, N1-3 en N1-5 weergegeven. De capaciteit geeft aan hoeveel voertuigen zich gelijktijdig op de aangegeven verbinding mogen bewegen. Default is 1000000, of liever gezegd: oneindig. De content geeft aan hoeveel voertuigen zich op dat moment op de aangegeven verbinding bewegen. De Content van 2 op N1-2 naar N1-5 behoort bij de momentopname in Figuur 11-11.

Laat het model een tijd runnen om te zien hoe de getallen in de rechterkolom van Figuur 11-12 voortdurend wijzigen. Bestudeer dit gedrag ook bij de andere punten van het netwerk.

Vraag 10 is nu gemakkelijk te beantwoorden door de capacity van N1-2 naar N1-5 te wijzigen in 1. Doe hetzelfde ook van N1-5 naar N1-2!

Bestudeer nu het gedrag van de transporters op het traject N1-2 naar N1-5. Wachten de voertuigen netjes op elkaar? Merk op dat verbindingen rood kleuren indien de capaciteitsgrens bereikt is! Dit model is te vinden onder *transporter7.mod*.

Hiermee sluiten we voorlopig de toepassing van Transporters met of zonder netwerk af.

Het is duidelijk dat we nu een aantal complexe problemen in magazijnomgevingen kunnen modelleren.