

Intro to agents and agent-tools

1. Volg de tutorial en omschrijf daarna in één paragraaf wat deze tool anders maakt dan andere programmeertalen, wat zijn de voor- en nadelen? Leg ook uit waarom je programma wel of niet agent-based is.

- Netlogo is een taal die vaak gebruikt voor ABM (Agent based modeling). Er worden agents in de vorm van turtles, patches, links en observer gemaakt. De implementatie talen die gebruikt werden om Netlogo te ontwerpen zijn Scala en Java. Netlogo wordt vaak door studenten, docenten en mensen die weinig/geen programmeer ervaring hebben. Dat komt omdat deze tool voor meerdere doelgroepen ontworpen is. Het is een prima tool die wereldwijd gebruikt wordt bij de modelleren van agents in verschillende situaties. **** vertel over voor en nadelen van deze taal en andere programmingtaal ****
- Voor deze opdracht heb ik een ziekteverspreiding simulatie gemaakt. Het doel van de simulatie is om te laten zien hoe snel een ziekte verspreid kan worden op basis van verschillende elementen zoals: het houden van 1,5 m afstand, de kans dat mensen elkaar kunnen besmetten, de incubatietijd van 5 dagen, hoe lang iemand ziek kan blijven en de sterfcijfers van de zieke patiënten.
Naast de linkje die op canvas staan heb ik drie andere bronnen gebruikt, zoals je hieronder kunt zien:

<http://ccl.northwestern.edu/netlogo/models/SpreadofDisease>

https://www.youtube.com/watch?v=9MdaROY-YEk&ab_channel=sagism

<http://www.netlogoweb.org/launch#http://ccl.northwestern.edu/netlogo/models/models/Sample%20Models/Biology/Virus.nlogo>

2. We definiëren een staat en drie functies waarmee we een stateful agent abstract omschrijven

Initiële staat: Hij begint bij 1 van de interne staten die het model bevat. Bij de ziekteverspreiding simulatie kan de Initiële staat een witte of een oranje agent zijn.

Een agent werkt als volgt: hij observeert de environment state met de *See* functie, update de state door gebruik van de *Next* functie te maken en op basis van de *action* functie, kiest hij een actie die hij op de state gaat uitvoeren.

Waarbij de genoemde functies in de formele notatie hieronder staan:

$$see : S \rightarrow P$$

$$action : I \rightarrow A$$

$$next : I \times P \rightarrow I$$

Aangezien de ziekteverspreiding model ook gebruik van deze functies maakt, kunnen we deze functies voor de simulatie definiëren.

See:

Een agent kan in ons voorbeeld alleen de omgeving van 1 radius zien (een cirkel om zich heen). Wat deze omgeving ontoegankelijk maakt. Op basis van het kleurtje van de agent, de tijd aspect, en andere bepaalde random kansen. kan er een actie uit de volgende verzameling uitgevoerd worden.

$$\text{action}(s) = \left\{ \begin{array}{ll} \text{Be orange} & \text{if } s = \text{in 1 radius a red or orange} \\ \text{Be red} & \text{if } s = \text{orange and it has been longer than 5 days} \\ \text{Be violet} & \text{if } s = \text{red, it has been longer than 10 days and don't} \\ & \text{have a chance to recover} \\ \text{Be gray} & \text{if } s = \text{violet, it has been longer than 10 days and have} \\ & \text{a chance to die} \\ \text{Be green} & \text{otherwise} \end{array} \right\}$$

Aangenomen dat de set of S alle mogelijke staten bevat. Om het nog duidelijker te maken gaan we staten, de verzameling of s en de functie See ook definiëren.

Statements:

x= "the color is red"

y= "the color is orange"

z= "the color is violet"

a= "it has been 5 days"

b= "it has been 10 days"

c= "chance to recover"

d= "chance to die"

$$s = \{s1, s2, \dots s6\} = \{\{x \vee y\}, \{y \wedge a\}, \{x \wedge b \wedge (\neg c)\}, \{x \wedge b \wedge c\}, \{z \wedge b \wedge d\}, \{z \wedge b \wedge (\neg d)\}\}$$

Percepts:

P1 = "Be orange"

P2 = "Be red"

P3 = "Be violet"

P4 = "Be green"

P5 = "Be gray"

Dus de see functie wordt als volgt:

$$\text{see}(s) = \left\{ \begin{array}{ll} P1 & \text{if } s = s1 \\ P2 & \text{if } s = s2 \\ P3 & \text{if } s = s3 \\ P4 & \text{if } s = s4 \text{ or } s = s6 \\ P5 & \text{if } s = s5 \end{array} \right\}$$

**** je moet nog over de functie action vertellen + meer uitleg nodig****

3. Beschrijf je omgeving op basis van de dichotomies die hier op pagina 6 beschreven staan, en licht toe (dus niet alleen termen opsommen):

De omgeving in de ziekteverspreiding model is inaccessible omdat de agent alleen een radius om zich heen kan zien. Dus hij heeft niet alle informatie van de omgeving tot beschikking .

Omdat we moeilijk de resultaten van deze simulatie kunnen vastleggen op basis van de acties die een agent uitvoert. Een persoon die een oranje of een rode persoon om zich heen heeft (dat zijn mensen die de ziekte kunnen verspreiden) kan ook ziek worden. Maar er zijn ook andere regels die de verspreiding van de ziekte controleren. Dus we niet heel goed het eindresultaat/de verspreiding van het model vastleggen. Daardoor kunnen we dit model als stochastisch model classificeren.

In deze simulatie kan een agent beslissen welke actie uitgevoerd moet worden op basis van de huidige situatie. Dat maakt deze simulatie episodisch.

Aangezien de omgeving van de agent blijft veranderen, omdat de agents zelf allemaal bewegen. Kunnen er altijd verschillende resultaten uit de simulatie leiden. Daardoor kunnen we zeggen dat deze simulatie dynamisch is.

Deze simulatie eindigt na een bepaalde stappen. Dus na het zien dat er geen mensen zijn die of ziek of infected met de ziekte zijn, eindigt de simulatie. Het doel is om te laten zien hoe snel een ziekte verspreid kan worden ondanks dat mensen soms afstand onderling houden. In dit geval kunnen we deze simulatie als een discrete simulatie beschouwen.

*** Je moet beter alles uitleggen+ sommige dingen kloppen niet helemaal***

4. Bedenk een voorbeeld waarbij minimaal 3 dichotomies precies tegenovergesteld zijn en beargumenteer waarom dit wel of niet van belang is om je simulatie nuttig te maken.