

Sammenligning af NetLogo og Mesa

15. april 2020

1 Oversigt

Agent-baserede modeller er modeller, som kan simulere interaktioner mellem flere forskellige, individuelle *agenter* i et system, typisk med det formål at danne en forståelse for systemet som en helhed. Simulationerne viser typisk, hvordan systemet ændrer sig over tid, afhængigt af vilkårlige startparametre og agenternes opførsel.

Vi sammenligner to værktøjer, som tilbyder simulationer af agent-baserede modeller. Det ene er **NetLogo**¹, som er et selvstændigt miljø der benytter sit eget domænespecifikke sprog. Det andet er **Mesa**², som er et framework til Python. Vi giver et overblik over hvert, og sammenligner deres fordele og ulemper.

2 NetLogo

NetLogo er et udviklingsmiljø til simulation af agent-baserede modeller. Miljøet tilbyder sit eget sprog (også kaldet **NetLogo**) til at opstille modellerne, samt et visuelt interface, som gør det muligt at simulere sin model og justere parametre hen ad vejen.

Sproget er forholdsvis nemt at læse og forstå. Mange af de mest nødvendige funktionaliteter eksisterer som indbyggede funktioner, så ens egen simple funktioner kan typisk holdes på under 10 linjer. Det kræves at man har forståelse for simple datatyper som integers, bools, strings etc., samt control-flow i form af if-sætninger. Løkker kan normalt undgås ved istedet at bruge `ask`-keywordet.

¹ccl.northwestern.edu/netlogo

²mesa.readthedocs.io/en/master/index.html

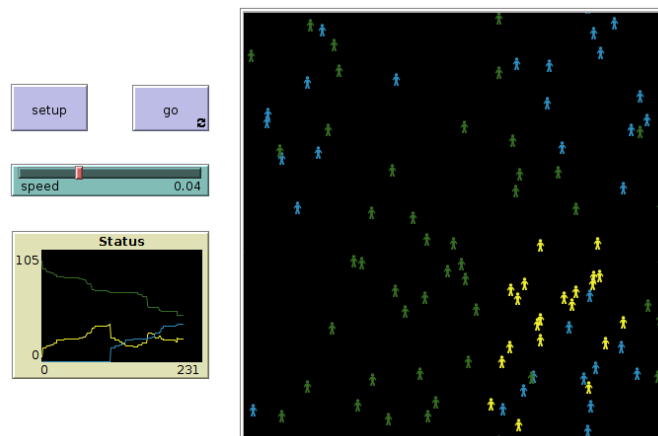
```

ask people [
  set shape "person"
  set is-immune false
  ifelse random-float 100 < 5
    [ infect ]
    [ cure ]
]

```

Figur 1: Kodeeksempel fra `epidemic.nlogo`.

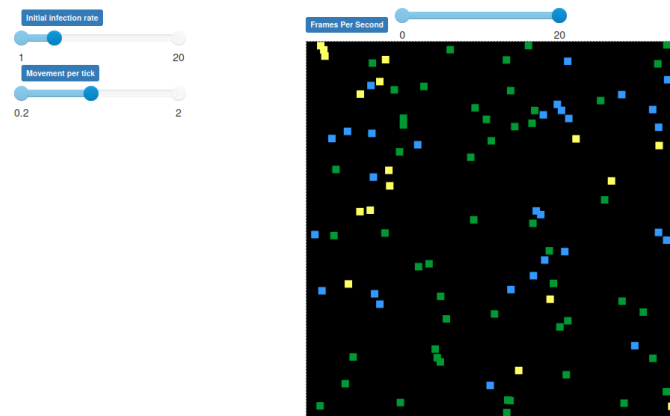
NetLogo-miljøet tilbyder også en visuel interface, hvor det er muligt at se agenternes opførsel i realtid. Her kan man også tilføje bl.a. knapper og "sliders", der tillader en at justere parametre undervejs, eller små graf-vinduer, der viser udvalgte datapunkter.



Figur 2: Visuel interface fra `epidemic.nlogo`.

3 Mesa

Mesa er et framework til Python. Det implementerer et sæt af basis-klasser, herunder `Model` og `Agent`, som man så skal tilføje funktionalitet til. Dette kræver en god forståelse for klasser i Python, afhængigt af, hvor meget man vil lade eleverne implementere selv. For at visualisere modellen i realtid kræves desuden, at man implementerer et `VisualizationElement`, eller benytter en af de eksisterende, som f.eks. en gitter-visualisering med `CanvasGrid`. Hvis man ønsker en mere specifik model, skal man, ud over at implementere et `VisualizationElement`, også kode en del af det visuelle i JavaScript, da Mesa bruger en web-server til visuel præsentation.



Figur 3: Visuel interface fra `EpidemicModelRun.py`.

4 Konklusion