

Onderzoek Sim Frameworks

Inhoud

Inleiding.....	3
Frameworks.....	4
Mesa.....	4
Agentpy.....	4
BPTK-Py	4
GAMA	4
Voor & Nadelen	6
Mesa.....	6
AgentPy.....	6
BPTK-Py	7
GAMA	7

Inleiding.

Dit document is opgesteld om te onderzoeken welke agens based modelling frameworks er bestaan voor python en te onderzoeken welke het best geschikt zou zijn voor onze casus.

Frameworks

Er zijn verschillende frameworks waar uitgekozen kan worden om onze simulatie mee op te zetten de eerste vraag is natuurlijk welke frameworks zijn er voor ons beschikbaar met onze criteria zoals open-source.

Mesa

Mesa is een open-source framework voor agent-based modeling in Python. Het biedt een flexibele en modulaire structuur waarmee gebruikers agenten kunnen definiëren, omgevingen kunnen modelleren en simulaties kunnen uitvoeren. Mesa bevat ingebouwde visualisatieopties, data-analysetools en ondersteuning voor parallele verwerking, waardoor het geschikt is voor zowel onderzoek als educatieve doeleinden. Dankzij de actieve community en uitbreidbaarheid is het een populaire keuze voor het ontwikkelen van agent gebaseerde simulaties.

Agentpy

AgentPy is een open-source framework voor agent-based modeling in Python, ontworpen voor onderzoek en onderwijs. Het biedt een eenvoudige en flexibele API waarmee gebruikers dynamische agenten en complexe systemen kunnen simuleren. AgentPy integreert goed met populaire wetenschappelijke Python-bibliotheken zoals NumPy, Pandas en Matplotlib, wat krachtige analysemogelijkheden biedt. Daarnaast ondersteunt het parallele verwerking en optimalisatie van modellen, waardoor het geschikt is voor grootschalige simulaties en experimentele modellering. [\[Link: AgentPy\]](#)

BPTK-Py

BPTK-Py (Business Prototyping Toolkit for Python) is een open-source framework voor systeemodynamica en agent-based modeling. Het is ontworpen voor bedrijfsmodellering en beleidsanalyse en biedt een gestructureerde aanpak om simulaties te bouwen met behulp van Python. BPTK-Py ondersteunt hybride simulaties waarin zowel agent-based als systeemdynamische modellen gecombineerd kunnen worden. Het integreert goed met Jupyter Notebooks en heeft ingebouwde functies voor scenario-analyse en optimalisatie, waardoor het een krachtig hulpmiddel is voor strategische planning en complexe besluitvorming.

GAMA

GAMA (GIS & Agent-based Modeling Architecture) is een krachtig open-source framework voor agent-based modeling, met een focus op complexe systemen en

ruimtelijke simulaties. Het biedt een visuele interface en een eigen scripttaal (GAML) waarmee gebruikers geavanceerde simulaties kunnen bouwen zonder diepgaande programmeerkennis. GAMA ondersteunt grootschalige, ruimtelijk-expliciete simulaties en integreert goed met GIS-data, waardoor het bijzonder geschikt is voor stedelijke modellering, milieusimulaties en besluitvorming. Dankzij zijn flexibiliteit en uitgebreide visualisatiemogelijkheden wordt GAMA veel gebruikt in academisch en toegepast onderzoek.

Voor & Nadelen

Mesa

Voordelen:

- Geschreven in Python, dus eenvoudig te gebruiken en integreren met andere Python-bibliotheken.
- Modulaire opzet met ingebouwde tools voor visualisatie en data-analyse.
- Actieve open-source community en goede documentatie.
- Ondersteuning voor parallele verwerking via multiprocessing.

Nadelen:

- Minder geschikt voor zeer grote simulaties vanwege performance beperkingen.
- Beperkte ondersteuning voor ruimtelijke en GIS-gebaseerde simulaties.
- Visualisaties kunnen minder geavanceerd zijn dan in sommige andere frameworks.

AgentPy

Voordelen:

- Lichtgewicht en flexibel framework dat goed samenwerkt met wetenschappelijke Python-bibliotheken zoals NumPy, Pandas en Matplotlib.
- Ondersteunt parallele verwerking en geavanceerde data-analyse.
- Eenvoudige API, waardoor het makkelijk te leren en toe te passen is.
- Geschikt voor snelle experimenten en academisch onderzoek.

Nadelen:

- Minder ingebouwde visualisatiemogelijkheden dan Mesa of GAMA.
- Nog een relatief jong project met een kleinere community en minder voorbeelden.
- Minder geoptimaliseerd voor grote schaal of high-performance simulaties.
- Al een lange tijd niet meer ondersteunt.

BPTK-Py

Voordelen:

- Combineert agent-based modeling met systeemodynamica, wat hybride simulaties mogelijk maakt.
- Sterke focus op bedrijfsmodellering, strategische planning en beleidsanalyse.
- Goede integratie met Jupyter Notebooks, wat het geschikt maakt voor interactief onderzoek.
- Ondersteuning voor scenario-analyse en optimalisatie.

Nadelen:

- Minder geschikt voor grootschalige of ruimtelijke simulaties.
- Heeft een iets steilere leercurve door de focus op systeemodynamica.
- Kleinere community en minder algemene documentatie dan andere frameworks

GAMA

Voordelen:

- Zeer krachtig voor ruimtelijke simulaties en GIS-integratie.
- Biedt een visuele interface en een eigen scripttaal (GAML), waardoor het toegankelijk is voor gebruikers zonder uitgebreide programmeerkennis.
- Geschikt voor grootschalige simulaties en gedetailleerde agentgedrag-modellering.
- Uitgebreide visualisatiemogelijkheden en scenario-analyse.

Nadelen:

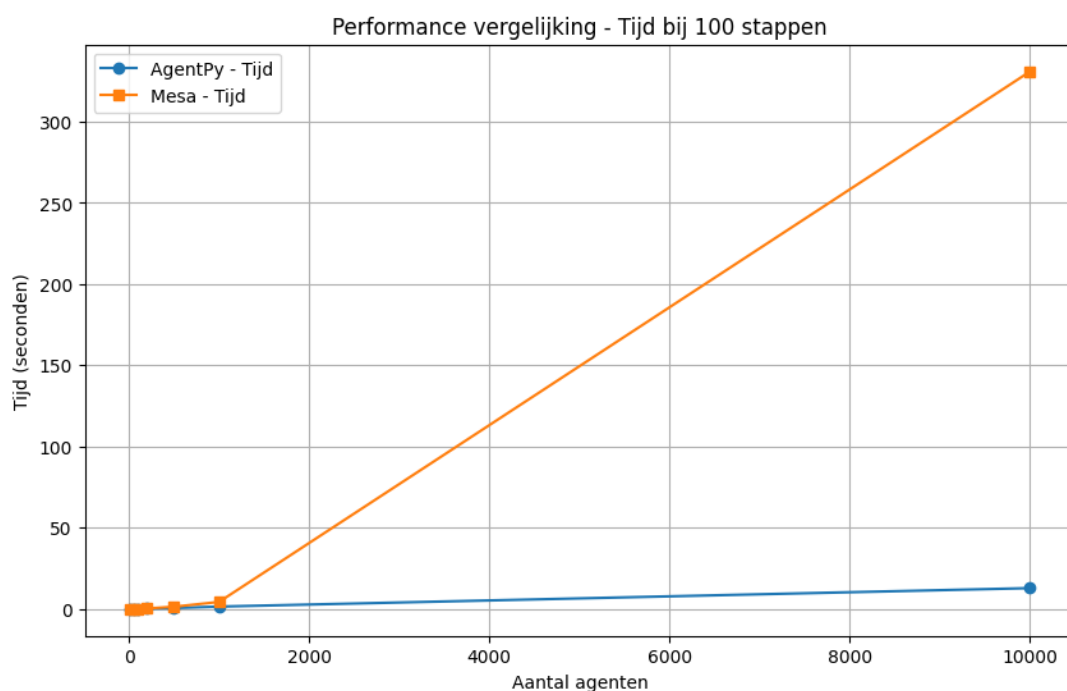
- GAML kan een leercurve hebben voor gebruikers die gewend zijn aan Python.
- Vereist meer rekenkracht, waardoor het minder efficiënt kan zijn voor eenvoudige modellen.
- Minder eenvoudig te integreren met de Python-ecosystemen zoals NumPy en Pandas.

Tests Notebook.

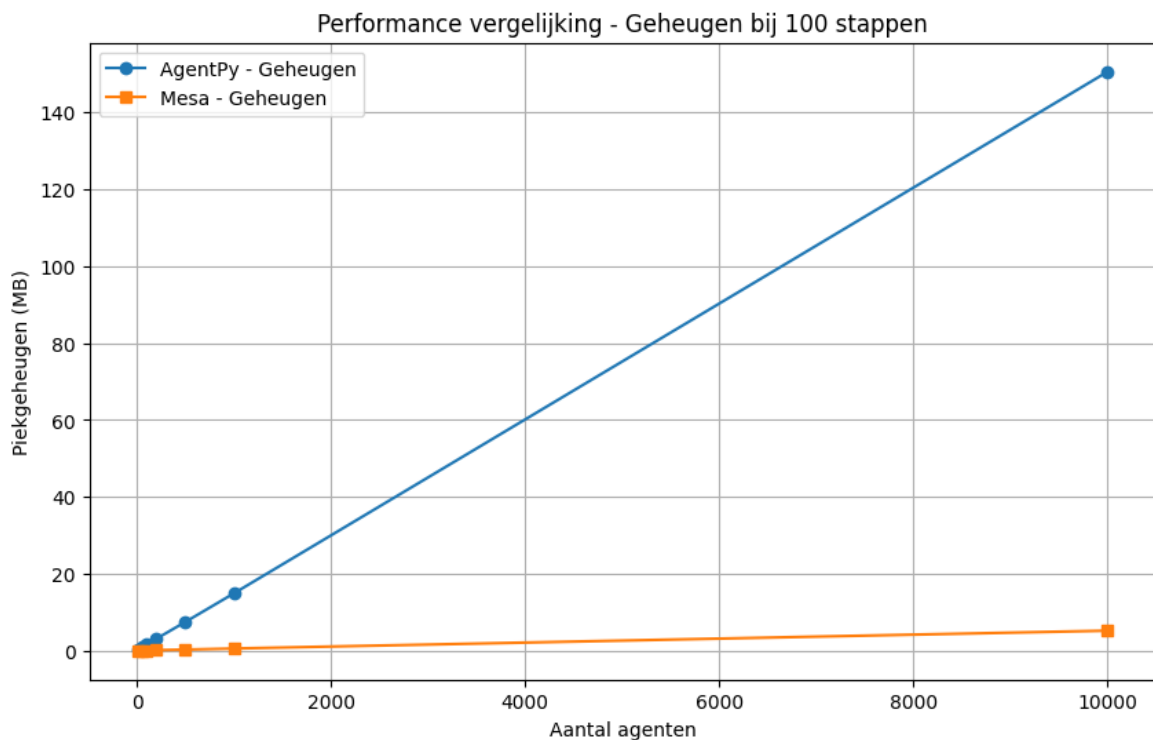
Om de verschillende frameworks objectief te kunnen vergelijken, is een Jupyter Notebook ontwikkeld waarin een reeks tests zijn uitgevoerd. Al snel werd duidelijk dat **BPTK-Py** en **GAMA** minder geschikt zijn voor de doeleinden van dit project. GAMA vereist het gebruik van een eigen programmeertaal, wat de integratie bemoeilijkt. BPTK-Py bleek lastig te installeren in de werkomgeving en is bovendien meer gericht op het modelleren van businesscampagnes dan op dynamische agent-based simulaties.

Hierdoor bleven twee frameworks over: **Mesa** en **AgentPy**. In het Jupyter Notebook is met beide frameworks een identieke simulatie opgezet: het bekende *Money Model*, waarin agenten willekeurig geld uitwisselen met elkaar. Dit model werd gebruikt om de prestaties van beide frameworks te vergelijken.

De simulaties zijn uitgevoerd met 100 tijdstappen en verschillende aantallen agenten (10, 50, 100, 200, 500, 1000 en 10.000). Daarbij is specifiek gekeken naar simulatiesnelheid en geheugengebruik, met als doel een schaalbare oplossing te vinden die in de toekomst gehost kan worden. De resultaten zijn weergegeven in twee overzichtelijke grafieken.



Figuur 1: Mesa VS Agent-py tijd per hoeveelheid agents



Figuur 2: Mesa vs AgentPy Geheugen gebruik

Conclusie

Na het vergelijken van de verschillende frameworks is uiteindelijk gekozen voor **Mesa** als basis voor dit project. Zowel Mesa als AgentPy zijn lichtgewicht, open-source en goed geïntegreerd met het Python-ecosysteem, wat ze geschikt maakt voor het opzetten van agent-based simulaties en het verzamelen van data.

Hoewel AgentPy in de praktijk **sneller** bleek bij het uitvoeren van simulaties met grote aantallen agents, heeft de uiteindelijke keuze voor Mesa te maken met de **toekomstbestendigheid** van het project. AgentPy wordt al geruime tijd niet meer actief onderhouden, wat risico's met zich meebrengt op het gebied van compatibiliteit en ondersteuning bij verdere ontwikkeling.

Mesa biedt daarentegen een actieve community, regelmatige updates en een modulair ontwerp met ingebouwde visualisatie-ondersteuning. Ondanks dat Mesa iets trager is in uitvoering, biedt het meer zekerheid op de lange termijn en betere ondersteuning voor toekomstige uitbreidingen van de simulatie.

Op basis van deze overwegingen is **Mesa** gekozen als het meest geschikte framework voor dit project.

Bronnen

1. Mesa: <https://mesa.readthedocs.io/latest/>
2. AgentPy: <https://agentpy.readthedocs.io/en/latest/>
3. BPTK-py: <https://bptk.transentis.com/quickstart/quickstart.html>
4. GAMA: <https://gama-platform.org/>