|  |
| --- |
| **BACHELORTHESIS** |
| Yasser Ibourk |
| Simulation urbaner Mobilität in Casablanca: Ein Multi-Agenten-System basierend auf dem MARS-Framework. |
| |  | | --- | |  | |
| **FAKULTÄT TECHNIK UND INFORMATIK** Department Informatik |
| Faculty of Computer Science and Engineering Department Computer Science |

Yasser Ibourk

Simulation urbaner Mobilität in Casablanca:

Ein Multi-Agenten-System basierend auf dem MARS-Framework.

Bachelorarbeit eingereicht im Rahmen der Bachelorprüfung

im Studiengang *Bachelor of Science Angewandte Informatik*

am Department Informatik

der Fakultät Technik und Informatik

der Hochschule für Angewandte Wissenschaften Hamburg

Betreuender Prüfer: Prof. Dr. Thomas Clemen

Zweitgutachter: XXX

Eingereicht am: XX. XXX 2025

**Yasser Ibourk**

**Thema der Arbeit**

Simulation urbaner Mobilität in Casablanca: Ein Multi-Agenten-System basierend auf dem MARS-Framework.

**Stichworte**

Mulitagentensystem (MAS), Agentenbasierte Modellierung (ABM), MARS-Framework, Casablanca,

**Kurzzusammenfassung**

Diese Bachelorarbeit beschreibt die Entwicklung eines Multiagentensystems zur Simulation des Verkehrs in Casablanca unter Verwendung des MARS (Multi-Agent Research and Simulation)-Frameworks. Das Ziel des Systems ist es, Entscheidungsträgern in der Stadtplanung und Mobilitätsverwaltung die Möglichkeit zu bieten, verschiedene Verkehrsszenarien zu simulieren und deren Auswirkungen zu analysieren. Es dient als Werkzeug zur Verbesserung der strategischen Planung und Entscheidungsfindung im Hinblick auf die Mobilität in der Metropolregion Casablanca.

Der Fokus liegt auf der Modellierung der multimodalen Mobilität in Casablanca, einschließlich Straßenbahn-, Bus- und Fußgängerverkehr sowie der Berücksichtigung informeller Verkehrsmittel wie Taxis und Minibusse. Das System modelliert die urbane Verkehrsumgebung, einschließlich Verkehrsflüsse, Straßennetzwerke und Interaktionen zwischen Agenten, die verschiedene Verkehrsteilnehmer repräsentieren. Die Agenten können kontextabhängige Entscheidungen treffen, indem sie ihre Modalwahl und Routenplanung dynamisch anpassen.

Die Arbeit diskutiert die Systemarchitektur, das Design und die Implementierung, einschließlich der Nutzung von Geodaten und Algorithmen zur multimodalen Wegfindung. Besondere Schwerpunkte liegen auf der Integration lokaler Verkehrsbesonderheiten, der Visualisierung von Simulationsergebnissen und der Analyse der Auswirkungen von Änderungen der Verkehrsinfrastruktur.

Die durchgeführten Experimente werden daraufhin ausgewertet, wie effektiv das System dabei ist, realitätsnahe Verkehrsströme zu simulieren und fundierte Erkenntnisse für die Verkehrsplanung in Casablanca zu liefern. Nach der Analyse der Ergebnisse wird festgestellt, dass das System bereits erste Erkenntnisse über die Mobilität in der Stadt liefern kann, jedoch weitere Verfeinerungen in der Modellierung informeller Verkehrsmittel und der Interaktionen zwischen Agenten erforderlich sind.

Die Arbeit zeigt auf, dass ein Multiagentensystem, das speziell an die Anforderungen Casablancas angepasst ist, ein vielversprechendes Werkzeug zur Unterstützung von Planungs- und Entscheidungsprozessen im urbanen Verkehrswesen darstellt. Gleichzeitig werden Verbesserungspotenziale identifiziert, um die Genauigkeit und Nutzbarkeit des Systems weiter zu steigern.

**Yasser Ibourk**

**Title of Thesis**

Simulation of Urban Mobility in Casablanca: A Multi-Agent System Based on the MARS Framework

**Keywords**

Multi-Agent System (MAS), Agent-based Modelling (ABM), MARS framework, Casablanca,

**Abstract**

This bachelor's thesis describes the development of a multi-agent system for simulating urban mobility in Casablanca using the MARS (Multi-Agent Research and Simulation) framework. The objective of the system is to provide decision-makers in urban planning and mobility management with the ability to simulate various traffic scenarios and analyze their impacts. It serves as a tool to enhance strategic planning and decision-making concerning mobility in the Casablanca metropolitan region.

The focus lies on modeling multimodal mobility in Casablanca, including tram, bus, and pedestrian traffic, while also considering informal transport modes such as taxis and minibuses. The system models the urban traffic environment, incorporating traffic flows, road networks, and interactions between agents representing different types of traffic participants. The agents can make context-sensitive decisions, dynamically adapting their mode of transport and route planning.

The thesis discusses the system architecture, design, and implementation, including the use of geospatial data. Key aspects include the integration of local traffic characteristics, visualization of simulation results, and analysis of the impacts of changes to the transportation infrastructure.

The conducted experiments are evaluated to determine how effectively the system simulates realistic traffic flows and provides actionable insights for urban transportation planning in Casablanca. The analysis of the results reveals that the system can already provide initial insights into urban mobility, though further refinements are needed, particularly in the modeling of informal transport modes and agent interactions.

This work demonstrates that a multi-agent system specifically tailored to Casablanca's requirements offers a promising tool to support planning and decision-making processes in urban transportation. At the same time, areas for improvement are identified to enhance the system's accuracy and usability further.

Inhaltsverzeichnis

[Inhaltsverzeichnis v](#_Toc187179096)

[Abbildungsverzeichnis viii](#_Toc187179097)

[Tabellenverzeichnis ix](#_Toc187179098)

[Listings x](#_Toc187179099)

[Abkürzungsverzeichnis xi](#_Toc187179100)

[Glossar xii](#_Toc187179101)

[1 Einleitung 1](#_Toc187179102)

[1.1 Hintergrund und Motivation der Arbeit 1](#_Toc187179103)

[1.2 Zielsetzung und Abgrenzung der Arbeit 2](#_Toc187179104)

[1.3 Aufbau der Arbeit 4](#_Toc187179105)

[2 Grundlagen 7](#_Toc187179106)

[2.1 Multiagentensysteme 7](#_Toc187179107)

[2.2 Das MARS-Framework 8](#_Toc187179108)

[2.2.1 Layers 9](#_Toc187179109)

[2.2.2 Agenten 9](#_Toc187179110)

[2.2.3 Entitäten 10](#_Toc187179111)

[2.2.4 Environments 10](#_Toc187179112)

[2.3 Casablanca als Stadt und Forschungsobjekt 10](#_Toc187179113)

[2.3.1 Casablanca als Multiagentensystem 11](#_Toc187179114)

[2.4 Vergleichbare Multiagenten-Systeme im Bereich Verkehrssimulation 13](#_Toc187179115)

[3 Systemanforderungsanalyse 15](#_Toc187179116)

[3.1 Funktionale Anforderungen 15](#_Toc187179117)

[3.2 Nicht-funktionale Anforderungen 17](#_Toc187179118)

[4 Entwurf der Simulations- und Modellumgebung 19](#_Toc187179119)

[5 Realisierung des Systems 20](#_Toc187179120)

[5.1 Wahl der Programmiersprache 20](#_Toc187179121)

[5.2 Umsetzung des Analyseprogramms 20](#_Toc187179122)

[5.3 Konfiguration und Ausführung des Systems 22](#_Toc187179123)

[5.4 Implementierung 26](#_Toc187179124)

[5.4.1 Abweichungen von dem Systementwurf 26](#_Toc187179125)

[5.4.2 Umsetzungsdetails 28](#_Toc187179126)

[5.4.3 Verhaltensweisen der Agenten 29](#_Toc187179127)

[PlanningGoalkeeperPlayerMind 30](#_Toc187179128)

[PlanningDefendingPlayerMind 32](#_Toc187179129)

[PlanningAttackingPlayerMind 33](#_Toc187179130)

[5.5 Tests 38](#_Toc187179131)

[Aufbau 39](#_Toc187179132)

[Zu testende Aspekte 39](#_Toc187179133)

[6 Experimente und Ergebnisse 41](#_Toc187179134)

[6.1 Experiment 1: Einzelverhalten 42](#_Toc187179135)

[Aufbau 42](#_Toc187179136)

[Ablauf 42](#_Toc187179137)

[Ergebnisse 43](#_Toc187179138)

[6.2 Experiment 2: Mannschaftsverhalten 44](#_Toc187179139)

[Aufbau 44](#_Toc187179140)

[Ablauf 45](#_Toc187179141)

[Ergebnisse 46](#_Toc187179142)

[6.3 Experiment 3: Nutzung des Systems 48](#_Toc187179143)

[Hintergrund 49](#_Toc187179144)

[Aufbau 50](#_Toc187179145)

[Ablauf 52](#_Toc187179146)

[Ergebnisse 52](#_Toc187179147)

[7 Diskussion der Ergebnisse und Bewertung des Systems 55](#_Toc187179148)

[8 Schlussfolgerungen und Ausblick 60](#_Toc187179149)

[Literaturverzeichnis 62](#_Toc187179150)

[A Anwendungsfälle 66](#_Toc187179151)

[B Dateien des Systems 75](#_Toc187179152)

[B.1 Program.cs-Datei 75](#_Toc187179153)

[B.2 config.json-Datei 76](#_Toc187179154)

[B.3 plays.json-Datei 78](#_Toc187179155)

[C GOAP-Aktionen der Angreifer 81](#_Toc187179156)

Abbildungsverzeichnis

[Abbildung 1: XXXXXX. 9](#_Toc175832305)

Tabellenverzeichnis

[Tabelle 1: XXXXX. 42](#_Toc175832316)

Listings

[Listing 1: XXXX xx2](#_Toc175832340)

Abkürzungsverzeichnis

|  |  |
| --- | --- |
| ABM | Agentenbasiertes Modell |
| MARS | Multi Agent Research and Simulation |
| MAS  GIS | Multiagenten Simulation  Geoinformationssystem |

Glossar

|  |  |
| --- | --- |
| MARS-Framework | Ein C# Framework für Multiagentensysteme, das zur Entwicklung und Simulation von Szenarien verwendet wird. |
| Multiagentensystem (MAS)  GIS (Geoinformationssystem)  Kepler.gl | Ein System, das aus mehreren Agenten besteht, die autonom handeln und miteinander interagieren, um Aufgaben zu lösen oder bestimmte Ziele zu erreichen.  Ein System zur Erfassung, Speicherung, Analyse und Darstellung räumlicher oder geografischer Daten. In diesem Projekt wird QGIS für die Verarbeitung von Geo-Daten verwendet.  Ein Open-Source-Tool zur Visualisierung geografischer Daten. Es wird verwendet, um die Simulationsergebnisse darzustellen. |

1. Einleitung

Diese Arbeit beschreibt ein im Rahmen der Bachelorthesis entwickeltes Multiagentensystem, das den urbanen Verkehr und die Mobilität in Casablanca simuliert. Das System modelliert verschiedene Verkehrsteilnehmer (Agenten), die unterschiedliche Modalitäten wie Straßenbahn, Bus, Fußgängerverkehr und private Fahrzeuge repräsentieren. Ein zentraler Aspekt des Systems ist die Möglichkeit, "Was-wäre-wenn"-Szenarien für die Stadtplanung zu simulieren, indem spezifische Änderungen an der Verkehrsinfrastruktur oder Mobilitätsstrategien konfiguriert werden, die anschließend von den Agenten ausgeführt werden. Nachfolgend beschreibt dieses Kapitel die Motivation für die Arbeit, definiert die Zielsetzung sowie die gesetzten Rahmenbedingungen. Abschließend wird der Aufbau der weiteren Arbeit erläutert.

* 1. Hintergrund und Motivation der Arbeit

Die Mobilität und der Verkehr stellen zentrale Herausforderungen für Städte weltweit dar. Casablanca, als größte Stadt Marokkos und wirtschaftliches Zentrum des Landes, kämpft mit erheblichen Verkehrsproblemen wie Staus, ineffizienten öffentlichen Verkehrsmitteln und der Notwendigkeit einer nachhaltigen Stadtentwicklung. Um diese Herausforderungen zu bewältigen, ist es entscheidend, Lösungen zu entwickeln, die verschiedene Verkehrsmittel miteinander verbinden und die Effizienz des Verkehrsnetzes steigern.

Ein essenzieller Aspekt bei der Stadtplanung und Verkehrsoptimierung ist die Möglichkeit, "Was-wäre-wenn"-Szenarien zu simulieren. So können Auswirkungen von Maßnahmen wie der Einführung neuer Straßenbahnlinien, der Optimierung von Busverbindungen oder der Förderung von Fuß- und Radverkehr im Vorfeld analysiert werden. Aktuelle Methoden, wie herkömmliche Verkehrsplanungssoftware oder isolierte Simulationen, bieten jedoch nur begrenzte Möglichkeiten, das komplexe Zusammenspiel von Verkehrsmitteln, Infrastruktur und menschlichem Verhalten realitätsnah abzubilden.

Multiagentensysteme bieten hier einen innovativen Ansatz, um städtische Mobilität umfassend zu modellieren. Sie ermöglichen die Simulation von individuellen Verkehrsteilnehmern (Agenten), die eigenständig Entscheidungen treffen und miteinander interagieren. Solche Systeme können eine wichtige Rolle spielen, um die Auswirkungen geplanter Maßnahmen auf den Verkehr und die Lebensqualität in Städten zu analysieren und fundierte Entscheidungen für die Stadtentwicklung zu treffen.

Im Rahmen dieser Arbeit wird ein Multiagentensystem namens *SmartOpenCasablanca* entwickelt, das auf dem erfolgreichen Modell *SmartOpenHamburg* basiert. Ziel ist es, die Mobilität in Casablanca durch die Simulation verschiedener Szenarien zu untersuchen und nachhaltige Lösungsansätze für die Verkehrsprobleme der Stadt zu identifizieren. Das System ermöglicht die Modellierung von Verkehrsteilnehmern, darunter Straßenbahnen, Busse, Fahrräder und Fußgänger, sowie die Analyse ihrer Interaktionen in einem urbanen Kontext.

Ein besonderer Fokus liegt dabei auf der Flexibilität des Systems, das es Entscheidungsträgern erlaubt, spezifische Mobilitätsmaßnahmen zu testen und deren Auswirkungen auf den Verkehrsfluss zu bewerten. Das System dient somit als Werkzeug zur Unterstützung der Stadtplanung und als Basis für datengestützte Entscheidungen zur Verbesserung der Lebensqualität in Casablanca.

* 1. Zielsetzung und Abgrenzung der Arbeit

Das Ziel dieser Arbeit ist die Entwicklung eines Multiagentensystems, das als Analysewerkzeug für die Mobilität und Verkehrsplanung in Casablanca dient. Wie in Kapitel 1.1 beschrieben, soll das System die Modellierung städtischer Mobilität ermöglichen und die Auswirkungen von Verkehrsmaßnahmen in einer komplexen urbanen Umgebung analysieren. Dabei bietet es die Möglichkeit, verschiedene Verkehrsszenarien zu simulieren und Entscheidungsträgern eine datengestützte Grundlage für die Verkehrsplanung und -optimierung bereitzustellen. Im Rahmen des Projekts **SmartOpenCasablanca** werden drei Hauptziele verfolgt: Erstens der Aufbau einer urbanen Simulationsumgebung, die das Verkehrsnetz Casablancas einschließlich Straßen, öffentlicher Verkehrsmittel, Fußgängerzonen und weiterer relevanter Elemente abbildet. Zweitens die Implementierung einer agentenbasierten Simulation, bei der Agenten realistische Verkehrsteilnehmer wie Fußgänger, Radfahrer, Autofahrer und öffentliche Verkehrsmittel repräsentieren und basierend auf ihrer Wahrnehmung sowie ihren Zielen Entscheidungen treffen. Drittens die Integration einer Visualisierungskomponente, die die Beobachtung und Analyse von Verkehrsszenarien ermöglicht und die Ergebnisse anschaulich und verständlich für Fachleute und Stadtplaner darstellt. Der Fokus der Arbeit liegt auf der Entwicklung eines funktionsfähigen Simulationssystems mit diesen Kernzielen, wobei aufgrund der Projektkomplexität und des begrenzten Zeitrahmens bewusst einige Aspekte ausgeklammert oder vereinfacht werden. So konzentriert sich die Simulation auf ausgewählte Verkehrsszenarien wie die Einführung neuer Straßenbahnlinien oder die Förderung des Fuß- und Radverkehrs, während eine vollständige Abdeckung aller Mobilitätsszenarien Casablancas nicht angestrebt wird. Die Verkehrsinfrastruktur wird auf wesentliche Elemente beschränkt, wobei Details wie Verkehrsampeln oder komplexe Kreuzungsregelungen nur vereinfacht dargestellt werden. Auch das Verhalten der Agenten wird auf grundlegende Entscheidungsprozesse reduziert, ohne soziale, kulturelle oder wirtschaftliche Faktoren umfassend zu berücksichtigen, wobei diese Aspekte in zukünftigen Erweiterungen einbezogen werden könnten. Das System wird zudem modular und erweiterbar gestaltet, um in Zukunft die Simulation zusätzlicher Verkehrsmittel oder die Einbindung externer Datenquellen zu ermöglichen. Diese Arbeit legt somit den Grundstein für ein erweitertes Simulationssystem, das nicht nur Entscheidungsträger in Casablanca unterstützt, sondern auch als Forschungswerkzeug für Multiagentensysteme und urbane Mobilität dient.

Aufgrund der Komplexität des Projekts und des begrenzten Zeitrahmens werden folgende Aspekte bewusst ausgeklammert oder vereinfacht:

1. **Beschränkung auf ausgewählte Verkehrsszenarien:** Die Simulation wird sich auf spezifische Anwendungsfälle wie die Einführung neuer Straßenbahnlinien oder die Förderung des Fuß- und Radverkehrs konzentrieren. Eine vollständige Abdeckung aller Mobilitätsszenarien Casablancas ist nicht möglich.
2. **Eingeschränkte Modellierung der Verkehrsinfrastruktur:** Die Darstellung der städtischen Infrastruktur wird sich auf die wichtigsten Verkehrsmittel und -wege beschränken. Details wie Verkehrsampeln oder komplexe Kreuzungsregelungen werden nur vereinfacht umgesetzt.
3. **Agentenverhalten:** Während die Agenten grundlegende Entscheidungsprozesse durchführen, wird eine detaillierte Modellierung von sozialen, kulturellen oder wirtschaftlichen Faktoren nicht Teil dieser Arbeit sein. Diese Aspekte könnten jedoch in zukünftigen Erweiterungen integriert werden.
4. **Erweiterbarkeit:** Das System wird modular und erweiterbar aufgebaut, um zukünftige Ergänzungen wie die Simulation weiterer Verkehrsmittel oder die Einbindung externer Datenquellen zu ermöglichen.

Die Arbeit legt somit den Grundstein für ein erweitertes System, das nicht nur Entscheidungsträgern in Casablanca dient, sondern auch als Forschungswerkzeug für Multiagentensysteme und urbane Mobilität eingesetzt werden kann.

* 1. Aufbau der Arbeit

Diese Arbeit ist in acht Kapitel unterteilt, die schrittweise aufeinander aufbauen und das Projekt **SmartOpenCasablanca** umfassend beschreiben. **Kapitel 2** legt die theoretischen und methodischen Grundlagen für das Verständnis des Projekts dar. Es bietet zunächst eine Einführung in die Konzepte von Multiagentensystemen, gefolgt von einer detaillierten Darstellung des **MARS-Frameworks** als technische Basis der Simulation. Darüber hinaus werden die städtischen Mobilitätsanforderungen der Stadt Casablanca erläutert, um den realweltlichen Kontext der Arbeit zu verdeutlichen. Ergänzt wird das Kapitel durch eine Übersicht über verwandte Arbeiten im Bereich der Mobilitätssimulation, wodurch der aktuelle Forschungsstand und die Positionierung der vorliegenden Arbeit im wissenschaftlichen Diskurs verdeutlicht werden.

In **Kapitel 3** werden die funktionalen und nicht-funktionalen Anforderungen an das System detailliert definiert. Hierzu zählen die Anforderungen an die Modellierung der komplexen Verkehrsinfrastruktur der Stadt Casablanca, das Verhalten der simulierten Agenten sowie die Visualisierungs- und Analysefähigkeiten des Systems. Besonderer Fokus liegt dabei auf der möglichst realistischen Abbildung der urbanen Verkehrsdynamiken, um eine fundierte Grundlage für spätere Experimente zu schaffen.

Das darauf aufbauende **Kapitel 4** beschreibt den **Systementwurf** im Detail. Es behandelt die grundlegende Architektur der Simulation, die Struktur und Integration der einzelnen Systemkomponenten sowie die Modellierung der urbanen Umgebung. Zudem wird die Konzeption der Agenten erläutert, insbesondere ihre Entscheidungsfindung, ihre Interaktionsmechanismen und die Schnittstellen zur Umgebung. Dieses Kapitel bildet den Übergang von den zuvor definierten Anforderungen zur konkreten technischen Umsetzung.

Die technische Umsetzung der entworfenen Systemkomponenten steht im Fokus von **Kapitel 5**. Hier wird die Implementierung des Systems detailliert beschrieben, einschließlich der Programmierung der Verkehrsinfrastruktur, der Agentenlogik und der Visualisierungskomponente. Es wird dargelegt, wie die in Kapitel 4 entwickelten Konzepte praktisch realisiert wurden, und es werden die verwendeten Technologien sowie spezifische Implementierungsentscheidungen erläutert.

In **Kapitel 6** werden die durchgeführten **Experimente** vorgestellt, die das Verhalten des Systems in verschiedenen Verkehrsszenarien analysieren. Dazu gehört die Konfiguration der Experimente, die Auswahl geeigneter Testfälle sowie die Analyse der erzielten Ergebnisse. Ziel ist es, das Verhalten der Agenten unter unterschiedlichen Bedingungen zu untersuchen und die Leistungsfähigkeit des Systems in Bezug auf die definierten Anforderungen zu evaluieren.

Die Ergebnisse der Experimente werden anschließend in **Kapitel 7** umfassend diskutiert. Hierbei wird die Aussagekraft der Ergebnisse analysiert und ihr Beitrag zur Zielerreichung der Arbeit bewertet. Besonderes Augenmerk liegt auf der Untersuchung der Eignung des Systems als **Entscheidungsunterstützungstool** für Verkehrsplaner, wobei auch mögliche Herausforderungen und Limitationen kritisch beleuchtet werden.

Den Abschluss der Arbeit bildet **Kapitel 8**, in dem die wichtigsten Erkenntnisse zusammengefasst werden. Darüber hinaus bietet dieses Kapitel einen fundierten **Ausblick** auf potenzielle Erweiterungen und Verbesserungen des Systems. Dazu gehören beispielsweise die Integration weiterer Verkehrsmittel, die Einbindung von Echtzeitdaten zur Steigerung der Simulationsgenauigkeit oder die Verbesserung der Agentenlogik durch fortschrittliche Algorithmen.

Die vorliegende Arbeit folgt somit einer klaren und logisch aufgebauten Struktur, die den Leser schrittweise durch die verschiedenen Entwicklungsphasen des Projekts führt. Beginnend mit der theoretischen Fundierung über die Entwicklung und technische Umsetzung des Systems bis hin zur Evaluierung und kritischen Betrachtung der Ergebnisse wird ein umfassender Überblick über das Projekt und dessen Beitrag zur Modellierung und Simulation urbaner Verkehrsdynamiken vermittelt.

1. Grundlagen

Dieses Kapitel befasst sich mit den theoretischen Grundlagen, die für die Entwicklung dieser Arbeit relevant sind. Es präsentiert die Ergebnisse einer umfassenden Literaturrecherche, die durchgeführt wurde, um eine fundierte Basis für die Konzeption und Realisierung des Systems sicherzustellen.

* 1. Multiagentensysteme

**Multiagentensysteme** (**MAS**), auch als agentenbasierte Modelle (ABM) bekannt, sind softwaregestützte Repräsentationen realer Systeme und Szenarien. In diesen Simulationen agiert eine Gruppe autonomer, adaptiver Entscheidungsträger – die sogenannten Agenten – innerhalb einer Simulationsumgebung, die das zu modellierende System abbildet. Diese Agenten verfolgen spezifische Ziele, die sie durch Interaktionen mit ihrer Umgebung und miteinander erreichen (Saeed Rahimi et al., 2022). Ein zentrales Merkmal von MAS ist die Autonomie der Agenten, welche ihnen ermöglicht, unabhängig Entscheidungen zu treffen und eigenständig zu handeln, anstatt deterministischen Mustern zu folgen. Ihre Entscheidungen basieren auf ihrem aktuellen internen Zustand, ihrer Wahrnehmung der Umgebung sowie den verfügbaren internen Ressourcen („MARS Group, Dokumentation“).

Durch ihr Verhalten nehmen die Agenten Einfluss auf ihre Umgebung und verändern deren Zustand, was wiederum das Verhalten anderer Agenten beeinflussen kann. Trotz ihrer Eigenständigkeit verfolgen die Agenten häufig übergeordnete Ziele, die eine Zusammenarbeit und Koordination zwischen ihnen erfordern. Hierfür kommunizieren sie untereinander und koordinieren ihre Handlungen (Abdollah Amirkhani & Amir Hossein Barshooi, 2021). Die Komplexität der Agenten reicht von einfachen Regelwerken – die leicht nachvollziehbar anhand von „Wenn-Dann“-Bedingungen beschrieben werden können – bis hin zu lernfähigen Agenten, die ihr Verhalten durch maschinelle Lernalgorithmen trainieren und anpassen („MARS Group, Dokumentation“).

Der Ursprung der Multiagentensysteme liegt in der Biologie, wo sie noch heute zur Modellierung und Analyse von Bewegungs- und Migrationsmustern von Tieren eingesetzt werden (Abdollah Amirkhani & Amir Hossein Barshooi, 2021). Darüber hinaus finden MAS Anwendung in vielfältigen weiteren Bereichen, beispielsweise bei der Modellierung von Bewegungs- und Verhaltensmustern der Einwohner von Städten. Hier dienen sie als digitale Zwillinge oder digitale Schatten, um regionale Verhaltensweisen zu simulieren und zu analysieren (Saeed Rahimi et al., 2022; Tolk et al., 2022).

* 1. Das MARS-Framework

Das **Multi-Agent Research and Simulation (MARS) Framework** ist eine spezialisierte Softwarebibliothek zur Entwicklung und Ausführung von Multiagentensystemen. Es bildet die technologische Grundlage des **SmartOpenCasablanca-Projekts**, das darauf abzielt, multimodale Verkehrsflüsse und urbane Dynamiken der Stadt Casablanca zu simulieren. Ursprünglich an der HAW Hamburg in der Programmiersprache C# entwickelt, wird MARS unter der **General Public License (GNU)** bereitgestellt. Dadurch steht es sowohl Einsteigern als auch erfahrenen Entwicklern von Multiagentensystemen als flexible und leistungsfähige Entwicklungsumgebung zur Verfügung. Die Bereitstellung als **NuGet-Paket** erleichtert den Einstieg und ermöglicht die Integration komplexer spatio-temporaler Daten für anspruchsvolle Simulationsprojekte.

Das MARS-Framework basiert auf zwei zentralen Konzepten: den **Layers** (Schichten) und den **Agents** (Agenten). Diese Struktur erlaubt es, reale Szenarien in konzeptionelle Modelle zu übertragen und als Simulation zu implementieren.

In Anlehnung an die Funktionsweise typischer Multiagentensysteme unterteilt MARS den Simulationsprozess in fest definierte Zeitschritte (**Ticks**) von konstanter Dauer **Δt**. Innerhalb jedes Simulationsschritts aktiviert das Framework die relevanten Systemkomponenten, indem es die **Tick-Methode** aufruft. Diese Methode wird sowohl von Agenten als auch von spezifischen Layer-Typen implementiert. Diese Architektur ermöglicht eine präzise Steuerung und Wiederholbarkeit der Simulationen und gewährleistet damit hohe Modellierungsgenauigkeit („MARS Group, Dokumentation“).

### Layers

Im **MARS-Framework** nehmen die **Layers** eine zentrale Rolle ein, da sie die Umgebung des simulierten Szenarios modellieren. Die Welt wird dabei durch eine Menge von spatio-temporalen Layers repräsentiert. Jede dieser Layers bildet spezifische Aspekte des Modells ab und enthält beispielsweise Agenten oder relevante Daten.

Die Layers sind für die Erstellung der Agenten und anderer Entitäten verantwortlich. Sie fungieren als Schnittstelle, über die Agenten Informationen über ihre Umwelt erhalten können. Darüber hinaus ermöglichen sie es den Agenten, mit ihrer Umgebung zu interagieren. Diese bidirektionale Beziehung zwischen Layers und Agenten ist ein wesentlicher Bestandteil der Simulationsfunktionalität und gewährleistet eine präzise Abbildung der dynamischen Wechselwirkungen im System („MARS Group, Dokumentation“).

### Agenten

Die **Agenten** bilden den zweiten fundamentalen Bestandteil einer Simulation im **MARS-Framework**. Sie repräsentieren die aktiven Komponenten eines Multiagentensystems (MAS), die ihre Umgebung individuell und eigenständig wahrnehmen. Auf Basis dieser Wahrnehmungen treffen sie Entscheidungen über ihr Verhalten und darüber, wie sie mit anderen Agenten, Entitäten und der Umwelt interagieren.

Jeder Agent ist ein autonomes, eigenständig laufendes Programm, das seine Verhaltenslogik in der bereits beschriebenen **Tick-Methode** definiert. Das Verhalten der Agenten wird typischerweise durch ihre Wahrnehmung des Systems, ihre individuellen sowie gemeinschaftlichen Ziele und andere Aspekte ihres internen Zustands bestimmt. Diese flexible Architektur ermöglicht es, komplexe und dynamische Interaktionen innerhalb des Systems zu modellieren und zu simulieren („MARS Group, Dokumentation“).

### Entitäten

Entitäten modellieren nicht-aktive Komponenten der Simulation. Das bedeutet, dass sie nicht eigenständig agieren und daher auch über keine Tick-Methode verfügen. Sie stellen Objekte oder Gegenstände dar, mit denen die Agenten interagieren können. Durch diese Interaktionen sind Änderungen des Zustands der Entität möglich, beispielsweise wenn sie durch einen Agenten bewegt werden („MARS Group, Dokumentation“).

### Environments

Die **Environments** sind wesentliche Datenstrukturen im **MARS-Framework**, die dazu dienen, das Positionieren, die Bewegung sowie die Erkundung der Umgebung zu ermöglichen. Sie sind Teil der **Layers** und bieten unterschiedliche Methoden, um Bewegungen innerhalb der Simulation zu realisieren.

Ein Beispiel ist das **GridEnvironment**, das die Umwelt in ein Raster unterteilt und eine strukturierte Bewegung zwischen den Rasterpunkten erlaubt. Eine weitere Option ist das **PhysicsEnvironment**, das realitätsnahe physikalische Interaktionen wie Gravitation oder Elastizität simuliert und dabei Kollisionen sowie dynamische Bewegungen im Raum berücksichtigt.

Zusätzlich erleichtern Environments die Wahrnehmung der Umgebung, indem sie relevante Informationen bereitstellen. Dies ermöglicht es den Agenten, den Zustand ihrer Umwelt zu erkennen und ihr Verhalten entsprechend anzupassen („MARS Group, Dokumentation“).

* 1. Casablanca als Stadt und Forschungsobjekt

**Casablanca** ist die größte Stadt Marokkos und ein bedeutendes wirtschaftliches, kulturelles und logistisches Zentrum. Mit über 3,5 Millionen Einwohnern in der Kernstadt und einer Metropolregion von mehr als 7 Millionen Menschen steht Casablanca vor einer Vielzahl von urbanen Herausforderungen, insbesondere im Bereich des Verkehrsmanagements. Die Stadt verfügt über ein wachsendes öffentliches Verkehrssystem, das Straßenbahnen, Busse und Taxis umfasst, die durch ein dichtes Netz von Straßen und Wegen ergänzt werden. Trotz der laufenden Infrastrukturverbesserungen ist der Verkehr in Casablanca oft durch Staus, ineffiziente Verbindungen und eine hohe Abhängigkeit von privaten Autos gekennzeichnet.

Die Untersuchung von Casablanca als Forschungsobjekt bietet eine einzigartige Möglichkeit, urbane Dynamiken in einer Stadt des globalen Südens zu analysieren, in der rasches Bevölkerungswachstum und Urbanisierung auf komplexe Mobilitätsanforderungen treffen. Besondere Herausforderungen ergeben sich durch die informellen Siedlungen und das Fehlen eines vollständig integrierten Verkehrssystems, wodurch die Notwendigkeit innovativer und angepasster Lösungen für die Verkehrssimulation steigt.

Das Ziel des Projekts *SmartOpenCasablanca* ist es, ein Multiagentensystem zu entwickeln, das diese komplexen Interaktionen und Strukturen abbildet. Dabei sollen multimodale Verkehrsmittel wie Straßenbahnen, Busse, Taxis und Fußgänger in die Simulation integriert werden, um eine realitätsnahe Nachbildung des städtischen Verkehrs zu ermöglichen. Casablanca eignet sich hervorragend als Forschungsobjekt, da die Ergebnisse der Simulation nicht nur die lokalen Gegebenheiten verbessern, sondern auch als Blaupause für ähnliche Städte mit vergleichbaren Herausforderungen dienen können.

Ein besonderes Merkmal von Casablanca ist die 2012 eingeführte Straßenbahn, die mittlerweile zwei Linien umfasst und eine zentrale Rolle im öffentlichen Verkehr spielt. Die Straßenbahn symbolisiert den Fortschritt hin zu nachhaltiger Mobilität und bildet eine der wichtigsten Modalitäten für das SmartOpenCasablanca-Projekt. Ziel ist es, die Effizienz der Straßenbahn mit anderen Verkehrsmitteln zu vergleichen und Maßnahmen zu simulieren, die den öffentlichen Verkehr für die Bewohner der Stadt optimieren könnten.

Zusammenfassend ist Casablanca ein ideales Forschungsobjekt, um innovative Simulationsmodelle zu entwickeln und neue Erkenntnisse über städtische Mobilität und Verkehrsmanagement zu gewinnen.

* + 1. Casablanca als Multiagentensystem

Das urbane Verkehrssystem von Casablanca bietet einen faszinierenden Anwendungsfall für die Modellierung als Multiagentensystem (MAS), da es durch seine Komplexität, Dynamik und die Interaktionen der verschiedenen Elemente geprägt ist. In einem solchen System werden einzelne Entitäten wie Fußgänger, Autos, öffentliche Verkehrsmittel und andere Infrastrukturelemente als autonome Agenten dargestellt. Diese Agenten agieren basierend auf definierten Zielen, wie der Minimierung von Reisezeiten, der Einhaltung von Verkehrsregeln oder der Optimierung von Ressourcen, während sie gleichzeitig auf Umwelteinflüsse und die Handlungen anderer Agenten reagieren.

Ähnlich wie die Dynamik im Mannschaftssport ist der Verkehr in Casablanca ein hoch interaktives und anpassungsfähiges System. Agenten, die individuelle Verkehrsmittel wie Straßenbahnfahrgäste, Radfahrer oder Autofahrer repräsentieren, passen ihre Aktionen ständig an, um Staus zu vermeiden, Routen zu optimieren oder Umstiege zwischen Modalitäten wie Gehen, Bussen und Straßenbahnen zu koordinieren. Daher ist die Integration von Multimodalität essenziell, um das reale Verhalten der Akteure präzise abzubilden.

Die räumlichen und zeitlichen Gegebenheiten der urbanen Umgebung Casablancas – darunter das dichte Stadtzentrum, gemischte Straßennetze und öffentliche Verkehrssysteme – machen die Stadt zu einem idealen Anwendungsfall für ein MAS. Die Agenten werden so modelliert, dass sie Entscheidungen auf Basis von Umweltdaten wie Verkehrsaufkommen, Nähe zum Ziel oder Fahrplänen treffen. Darüber hinaus ermöglichen die Interaktionen zwischen diesen Agenten emergentes Verhalten, wie die Entstehung von Stauzonen oder Moduswechsel, was wertvolle Einblicke in mögliche Verbesserungsbereiche für die Stadtplanung liefert.

Das System integriert spezifische Funktionen, um den multimodalen Verkehr zu simulieren. Beispielsweise:

* **Fußgänger-Agenten** wählen die kürzesten fußgängerfreundlichen Routen und meiden stark befahrene Gebiete.
* **Agenten des öffentlichen Nahverkehrs** navigieren durch Fahrpläne, priorisieren den Zugang zu Straßenbahnhaltestellen und koordinieren Umstiege, um effizient ihre Ziele zu erreichen.
* **Auto-Agenten** werden mit dynamischen Parkmöglichkeiten, Verkehrsstaus und Routenoptimierungen modelliert.
* **Fahrrad-Agenten**, sowohl private als auch Mietfahrräder, berücksichtigen Faktoren wie die Verfügbarkeit an Stationen oder die Nähe zu fahrradfreundlichen Wegen.

Dieser MAS-Ansatz für Casablanca ermöglicht eine "Was-wäre-wenn"-Analyse, bei der beispielsweise die Einführung neuer Straßenbahnlinien, Änderungen an Straßenlayouts oder Anpassungen der Busfahrpläne getestet werden können, um deren Auswirkungen auf Verkehrsfluss, Umweltbelastung und die Zufriedenheit der Pendler zu prognostizieren. Solche Simulationen bieten Stadtplanern wertvolle Werkzeuge, um die Wirksamkeit vorgeschlagener Änderungen vor der Umsetzung zu bewerten.

Zusammenfassend ermöglicht die Behandlung Casablancas als Multiagentensystem eine detaillierte, realistische und flexible Simulation der urbanen Verkehrsdynamik. Dies unterstützt eine fundierte Analyse und die Entwicklung optimierter Lösungen, die auf die spezifischen Herausforderungen der Stadt zugeschnitten sind, und ebnet den Weg für eine intelligentere und nachhaltigere urbane Mobilität.

* 1. Vergleichbare Multiagenten-Systeme im Bereich Verkehrssimulation

In der Verkehrsforschung und -planung bietet sich die Modellierung von Verkehrssystemen als Multiagentensystem (MAS) an. Vergleichbare Projekte wie SmartOpenHamburg (SOH) haben gezeigt, wie MAS eingesetzt werden können, um multimodale Verkehrsszenarien zu modellieren und zu analysieren. Diese Ansätze bieten wertvolle Einblicke in die Auswirkungen städtischer Maßnahmen und helfen bei der Entwicklung nachhaltiger Mobilitätslösungen.

Im Kontext von SmartOpenHamburg wurden Agenten eingesetzt, um verschiedene Modalitäten wie Gehen, Radfahren, Autofahren und öffentliche Verkehrsmittel in der Metropolregion Hamburg abzubilden. Dabei standen sowohl individuelle Entscheidungen der Verkehrsteilnehmer als auch ihre Interaktionen im Fokus. Das System ermöglicht die Simulation komplexer Szenarien, in denen die Agenten auf Basis von vorgegebenen Bedingungen, Verkehrsregeln und verfügbaren Ressourcen Entscheidungen treffen.

**Übertragbarkeit auf SmartOpenCasablanca**

Für SmartOpenCasablanca, das von SmartOpenHamburg inspiriert ist, wird ein ähnlicher Ansatz verfolgt. Ziel ist es, ein MAS zu entwickeln, das die spezifischen Anforderungen und Herausforderungen des Verkehrssystems in Casablanca berücksichtigt. Während SOH auf die dichte Infrastruktur und multimodale Mobilität Hamburgs ausgelegt ist, liegt der Fokus bei SmartOpenCasablanca auf den besonderen Bedingungen einer aufstrebenden Großstadt in einem Schwellenland.

**Vergleich mit bestehenden Systemen**

Neben SmartOpenHamburg gibt es weitere relevante Systeme, wie den **MATSim** (Multi-Agent Transport Simulation), der weltweit für Verkehrssimulationen genutzt wird. MATSim bietet ähnliche Funktionen, jedoch mit einem stärkeren Fokus auf großskalige Simulationen. SmartOpenCasablanca verfolgt hingegen das Ziel, kleinräumige Szenarien präzise abzubilden, um realistische Verkehrsflüsse und multimodale Wechselwirkungen zu simulieren.

Zusammenfassend bietet SmartOpenCasablanca eine Gelegenheit, bewährte Ansätze aus vergleichbaren MAS-Projekten wie SOH auf eine neue, kulturell und geografisch unterschiedliche Stadt anzuwenden. Durch die Berücksichtigung lokaler Besonderheiten kann das System wertvolle Einblicke in die Herausforderungen und Chancen des Verkehrssystems in Casablanca liefern.

1. Systemanforderungsanalyse

In diesem Kapitel werden die Anforderungen an das zu entwickelnde System umfassend beschrieben. Dabei erfolgt eine Unterteilung in **funktionale** und **nicht-funktionale Anforderungen**, um eine strukturierte und präzise Analyse sicherzustellen. Die definierten Anforderungen bilden nicht nur die Grundlage für die Systemkonzeption, sondern dienen auch als Maßstab zur abschließenden Bewertung des Umsetzungserfolgs.

## Funktionale Anforderungen

Die Anforderungen an das System sind hauptsächlich funktionaler Art. Diese werden hier aufgeführt.

1. **Simulation urbaner Mobilität**

Das System ermöglicht die Simulation von urbanen Mobilitätsszenarien in Casablanca durch den Einsatz von Agenten, die das Verhalten von Fußgängern, Fahrzeugen und anderen Verkehrsteilnehmern repräsentieren.

1. **Zeitliche Planung**

Agenten können zeitabhängige Ziele verfolgen, wie z. B. das Erreichen eines bestimmten Ziels zu einer definierten Uhrzeit, und planen ihre Routen entsprechend.

1. **Grafische Ausgabe**

Das System erzeugt GeoJSON-Dateien als grafische Ausgabe, die in Tools wie Kepler.gl importiert werden können, um die Trajektorien und Simulationsergebnisse visuell darzustellen.

1. **Definition und Anpassung von Auftakthandlungen**

Benutzer können Auftakthandlungen für die angreifende Mannschaft einfach definieren und anpassen. Die definierten Auftakthandlungen können dann von den Spielern in dem System umgesetzt werden.

1. **Kommunikation zwischen Agenten**

Das System schafft eine Möglichkeit für die Agenten, um miteinander zu kommunizieren. Die Agenten können diese Funktion nutzen, um Informationen untereinander auszutauschen, Auftakthandlungen anzusagen oder auch für andere Zwecke.

1. **Adaptive Verhaltensweisen der Agenten**

Agenten nutzen adaptive Verhaltensweisen, um auf den von ihnen wahrgenommenen Zustand ihres Umfelds zu reagieren und ihr Verhalten entsprechend anzupassen.

1. **Berücksichtigung der Höhe in dem System**

Das System soll nicht nur eine zweidimensionale Simulation umsetzen, sondern auch die Höhe simulieren. Diese Dimension ist im Handball ein wichtiger Aspekt des Spiels, da Spieler beispielsweise durch Sprungwürfe eine bessere Wurfposition erreichen können und somit den Ball über die Abwehr auf das Tor werfen können, selbst wenn sich Gegenspieler im Weg befinden.

1. **Grafische Oberfläche zur Analyse von Spielsituationen**

Das System bietet einem Benutzer eine grafische Nutzeroberfläche, mittels der die Ergebnisse eines Simulationslaufs betrachtet werden können. Diese dient der Analyse der gegebenen Simulation, sowie der Verwendbarkeit des Systems als Mittel für Trainer für das Aufzeigen und Austesten von Auftakthandlungen.

Als Erweiterung der funktionalen Anforderungen dienen zudem auch Anwendungsfälle, die für dieses System definiert wurden. Dabei handelt es sich um Aktionen, die aus Sicht der Akteure (Spieler) innerhalb der Simulation verfügbar sein sollten, um mit der Umgebung, den anderen Akteuren und dem Ball interagieren zu können. Diese Liste von Anwendungsfällen umfasst 14 solcher Fälle, die in Anhang A in tabellarischer Form dargestellt werden.

**REQ-4. Benutzerdefinierte Szenarien**

Das System ermöglicht Nutzern die Erstellung individueller Simulationsszenarien, wie die Simulation bestimmter Stadtteile oder spezifischer Verkehrssituationen. Geo-Daten wie Straßennetze, Haltestellen und POIs können gezielt integriert werden. Nutzer können zudem spezielle Bedingungen wie Stoßzeiten, Baustellen oder Ereignisse definieren, die den Verkehrsfluss beeinflussen. Weiterhin lassen sich Verkehrsmittel sowie das Verhalten der Agenten anpassen, z. B. durch Auswahl von Abfahrtszeiten, Zielorten oder bevorzugten Modalitäten. Die erstellten Szenarien sind speicherbar, wiederverwendbar und erlauben den Vergleich verschiedener Ergebnisse. Dadurch bietet das System eine flexible Modellierung urbaner Mobilität für gezielte Analysen.

## Nicht-funktionale Anforderungen

Neben den **funktionalen Anforderungen** spielen auch **nicht-funktionale Anforderungen** eine wichtige Rolle, da sie qualitative Eigenschaften und Rahmenbedingungen des Systems beschreiben. Dazu zählen Aspekte wie Performance, Skalierbarkeit oder Benutzerfreundlichkeit. Da diese Anforderungen oft abstrakt formuliert sind, sollten sie

1. **Verwendung des MARS-Frameworks**

Das System soll unter Verwendung des MARS-Framework entwickelt werden. Dieses an der HAW Hamburg entwickelte Framework für Multiagentensysteme bietet eine Fülle an Möglichkeiten, um die funktionalen Anforderungen zu erfüllen.

1. **Modularer Aufbau des Systems**

Das System soll modular aufgebaut sein, um spätere Erweiterungen wie zusätzliche Modalitäten (z. B. neue Verkehrsmittel) oder die Integration neuer Stadtteile zu ermöglichen.

1. **Dokumentation**

Das Projekt soll eine umfassende Dokumentation umfassen. Diese Dokumentation soll alle Aspekte des Systems abdecken, einschließlich der Implementierung, Nutzung und Erweiterungsmöglichkeiten. Ziel ist es, den Einstieg für Entwickler und Nutzer zu erleichtern.

1. Entwurf der Simulations- und Modellumgebung

Um die in Kapitel 3 definierten Anforderungen zu erfüllen, wurde ein Konzept für das System entwickelt, das die vorgegebenen Rahmenbedingungen berücksichtigt. In diesem Kapitel werden die Spezifikationen ausführlich erläutert und aufgezeigt, wie diese die gestellten Anforderungen erfüllen. Dabei wird zunächst eine Übersicht über das gesamte Modell präsentiert, bevor ausgewählte Aspekte detailliert betrachtet werden.

Zur Erstellung des Konzepts wurden vergleichbare Systeme analysiert, wie in Kapitel 2.5 beschrieben. Zudem diente das öffentlich zugängliche **SOH-Modell** als Grundlage. Ein Fork der bestehenden Codebasis wurde erstellt und als Ausgangspunkt für die Konzeption des Systems verwendet.

## Agententypen

Das System nutzt Agenten, um Mobilitätsverhalten zu modellieren. Es bietet verschiedene vordefinierte Agententypen, die zur Untersuchung unterschiedlicher Szenarien verwendet werden können. Bestehende Typen lassen sich wiederverwenden, und es können neue Klassen für spezifische Anforderungen definiert werden. Folgendes zeigt die Agententypen mit ihren verschiedenen Rollen und Mobilitätsmuster.

**MultimodalAgent**

Der **MultimodalAgent** ist ein vielseitiger Agent, der verschiedene Transportmodalitäten bewältigen und nahtlos zwischen ihnen wechseln kann. Seine Hauptaufgabe besteht darin, multimodale Mobilität zu ermöglichen, indem er eine repräsentative Entität, den **MultimodalRoadUser**, nutzt. Dieser enthält physische Attribute (z. B. Größe, Position) und positioniert den Agenten in der Simulationsumgebung. Der MultimodalAgent dient als Grundlage für alle Mobilitätsagenten, die zwischen unterschiedlichen Modalitäten wechseln können. Ein klassisches Beispiel eines Subtyps ist der **Pedestrian** (Fußgänger), der sich zu Fuß entlang eines graphbasierten Modells bewegen kann.

Die Stärke des **MultimodalAgent** liegt in der Fähigkeit, verschiedene Steuerungsmodi zu implementieren, die spezifisch auf einzelne Transportmodalitäten zugeschnitten sind. Beispielsweise erweitert der **MultiCapableAgent** den MultimodalAgent, um Modalitäten wie **Gehen**, **Autofahren**, **Radfahren** und **Fährefahren** zu unterstützen. Jede dieser Modalitäten erfordert die Implementierung eines spezialisierten Steuerungs-Interfaces, wie z. B. **ICarSteeringCapable**, wie im Diagramm dargestellt. Diese Interfaces definieren die notwendigen Methoden und Eigenschaften, die der Agent implementieren muss, um das jeweilige Fahrzeug effektiv zu steuern.

Zusätzlich zu den Steuerungsfähigkeiten unterstützt der MultimodalAgent auch Interaktionen als Passagier. Der Agent kann bestimmte Fahrzeuge nutzen, indem er z. B. auf eine Fähre ein- und wieder aussteigt. Dies wird über das **IPassengerCapable**-Interface realisiert, das dem Agenten mittels **PassengerMessages** Echtzeit-Feedback über den Status der Fahrt liefert. Solche Nachrichten informieren den Agenten beispielsweise bei jedem Stopp (z. B. an einer Fährstation). Auf Basis dieser Benachrichtigungen kann der Agent dynamisch entscheiden, ob er aussteigen oder die Fahrt fortsetzen möchte.

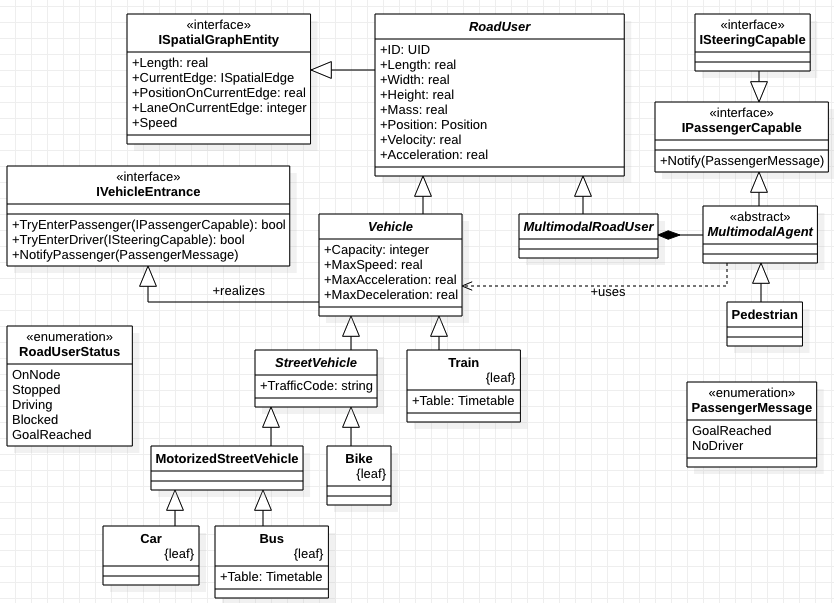


Abbildung X: Klassendiagramm

**MultiCapableAgent**

Der **MultiCapableAgent** ist eine Erweiterung des MultimodalAgent, der unterschiedliche Steuerungsfähigkeiten implementiert. Aktuell unterstützt dieser Agent die Steuerung von Fahrzeugen über das Interface ICarSteeringCapable sowie das Radfahren über das Interface IBicycleSteeringAndRentalCapable. Zusätzlich kann der Agent Fähren nutzen und als Fußgänger agieren (Pedestrian). Er zeichnet sich durch seine konfigurierbaren Fähigkeiten aus, die es ermöglichen, verschiedene Modalitäten dynamisch zu aktivieren oder zu deaktivieren. Dadurch können mehrere Agenten desselben Typs über unterschiedliche Transportmöglichkeiten verfügen.

* **Fahrzeugsteuerung (Auto):** Die Fähigkeit zum Autofahren wird über die Eigenschaft **CapabilityDriving** gesteuert.
* **Radfahren:** Die Fähigkeit zum Radfahren wird über die Eigenschaft **CapabilityCycling** eingestellt.

Diese Konfigurationsmöglichkeiten machen den **MultiCapableAgent** besonders flexibel, da er an unterschiedliche Szenarien und Anforderungen angepasst werden kann. Die spezifische Aktivierung oder Deaktivierung der Fähigkeiten ermöglicht die Modellierung vielfältiger Mobilitätsprofile innerhalb eines Simulationsmodells.

A screenshot of a black screen

Description automatically generated

1. Realisierung des Systems

Nachdem in Kapitel 4 der Modellaufbau ausführlich erläutert wurde, widmet sich das folgende Kapitel den spezifischen Implementierungsdetails. Dabei werden insbesondere jene Aspekte näher betrachtet, die von der ursprünglichen Modellierung abweichen oder bei denen eine detaillierte Beschreibung der Umsetzung aus technischen oder konzeptionellen Gründen erforderlich ist.

* 1. Wahl der Programmiersprache

Das **MARS-Framework** wurde in der Programmiersprache **C#** entwickelt, weshalb auch die Implementierung des Systems in C# realisiert wurde. Für die Visualisierung der Ergebnisse wurde vorwiegend auf **Jupyter Notebooks** zurückgegriffen, da diese eine flexible Analyse und anschauliche Darstellung der Agentendaten ermöglichen. Die interaktive Entwicklungsumgebung von Jupyter Notebooks erlaubt es, Visualisierungen effizient zu erstellen und Ergebnisse klar strukturiert zu präsentieren. Diese Kombination aus der leistungsstarken Simulation in C# und der benutzerfreundlichen Visualisierung mit Jupyter Notebooks bietet eine optimale Grundlage für die Analyse und Auswertung der Simulationsergebnisse.

.

* 1. Konfiguration und Ausführung des Systems
     + 1. Die Klasse **Program** dient als Einstiegspunkt des Multiagentensystems und enthält die zentrale **Main-Methode**, die für die Initialisierung und den Start der Simulation verantwortlich ist. Ihre Hauptaufgabe besteht darin, die erforderlichen Layer, Agenten und Entitäten zu definieren sowie die Konfiguration der Simulation bereitzustellen. Die Implementierung dieser Klasse ist in der Datei **Program.cs** auf Root-Ebene des Projekts zu finden. Zusätzliche Konfigurationsmöglichkeiten für Agenten und Layer werden durch weitere Dateien bereitgestellt, die im Folgenden beschrieben werden. Manche Dateien sind im **Resources-Verzeichnis** des Systems abgelegt.
       2. **config.json**

Die JSON-Datei bietet eine detaillierte Konfigurationsmöglichkeit für die Simulation und wird in der Main-Methode eingelesen, um die wesentlichen Parameter, Agenten, Layer und Entitäten der Simulation zu definieren. Nachfolgend werden die wichtigsten Bestandteile der Datei erläutert:

**1. Globale Einstellungen**

Im Abschnitt **globals** werden die globalen Parameter der Simulation festgelegt, die die zeitliche Dynamik und die Ausgabe steuern. Dazu gehört **deltaT**, der die Länge eines Simulationsschritts (Tick) definiert. Je nach gewähltem Zeitmodell beeinflusst **deltaT** die zeitliche Granularität und Ausführung der Simulation:

1. **Schrittbasierte Ausführung**: Hier wird die Simulation in einer festen Anzahl von Schritten ausgeführt, wobei **deltaT** die Länge eines Schritts angibt (z. B. 1 Sekunde). Die gesamte Anzahl der Schritte wird durch den Parameter **steps** festgelegt. Dieses Modell eignet sich für Simulationen, die keine zeitlichen Abhängigkeiten berücksichtigen.
2. **Echtzeitbasierte Ausführung**: In diesem Modell wird ein Start- und Endzeitpunkt **(startPoint, endPoint)** definiert, und **deltaT** bestimmt in Kombination mit **deltaTUnit** die Länge eines Zeitschritts (z. B. 1 Minute). Dieses Modell ist ideal für Simulationen, die zeitabhängige Ereignisse oder Kalenderbezüge enthalten.
3. **Individuelle Ausführungsfrequenz**: Zusätzlich kann die Ausführungsfrequenz für Agenten und Layer über **executionFrequency** festgelegt werden. Dieser Wert gibt an, nach wie vielen Ticks ein Agent oder Layer ausgeführt werden soll.

Die Zeiteinheit für **deltaT** wird durch **deltaTUnit** **Parameter** definiert und kann zwischen Mikrosekunden und Jahren variieren. Im Ergebnis wird bei einer echtzeitbasierten Ausführung eine zusätzliche **DateTime**-Angabe generiert, die jedem Simulationsobjekt eine konkrete Gültigkeitszeit zuordnet.

Zuletzt wird mit **console Parameter** angegeben, ob Konsolenausgaben aktiviert sind.

**2. Agentenkonfiguration**

Im Abschnitt „**agents**“ werden die Agententypen spezifiziert, die in der Simulation zum Einsatz kommen. Es ist darauf zu achten, dass sämtliche im Modell definierten Agententypen auch in der Konfigurationsdatei referenziert werden müssen. Der Parameter „**file**“ verweist auf eine CSV-Datei, die zur Initialisierung der Eigenschaftswerte der Agenten zu Beginn der Simulation dient. Diese Vorgehensweise ermöglicht eine individuelle Konfiguration der Eigenschaften jedes instanziierten Agenten. Der optional verwendbare Parameter „**mapping**“ bietet eine zusätzliche Möglichkeit, die Eigenschaften der Agenten zu Beginn der Simulation festzulegen. Diese Methode hat Einfluss auf alle instanziierten Agenten eines bestimmten Agententyps. Sollte für eine Eigenschaft sowohl in der Initialisierungsdatei des Agenten als auch in der Mapping-Konfiguration ein Wert angegeben sein, so wird der Wert aus der Konfiguration bevorzugt.

**3. Layer-Konfiguration**

„layers“ Abschnitt enthält eine Liste aller im Modell definierten Layer. Es ist zu beachten, dass alle im Modell definierten Layer auch in der Konfiguration referenziert werden müssen. Der Schlüssel „file“ verweist auf eine Datei (z. B. eine .geojson- oder .graphml-Datei), die geospatiale Informationen enthält, die zur Instanziierung des Layers verwendet werden. Der Schlüssel „mapping“ ist optional und ermöglicht die Zuweisung von Werten zu den Eigenschaften des Layers (wie sie in der Layer-Klasse definiert sind).

* 1. Implementierung

1. Experimente und Ergebnisse

1. Diskussion der Ergebnisse und Bewertung des Systems

1. Schlussfolgerungen und Ausblick

Literaturverzeichnis

Erklärung zur selbstständigen Bearbeitung einer Abschlussarbeit

Hiermit versichere ich, dass ich die vorliegende Arbeit ohne fremde Hilfe selbständig verfasst und nur die angegebenen Hilfsmittel benutzt habe. Wörtlich oder dem Sinn nach aus anderen Werken entnommene Stellen sind unter Angabe der Quellen kenntlich gemacht.

|  |  |  |
| --- | --- | --- |
| \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |
| Ort | Datum | Unterschrift im Original |