

AUFGABENBLATT 3

Modellierungsbeschreibung

abs16
11. April 2016

1 ALLGEMEINES

Dieses Dokument dient dazu, das Multiagentensystem des Projekts abs16 in seinen einzelnen Paketen darzustellen. Die Pakete werden hinsichtlich ihrer eigenen Funktionalität, ihrer Voraussetzungen an andere Pakete und ihren angebotenen Funktionalitäten beschrieben.

Das Endprodukt wird das Framework JADEx verwenden. Auf konkrete Beschreibung der Implementierung wird in diesem Dokument der Übersicht halber jedoch verzichtet.

2 PRODUKTÜBERSICHT

In diesem Abschnitt soll es darum gehen, die Produktfunktionen zu beschreiben. Dies geschah bereits ausführlich im Arbeitsplan. Daher werden die Funktionen nur nach Schlagwörtern aufgegriffen, geordnet und in verschiedene Gruppen eingeteilt, die dann von den Paketen realisiert werden.

Ziel dieses Multiagentensystems ist es, über proaktiv handelnde Agenten die Diffusion am Energiemarkt zu simulieren.

2.1 PRODUKTFUNKTIONEN

Bevor die oben angesprochene Gruppierung der Funktionen vorgenommen wird, werden die Produktfunktionen rekapituliert:

Das Produkt umfasst Produkte mit Eigenschaften.

Das Produkt umfasst Agenten mit Eigenschaften, Präferenzen/Wunschprodukten und Verhalten. Agenten werden unter Kundengruppen zusammengefasst, die Wertebereiche für Eigenschaften und Wunschprodukte festlegen.

Agentenverhalten umfasst Interaktion zwischen Agenten in Form von Kommunikation und Interaktion mit der Welt in Form eingehender Kommunikationsevents (z. B. Werbung) und eingehender Events, die Agentenattribute verändern.

Die Konfiguration des Produkts lässt sich sowohl über eine GUI als auch über Konfigurationsdateien vornehmen. Konfigurationsdateien lassen sich aus der GUI heraus laden und speichern.

Am Ende der Simulation wird das gespeicherte Verhalten der Agenten in Form von Graphen ausgegeben.

2.2 FUNKTIONSGRUPPEN

Um die oben aufgezählten Produktfunktionen sinnvoll zu implementieren, werde sie in mehrere funktionale Gruppen eingeteilt, die dann von verschiedenen Komponenten und Paketen der Implementierung des Endprodukts realisiert werden.

Die Funktionen werden in folgende Gruppen unterteilt:

Simulationsfunktionen Funktionen der tatsächlichen Simulation innerhalb des Gesamtprodukts

Konfigurationsfunktionen Funktionen der Konfiguration beschreiben alle Ansprüche an Parametrisierung der Simulation

Ausgabefunktionen Funktionen der Ausgabe betreffen alle Ansprüche an Ausgabe von Ergebnissen der Simulation (Anm.: dies umfasst nicht die Darstellung der Ergebnisse)

Nutzer-Funktionen Funktionen für den Nutzer betreffen Ein- und Ausgabemöglichkeiten des Gesamtprodukts aus Nutzersicht

2.2.1 SIMULATIONSFUNKTIONEN

Funktionen der Simulation greifen Teile der Absätze über Produkte, Agenten und Agentenverhalten auf.

Die Simulation muss...

- ... Agenten mit Eigenschaften und Wunschprodukten verarbeiten.
- ... Agentenverhalten verarbeiten.
- ... Produkte mit Eigenschaften und Kommunikations-Events verarbeiten.

2.2.2 KONFIGURATIONSFUNKTIONEN

Funktionen der Konfiguration greifen Teile der Absätze über Produkte, Agenten und Konfigurationen auf.

Die Konfiguration der Simulation muss...

- ... Agentengruppen erstellen.
- ... Agenten aus Agentengruppen erstellen.
- ... Produkte erstellen.
- ... das Erstellen von Objekten über Methoden der Programmiersprache ermöglichen.
- ... das Erstellen von Objekten über Konfigurationsdateien ermöglichen.
- ... das Erstellen von Konfigurationsdateien ermöglichen.

2.2.3 AUSGABEFUNKTIONEN

Funktionen der Ausgabe greifen den Absatz über das Ende der Simulation auf.

Die Ausgabe muss...

- ... das Speichern von Agentenverhalten ermöglichen.
- ... das Speichern von Agenten-Zuständen ermöglichen.
- ... die Ausgabe gespeicherter Daten über Methoden der Programmiersprache ermöglichen.

2.2.4 NUTZER-FUNKTIONEN

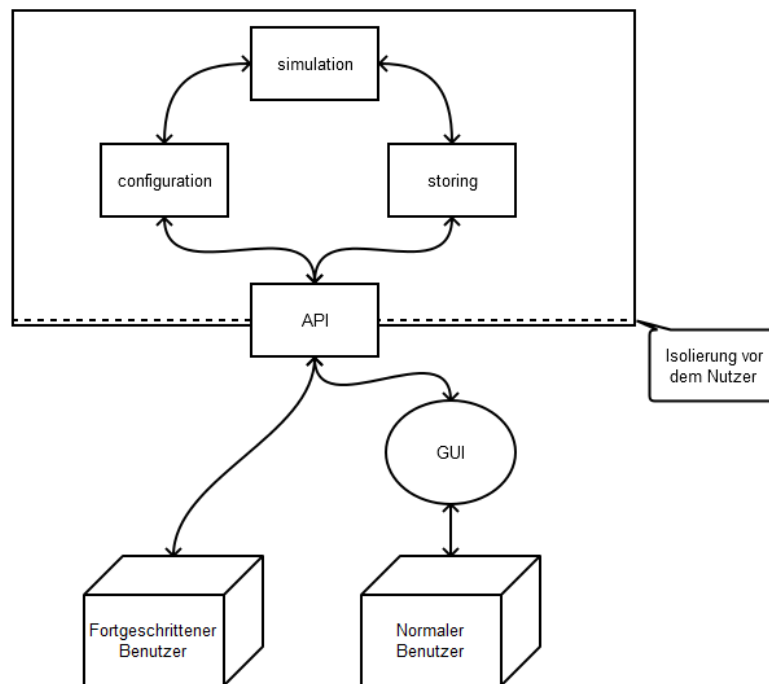
Funktionen für den Nutzer greifen Teile der Absätze über Konfiguration und das Ende der Simulation auf.

Der Nutzer kann...

- ... die Simulation über eine GUI konfigurieren.
- ... die Simulation über Konfigurationsdateien konfigurieren.
- ... Konfigurationen der Simulation in Konfigurationsdateien speichern.
- ... die Ergebnisse der Simulation über eine GUI einsehen.
- ... die Ergebnisse der Simulation in Dateien speichern lassen.

3 GRUNDSÄTZLICHE STRUKTUR- UND ENTWURFSPRINZIPIEN

Das Endprodukt wird in seiner Implementierung in folgende Pakete eingeteilt:

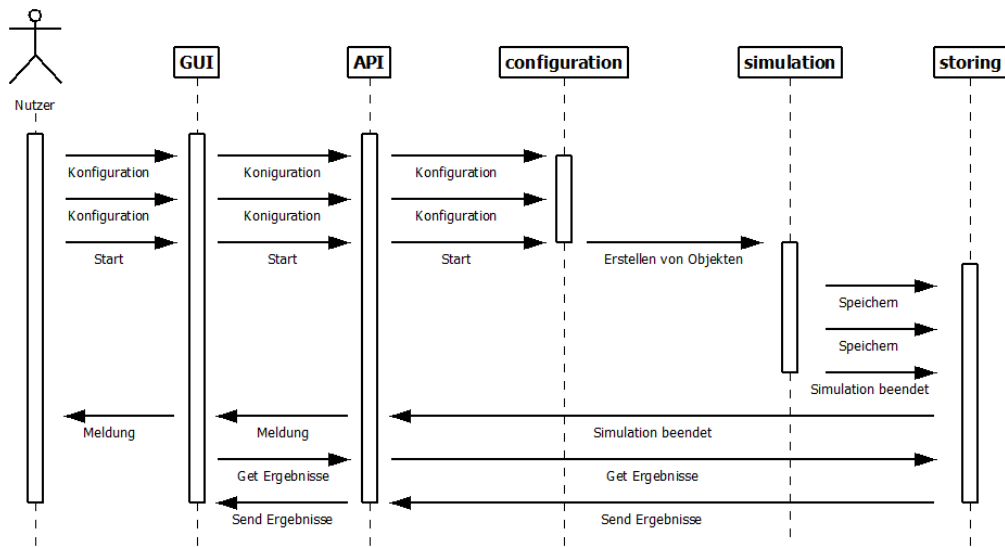


Dabei übernehmen Pakete jeweils die Implementierung von nicht unbedingt verschiedenen Funktionsgruppen.

Die oberen, eingeschlossenen vier Pakete umfassen den Teil des Produkts, der sich insgesamt um die Simulation kümmert. Durch die API ist es möglich, diesen Teil des Produkts auch ohne eine GUI zu nutzen.

Die GUI dient somit nur dazu, dem Nutzer eine einfachere Steuerung der API zu geben, sodass auch technisch unerfahrene Benutzer das Produkt verwenden können.

Um die grobe Funktionalität der einzelnen Pakete zu umreißen, wird folgendes Sequenzdiagramm erläutert:



Nach dem Programmstart führt der Nutzer nacheinander durch mehrere Aktionen *Konfiguration* Einstellungen an der Simulation durch. Jede dieser Einstellungen wird von der API an das Paket *configuration* weitergereicht.

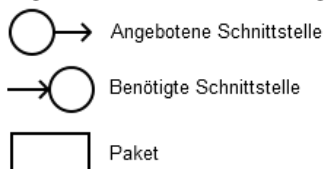
Nachdem die Konfiguration beendet wurde, startet der Nutzer durch die Aktion *Start* die Simulation. Die API reicht auch diese Aktion an das *configuration*-Paket weiter, das aus den Einstellungen die eigentliche Simulation startet.

Das *simulation*-Paket übernimmt nun die Berechnung des Multiagentensystems und speichert Ergebnisse der Berechnung über die Aktion *Speichern* in dem Paket *storing*. Nach Beenden der Simulation benachrichtigt das Paket *simulation* das Paket *storing* darüber. Dieses Paket leitet die Nachricht über die API bis zum Nutzer weiter.

Die GUI fragt über die Aktion *Get Ergebnisse* die Simulationsergebnisse selbstständig an, die am Ende dargestellt werden.

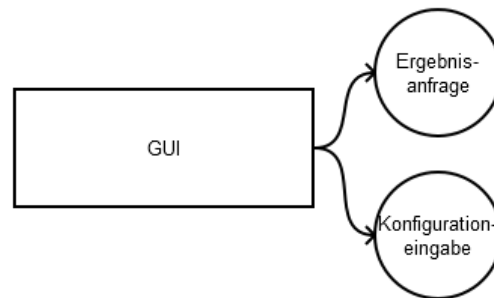
Die Lebenszyklen der Pakete werden sich in der Implementierung von der obigen Darstellung unterschieden. Dieses Sequenzdiagramm wurde für den Standardfall der Benutzung des Endprodukts entworfen und zeigt die Lebenszyklen der Pakete nur für ihre Hauptfunktionalität an.

Legende zur Visualisierung einzelner Pakete (folgt auf nächster Seite):



4 STRUKTUR- UND ENTWURFSPRINZIPIEN EINZELNER PAKETE

4.1 PAKET GUI



GRUNDSÄTZLICH Die GUI dient als Nutzer-Schnittstelle für unerfahrene Benutzer. Sie greift allein auf Funktionen der API zu. Über sie lässt sich das gesamte Produkt verwenden.

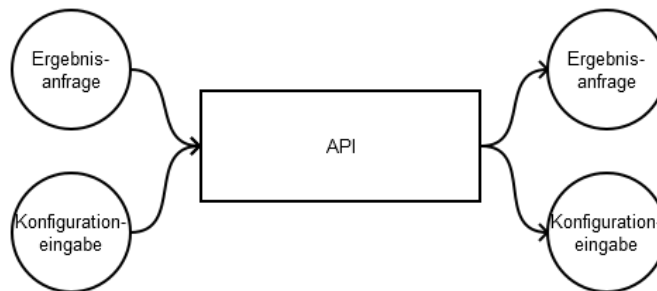
Damit implementiert sie einen Teil der Nutzer-Funktionen.

Die GUI wird systemunabhängig implementiert, damit keine Nutzergruppe ausgeschlossen wird.

ANGEBOTENE SCHNITTSTELLEN Aus technischer Sicht bietet die GUI keine Schnittstellen an.

BENÖTIGTE SCHNITTSTELLEN Die GUI benötigt Schnittstellen zur Konfigurationseingabe für eine Simulation und das Anfragen von Ergebnissen einer Simulation. Diese Schnittstellen werden in dem Paket API realisiert.

4.2 PAKET API



GRUNDSÄTZLICH Die API dient der tatsächlichen Steuerung der Simulation und dient somit auch als Nutzer-Schnittstelle. Sie ist Teil des Produkts, damit die tatsächliche Funktionalität gegenüber dem Anwender verborgen bleiben kann und Anpassungen aufgrund von Fehlerbehebungen (o.ä.) nicht notwendig Anpassungen der GUI nach sich ziehen.

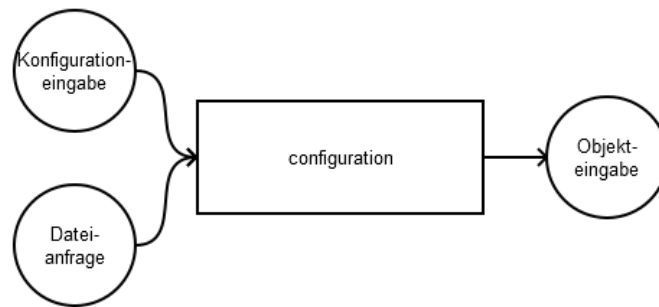
Die Programmlogik kann somit ein Stück weit abstrahiert werden.

Die API implementiert damit Nutzer-Funktionen.

ANGEBOTENE SCHNITTSTELLEN Die API bietet eine Schnittstelle zur Konfiguration einer Simulation und zur Anfrage von Ergebnissen dieser an bzw. realisiert diese Schnittstellen.

BENÖTIGTE SCHNITTSTELLEN Die API benötigt Schnittstellen um Konfigurationen und Ergebnisanfragen weiterzuleiten. Diese Schnittstellen werden in den Paketen `storing` und `configuration` angeboten.

4.3 CONFIGURATION



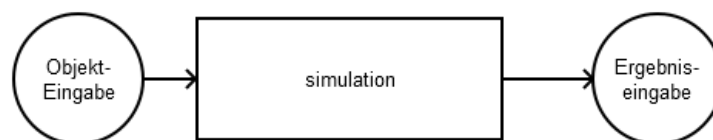
GRUNDSÄTZLICH Dieses Paket speichert Konfigurationen und erstellt aus diesen Datenobjekte für die Simulation.

Es implementiert damit alle Konfigurationsfunktionen.

ANGEBOTENE SCHNITTSTELLEN `configuration` bietet eine Schnittstelle zur Eingabe von Konfigurationen und eine zur Anfrage von Konfigurationsdateien an.

BENÖTIGTE SCHNITTSTELLEN `configuration` benötigt eine Schnittstelle zur Weitergabe von erstellten Datenobjekten zur Simulation. Diese Schnittstelle wird durch das Paket `simulation` realisiert.

4.4 SIMULATION



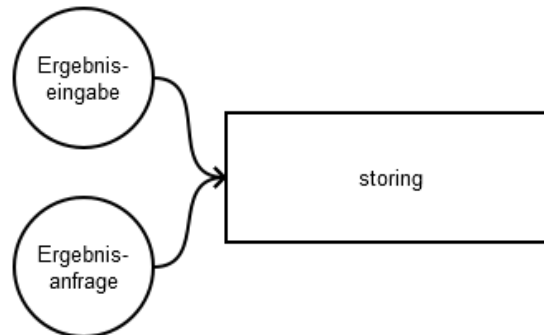
GRUNDSÄTZLICH Dieses Paket führt die Berechnung der Simulation und das Speichern von Ergebnissen dieser durch.

Damit implementiert es alle Funktionen der Gruppe Simulation.

ANGEBOTENE SCHNITTSTELLEN Dazu bietet es eine Schnittstelle an, über die Datenobjekte zur Simulation weitergegeben werden.

BENÖTIGTE SCHNITTSTELLEN Dieses Paket benötigt eine Schnittstelle, um Ergebnisse der Simulation zu speichern. Diese Schnittstelle wird von dem Paket storing angeboten.

4.5 STORING



GRUNDSÄTZLICH Dieses Paket erhält während der Berechnung der Simulation Ergebnisse von dieser, um sie zu speichern und am Ende auszugeben.

Das Paket storing implementiert damit die Ausgabefunktionen.

ANGEBOTENE SCHNITTSTELLEN Dazu bietet es zwei Schnittstellen an: Zum einen können Ergebnisse gespeichert, zum anderen können Ergebnisse angefragt werden.

BENÖTIGTE SCHNITTSTELLEN Das Paket storing benötigt keine Schnittstellen.

5 DATENMODELL

Unser Multiagentensystem wird bei der Speicherung aller Daten auf Datenbanksysteme verzichtet. Dafür gibt es drei Gründe:

- Ein Datenbanksystem würde eine eigene Anwendung zur Ausgabe der Daten erfordern
- Ein Datenbanksystem würde während der Laufzeit eine eigene Schnittstelle erfordern
- Daten in einem Datenbanksystem lassen sich nicht ohne weiteres durch den Endnutzer modifizieren und sind im hohen Grade von den Anwendungsprogrammen abhängig

Wir wählen - je nach Szenario - verschiedene *sprechende* Datentypen. Sie erlauben dem Nutzer auch ohne spezielle Anwendungsprogramme, Aufbau und Ergebnis der Simulation nachzuvollziehen oder zu bearbeiten.

5.1 EINGABEDATEN

Konfigurationsdateien werden über JSON-Dateien gespeichert. Dieses Format erlaubt Bearbeitung durch den Nutzer, z. B. in Texteditoren. Mit entsprechendem Vorwissen ist es somit möglich, manuell Konfigurationen zu erstellen oder zu modifizieren.

5.2 AUSGABEDATEN

Ausgabedaten werden in Form von Bilddateien oder CSV-Dateien ausgegeben. Sie sind im hohen Maße anwendungs- und systemunabhängig und erlauben es auch unerfahrenen Benutzern, Ergebnisse nachzuvollziehen und weiter zu verarbeiten.

6 GLOSSAR

Modell “Modelle sind stets Modelle von etwas, nämlich Abbildungen, Repräsentationen natürlicher oder künstlicher Originale, die selbst wieder Modelle sein können.

[...]

Modelle erfassen im allgemeinen nicht alle Attribute des durch sie repräsentierten Originals, sondern nur solche, die den jeweiligen Modellerschaffern und/oder Modellbenutzern relevant scheinen.” [3]

Agent “An intelligent agent is a computer system that is capable of flexible autonomous action in order to meet its design objectives. By flexible we mean that the system must be: responsive [,] proactive [and] social.” [4]

Ein intelligenter Agent ist ein Computersystem, das zu flexiblen, autonomen Handlungen fähig ist, um seine angestrebten Entwurfsziele zu erreichen. Mit flexibel meinen wir, dass das System ansprechbar, proaktiv und sozial ist.

Multiagentensystem “By an agent-based system, we mean one in which the key abstraction used is that of an agent.” [4]

Mit einem agentenbasierten System meinen wir eins, dessen Hauptabstraktion ein Agent ist.

Simulation System, das mit ähnlichen Eigenschaften wie ein Original erstellt wurde um Reaktionen und Veränderungen des Originals zu ermitteln. [5]

Innovationsdiffusion “Diffusion is the process by which an innovation is communicated through certain channels over time among the members of a social system.” [6]

Diffusion ist der Prozess, durch den eine Innovation über bestimmte Kanäle an die Mitglieder einer Gesellschaft kommuniziert wird.

Paket Pakete sind Ansammlungen von Modellelementen beliebigen Typs, mit denen das Gesamtmodell in kleinere überschaubare Einheiten gegliedert wird. [7]

GUI Graphical User Interface - Graphische Benutzeroberfläche

Konfiguration Eine Konfiguration beschreibt einen abgeschlossenen Zustand des Endprodukts, aus dem eine Simulation generiert werden kann.

7 QUELLEN

- [1] Goll J. Methoden und Architekturen der Softwaretechnik. 2011; 725.
- [2] Goll J, Dausmann M. Architektur- und Entwurfsmuster der Softwaretechnik. Mit lauffähigen Beispielen in Java 2013; 30.
- [3] Herbert Stachowiak: Allgemeine Modelltheorie, 1973; 131,132.
- [4] Jennings N. R, Michael J. Wooldridge M. J. Agent Technology: Foundations, Applications, and Markets, 1998; 4,5.
- [5] <https://de.wiktionary.org/wiki/Simulation>, 11.04.2016.
- [6] Rogers E. M. Diffusion of Innovations, 1995; 5.
- [7] <http://www.oose.de/glossar/paket-2/>, 11.04.2016