Architektur und Umsetzung des Multi-Agenten-Modells zur Diffusion von Nachhaltigkeitsinnovationen

Daniel Abitz[1,2], *Simon Johanning*[3]

Highlights

- Für das Agentenmodell-Framework (`IRPact`) zur Simulation der Diffusion von Innovationen sollte eine leicht erweiterbare Implementierung erfolgen.
- Mittels eines modularen Design Patterns, welches die Austauschbarkeit von Komponenten ermöglicht, wurde diese Anforderung umgesetzt.
- Das Framework `IRPact` ermöglicht den Nutzenden über einen Baukasten an Komponenten und Funktionen (von der Parametrisierung der Agenten, über die Bereitstellung von Eingabedaten und deren Verarbeitung bis hin zur Ergebnisdarstellung) problemorientierte Szenarien zu erzeugen und zu simulieren.
- Existierende Szenarien lassen sich dank der starken Modularisierung leicht um neue Aspekte erweitern und parametrisieren.

Einführung

Zur Abbildung der im Projekt entwickelten fachlichen Modellierung im Rahmen eines agentenbasierten Modells zur Diffusion nachhaltiger Innova-

[1]Institut für Informatik, Universität Leipzig
[2]Universitätsrechenzentrum, Universität Leipzig
[3]Institut für Infrastruktur und Ressourcenmanagement, Universität Leipzig

tionen wurde im Vorfeld des Projektes ein strukturell flexibles, modulares und erweiterbares Framework zur Beschreibung und Implementierung bestehender und zukünftiger Modelle entwickelt.[4]

Die softwaretechnische Umsetzung der einzelnen Modellkomponenten wurde in diesem Ansatz jedoch stark zusammenhängend entwickelt. Dies hatte den Nachteil, dass sich einzelne Komponenten teilweise nur umständlich um neue Funktionalitäten erweitern ließen. Aus diesem Grund wurde der Ansatz zum modularen Agentensimulations-Framework IRPact weiterentwickelt. Dabei sollen Benutzer:innen bei der Konfiguration ein maximaler Grad an Freiheit geboten und über ein Baukastenprinzip die Modifikation von Komponenten zu ermöglicht werden.

Ein weiteres Ziel bei der Neuimplementierung von IRPact war zudem die Überarbeitung des grundlegenden Prinzips der Verhaltenssteuerung der Agenten. Dieses bestimmt wie die Agenten innerhalb der Simulation untereinander und mit der Simulationsumgebung (unter anderem mit Produkten und Ereignissen) interagieren. Zu diesem Zweck wurde das belief-desire-intention (BDI) Modell[5] als grundlegender Architekturansatz für die Agentensimulation gewählt und implementiert. In den folgenden beiden Abschnitten wird zu diesem Zweck auf das Modell selber und die grundlegende Architektur sowie deren Kopplung eingegangen. Am Beispiel des Aufdach-Photovoltaik-Diffusionsmodells PVact wird die Anwendung des Frameworks verdeutlicht.

Belief-desire-intention Modell

Menschliche Entscheidungen werden häufig auf Basis mangelnder Informationen getroffen (*bounded rationality*).[6] Ein geeignetes Modell für die Abbildung menschlicher Eigenschaften in Simulationen stellt das belief-desire-intention (BDI) Modell dar, welches für die Abstraktion menschlicher Entscheidungsfindung geeignet ist.[7] Daher wird dieser Ansatz oft bei sozialwissenschaftlichen Simulationen verwendet.[8] Beispiele hierfür

[4] Vgl. Johanning u. a., 2020.
[5] Vgl. Bratman, 1987.
[6] Vgl. Simon, 1955.
[7] Vgl. Norling, 2004.
[8] Vgl. Adam und Gaudou, 2016.

sind Verhaltensanalysen von Menschenmengen in Gefahrensituationen[9] oder bei Evakuierungen[10]. Aber auch in Bezug auf ökonomische Aspekte findet die BDI-Architektur Anwendung. So wird es für die Simulation des Kaufverhaltens von Konsumenten herangezogen.[11]

Das belief-desire-intention Modell beschreibt die menschliche Entscheidungsfindung mittels der folgenden drei Komponenten:

belief:	Das *Wissen* über die Welt und das System an sich.
desire:	Die *Motivation* ein bestimmtes *Ziel* zu erreichen.
intention:	Die *Absicht* das gesetzte Ziel mittels der Durchführung von *Plänen* zu erreichen.

Dieses Konzept wurde für die Entwicklung intelligenter Agenten adaptiert.[12] BDI-Agenten besitzen *Wissen* über die Welt, in der sie sich befinden (**belief**). Dieses Wissen lässt sich durch *Kommunikation* mit anderen Agenten oder durch Schlussfolgerungen erweitern. Mittels *Plänen* versuchen sie ihr aktuelles Ziel zu erreichen (**desire** und **intention**). Pläne stellen dabei *Handlungsanweisungen* dar, welche von den Agenten durchgeführt werden. Als Resultat kann das Ziel nun entweder erfolgreich abgeschlossen oder verworfen werden. Der Aspekt, dass ein Agent überhaupt erst dazu animiert wird, ein Ziel zu verfolgen, wird über Ereignisse gesteuert. Diese können zum einen extern sein, zum Beispiel durch das Einbringen neuer Informationen in das System, oder durch eine Änderung ihrer Wissensbasis. Solche internen Ereignisse können beispielsweise durch die Kommunikation mit anderen Agenten ausgelöst werden.

Ein wichtiger Aspekt bei der Entscheidung für das BDI-Modell im Rahmen von IRPact war, dass es sich aufgrund seiner alltagspsychologischen Basis sehr gut auf Entscheidungsfindungsprobleme anwenden lässt. Es ermöglicht das Einbeziehen von verschiedenen Einflüssen, Faktoren und Schlussfolgerungen bei der Wahl der zu treffenden Entscheidung. Somit verspricht das BDI-Modell einen hohen Grad an Realitätsnähe.

[9] Vgl. Shendarkar u. a., 2008.
[10] Vgl. Bulumulla u. a., 2018.
[11] Vgl. Baptista, C. Martinho u. a., 2013; Baptista, C. R. Martinho u. a., 2014.
[12] Vgl. Rao, Georgeff u. a., 1995.

Umsetzung des BDI-Modells in IRPact

Für die Umsetzung des BDI-Modells wurden dessen Konzepte in den entsprechenden Komponenten der Simulation umgesetzt. Das Wissen (**belief**) der Agenten wurde dadurch realisiert, dass sie Informationen über sich selbst, andere Agenten als auch Produkte haben und diese durch Aktionen, wie Kommunikation (internes Ereignis), verändern und erweitern können. Die Motivation (**desire**) wurde durch den Wunsch nach Produktadoption operationalisiert. Der letzte Bestandteil, die auf Plänen basierende Zielrealisierung (**intention**), wurde durch das Prozessmodell umgesetzt.

Als eine Simulation für die Diffusion von Innovationstechnologien lässt sich das grundlegende Prinzip von `IRPact` somit folgendermaßen zusammenfassen:
Innerhalb einer simulierten Umgebung können Verbraucheragenten angebotene Produkte adoptieren. Mittels Kommunikation mit anderen Verbrauchern tauschen sie Wissen aus und wecken somit potentiell bei diesen Interesse für einen Erwerb. Über den Simulationszeitraum betrachtet lässt sich somit die Diffusion des Produktes innerhalb der Agentenpopulation analysieren.

Eine Übersicht über die Hauptkomponenten ist in der Abbildung 7.1 dargestellt. Die fundamentale Modellkomponente ist das zeitliche Modell, welches den chronologischen Verlauf der Simulation beschreibt. Verbraucher, welche intern durch das Konzept `ConsumerAgents` repräsentiert werden, gehören Gruppen an, mit denen sie grundlegende Eigenschaften teilen. So besitzen alle Mitglieder einer Gruppe dieselben Typen von Eigenschaften, wenn auch mit unterschiedlicher Ausprägung. Durch das räumliche Modell können sie zusätzlich um positionsbezogene Informationen angereichert werden. Die zu adoptierenden Produkte gehören ebenfalls Gruppen an, um unterschiedliche Konfigurationen realisieren zu können. Das Kommunikationsverhalten beschreibt, wie Agenten Informationen austauschen und somit Bewusstsein und Interesse für neue Produkte erzeugen können. Zusätzlich bietet `IRPact` noch weitere Agententypen wie den `CompanyAgent` oder `PointOfSaleAgent`, welche Produkte herstellen bzw. zum Verkauf anbieten können oder den `PolicyAgent`, der politische Institutionen repräsentiert und simulationsweiten Einfluss haben kann. Weiterhin existieren Konzepte, welche die Produktwahrnehmung der Verbraucher modifizieren

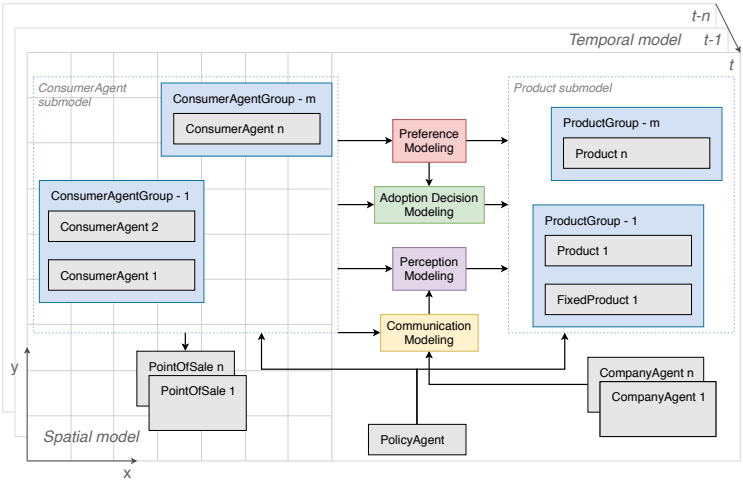

Abbildung 7.1.: Grundlegende Architektur von IRPact. Quelle: Johanning u. a., 2020.

können. In der Re-implementierung von IRPact wurde diese Möglichkeit der Wahrnehmung, sowie das Kommunikationsverhalten in einem Prozessmodell integriert und daraufhin modularisiert.

Design und Implementierung der Agenten

In Agentensimulationen gibt es eine Vielzahl verschiedener Agenten, welche sich in Bezug auf Eigenschaften, Fähigkeiten und Aufgaben unterscheiden. Technisch ist es häufig darüber hinaus nötig, sogenannte Systemagenten zu verwenden, die keine realen Akteure repräsentieren. Ihre Aufgaben beziehen sich auf die Steuerung und den korrekten Ablauf der Simulation bzw. das System.

Verbraucheragenten stellen die wichtigste Agentenkategorie innerhalb von IRPact dar. Analog zu realen Verbrauchern benötigen sie die Fähigkeit, Aktionen durchführen können, sowie sich mit anderen Agenten in Netz-

werken zu verknüpfen. Die wichtigste Aktion ist dabei die Kommunikation mit anderen Agenten.

Verbraucher als Agenten

Der Modellierungsansatz von IRPact folgt dem Paradigma der Objektorientierten Programmierung. Im Rahmen dessen wird den ConsumerAgents eine räumliche Verortung (Wohnsitz), sowie die Weitergabe von Informationen (Informationslegitimation) zugesprochen. In diesem Sinne gelten sie sowohl als SpatialAgents, als auch als InformationAgent. Dieser als Generalisierung bezeichnete Ansatz ist ein wichtiger Bestandteil der Objektorientierten Programmierung; er verleiht den ConsumerAgent beide Eigenschaften.

Soziale und lokale Netzwerke

Verbraucher sind in soziale Netzwerke eingebettet, welche auf Freundschaften, Familienzugehörigkeit, dem sozialen Milieu oder anderen Aspekten beruhen. Diese Netzwerke besitzen dabei einen großen Einfluss auf die Haltung der jeweiligen Individuen, weshalb sie in IRPact ebenfalls umgesetzt sind. Die Implementierung erfolgte über gerichtete Graphen, deren Verbindungen Kommunikationsmöglichkeiten repräsentieren. Kanten von Agent A nach Agent B werden so operationalisiert, dass A mit B eine Kommunikation initiieren kann. In Anbetracht dessen, dass der Graph gerichtet ist, kann jedoch B keine Kommunikation mit A starten, wenn nicht auch eine Kante von B nach A existiert. Das implementierte soziale Netzwerk ermöglicht somit Kommunikation zwischen Agenten.

Neben dem sozialen existiert auch ein räumlich-lokales Netzwerk. Die Idee basiert dabei auf der realen Sichtbarkeit der Agenten. Die durch ConsumerAgents repräsentierten Verbraucher besitzen über den oben genannten SpatialAgent eine räumliche Zuordnung. Durch das räumliche Modell können auf Basis dieser Informationen Entfernungen berechnet und eine Benutzer:innen-spezifische Sichtbarkeit definiert werden. Im Gegensatz zum sozialen Netzwerk ist das räumlich-lokale somit nicht explizit implementiert. Es entsteht implizit durch das räumliche Modell und die entsprechend zugeordneten Informationen der Agenten.

Kommunikation zwischen Agenten

Einer der wichtigsten Aspekte einer Agentensimulation ist die Fähigkeit der Kommunikation zwischen Agenten. Sie ermöglicht den direkten Wissens- und Informationsaustausch und bildet somit die Grundlage für die Verbreitung von Bewusstsein (über die Existenz betrachteter Produkte und deren Eigenschaften) und Interesse (an dem möglichen Erwerb eines Produktes) innerhalb der Simulation. In IRPact findet die Kommunikation dabei immer bilateral statt. Ein Agent kann zur selben Zeit immer nur mit exakt einem anderen Agenten kommunizieren. In diesem Zusammenhang erforderte der parallele Implementierungsansatz von IRPact und somit die potenziell parallele Ausführung von Kommunikationsereignissen besondere Achtsamkeit. Da ein Agent nur immer mit exakt einem anderen Agenten kommunizieren und nur eine begrenzte Anzahl an Aktionen ausführen kann, wurde der Ablauf der Kommunikation entsprechend koordiniert, da eine mangelhafte Implementierung bei parallelen Prozessen zu race conditions (deut. Wettlaufsituationen) führen kann, wodurch ein Agent mehr Aktionen durchführen könnte, als ihm zusteht. Um race conditions zu vermeiden, wurde ein synchronisierter Kommunikationsmechanismus implementiert. Dabei versetzt sich der Kommunikationsinitiator als erstes in den Kommunikationsmodus. Sollte der Initiator nun von einem anderen Agenten angesprochen werden, wird er das Gespräch ablehnen, da er sich bereits in einem befindet. Anschließend sucht er in seinem sozialen Umfeld zufällig nach einem potentiellen Kommunikationspartner. Sobald er diesen gefunden hat, wird eine Kommunikationsanfrage gesendet. Falls der Zielagent zu dem Zeitpunkt keine Aktion ausführt und noch freie Aktionen zur Verfügung hat, sendet dieser ein positives Feedback zurück und versetzt sich zudem selbst in den Kommunikationsmodus. Anschließend kann der Informationsaustausch durchgeführt werden. Zum Ende wird der Kommunikationsmodus beendet und die Agenten sind für weitere potentielle Aktionen verfügbar. Sollte der Initiator keinen validen Kommunikationspartner finden, gilt der Kommunikationsversuch als fehlgeschlagen und die Aktion wird erfolglos beendet.[13]

[13] Hierdurch tritt ein Verhalten äquivalent mit der Operation 'NOP' (keine Aktion) auf, da die Aktion hiermit aufgebraucht wird.

Prozessmodell

Das Prozessmodell in IRPact beschreibt die Ausgestaltung von Prozessen, insbesondere des Entscheidungsprozesses. Der Aufbau eines Prozessmodells kann komplex sein und aus mehreren Teilkomponenten bestehen. Ein solches kann integral (also das Modell als Ganzes beschreibend) oder modular aufgebaut sein (bei welchem Komponenten unabhängig voneinander nutzbar und miteinander verbunden sind). Ein integraler Ansatz hat dabei den Nachteil, dass Benutzer:innen das Prozessmodell nur als Einheit nutzen können. Optionen zur Aktivierung oder Deaktivierung von Funktionalitäten müssen individuell implementiert sein und können durch Benutzer:innen nicht nachträglich hinzugefügt werden. Um Flexibilität in der Gestaltung verschiedener Prozesse zu ermöglichen und Benutzer:innen zu ermächtigen, verschiedene Prozesse auch entwicklerunabhängig umzusetzen, wurde das ursprünglich integrale Prozessmodell zu einem modularen Prozessmodell weiterentwickelt. Den Benutzer:innen steht nun eine Vielzahl verschiedener Module zur Verfügung, welche zu komplexen Strukturen kombiniert werden können. Module unterscheiden sich dabei bezüglich ihres Ausgabetypes und lassen sich diesbezüglich gruppieren. Die folgende Tabelle legt die unterstützten Arten mit der jeweiligen Ausgabe dar.

Tabelle 7.1.: Darstellung der Modul- und Ausgabetypen in IRPact.

Modultyp	**Ausgabetyp**
Aktion	-
Berechnung	Zahlwert
Entscheidung	ja/nein
PVact	Phase

Aktionsmodule haben keine definierte Ausgabe und können als *finale* Handlung interpretiert werden. Die folgende Tabelle stellt für jeden Typ beispielhaft einen entsprechenden Kandidaten und seine Funktion vor.

Neben einer Ausgabe können Module auch Eingaben besitzen. Diese sind für jedes Module individuell definiert. So besitzt der *Addierer* eine Liste von Berechnungsmodulen als Eingabe, deren Werte er addiert und die

Tabelle 7.2.: Übersicht über die Modultypen in IRPact.

Modultyp	**Beispiel**	**Funktion**
Aktion	Kommunikator	Startet Kommunikation mit anderen Agenten
Berechnung	Addierer	Berechnung der Summe
Entscheidung	Grenzwerttest	Bestimmt, ob der Eingabewert über einem Grenzwert liegt
PVact	Adoptionsentscheider	Bestimmt Adoptionsstatus basierend auf dem Eingabewert

Summe als Ergebnis weitergibt. Die Verknüpfung von Modulen ist dabei immer unidirektional, wodurch die Modulstruktur als gerichteter Graph interpretiert werden kann.

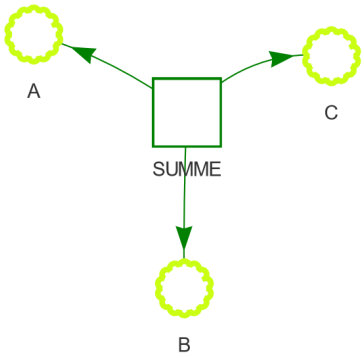

Abbildung 7.2.: Modulgraph für die Addition von drei Werten A, B und C. Eigene Darstellung.

Mit dieser Modularisierung können Benutzer:innen nun ein Prozessmodell erstellen und modulweise definieren, wo Daten herkommen, wie sie verarbeitet und anschließend final ausgewertet werden sollen. Äquivalent zum komponentenbasierten Ansatz von IRPact wurde das modulare

Prozessmodell so konzipiert, dass es leicht durch neue Module erweitert werden kann. Die Weboberfläche (vgl. dazu das Kapitel über die Benutzeroberfläche in diesem Band) ermöglicht eine Visualisierung konstruierter Prozessmodelle, um die Gesamtübersicht zu vereinfachen. Ein Beispiel dafür ist in Abbildung 7.2 mittels der Addition von drei Wert gegeben. Die Pfeilrichtung der Kanten vom Addierer **SUMME** zu den Modulen **A**, **B** und **C** zeigt die Richtung der Befehlsaufrufe. Der Informationsfluss findet entgegen der Pfeilrichtung statt. Der Addierer *fragt* bei den drei Modulen an und *erhält* die zu addierenden Werte. Die berechnete Summe kann nun an andere gekoppelte Module für weitere Analysen übermittelt werden.

Umsetzung und Instanziierung am Beispiel von PVact

Für die Verwendung von IRPact müssen Benutzer:innen eine parametrisierte Konfiguration aus den zur Verfügung stehenden Agententypen und Komponenten erstellen. Eine solche Konfiguration wird im Kontext von IRPact als Szenario bezeichnet. Im Rahmen des Projektes ist auf Basis der Fallstudien (Kapitel Fallstudien), dem Entscheidungsverhalten (Kapitel Entscheidungsverhalten Energieakteure) sowie der empirisch basierten Haushaltsagenten (Kapitel Empirische Verankerung) das PVact Szenario konfiguriert und parametrisiert worden.

Konfiguration und Parametrisierung

Die Agentengruppen im PVact Szenario wurden entsprechend den Sinus-Milieus erstellt und parametrisiert (Vergleich Kapitel Empirische Verankerung). Es existiert nur ein PV Produkt, dessen Wahrnehmung trivial ist.[14] Die Adoption erfolgt direkt, wodurch Hersteller und Verkäufer wegfallen. Das zeitliche Modell basiert auf wöchentlichen diskreten Zeitschritten. Somit besteht ein Simulationsjahr, welches am 01.01. beginnt, aus 52 Schritten.

[14] Dieses bedeutet, dass Produktattribute direkt beobachtet werden können (also die Wahrnehmung dieser den tatsächlichen Produkteigenschaften entspricht). Weiterhin sind sich alle ConsumerAgents der Eigenschaften des Produktes bewusst.

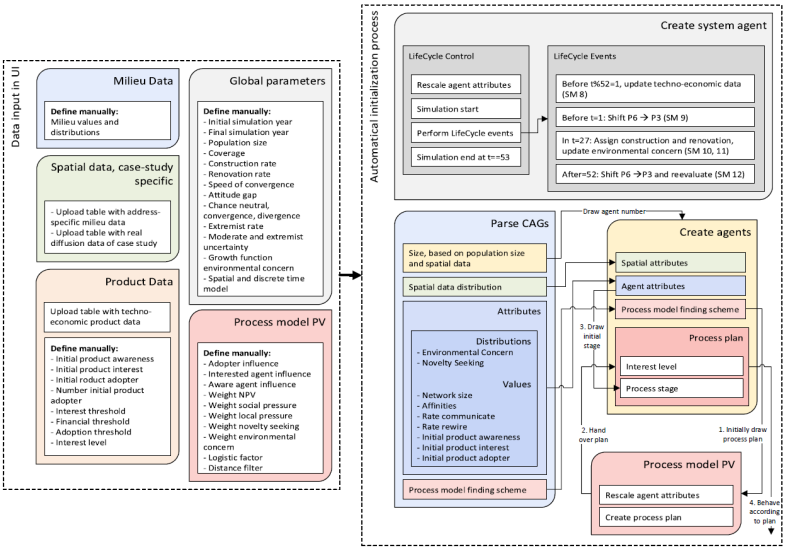

Abbildung 7.3.: Instanziierungsprozess von IRPact am Beispiel von PVact. Eigene Darstellung.

Eine Gesamtübersicht über alle Eingaben ist auf der linken Seite von Abbildung 7.3 dargestellt. Für die Umsetzung des Prozessmodells aus dem Kapitel über das Entscheidungsverhalten von Energieakteuren wurden die einzelnen Phasen (Vergleich Abbildung 3.1) als Module gekapselt und implementiert. Dabei lag das Hauptaugenmerk auf einer möglichst feingranularen Umsetzung. Bei der Modularisierung hat sich gezeigt, dass die Bewusstseins- und Machbarkeitsphase in ihrer Funktionalität kompakt sind und als eigenständige Module implementiert werden können. Im Gegensatz dazu wurde die Evaluierungsphase sehr stark modularisiert. Dies hatte zum Ziel, die zugrunde liegenden mathematischen Beschreibungen komplett über Module zu realisieren. Für die grundlegende Phasensteuerung wurde ein spezielles Verwaltungsmodul implementiert. Insgesamt besteht der resultierende Modulgraph aus über 30 Modulen, wobei ein Großteil auf die Evaluierungsphase entfallen.

Initialisierungsprozess

Mit dem Start und der Szenarioübergabe an IRPact wird ein komplexer Initialisierungsprozess gestartet, welcher am Beispiel von PVact in der rechten Hälfte von Abbildung 7.3 dargestellt ist. Im ersten Schritt wird das übergebende Szenario eingelesen und validiert, um fehlerhafte Konfigurationen vor dem eigentlichen Simulationsstart erkennen zu können. Im Anschluss an die erfolgreiche Validierung werden alle benötigten Agenten und Komponenten auf Basis der übergebenden Parametrisierung instanziiert. Während dieses Prozesses können die einzelnen Instanzen auf Basis verschiedener Eingabedaten konfiguriert werden. Die Konsumentengruppen werden über die im Szenario direkt definierten Milieudaten erstellt. Mittels der tabellarisch definierten Adressinformationen erhalten die Verbraucher ihre räumlichen Koordinaten sowie Informationen zu Dachausrichtung und -neigung. Äquivalent erfolgt die Instanziierung des PV Produktes, des Prozessmodells sowie weiterer Teilmodelle. Im Zusammenhang mit dem Prozessmodell wird für jeden Agenten ein sogenannter Prozessplan erstellt. Bei diesem handelt es sich um die Umsetzung des *Planes* aus dem BDI-Modell. Der Prozessplan beschreibt für jeden Agenten, welches Prozessmodell für die Zielerfüllung (= Produktadoption) verwendet werden soll. Neben den durch Benutzer:innen im Szenario definierten Agenten instanziiert IRPact auch noch einen speziellen Systemagenten. Dieser überwacht und kontrolliert den Verlauf, indem er für den Start und die Beendigung der Simulation sowie den chronologisch korrekten Ablauf von zeitlichen Ereignissen verantwortlich ist.

Mit Beendigung des Initialisierungsprozesses erhält der Systemagent das Signal zum Starten der Simulation und es beginnt der diskrete Ablauf. In jedem Zeitschritt durchlaufen die Verbraucher ihren Prozessplan und führen, basierend auf dem Entscheidungsverhalten (Vergleich Abbildung 3.1), entsprechende Aktionen durch. Nach Beendigung des letzten validen Zeitschritts beendet der Systemagent die Simulation und die Phase der Auswertung und Nachbearbeitung beginnt. Im Zuge dieser Nachanalyse werden verschiedene Statistiken über das Adoptionsverhalten sowie die Verläufe des Produktinteresses und der Attribute der Agenten erstellt. Diese werden für die Agentenpopulation als Ganzes oder aufgeschlüsselt nach Milieus bzw. Quantilen visualisiert. Auf Basis der Ergebnisse werden abschließend die konfigurierten Abbildungen erzeugt.

Danksagung

Dieser Beitrag wurde finanziert durch das Projekt „Smart Utilities and Sustainable Infrastructure Change“ (Antragsnummer 100378087 (SAB)).

Diese Maßnahme wird mitfinanziert durch Steuermittel auf der Grundlage des vom Sächsischen Landtag beschlossenen Haushaltes.

DOI: https://doi.org/10.30819/5413.07

Literatur

Adam, C. und B. Gaudou (2016). »BDI agents in social simulations: a survey«. In: *The Knowledge Engineering Review* 31.3, S. 207–238.

Baptista, M. L., C. Martinho, F. Lima, P. A. Santos und H. Prendinger (2013). »A Business Simulation with an Agent-Based Deliberative Model of Consumer Behaviour«. In: *International Conference on Games and Learning Alliance*. Springer, S. 215–223.

Baptista, M. L., C. R. Martinho, F. Lima, P. A. Santos und H. Prendinger (2014). »An agent-based model of consumer behavior based on the BDI architecture and neoclassical theory«. In: *Developments in Business Simulation and Experiential Learning: Proceedings of the Annual ABSEL conference*. Bd. 41.

Bratman, M. (1987). *Intention, plans, and practical reason*. Bd. 10. Harvard University Press Cambridge, MA.

Bulumulla, C., J. Chan und L. Padgham (2018). »Enhancing diffusion models by embedding cognitive reasoning«. In: *2018 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*. IEEE, S. 744–749.

Johanning, S., F. Scheller, D. Abitz, C. Wehner und T. Bruckner (2020). »A modular multi-agent framework for innovation diffusion in changing business environments: conceptualization, formalization and implementation«. In: *Complex Adaptive Systems Modeling* 8.1. DOI: 10.1186/s40294-020-00074-6.

Norling, E. (2004). »Folk psychology for human modelling: Extending the BDI paradigm«. In: *Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems-Volume 1*. IEEE Computer Society, S. 202–209.

Rao, A. S., M. P. Georgeff u. a. (1995). »BDI agents: from theory to practice.« In: *ICMAS*. Bd. 95, S. 312–319.

Shendarkar, A., K. Vasudevan, S. Lee und Y.-J. Son (2008). »Crowd simulation for emergency response using BDI agents based on immersive virtual reality«. In: *Simulation Modelling Practice and Theory* 16.9, S. 1415–1429.

Simon, H. A. (1955). »A behavioral model of rational choice«. In: *The quarterly journal of economics* 69.1, S. 99–118.