



Hochschule für Angewandte Wissenschaften Hamburg
Hamburg University of Applied Sciences

Ausarbeitung Anwendungen 2
SoSe 2012
Stefan Münchow

Eine flexible Architektur für die Agenten einer
Fußgängersimulation

Inhaltsverzeichnis

1	Einleitung	1
1.1	Ziele	1
1.2	Aufbau der Arbeit	2
2	Related Work	2
2.1	ACT-R	2
2.1.1	Aufbau	2
2.1.2	Repräsentation von Wissen	3
2.1.3	Bewertung & Relevanz	4
2.2	C4	5
2.2.1	Aufbau	5
2.2.2	Working Memory & Blackboard	5
2.2.3	Aktionen	6
2.2.4	Bewertung & Relevanz	7
2.3	TOK	7
2.3.1	Aufbau	8
2.3.2	Pläne & Ziele	8
2.3.3	Emotionen & Sozialverhalten	9
2.3.4	Bewertung & Relevanz	9
3	Eigenes Vorgehen	10
3.1	Agentenarchitektur	10
3.2	Sensitivitätsanalyse	10
4	Ausblick	11
	Literatur	12

1 Einleitung

Im Projekt WALK der Hochschule für angewandte Wissenschaften Hamburg wird eine agentenbasierte Fußgängersimulation entwickelt. Diese soll die Simulation von Menschenmengen in Evakuierungsszenarien unter Berücksichtigung von individuellem Verhalten ermöglichen. Im Gegensatz zu *makroskopischen Ansätzen*, bei denen eine Menschenmenge als eine Entität (als „Ganzes“) betrachtet wird, sollen Massenphänomene bei diesem Ansatz als Emergenz aus den individuellen Entscheidungen der einzelnen Individuen entstehen. Diese Betrachtungsweise, bei der einzelne Individuen in der Simulation unterschieden werden, wird auch als *mikroskopischer Ansatz* bezeichnet (vgl. [RiMEA 2009](#), S. 6).

Um die individuellen Entscheidungen einzelner Agenten möglichst realitätsnah abzubilden, müssen sozio-psychologische Faktoren, wie z.B. Persönlichkeitseigenschaften, Emotionen und soziale Normen, berücksichtigt werden. Allerdings herrscht in der Psychologie und den Sozialwissenschaften Uneinigkeit darüber, wie bestimmte Verhaltensweisen innerhalb einer Gefahrensituation entstehen. Besonders umstritten ist hier, wie stark unterschiedliche Faktoren, z.B. Emotionen, eine Entscheidung beeinflussen. Eine frühe Festlegung auf bestimmte Modelle zur Abbildung von Entscheidungen in den Agenten ist also nicht sinnvoll möglich oder schränkt die Verallgemeinerbarkeit einer Lösung deutlich ein.

Im Bereich der Massensimulation existieren bereits viele Lösungsansätze, diese sind jedoch meist an spezielle Szenarien gebunden und nicht verallgemeinerbar (vgl. [Zhou u. a. 2010](#), S. 8, 29 ff.). Eines der Hauptziele von WALK besteht darin, einen allgemeineren Lösungsansatz bereitzustellen, der innerhalb verschiedener Szenarien eingesetzt werden kann. Um dies zu erreichen, soll das System zunächst als Experimentierumgebung verwendet werden, um verschiedene Entscheidungsmodelle und sozio-psychologische Theorien experimentell zu untersuchen und ihre Auswirkung auf das Simulationsergebnis zu beurteilen. Auf diese Weise kann die Eignung verschiedener Modelle für die Simulation von Menschenmengen analysiert und das Ergebnis der Analyse zur Entwicklung eines realistischen Simulationssystems herangezogen werden.

1.1 Ziele

Um die experimentelle Erforschung verschiedener Entscheidungsmodelle und sozio-psychologischer Theorien mithilfe der WALK-Agenten zu ermöglichen, müssen diese mit einer flexiblen Agentenarchitektur ausgestattet werden, welche die Umsetzung verschiedener Modelle ermöglicht. In einer Vielzahl der wissenschaftlichen Arbeiten zum Thema Massensimulation existieren nur wenige, die sich ausführlich mit der Gesamtarchitektur der Agenten beschäftigen. Stattdessen werden meist einzelne Komponenten oder Details, z.B. die Modellierung von Emotionen, sehr detailliert erläutert.

Das Ziel des Autors besteht zunächst darin, eine flexible Agentenarchitektur für das Simulationssystem WALK zu schaffen. Hierzu ist eine komponentenorientierte Vorgehensweise von zentraler Bedeutung. Komponenten zur Erzeugung von Emotionen, zur Abbildung von Persönlichkeitseigenschaften und zur Auswahl von Zielen müssen so gestaltet sein, dass sie einfach ausgetauscht werden können. Dies ermöglicht den Einsatz des Systems als „Framework“ für weitere Experimente.

1.2 Aufbau der Arbeit

In Abschnitt 2 werden einige bekannte Agentenarchitekturen beschrieben und für WALK interessante Konzepte erläutert. Abschnitt 3 gibt einen Überblick über den aktuellen Stand der Entwicklung der WALK-Agenten und beschreibt den möglichen Einsatz der geschaffenen Experimentierumgebung. In Abschnitt 4 wird ein Ausblick auf das weitere Vorgehen des Autors gegeben.

2 Related Work

In diesem Abschnitt werden verschiedene Agentenarchitekturen und ihre Besonderheiten vorgestellt. Dabei werden nicht alle Aspekte beschrieben, sondern nur die für die WALK-Agenten relevanten Konzepte.

2.1 ACT-R

ACT-R steht für *Adaptive Control of Thought - Rational*. Es wurde in den 1990er Jahren an der Carnegie Mellon University in Pittsburgh entwickelt und ist in [Anderson u. a. \(2004\)](#) genau beschrieben.

2.1.1 Aufbau

Der Aufbau der Architektur ist in Abbildung 1 zu sehen. ACT-R verwendet sogenannte *Module* zur Nachbildung der unterschiedlichen Funktionen des menschlichen Gehirns. So werden in der Originalarbeit von [Anderson u. a.](#) vier mindestens erforderliche Module vorgeschlagen: Das *Visual Module* für die visuelle Wahrnehmung, das *Intentional Module* für die Auswahl eines Ziels, das *Declarative Module* zur Abbildung eines Langzeitgedächtnisses und das *Manual Module* zur Modellierung der Aktoren eines Agenten. Jedes dieser Module ist mit einem eigenen Zwischenspeicher verbunden, in den es Informationen schreiben und aus dem

es Informationen lesen kann. So kann z.B. das Visual Module aktuelle Wahrnehmungen zur weiteren Verarbeitung in den zugehörigen Zwischenspeicher schreiben und das Manual Module Befehle zur Ausführung aus seinem Zwischenspeicher lesen. Alle Module in ACT-R sind nebenläufig und können jederzeit ihre Zwischenspeicher verändern oder auslesen.

Den Kern der Architektur bildet das *Produktionssystem*. Es bildet die Steuerung des Agenten, indem es die einzelnen Module steuert und das vorhandene Wissen verarbeitet. Das Produktionssystem überwacht ständig die Zwischenspeicher der Module und kann in ihrer Belegung mithilfe eines Matching-Verfahrens Muster erkennen. Durch das Matching werden *Produktionsregeln* aus dem *prozeduralen Gedächtnis*, einem Speicher für das Wissen über mögliche Aktionen des Agenten, ausgewählt. Zu einem Zeitpunkt können beliebig viele dieser Regeln aktiviert werden, wobei die Nützlichkeit jeder Regel mithilfe einer Utility-Funktion bewertet wird. Die Utility-Funktion berücksichtigt die Wahrscheinlichkeit, dass das angestrebte Ziel erreicht wird, den „Wert“ des Ziels und die Kosten der Aktion (meist in Form von Zeitaufwand). Anschließend wird durch eine weitere Funktion genau eine Regel ausgewählt, die dann als nächstes angewandt wird.

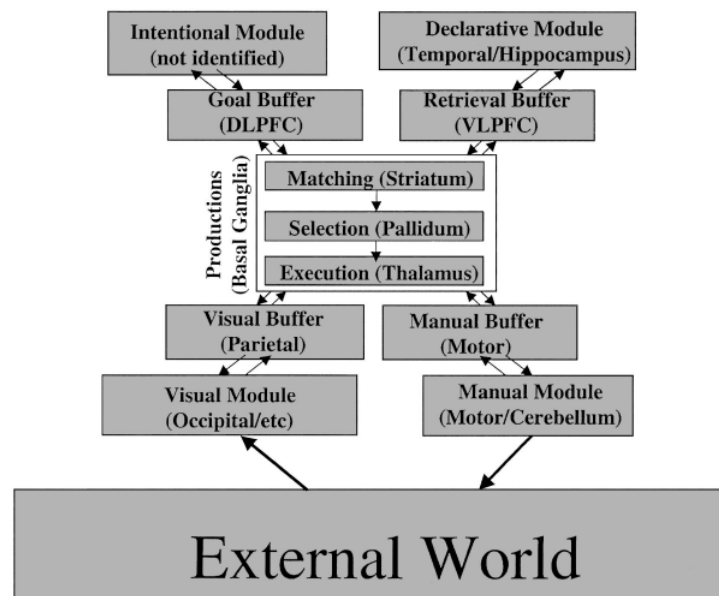


Abbildung 1: Die ACT-R-Architektur

2.1.2 Repräsentation von Wissen

Der ursprüngliche Fokus von ACT-R lag auf der Modellierung des Gedächtnisses. Das Abrufen von explizitem und implizitem Wissen, das Lernen und Vergessen von Informationen

sowie semantische Zusammenhänge können mithilfe des Declarative Module sehr detailliert abgebildet werden (vgl. [Anderson u. a. 2004](#), S. 7 - 10). Alle Informationen werden als *Chunks* im deklarativen Gedächtnis abgelegt. Ein Chunk repräsentiert grundsätzlich einen Fakt, der mehrere Konzepte miteinander verbindet, z.B. „4 plus 8 ergibt 12“. Die Beziehungen zwischen den Konzepten sind dabei mit Gewichten versehen, die durch Lernen verändert werden können. Die Produktionsregeln im prozeduralen Gedächtnis haben die Form einfacher Wenn-Dann-Regeln. Die Wenn-Teile der Regeln werden vom Produktionssystem mit dem aktuellen Zustand der Modul-Zwischenspeicher verglichen, wodurch alle für die aktuelle Situation passenden Regeln ausgewählt werden.

Das Intentional Module ist dafür verantwortlich, dass die ausgeführten Aktionen dem aktuellen Ziel dienen. Wie dies im Detail umgesetzt wird, ist in [Anderson u. a. \(2004\)](#) nicht im Detail beschrieben. Es wird jedoch erwähnt, dass ein Ziel als Menge von Subzielen repräsentiert wird.

Es existiert eine weitere, sehr ähnliche Agentenarchitektur namens Soar. Diese verfolgt einen ähnlichen Ansatz, jedoch wurden einige Details anders umgesetzt oder unterschiedlich stark priorisiert. Genauere Informationen hierzu sind in [Laird u. a. \(1987\)](#) nachzulesen.

2.1.3 Bewertung & Relevanz

ACT-R wurde mit dem Ziel entwickelt, die menschlichen Denkprozesse möglichst detailliert abzubilden. Ein besonderes Augenmerk liegt hier auf dem Gedächtnis des Agenten und der Abrufbarkeit von Wissen. Die damit verbundenen Abläufe werden sehr detailliert abgebildet, wobei die Erfahrung des Agenten und die bisherige Nützlichkeit von Informationen berücksichtigt werden. Die Abbildung des Gedächtnisses in diesem Detailgrad würde im Rahmen einer Massensimulation wie WALK zu weit gehen. Im Kontext einer Fußgängersimulation werden üblicherweise kurze Zeitdauern von mehreren Minuten oder maximal Stunden simuliert. Dabei kann davon ausgegangen werden, dass die Agenten, abgesehen von Umgebungsinformationen, bestimmtes abrufbares Wissen besitzen, welches sich im Laufe der Simulation kaum verändert.

Sehr nützlich erscheint hingegen das Konzept, das deklarative und prozedurale Gedächtnis voneinander zu trennen. Dies könnte besonders im Hinblick auf die angestrebte Flexibilität der WALK-Agenten einen entscheidenden Vorteil bieten. Auch sind die Konzepte der Wissensrepräsentation in ACT-R sehr ausgeprägt und könnten im Ansatz in den WALK-Agenten adaptiert werden. So werden für die Wissensbasis ebenso Fakten, ähnlich den Chunks aus ACT-R, benötigt. Die Mechanismen zur Auswahl einer Produktionsregel sind sehr durchdacht und formal ausgedrückt und könnten als Ansatz für eine Aktions- oder Planauswahl innerhalb der Agenten dienen.

2.2 C4

Die C4-Architektur wurde am MIT Media Lab mit dem Ziel entwickelt, die Intelligenz eines durchschnittlichen Hundes abzubilden. Sie ist in [Burke u. a. \(2001\)](#) genau beschrieben.

2.2.1 Aufbau

Die Grundidee der Architektur basiert auf einer Idee aus [Minsky \(1986\)](#): Das menschliche Denken lässt sich in verschiedene Funktionen aufteilen, die von unabhängigen „Agenten“ durchgeführt werden. Jeder dieser Agenten ist für sich genommen eher einfach, komplexes Verhalten entsteht durch ihre Interaktion.

Die Architektur ist in [Abbildung 2](#) dargestellt. Die Verarbeitung innerhalb des Agenten geschieht hier, anders als bei ACT-R, sequentiell von oben nach unten. Zunächst nimmt der Agent über seine Sensoren die Umgebung wahr. Die aufgenommenen „Rohdaten“ werden vom *Perception System* in Objekte umgewandelt, die der Agent intern zur Findung seiner Entscheidungen verwenden kann. Dazu durchläuft jede Wahrnehmung einen *Percept Tree*, der von der Wurzel zu den Blättern immer spezifischer werdende Objekte enthält. Die aktuelle Wahrnehmung wird mit jedem Knoten verglichen und eine *Match Probability* und ein *Confidence*-Wert errechnet. Überschreiten diese Werte einen bestimmten Schwellenwert, so wird ein dem Knoten entsprechendes Objekt im Gedächtnis des Agenten erzeugt. Der Entscheidungszyklus entspricht mit identifizierbaren Wahrnehmungs-, Aktionsauswahl- und Ausführungsphasen eher den klassischen Agentenarchitekturen (siehe z.B. [Wooldridge \(2002\)](#)).

2.2.2 Working Memory & Blackboard

Das Gedächtnis des Agenten wird durch einen *Working Memory* modelliert. Er enthält alle aktuellen Wahrnehmungen des Agenten in Form von *PerceptMemory*-Objekten und wird vom *Action System* für die Zielauswahl und Planung verwendet. Hat das Action System eine Aktion ausgewählt, so werden die für die Ausführung relevanten Informationen auf ein internes *Blackboard* geschrieben. Sie werden vom *Navigation System* und dem *Motor System* von dort gelesen und zur Ausführung der nächsten Aktion verwendet. Durch die Kommunikation über den Working Memory und das Blackboard ist die Architektur flexibel und leicht erweiterbar.

Eine weitere wichtige Komponente ist *Proprioception System*: Es handelt sich hierbei ebenfalls um eine Art internes Blackboard, auf dem interne Ereignisse des Agenten veröffentlicht werden können. Sie werden vom *Sensory System* analog zu externen Ereignisse verarbeitet.

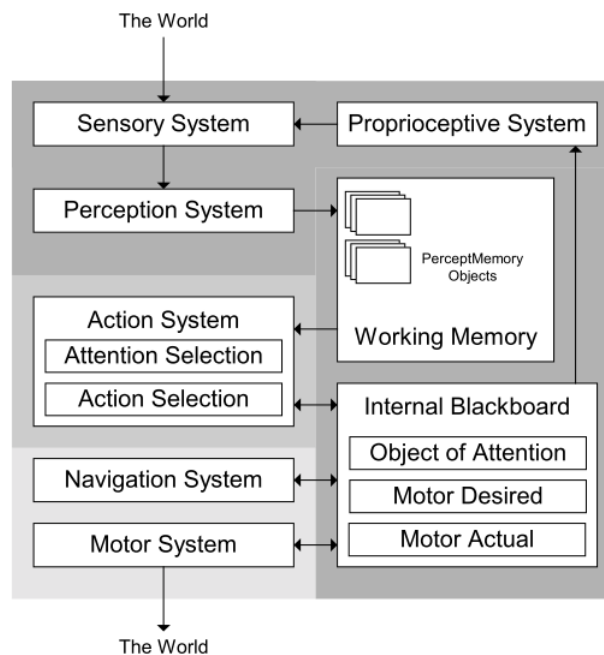


Abbildung 2: Die C4-Architektur

2.2.3 Aktionen

Zur Repräsentation von Aktionen verwendet die C4-Architektur sogenannte *Action Tuples*, die aus jeweils vier Teilen bestehen:

- **TriggerContext:** Gibt Vorbedingungen an, unter denen eine Aktion ausgeführt wird.
- **Action:** Beschreibt die auszuführende Aktion.
- **ObjectContext:** Enthält ein Objekt oder einen anderen Agenten mit oder an dem die Aktion ausgeführt wird.
- **DoUntilContext:** Gibt Bedingungen an, die erfüllt sein müssen, damit die Aktion beendet wird.

Die jeweils nächste Aktion wird ausgewählt, indem unter Verwendung des TriggerContexts ein Relevanzwert für jede mögliche Aktion berechnet wird. Dieser Wert ergibt sich aus dem aktuell bekannten Umweltzustand im Working Memory. Zudem erhält jede Aktion einen *Intrinsic Value*, der angibt, wie günstig eine Aktion ist. Mithilfe von Reinforcement Learning kann dieser Wert für eine Aktion über die Zeit angepasst werden, so dass Aktionen mit einem günstigen Ausgang einen höheren Intrinsic Value erhalten. Der Wert einer Aktion ergibt sich aus dem Produkt der Relevanz und dem Intrinsic Value. Es wird stets die Aktion mit dem höchsten Wert ausgeführt.

Die simultane Ausführung mehrerer Aktionen und der wechselseitige Ausschluss bestimmter Aktionen werden über *Action Groups* realisiert: Sollen mehrere Aktionen simultan ausgeführt werden, so werden alle Aktionstupel in der Gruppe durchlaufen und jede Aktion mit einem echt positiven Wert aktiviert. Ist ein wechselseitiger Ausschluss erforderlich, so wird aus einer Gruppe die Aktion mit dem höchsten Wert ausgewählt. Ist keine der Aktionen aktiviert (da der Umweltzustand dies nicht zulässt), so wird eine Aktion zufällig aus einer Menge von Aktionen ausgewählt.

2.2.4 Bewertung & Relevanz

Die C4-Architektur bietet ein durchdachtes Gesamtkonzept für die Entwicklung eines Agenten. Die Funktionsweise der einzelnen Komponenten ist sehr genau beschrieben, was eine Umsetzung erleichtert. Die Architektur ist in Komponenten mit klaren Verantwortlichkeiten untergliedert, genauso wie es für die WALK-Agenten erstrebenswert ist.

Die Verwendung eines „Kurzzeitgedächtnisses“ und eines internen Blackboards, in denen alle für die Entscheidungsfindung relevanten Informationen liegen, fördert Flexibilität und Erweiterbarkeit. Die Kategorisierung von Wahrnehmungen mithilfe eines PerceptionTrees ermöglicht die Abbildung der nicht immer zuverlässigen Wahrnehmung des Menschen.

Das Konzept der Action Tuples scheint mächtig genug zu sein, um viele verschiedene Verhaltensweisen abzubilden. Die Bestandteile der Tupel sind leicht nachvollziehbar und vermitteln eine Vorstellung davon, welche Informationen zur Aktionsauswahl erforderlich sind. Zwar sind sozio-psychologische Faktoren, wie z.B. Emotionen, im Modell nicht direkt vorgesehen, jedoch wurde in [Klingenberg \(2010\)](#) gezeigt, dass eine Erweiterung, z.B. um ein Emotionsmodell, sehr einfach möglich ist.

2.3 TOK

TOK wurde in den 1990er Jahren an der Carnegie Mellon University, Pittsburgh, in der Oz Group entwickelt und erstmals in [Bates u. a. \(1994\)](#) veröffentlicht. Die Oz Group erforschte die Möglichkeiten, virtuelle Welten mit *Broad Agents*, d.h. Agenten mit facettenreichem und realistischem Verhalten, zu schaffen. Als einzige der hier vorgestellten Architekturen enthält TOK von vornherein ein vollständiges Konzept zur Abbildung von Emotionen und Sozialverhalten.

2.3.1 Aufbau

Die Architektur ist in Abbildung 3 dargestellt. Sie besteht im wesentlichen aus den drei Komponenten *Sensory Routines*, *Hap* und *Em*. Die Sensory-Routines-Komponente bildet zusammen mit dem *Integrated Sense Modul (ISM)*, welches die Wissensbasis des Agenten darstellt, eine Komponente. *Hap* ist für die Planung und Zielauswahl des Agenten verantwortlich. Es steht in Wechselwirkung mit *Em*, der Komponente zur Abbildung von Emotionen und Sozialverhalten.

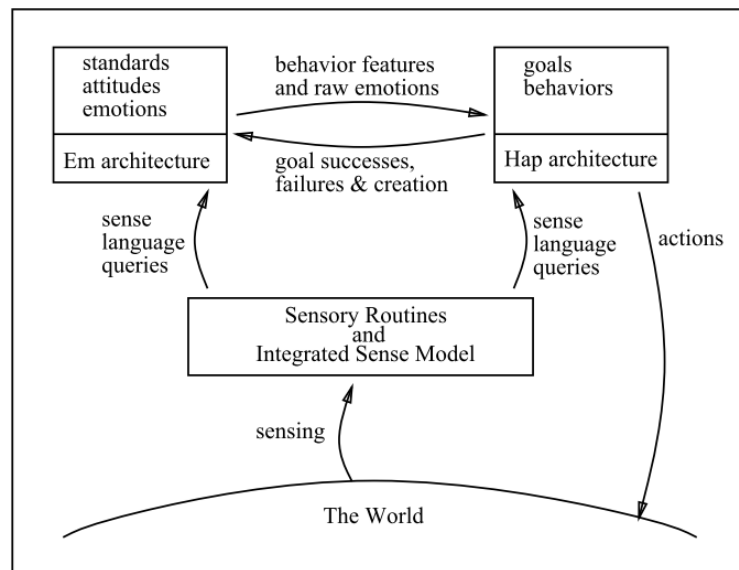


Abbildung 3: Die TOK-Architektur

2.3.2 Pläne & Ziele

Alle Entscheidungen des Agenten basieren auf dem Wissen des Agenten, welches im ISM abgelegt ist. Das ISM enthält einen topologischen Graphen der Umwelt, in dem alle bekannten Objekte und ihre Beziehungen gespeichert werden. Mögliche Beispiele für Objektbeziehung sind „Glas X steht auf dem Tisch Y“ oder „Tisch Y steht in Raum Z“.

Die Hap-Komponente wählt auf Basis dieses Wissens und eventuell vorhandenen Emotionen ein neues Ziel aus. Ziele besitzen in TOK einen eindeutigen Namen und beliebig viele Parameter, z.B. `goto <object>`. Wird das `goto`-Ziel aktiviert, dann wird der Parameter `<object>` mit einem Wert belegt. Jedes Ziel ist mit einem *Success Test* versehen. Dieser wird in jedem Zeitschritt ausgewertet und gibt an, ob ein Ziel erreicht wurde. Der Agent besitzt zudem eine Menge an vordefinierten Plänen, die jeweils einem Ziel zugeordnet sind und

aus einer Sequenz von Subzielen bestehen. Pläne besitzen eine *Context Condition*, welche die Erfüllbarkeit eines Plans angibt. Ein Plan, dessen Context Condition nicht erfüllt ist, wird nicht weiter verfolgt, da er unmöglich erfolgreich abgeschlossen werden kann.

2.3.3 Emotionen & Sozialverhalten

Die Hap-Komponente meldet (Mis-)Erfolge bei der Ausführung von Plänen an die Em-Komponente. Diese modifiziert die aktuelle Emotion des Agenten unter Verwendung des Emotionsmodells aus Ortony u. a. (1988). Dazu nutzt sie die von Ortony u. a. vorgeschlagenen *Standards* und *Attitudes*. Standards stellen so etwas wie Moralvorstellungen dar, welche die Richtigkeit einer Handlung bewerten. Attitudes beziehen sich auf Objekte oder andere Agenten und bestimmen, ob ein Agent anderen Objekten oder Agenten gegenüber positiv oder negativ eingestellt ist.

Die Em-Komponente generiert aus einer Emotion und den aktuellen Zielen des Agenten *Behavioral Features*. Diese stellen eine Abstraktion zwischen Emotionen und Verhalten dar, z.B. kann sowohl aus Wut als auch aus Angst aggressives Verhalten resultieren. In diesem Fall würde die Em-Komponente aus einer der beiden Emotionen das Behavioral Feature „Aggressive“ generieren. Dieses wird an die Hap-Komponente übergeben und beeinflusst dort die Zielauswahl und die Ausführung von Plänen.

2.3.4 Bewertung & Relevanz

TOK bietet, wie die C4-Architektur, ein umfassendes Konzept für alle relevanten Funktionen des Agenten. Es ist die einzige hier vorgestellte Architektur, die Emotionen und Sozialverhalten von vornherein vorsieht. Insgesamt wirkt die Struktur von TOK starrer als die der bisher vorgestellten Architekturen. Die Komponenten sind nicht so klar voneinander abgegrenzt wie z.B. in der C4-Architektur. Die grundsätzliche Struktur von TOK ist nicht so modular, wie es für die WALK-Agenten erforderlich ist.

Zwei für die WALK-Agenten nützliche Konzepte sind die vorgestellten Umsetzungen von Zielen und Plänen. Die Verwendung einer festen Menge von Zielen, die entsprechend einer Situation parametrisiert werden, eignet sich für die WALK-Simulation, da die Menge möglicher Ziele bekannt ist und sich grundsätzlich nicht verändert. Durch die Repräsentation von vordefinierten Plänen durch Subziele reduziert sich die Komplexität auf zwei wesentliche Abstraktionen, was die Verständlichkeit der Lösung fördert. Zudem könnte sich eine dynamische Berechnung von Plänen in jedem Agenten negativ auf die Performanz des Systems auswirken.

3 Eigenes Vorgehen

In diesem Abschnitt wird der aktuelle Stand der Entwicklung der WALK-Agenten beschrieben und ein Ausblick auf die Nutzung der zu schaffenden Experimentierumgebung gegeben.

3.1 Agentenarchitektur

Es soll eine Agentenarchitektur entwickelt werden, die den menschlichen Entscheidungsfindungsvorgang optimal abbildet. Dieser besteht nach [Kuligowski \(2009\)](#) aus vier Phasen:

1. Wahrnehmung der Umgebung und der Situation.
2. Bewertung der aktuellen Situation. In diesem Moment entstehen Emotionen und Persönlichkeitseigenschaften bestimmen maßgeblich die spätere Entscheidung.
3. Treffen der Entscheidung darüber, welche Aktion als nächstes ausgeführt wird. Auch die Planung von Teilschritten geschieht in dieser Phase.
4. Die Entscheidung wird in eine konkrete Handlung umgesetzt.

Diese vier Phasen finden sich in der Architektur der Agenten als Komponenten wieder. Sie werden sequentiell nacheinander ausgeführt und enthalten jeweils Subkomponenten für jeden relevanten Aspekt der entsprechenden Phase. Die Architektur ist in [Abbildung 4](#) dargestellt. Ihre grundsätzlichen Konzepte sowie die Gesamtarchitektur von WALK werden in [Münchow u. a. \(2012\)](#) genauer erläutert.

3.2 Sensitivitätsanalyse

Durch die Realisierung einer flexiblen Agentenarchitektur wird eine Experimentierumgebung geschaffen, die einen Vergleich verschiedener Modelle und ihrer Auswirkungen auf die Simulationsergebnisse ermöglicht. Das nächste Ziel besteht darin, eine Sensitivitätsanalyse bezogen auf einzelne sozio-psychologische Faktoren durchzuführen. In dieser könnten folgende Aspekte untersucht werden:

- Vergleich der Simulationsergebnisse mit und ohne Berücksichtigung bestimmter Faktoren, z.B. Persönlichkeitseigenschaften der Agenten.
- Vergleich unterschiedlicher Modelle für einen bestimmten Faktor, z.B. Implementierung und Vergleich unterschiedlicher Emotionsmodelle in den Agenten.
- Untersuchung von Wechselwirkungen verschiedener Faktoren, z.B. Sozialverhalten und Emotionen.

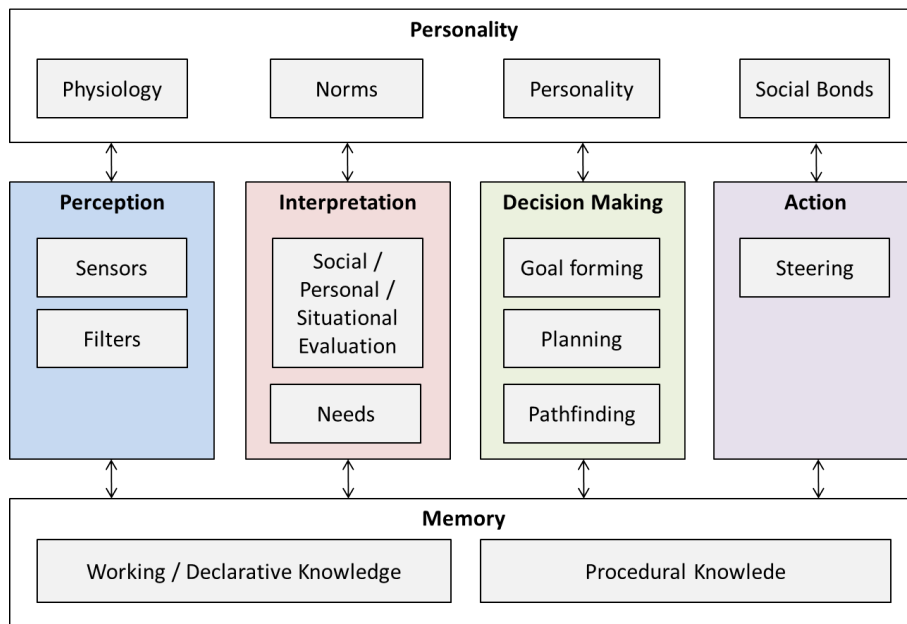


Abbildung 4: Die WALK-Agentenarchitektur

Das Ergebnis dieser Analyse ermöglicht allgemeine Aussagen über die Bedeutung verschiedener sozio-psychologischer Faktoren für den Realismus der Simulation. Die konkrete Auswahl geeigneter Modelle geschieht dadurch nicht willkürlich, sondern auf Basis der erzielten Ergebnisse. Zudem wird verhindert, dass ein Modell gewählt wird, welches sich nur für ausgewählte Szenarien eignet.

4 Ausblick

In dieser Arbeit wurden bestehende Agentenarchitekturen vorgestellt und die für die WALK-Agenten nutzbaren Konzepte erläutert. Die in Abschnitt 3 vorgestellte Architektur dient als Ausgangspunkt für die weitere Entwicklung und greift einige der vorgestellten Konzepte auf. Das Verhalten der Simulation wird mit realen Szenarien verglichen und die Agenten in mehreren Schritten iterativ verfeinert.

Sobald sichergestellt ist, dass die Simulation ohne die Berücksichtigung sozio-psychologischer Faktoren dem erwarteten Verhalten entspricht, werden ausgewählte Faktoren prototypisch implementiert und im bestehenden System getestet. Das Hauptziel soll hierbei die beschriebene Sensitivitätsanalyse bilden. Ein interessanter Aspekt, zu dem ausreichend empirische Daten existieren und der eine unmittelbare Auswirkung auf das Verhalten der Simulation hat, ist z.B. die Berücksichtigung sozialer Gruppen.

Literatur

- [Anderson u.a. 2004] ANDERSON, John R. ; BOTHELL, Daniel ; BYRNE, Michael D. ; DOUGLASS, Scott ; LEBIERE, Christian ; QIN, Yulin: An integrated theory of the mind. In: *Psychological review* 111 (2004), Oktober, Nr. 4, S. 1036–60. – URL <http://www.ncbi.nlm.nih.gov/pubmed/15482072>. – ISSN 0033-295X
- [Bates u.a. 1994] BATES, Joseph ; LOYALL, A. B. ; REILLY, W. S.: An architecture for action, emotion, and social behavior. In: *Artificial social systems* (1994). – URL <http://www.springerlink.com/index/70502844138hu937.pdf>
- [Burke u.a. 2001] BURKE, Robert ; ISLA, Damian ; DOWNIE, Marc ; IVANOV, Yuri ; BLUMBERG, Bruce: Creature smarts: The art and architecture of a virtual brain. In: *Proceedings of the Computer Game Developers Conference* (2001). – URL <http://yayamoose.homelinux.com/~ripper/mirrors/www.gdconf.com/archives/proceedings/2001/burke.pdf>
- [Klingenberg 2010] KLINGENBERG, Arne: Prototypische Entwicklung eines emotionalen Agenten auf der Basis des Goal Oriented Action Plannings. (2010). – URL <http://opus.haw-hamburg.de/volltexte/2010/964/pdf/Bachelorarbeit.pdf>
- [Kuligowski 2009] KULIGOWSKI, Erica D.: The process of human behavior in fires. In: *NIST Technical Note 1632* (2009). – URL <http://fire.nist.gov/bfrlpubs/fire09/PDF/f09027.pdf>
- [Laird u.a. 1987] LAIRD, John ; NEWELL, Allen ; ROSENBLOOM, Paul S.: SOAR: An architecture for general intelligence. In: *Artificial Intelligence* 33 (1987), Nr. 1, S. 1–64. – URL <http://linkinghub.elsevier.com/retrieve/pii/0004370287900506>. – ISSN 00043702
- [Minsky 1986] MINSKY, Marvin: *The society of mind*. New York : Simon and Schuster, 1986
- [Münchow u.a. 2012] MÜNCHOW, Stefan ; ENUKIDZE, Ia ; SARSTEDT, Stefan ; THIEL-CLEMEN, Thomas: WALK: A Modular Testbed for Crowd Evacuation Simulation. In: *Weidman et al. (eds.), Proc. of the 6th Int. Conf. on Pedestrian Evacuation Dynamics (in print)* (2012)
- [Ortony u.a. 1988] ORTONY, Andrew ; CLORE, Gerald L. ; COLLINS, Alan: *The Cognitive Structure of Emotions*. URL [http://books.google.de/books?hl=de&lr=&id=dA3JEEAp6TsC&oi=fnd&pg=PR9&dq=Ortony,+A.,+Clare,+G.L.,+%26+Collins+\(1988\):+The+Cognitive+Structure+](http://books.google.de/books?hl=de&lr=&id=dA3JEEAp6TsC&oi=fnd&pg=PR9&dq=Ortony,+A.,+Clare,+G.L.,+%26+Collins+(1988):+The+Cognitive+Structure+)

of+Emotions&ots=HULEZo7Qn5&sig=R1P1YuVWTTok_2dtNS425WQSsX8#v=onepage&q&f=false, 1988

[RiMEA 2009] RIMEA: Richtlinie für Mikroskopische Entfluchtungsanalysen. (2009). – URL <http://www.rimea.de/downloads/richtlinien/r2.2.1.pdf>

[Wooldridge 2002] WOOLDRIDGE, Michael J.: *An introduction to multiagent systems*. Chichester, U.K. : John Wiley & Sons, 2002

[Zhou u. a. 2010] ZHOU, Suiping ; CHEN, Dan ; CAI, Wentong ; LUO, Linbo ; LOW, Malcolm ; TIAN, Feng ; TAY, Victor ; ONG, Darren ; HAMILTON, Benjamin: Crowd modeling and simulation technologies. In: *ACM Transactions on Modeling and Computer Simulation* 20 (2010), Oktober, Nr. 4, S. 1–35. – URL <http://portal.acm.org/citation.cfm?doid=1842722.1842725>. – ISSN 10493301