



## Diplomarbeit

Umsetzung des PECS-Referenzmodells zur Abbildung psychosomatischer Prozesse in der Simulation von Personenströmen und Vorbereitung der Integration in das Simulationstool SimWalk

vorgelegt von:

Kristina Dammasch

Betreuer:

Prof. Dr. Graham Horton

Otto-von-Guericke-Universität Magdeburg

Fakultät für Informatik

Institut für Simulation und Grafik

Dr. Karl Alfred Schniz

Dr. Schniz GmbH

Wasserburger Landstraße 264

81827 München

13. April 2006

Kristina Dammasch

Umsetzung des PECS-Referenzmodells zur Abbildung psychosomatischer Prozesse in der Simulation von Personenströmen und Vorbereitung der Integration in das Simulationstool SimWalk

Diplomarbeit, Otto-von-Guericke-Universität Magdeburg, 2006

# Zusammenfassung

Die Simulation von Personenströmen mithilfe der Agententechnologie ist ein noch junges Fachgebiet. Die bisher entwickelten Simulatoren, die die Möglichkeit bieten, beliebige Umgebungen abzubilden, verfügen nur über sehr einfache Modelle menschlichen Verhaltens.

Ziel der vorliegenden Diplomarbeit ist es, aufzuzeigen, wie die Simulationsergebnisse durch Integration menschlicher Parameter und Entscheidungsprozesse verbessert werden können, um damit auch die Akzeptanz der Resultate beim Anwender zu steigern. Dazu wurden Anforderungen an ein entsprechendes Agentenmodell entworfen und auf der Basis des PECS-Referenzmodells ein Modul für das Simulationswerkzeug SimWalk entwickelt. Dieses Modul bildet die Grundlage für die zukünftige vollständige Umsetzung des Entwurfes.

Da sich das Modell nicht auf interne Zustände und Prozeduren des Agenten beschränkt, sondern ebenso Anforderungen an die Umwelt stellt, werden Vorschläge für die Integration in den Simulator gemacht, die es ermöglichen sollen, den vollen Funktionsumfang des entwickelten Agentenmodells nutzen zu können.

Zur Veranschaulichung werden exemplarisch in den Entwurf integrierte Parameter und Prozesse von Umwelt und Agenten vorgestellt, auf Grundlage derer verschiedene Experimente durchgeführt werden konnten, die die zukünftigen Einsatzmöglichkeiten des Moduls aufzeigen.



---

# Inhaltsverzeichnis

1	Einleitung.....	<b>1</b>
1.1	Stand der Technik .....	1
1.2	Erläuterungen zur Problematik .....	2
1.3	Zielsetzung und Rahmenbedingungen .....	3
2	Grundlagen .....	<b>6</b>
2.1	Die Simulation von Personenströmen .....	6
2.2	Der agentenbasierte Simulator SimWalk .....	9
2.3	Der Faktor Mensch.....	12
2.4	Das PECS-Referenzmodell .....	14
3	Adaption des Referenzmodells an SimWalk .....	<b>20</b>
3.1	Anforderungen an einen Agenten in SimWalk.....	20
3.2	Anpassung der Komponente Agent .....	25
3.3	Zur Implementierung des PECS-Moduls.....	28
3.4	Überlegungen zur Integration in den Simulator .....	32
4	Parameter- und Prozessbeispiele.....	<b>37</b>
4.1	Externe und interne Zustände.....	37
4.2	Das Zusammenspiel psychosomatischer Parameter und Prozesse .....	41
4.3	Autonome Optimierung des Pfades.....	43
4.4	Berechnete Parameter als Handlungsauslöser: Das Beispiel Nahrungssuche .....	47
4.5	Die Bestimmung eines geeigneten Entfluchtungspunktes .....	51
5	Resümee .....	<b>58</b>
5.1	Zusammenfassung der Arbeit.....	58
5.2	Bewertung der Entwicklung.....	60
5.3	Ausblick .....	63
	Referenzen .....	<b>65</b>

---

## Abbildungsverzeichnis

Abbildung 2.1 - Werte in einem einfachen Potenzialfeld.....	8
Abbildung 2.2 - Potenzialfeld und Hindernisse in SimWalk.....	10
Abbildung 2.3 - Object Pressure.....	11
Abbildung 2.4 - systemtheoretisches Konzept.....	14
Abbildung 2.5 - grafische Darstellung der Kommunikation zwischen Komponenten.....	15
Abbildung 2.6 - Übersicht über die Komponenten des PECS-Referenzmodells und deren Interaktion.....	15
Abbildung 2.7 - Aufbau eines PECS-Agenten.....	17
Abbildung 3.1 - Berechnungsabläufe im Agenten.....	25
Abbildung 3.2 - Klassendiagramm des PECS-Moduls.....	28
Abbildung 3.3 - Kapselung der Daten im Agenten.....	30
Abbildung 3.4 - Umstrukturierung von PECS_World.....	35
Abbildung 4.1 - Verwaltung der externen Informationen im Connector.....	38
Abbildung 4.2 - an Actor übergebene Daten.....	39
Abbildung 4.3 - interne Parameter des Agenten.....	40
Abbildung 4.4 - Problemstellung bei der Pfadwahl.....	44
Abbildung 4.5 - im PECS-Modul berechnete Änderung des Pfades.....	46
Abbildung 4.6 - Abschätzen eines Zieles anhand von Beschilderung.....	49
Abbildung 4.7 - Auswahl einer Nahrungsquelle als Zielpunkt.....	50
Abbildung 4.8 - Einflussnahme der Position der Gefahrenquelle auf die Zielwahl.....	52
Abbildung 4.9 - Möglichkeiten der Lage von Ausgang, Gefahrenquelle und Agent zueinander.....	52
Abbildung 4.10 - im Modul berechnete Entfluchtungspunkte.....	54
Abbildung 4.11 - Ergebnisse bei verschiedenen Positionen.....	55
Abbildung 4.12 - Ergebnisse bei verschiedenen Umweltzuständen.....	55
Abbildung 4.13 - Prognose über den zusätzlichen Berechnungsaufwand.....	57

## Abkürzungsverzeichnis

CAD	Computer Aided Design
PECS	Physis, Emotion, Cognition, Social Characteristics
OO	Objektorientierung

» Der Mensch ist eine seltsame und interessante Erfindung. «

Mark Twain (1835 - 1910)

# 1 Einleitung

Die Sicherheit von Personen in öffentlichen Gebäuden oder Anlagen hat in den vergangenen Jahren immer mehr an Bedeutung gewonnen, und das nicht erst seit den Terroranschlägen der vergangenen fünf Jahre. Bereits seit 1994, dem Jahr des Unterganges der Fähre Estonia, arbeiten Forscher an der computergestützten Simulation von Menschenmassen, um Entfluchtungszeiten und optimale Rettungswege prognostizieren zu können. Bisher hat sich dabei fast einzig im Bereich der Passagierschiffahrt die simulationsgestützte Planung und Entwicklung durchgesetzt.

Doch Evakuierungsszenarien sind nicht die einzigen Anwendungsgebiete für derartige Simulationen. Sie können ebenso zur Analyse und Optimierung von Personenströmen herangezogen werden, um eine komfortable Bewegung und kurze Wege zu ermöglichen, kreuzende Ströme zu vermeiden oder Menschenmengen zu einem anderen Zweck, zum Beispiel auf Flughäfen oder in Einkaufszentren, direkt zu lenken. Die Anwendungsgebiete unterscheiden sich dabei in der Art und Weise, wie Menschen in einer entsprechenden Situation handeln und mit der Umwelt oder Mitmenschen interagieren.

## 1.1 Stand der Technik

Die ersten Standards im Bereich der Simulation von Menschenmengen wurden von der Fire Safety Engineering Group der Universität Greenwich mit ihrer Software EXODUS gesetzt. Diese verbindet verschiedene Distributionen, unter anderem die Lösungen maritimeEXODUS für Passagierschiffe oder auch buildingEXODUS für Gebäudesimulationen, ist aber ausschließlich für Evakuierungsszenarien einsetzbar. Anstelle von Evakuierung ist auch der Begriff der »Entfluchtung« gebräuchlich. Die Bewegungen der Personen in ihrer modellierten Umgebung erfolgt dabei aufgrund physikalischer Gesetzmäßigkeiten; das menschliche Verhalten in Gefahrensituationen ist dabei nur ansatzweise beschrieben.

Der Vorteil dieser Technik ist die geringe Rechenzeit, die für einen einzelnen Simulationslauf benötigt wird. Jedoch werden derartige Modelle seit einigen Jahren als veraltet betrachtet, da sie die Eigenarten menschlichen Handelns und Interagierens, vor allem in derartigen Ausnahmesituationen, fast vollständig vernachlässigen.

Eine Lösungsmöglichkeit für dieses Problem bietet die Agententechnologie als Teilgebiet der Künstlichen Intelligenz. Hier übernehmen intelligente Software-Agenten alle Fähigkeiten und Aufgaben der in der Simulation abzubildenden Personen. Ein auf dem Markt erhältliches Simulationswerkzeug, das diese Techniken bereits einsetzt, ist SimWalk, entwickelt vom Schweizer Unternehmen Savannah Simulations. Doch die Agententechnologie steht im Allgemeinen noch am Anfang ihrer Entwicklung. Insbesondere der Einsatz zur Abbildung menschlichen Handelns und Entscheidens sowie die Idee, ein komfortables Werkzeug zu liefern, mit dem beliebige Umgebungen aufgebaut werden können, findet bisher kaum Anwendung. So sind die in SimWalk und den wenigen vergleichbaren Simulatoren verwendeten Agentenmodelle nur wenig umfangreicher als der in EXODUS angewandte Entwurf.

## 1.2 Erläuterungen zur Problematik

Die bisherige Implementierung des Simulators SimWalk betrachtet weder mentale und körperliche Zustände des Menschen, wie zum Beispiel Angst oder Panik in Evakuierungssituationen, noch Prozesse, die dem Agenten mehr als nur ein rein reaktives Verhalten ermöglichen. Diese Einflüsse spielen bei der Simulation von Personenströmen bezüglich der Genauigkeit der Simulationsergebnisse jedoch eine entscheidende Rolle. Und auch die Akzeptanz der Agententechnologie beim Anwender kann durch eine differenziertere Abbildung menschlicher Verhaltensweisen gesteigert werden. Aufgrund dessen befasst sich die vorliegende Arbeit damit, wie derartige Prozesse und Zustände in die Simulation integriert werden können.

Die Hauptproblematik, mit der sich Forscher in diesem Bereich seit einigen Jahren befassen, ist zum einen die Frage, wie sich Menschen in unterschiedlichen Bewegungssituationen verhalten. Zum anderen beschäftigt man sich damit, wie das Verhalten am Rechner nachgebildet werden kann und in welcher Form und in welchem Umfang dies erforderlich ist. Ersteres ist als Teilgebiet der Soziologie und Psychologie anzusehen und wird in Zusammenarbeit mit Experten in Studien und Versuchen evaluiert.

Mit der zweiten Problematik befassen sich Forscher aus den Bereichen der Künstlichen Intelligenz und Simulationstechnik. Sie haben bereits verschiedenste Konzepte entwickelt, die versuchen, eine Struktur für ein geeignetes Modell eines Agenten, seiner Umwelt oder beides in Kombination vorzugeben. Diese Konzepte werden als Referenzmodelle bezeichnet und sind als eine Art Konstruktionsschema zu verstehen. Die genauen Inhalte und der Detaillierungsgrad der Abbildung wird nicht vorgegeben, sondern kann in Abhängigkeit der durch die Simulation zu beantwortenden Fragestellung näher definiert werden.

Das PECS-Referenzmodell ist ein solches Modell und wurde speziell für die agentenbasierte Modellierung menschlichen Handelns, Entscheidens und Verhaltens entwickelt [Urb04]. Es ist ein hierarchisches komponentenbasiertes Modell, welches auf der obersten Ebene die Einheiten angibt, aus denen ein vollständiges Simulationsmodell bestehen könnte. Dazu zählen neben der Komponente für den Agenten eine weitere für die Abbildung der Umwelt und eine für den Nachrichtenaustausch zwischen den Agenten. Ein Agent ist zusätzlich in weitere Komponenten untergliedert, wovon jeder eine vorgegebene Aufgabe im Zusammenhang mit der mentalen Verarbeitung von Informationen zugewiesen ist. Die Komponente Agent befindet sich in Abhängigkeit der gewünschten Zahl von vorhandenen Personen beliebig oft im Modell. Das bedeutet, dass jeder Agent im System exakt wie jeder andere aufgebaut ist und somit prinzipiell die gleichen Fähigkeiten besitzt. Variationen zwischen den Agenten können durch unterschiedliche Eigenschaften, sprich Parameterwerte, in der Komponente umgesetzt werden. So kann sowohl die strukturelle Integrität und Übersichtlichkeit beim Modellaufbau als auch die Abbildung verschiedener menschlicher Persönlichkeitsmerkmale gewährleistet werden.

Der Name des Referenzmodells ergibt sich aus den vier großen Komponenten im Agenten, die für die Abbildung psychosomatischer Zustände und Prozesse vorgegeben sind: PECS steht dabei für »Physis«, »Emotion«, »Cognition« und »Social Characteristics«.

### 1.3 Zielsetzung und Rahmenbedingungen

In der vorliegenden Arbeit soll gezeigt werden, auf welche Art und Weise psychosomatische Zustände und Prozesse des Menschen in den Personenstromsimulator Sim-

Walk integriert werden können. Als psychosomatisch werden gemeinhin körperliche, seelische und soziale Vorgänge sowie deren Wechselwirkungen untereinander bezeichnet. Das PECS-Referenzmodell wird dabei den konzeptionellen Rahmen liefern, um eine Richtung für die späteren Inhalte vorzugeben und die strukturelle Integrität der einzelnen Bestandteile des Modells zu gewährleisten. Da dieses Referenzmodell auch von Savannah Simulations vorgeschlagen wurde, werden keine weiteren Alternativen betrachtet. Es ist jedoch zu sagen, dass kaum ein anderes Modell die Möglichkeit bietet, die komplexen Wechselwirkungen psychosomatischer Prozesse derartig detailliert und strukturiert abzubilden.

Der Schwerpunkt der Einsatzmöglichkeiten für SimWalk liegt auf der Analyse und Optimierung von reinen Personenströmen, bei denen die Agenten ohne umfassende Vorgaben vom Anwender einen Weg in der Umwelt finden müssen. So muss zuerst geklärt werden, durch welche Eigenschaften ein Agent in derartigen Situationen sinnvoll und ausreichend charakterisiert wird. Dies bildet die Grundlage dafür, welche Komponenten des Referenzmodells in welchem Detaillierungsgrad abgebildet werden und wie ihr Inhalt zu definieren ist. Weiterhin muss der Umfang der Fähigkeiten geklärt werden, die den Agenten die ihm gestellten Aufgaben lösen lassen. Das setzt eine Einschätzung dessen voraus, welchen Situationen sich der Agent gegenübergestellt sehen kann und welche Fähigkeiten wiederum für die Simulationsergebnisse keine nennenswerten Verbesserungen bringen werden. Es sollte menschliches Verhalten insofern abgebildet werden, dass der Verlauf der Personenströme von einem Beurteilenden als natürlich eingeschätzt wird. Beim momentanen Umfang der Implementierung des Simulators ist dies noch eine der offensichtlichsten Schwächen. Das Fehlen menschlicher Verhaltensweisen, die die Wahl des Weges in der Umgebung beeinflussen, weckt bei Interessenten und Anwendern die größten Zweifel an der Glaubhaftigkeit der in SimWalk integrierten Agententechnologie. Zu diesen Verhaltensweisen gehört in erster Linie die Wahrnehmung der Umwelt, die es dem Agenten ermöglicht, aufgrund kognitiver Fähigkeiten aktiv eine Entscheidung bezüglich des zu laufenden Weges zu treffen und nicht nur passiv zu reagieren, wie es momentan der Fall ist. Das setzt auch voraus, dass das Umweltmodell entsprechende Informationen bereitstellt. Aus diesem Grund werden im Hinblick darauf Möglichkeiten und Umfänge vorgeschlagen, die auf das entwickelte Agentenmodell abgestimmt sind.

Da die Funktionalität zur Agentenbewegung von SimWalk in C++ implementiert ist, wird auch die Realisierung des PECS-Referenzmodells in einem zusätzlichen Modul

in dieser Programmiersprache erfolgen. Dadurch ist eine schnelle Integration in den Simulator gewährleistet und eventuelle Schnittstellen sind ohne Aufwand umzusetzen. Die objektorientierte Programmierung in C++ bietet auch die Möglichkeit, den Programmcode stark zu modularisieren. Aufgrund dessen sind die entwickelten Komponenten einfach zu pflegen und anzupassen, ohne dass Änderungen am Gesamtkonzept erforderlich sind. Auch die Abläufe im Modul können auf diese Weise bei Bedarf ohne größeren Zeitaufwand verändert werden. Des Weiteren können die einzelnen Komponenten gezielt mit Zugriffsrechten und Mechanismen zum Schutz der enthaltenen Parameter ausgestattet werden, um eine ungewollte Manipulation zu verhindern und den Zugriff zu kontrollieren.

Um aufzeigen zu können, dass eine Integration psychosomatischer Parameter in die Simulation die erforderlichen Verbesserungen bringen kann, werden verschiedene Experimente entworfen, in denen Agenten Situationen bewältigen müssen, in denen sie bisher kaum eine angepasste Verhaltensweise zeigen könnten. Es handelt sich dabei um Situationen, in denen mehr als Reagieren vom Agenten erwartet wird, da nur eine aktive Verarbeitung der zur Verfügung stehenden Informationen die gewünschten menschlichen Verhaltensweisen approximieren kann.

Da die Integration der Entwürfe in SimWalk nicht simultan erfolgen kann, steht der Simulator nicht zu Testzwecken zur Verfügung. So muss für die entwickelten Programmcodes ein Rahmen geschaffen werden, der die Aufgaben von SimWalk wahrnimmt. Dies bietet gleichzeitig die Möglichkeit, die genauen Vorstellungen zur Integration und erforderlichen Anpassungen am Simulator zu veranschaulichen. Für eine korrekte Umsetzung psychosomatischer Parameter und Prozesse wäre jedoch zusätzlich eine umfassende Ausarbeitung erforderlich, welche Größen auf welche Art und Weise eine Rolle spielen. Da dies nicht Inhalt dieser Arbeit sein soll, wurden alle in den Experimenten verwendeten Parameter von Umwelt und Agenten sowie die relationalen und quantitativen Zusammenhänge rein für diese Szenarien definiert. Sie entbehren jeglicher wissenschaftlicher Basis, auch wenn eine für Personenströme als sinnvolle erachtete Auswahl getroffen wurde. Sie sollen lediglich aufzeigen, wie eine zukünftige Umsetzung erfolgen kann.

Doch selbst diese Beispiele lassen bereits deutlich werden, welches Potenzial in einem umfassenderen Agentenmodell steckt und inwiefern die Simulationsergebnisse damit verbessert werden können.

## 2 Grundlagen

Das Gebiet der Personenstromsimulation ist interdisziplinär und erfordert Kenntnisse aus den Bereichen Physik, Simulation und Künstliche Intelligenz, sowie Wissen über psychologische und soziologische Faktoren menschlichen Handelns und Entscheidens. Im Folgenden soll ein kurzer Einblick in diese Themengebiete gegeben werden.

### 2.1 Die Simulation von Personenströmen

Die computergestützte Simulation ermöglicht die Optimierung eines Objektes hinsichtlich des Verlaufes der Personenströme oder der Lage und Kapazität der Rettungswege bereits in der frühen Planungsphase. Eine spätere Anpassung der Architektur ist oftmals nicht nur schwierig sondern auch kostenintensiv. Auch die mit Sicherheitsrisiken verbundenen und mehrfach ungenauen Tests mit Versuchspersonen unter nachgestellten Bedingungen können dank entsprechender Simulationsanalysen eingeschränkt werden.

Bisher werden für solche Simulationen häufig Werkzeuge genutzt, welche auch für die zeitdiskrete Abbildung von Materialflüssen oder ähnliche Aufgabenstellungen verwendet werden. Der Mensch wird dort als ein Objekt betrachtet, welches nur durch eine geringe Anzahl von »technischen« Parametern charakterisiert ist und von einem Ort im Raum zum nächsten mit einem bestimmten Zeitverbrauch transferiert wird. Dabei sind die Pfade, also die anzusteuern Punkte in der Umgebung und ihre Reihenfolge, fest vorgegeben. Dies ermöglicht zwar die Bewertung von Warteschlangen oder Durchsätzen an den definierten Punkten, bietet aber keine Möglichkeit, die gelaufenen Wege und damit zum Beispiel den Komfort, räumliche Engpässe oder die zurückgelegte Wegstrecke zu beurteilen.

Solche zeitdiskreten Simulationen bieten die Möglichkeit, mit wenig Aufwand und Rechenzeit eine gewisse Zahl von Ergebnissen zu erhalten, sind jedoch nicht für die qualitative Bewertung von Personenströmen geeignet. Des Weiteren ist das Ausblen-

den von Fußgängerverhalten und der dabei auftretenden Phänomene eine Quelle für Ungenauigkeiten in den Simulationsergebnissen. Solche Phänomene können zum Beispiel das Ausweichverhalten und der spontane Richtungswechsel sein, welche das Aussehen der Bewegung im freien Raum, den zurückgelegten Weg oder die benötigte Zeit beeinflussen.

In den so genannten hydraulischen Modellen wird ein anderer Ansatz verfolgt, da hier die Bewegung der abzubildenden Personen als Partikelfluss dargestellt wird. Um die Partikel »fließen« zu lassen muss ein Gefälle konstruiert werden, welches seine Tiefen an den anzusteuern den Punkten im Raum hat. Dies wird häufig über ein Potenzialfeld realisiert. Dazu wird der Raum in diskrete Zellen unterteilt und jeder Zelle ein Potenzial in Abhängigkeit ihrer Entfernung zu den verschiedenen Punkten zugewiesen. Die Ziele bekommen das Potenzial null und die direkt benachbarten Zellen das Potenzial der Zielzelle um eine Einheit inkrementiert. In den indirekten Nachbarn wird das Potenzial auf den Wert der Zielzelle plus zwei Einheiten gesetzt. Danach werden für jede der soeben berechneten Zellen deren direkte oder indirekte Nachbarn aktualisiert und die Berechnung so lange fortgesetzt, bis jeder Zelle im Raum ein Potenzialwert zugewiesen ist. Stellt sich dabei bei einer Zelle heraus, dass diese bereits einmal berechnet wurde, wird der kleinere der beiden Werte übernommen (Abbildung 2.1). So erhält jeder Zielpunkt eine Art Wirkungsgebiet.

Diese Berechnung wird als »Wave Algorithm« bezeichnet, da sich die neuen Werte wie eine Flüssigkeit wellenartig über den Raum ausbreiten. Sie ist eine gebräuchliche Methode zur Aufstellung von Potenzialfeldern und ihr Vorteil liegt in der Einfachheit und Geschwindigkeit, mit der der gesamte Raum berechnet werden kann. Der Algorithmus ist nicht von den Positionen der Partikel abhängig und muss so auch nur einmal zu Simulationsbeginn ausgeführt werden [Fer01].

Wenn man die berechneten Potenziale als Höhenwerte betrachtet, ergibt sich eine Art »Gebirge«, in welchem sich die Partikel – die abzubildenden Personen – Richtung Tal bewegen können.

	2	3	4	5	6	6	5	4	3	2	
	1	2	3	4	5	5	4	3	2	1	
Ziel A	0	1	2	3	4	4	3	2	1	0	Ziel B
	1	2	3	4	5	5	4	3	2	1	
	2	3	4	5	6	6	5	4	3	2	

Abbildung 2.1 - Werte in einem einfachen Potenzialfeld

Jede der Zellen im Potenzialfeld kann immer nur von einem Partikel besetzt werden. Folglich wird sich jede Zelle in der Simulation stets in genau einem von zwei möglichen Zuständen befinden: frei oder belegt. Eine Fortbewegung der Partikel im Raum erfolgt durch die Suche nach der nächsten benachbarten freien Zelle mit dem niedrigsten Potenzial. Die Partikel »strömen« auf diese Weise in Richtung des nächstgelegenen Ziels.

Dieses rein hydraulische Verhalten ist in der Lage nachzubilden, dass Menschen stets den kürzesten Weg zum Ziel zu wählen. Jedoch ist die Entfernung nicht das einzige Kriterium für die Wahl eines Pfades im Raum.

Um menschliche Verhaltensweisen approximieren zu können ist es erforderlich, den bisher betrachteten Partikeln Eigenschaften und Fähigkeiten zu verleihen. Dazu bietet sich die Agententechnologie als Fachgebiet der Künstlichen Intelligenz an. Als Agenten bezeichnet man dort ein Software-Programm, das in einer definierten statischen oder dynamischen Umgebung arbeitet, in welcher es Aktionen autonom ausführen kann, um die ihm gestellten Aufgaben zu erfüllen [ISE]. Für die Simulation von Personenströmen bedeutet dies, dass jeder Fußgänger im Modell individuell über einen Agenten abgebildet wird. Dieser übernimmt alle Eigenschaften, Fähigkeiten und Aufgaben der Person. Ein System, in welchem mehrere Personen abgebildet werden sollen und demzufolge auch eine entsprechende Zahl von Agenten vorhanden ist, wird als Multiagentensystem bezeichnet.

Die Klassifizierung von Agenten ist schwierig, da die Grenzen zwischen verschiedenen Konzepten oftmals schwer zu ziehen sind. Fast immer eindeutig lassen sich jedoch intelligente Agenten von nicht-intelligenten unterscheiden. Letztere werden als reaktive Agenten bezeichnet, da sie ausschließlich auf Veränderungen ihrer Umwelt reagieren. Sie besitzen für gewöhnlich eine sehr einfache Struktur.

Um ein Stück Software als intelligenten Agenten klassifizieren zu können, muss es einige Anforderungen erfüllen. Die genauen Definitionen gehen in der Literatur auseinander, zu den wichtigsten Aspekten zählen jedoch Autonomie und Proaktivität. Das bedeutet, dass der Agent erstens weitgehend unabhängig von Benutzereingriffen arbeitet und zweitens Aktionen aufgrund eigener Initiative auslöst. Häufig wird die Fähigkeit zur Kommunikation mit anderen Agenten als Anforderung genannt, um soziale Faktoren intelligenten Verhaltens zu berücksichtigen. Des Weiteren sollte er auf Änderungen in der Umwelt reagieren und in der Lage sein, aufgrund zuvor getätigter Beobachtungen und Entscheidungen zu lernen, um sich auf diese Weise an Bedingungen anzupassen. Denn dies sind Fähigkeiten, aufgrund derer auch das Verhalten und Handeln von Lebewesen als intelligent bezeichnet wird. Mithilfe solcher Agenten kann simuliert werden, wie sich Menschen in einer definierten Umgebung autonom bewegen.

## 2.2 Der agentenbasierte Simulator SimWalk

Die Entwicklung von Software-Agenten hat verstärkt zu Beginn der 90er Jahre mit den ersten Einsätzen im World Wide Web begonnen. In den vergangenen Jahren kam die Agententechnologie auch in anderen Anwendungsgebieten, wie zum Beispiel in der Biologie oder Soziologie, zum Einsatz. Seitdem gab es auch erste Bestrebungen, menschliche Verhaltensweisen anhand von Multiagentensystemen nachzubilden und zu prognostizieren. Doch die Komplexität und die so häufige scheinbare Nicht-Prognostizierbarkeit der menschlichen Psyche lässt die Arbeiten in diesem Forschungsgebiet nur langsam vorankommen. Aufgrund der Schwierigkeiten beim Entwurf »umfassender« Modelle wurden zahlreiche problemspezifische Multiagentensimulationen durchgeführt, bei denen jeweils nur ein kleiner, für die aktuelle Problematik als relevant erachteter Teil menschlicher Eigenschaften und Fähigkeiten abgebildet wurde.

Solche Modelle müssen von Experten entwickelt und implementiert werden. Sie sind daher nur schwer wiederverwendbar oder auf andere Fragestellungen adaptierbar. Die Auftraggeber für derartige Simulationen, häufig Ingenieure oder Soziologen, sind jedoch für gewöhnlich Laien auf dem Gebiet der Künstlichen Intelligenz und Programmierung. Trotzdem erwartet ein Auftraggeber, dass er das Simulationsmodell

selbstständig anpassen kann, um verschiedene Szenarien zu testen. So liegt es nahe, auch für die agentenbasierte Simulation von Personenströmen schnell erlernbare Tools bereitzustellen, mit denen jedes beliebige Modell auch von einem für die Problematik ungeschulten Anwender erstellt werden kann, so wie es in anderen Simulationsgebieten bereits Standard ist. Der Simulator SimWalk ist ein solches Tool.

Die Idee der Entwickler war es, ein Werkzeug zu schaffen, welches die Möglichkeit bietet, beliebige Umweltmodelle und Agenten zu definieren, ohne dass dafür Kenntnisse im Bereich der Agententechnologie oder Simulation bzw. im Umgang mit gängigen Simulationstools erforderlich sind. Aus diesem Grund lässt sich ein Modell auch in wenigen einfachen Arbeitsschritten erstellen. Die Beschaffenheit der Umwelt in Form von Grenzen und Hindernissen kann direkt aus einer CAD-Datei eingelesen werden. Anschließend werden Start- und Endpunkte für die Personenströme sowie markante Punkte im Raum mit Zeitverbrauch, zum Beispiel ein Ticketschalter oder die Sicherheitskontrolle an einem Flughafen, direkt in diesem Layout angegeben. Für jeden dieser Punkte wird ein Potenzialfeld berechnet, ähnlich wie in Kapitel 2.1 beschrieben. Für Zellen, die zu einem Hindernis gehören, wird kein Potenzial berechnet – die wellenförmige Ausbreitung der berechneten Potenziale bewegt sich um die Hindernisse herum.

Die Potenziale können in SimWalk farbkodiert angezeigt werden (Abbildung 2.2). Grün symbolisiert dabei das niedrigste Potenzial, also den Zielpunkt, und der Farbverlauf nach Blau die zunehmende Entfernung. Zellen die nicht berechnet wurden – hier die rechteckigen Hindernisse – werden schwarz dargestellt.

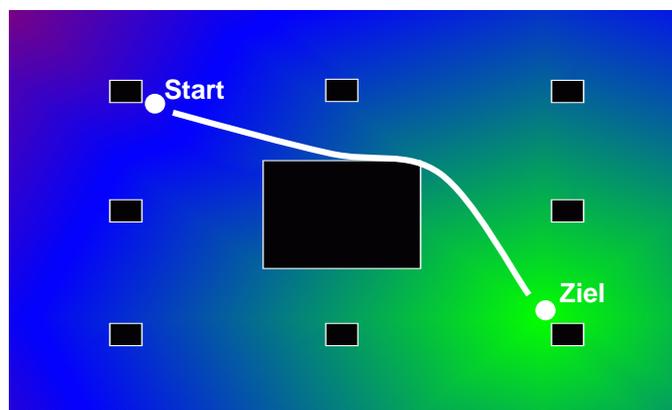


Abbildung 2.2 - Potenzialfeld und Hindernisse in SimWalk

Ein Mensch, der sich in einer derartigen Umgebung vom angegebenen Start- zum Zielpunkt bewegen müsste, würde in etwa den dargestellten Weg wählen, da er die kürzeste Verbindung beider Punkte ist, beim gleichzeitigen Einhalten eines Mindestabstandes zu den vorhandenen Hindernissen. Dieser Abstand wird in Studien auf etwa 0,2 bis 0,5 Meter geschätzt [Stu03]. Um zu verhindern, dass auch ein Agent, der sich von Start zu Ziel bewegt, nicht direkt an der Wand des Objektes in der Mitte des Raumes entlang läuft, werden die Potenziale um das Objekt herum erhöht und üben somit einen abstoßenden Druck auf den Agenten aus. Diese Kraft wird im Simulator als »Object Pressure« bezeichnet (Abbildung 2.3).

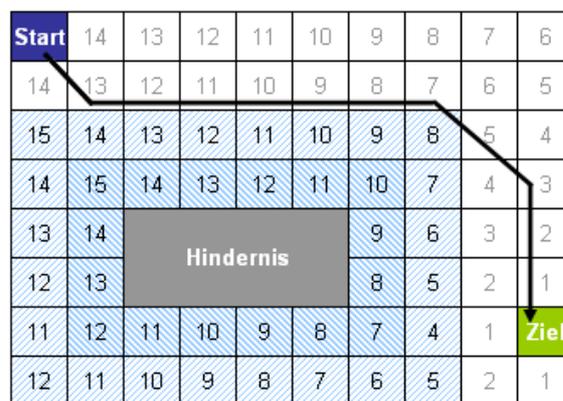


Abbildung 2.3 - Object Pressure

Auf den Agenten wirken nun zwei Kräfte: der beschriebene Druck, den Hindernisse ausüben, und die Kraft, die den Agenten zum Zielpunkt zieht, modelliert durch die Potenziale der Zellen im Raum. Letztere wird aus diesem Grund als »Potential Force« bezeichnet. Doch es gibt noch eine dritte Einflussgröße. Sie beschreibt die räumliche Interaktion mit anderen Agenten und hat als »Pedestrian Pressure« eine ähnliche Wirkweise wie der Objektdruck. Sie beruht auf der Tatsache, dass Menschen soweit möglich einen gewissen Abstand zu anderen, ihnen unbekanntem einhalten. Dieser Abstand wird im Simulator als »Pedestrian Interaction Range« bezeichnet. Aufgrund der Auswertung verschiedener Studien wurde er standardmäßig auf 1,71 Meter gesetzt.

Auf diese Art und Weise wird der Agent auf der Suche nach dem kürzesten Weg zum Zielpunkt sowohl Hindernissen als auch anderen Agenten ausweichen. In Abhängigkeit der räumlichen Gegebenheiten, zum Beispiel in einer engen Passage oder bei ho-

hem Agentenaufkommen, können die Distanzen zu beiden – Agenten und Objekten – unterschritten werden.

Dieser Bewegungsalgorithmus der Software, über den sich die Agenten in Gebäuden oder Anlagen fortbewegen, basiert auf der Diplomarbeit von Pascal Stucki aus dem Jahr 2003. Seitdem wurde zu diesem Algorithmus ein Gesamtkonzept für ein Simulationstool inklusive der entsprechenden Programmoberfläche entwickelt. Das Agentenmodell wurde bisher noch nicht modifiziert. Der momentane Stand umfasst einen Agenten, der in der Lage ist, ein Ziel zu finden und auf räumliche Veränderungen in der Umwelt in Form auftauchender Hindernisse zu reagieren, genauer gesagt, diesen auszuweichen. Derartige reaktive Agenten in einer dynamischen Umwelt werden als situierte Agenten bezeichnet [Fer01]. Die einzige Aktion, die diese Agenten in SimWalk ausführen, ist die physikalische Bewegung. Sie können nicht die Anforderungen erfüllen, die an einen intelligenten Agenten zur Modellierung menschlicher Verhaltensweisen gestellt werden.

Dies hat zur Folge, dass das Verhalten der Agenten vom Anwender oftmals als unrealistisch kritisiert wird, da ein Agent nicht in der Lage ist, bewusst eine Entscheidung zu treffen. Doch selbst in einfachen Bewegungssituationen handeln Menschen nicht ausschließlich reaktiv wie die Agenten in SimWalk, sondern treffen innerhalb kürzester Zeit zahlreiche Entscheidungen, die ihren Weg und damit auch das Verhalten anderer beeinflussen.

## 2.3 Der Faktor Mensch

Die Schwierigkeit bei der Abbildung menschlichen Verhaltens besteht in der Komplexität menschlicher Entscheidungs- und Bewertungsprozesse. Diese Prozesse weisen häufig eine komplizierte Gestalt auf und werden in hohem Maße durch verschiedenste Faktoren beeinflusst. Aus diesem Grund muss ein Mensch als Einheit gesehen werden und nicht als Objekt, losgelöst von internen oder externen Einflüssen [Sch01]. »Intern« bezeichnet dabei alle mentalen und körperlichen Größen, aber auch Prozesse beziehungsweise Aktionen, die interne Größen verändern. Als »extern« werden die Zustände der Umwelt bezeichnet, sowie Aktionen, die in der Umwelt ausgeführt werden oder dem Austausch von Informationen dienen. Das menschliche Verhalten wird allgemein als die Summe aller internen und externen Aktionen definiert und wird – da

es die Grundlage dafür bildet – in ähnlicher Weise klassifiziert wie die bereits beschriebenen Agententypen.

Die einfachste Form ist das reaktive Verhalten, welches weitestgehend automatisch und ohne bewusste Kontrolle aufgrund angeborener oder erlernter Reiz-Reaktions-Muster erfolgt. Das bedeutet, dass dieses Verhalten so lange erfolgreich ist, wie eine gegebene Situation einem bekannten Reiz entspricht. In einer unbekanntem Situation stößt es an seine Grenzen, da kein passender Reiz und somit auch kein Reaktions-schema gefunden werden kann. Auch bei der Nichterfüllung von Erwartungen, die an eine ausgeführte Aktion geknüpft wurden, kann keine entsprechende Reaktion ausgelöst werden. Dadurch ist eine Bewertung von Erfolg oder Misserfolg nicht möglich.

Beim deliberativen Handeln und Entscheiden wird das Verhalten bewusst gelenkt, um definierte Ziele zu erreichen. Die in einer vorgegebenen Situation auszuführenden Aktionen werden durch die Motive und die Motivstärke bestimmt. Dabei wird als Motiv eine innere Kraft verstanden, die den Menschen zur Ausführung einer Handlung lenkt. Beim Vorhandensein mehrerer Motive wird das Motiv mit der jeweils höchsten Motivationsstärke handlungsleitend und legt so die Aktionsfolge fest. Durch bewusstes Nachdenken und Probehandeln anhand von Modellen wird eine erfolgversprechende Aktionsfolge ermittelt, die auf das Ziel zuführt. Sofern ein Mensch nicht nur deliberative Entscheidungen trifft und entsprechende Handlungen ausführt, sondern auch die damit in Verbindung stehenden Prozesse und Ergebnisse beobachtet und beeinflusst, spricht man von reflektivem Verhalten. Jede dieser Verhaltensweisen bedeutet eine Weiterentwicklung der Fähigkeiten zur Anpassung an interne und externe Faktoren.

Wenn die mit derartigem Verhalten in Zusammenhang stehenden Einflussgrößen und Entscheidungen mithilfe eines Agenten abgebildet werden sollen, ist es sinnvoll, einen festen Rahmen zu verwenden. Nicht nur, um die Struktur der Agenten zu vereinheitlichen, sondern auch, um die Konzeption für das Gesamtsystem vorzugeben. So kann einerseits sichergestellt werden, dass jeder Agent grundsätzlich dieselben Fähigkeiten besitzt wie jeder andere, und somit dieselben Möglichkeiten zur Adaption hat. Ferner kann in einem die Umwelt mit einbeziehenden Modell die Repräsentation externer Faktoren bezüglich der Anforderungen des Agenten angepasst, sowie die wechselseitigen Beziehungen zwischen Umwelt und Agenten modelliert werden.

Ein Referenzmodell bietet diese Möglichkeit, jedoch sind nicht alle Modelle derart umfangreich konzipiert. Des Weiteren wird die Bedeutung des Begriffes »Referenzmo-

dell« zurzeit in der Literatur nicht einheitlich verwendet. Die Arbeitsgemeinschaft Simulation ASIM – eine der führenden Forschungsgruppen im Bereich der Referenzmodelle – schlägt die folgende Definition vor [Urb04], welche für die Entwicklung des PECS-Referenzmodells und somit auch für die vorliegende Arbeit als Richtlinie gesehen wird:

Ein Referenzmodell umfasst eine systematische und allgemeingültige Beschreibung eines definierten Bereiches der Realität mit den für eine vorgegebene Aufgabenstellung relevanten charakteristischen Eigenschaften und legt das zugehörige Modellierungskonzept fest. Im Bereich der Simulation dienen Referenzmodelle als Konstruktionsschemata für den Entwurf von Simulationsmodellen.

So ist das PECS-Referenzmodell als domänenunabhängiges Konstruktionsschema für agentenbasierte Simulationsmodelle definiert, bei denen menschliches Handeln, Entscheiden und Verhalten von Bedeutung sind. Es gibt lediglich den konzeptionellen Rahmen für den Modellentwurf vor, welcher dann in Abhängigkeit der gegebenen Problemstellung mit Inhalt gefüllt werden kann.

## 2.4 Das PECS-Referenzmodell

Das Prinzip des Modells beruht auf einem klassischen systemtheoretischen Ansatz. Das System ist dabei durch interne Zustandsgrößen gekennzeichnet, welche sich sowohl durch Eigendynamik als auch durch extern induzierte Dynamiken in Form verschiedener Inputs ändern. Aufgrund dessen erzeugt es nach außen hin sichtbare Outputs (Abbildung 2.4).

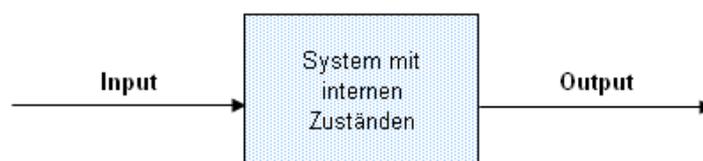


Abbildung 2.4 - systemtheoretisches Konzept

Des Weiteren ist das gesamte Modell komponentenbasiert und hierarchisch aufgebaut. Die verschiedenen Komponenten stehen dabei in Beziehung zueinander. Zum einen kann eine kausale Abhängigkeit bestehen, wenn das dynamische Verhalten einer

Komponente vom Zustand einer anderen Komponente abhängig ist. Die Informationsübertragung erfolgt ohne Zeitverzögerung oder Modifikation. Die abhängige Komponente greift dabei lesend auf den Zustand der unabhängigen Komponente zu. Des Weiteren können diskrete Informationsflüsse zwischen zwei Komponenten modelliert werden. Dies erfolgt über die Transferierung von Datenobjekten. Beide Beziehungen werden grafisch wie folgt veranschaulicht:

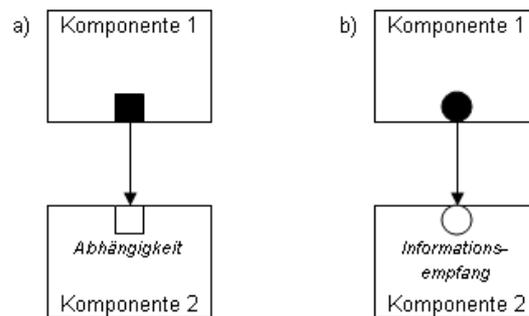


Abbildung 2.5 - grafische Darstellung der Kommunikation zwischen Komponenten: (a) kausale Abhängigkeit (b) diskreter Informationsfluss

Das Referenzmodell selbst kann als geschlossenes System PECS\_World betrachtet werden (Abbildung 2.6 [Urb04]), welches die Komponente Environment für die Umweltrepräsentation und die Komponente Connector für den Nachrichtenaustausch zwischen den Agenten enthält. Weiterhin ist eine definierte Zahl der Komponente Agent vorhanden, wobei diese der Anzahl der abzubildenden Personen im Modell entspricht.

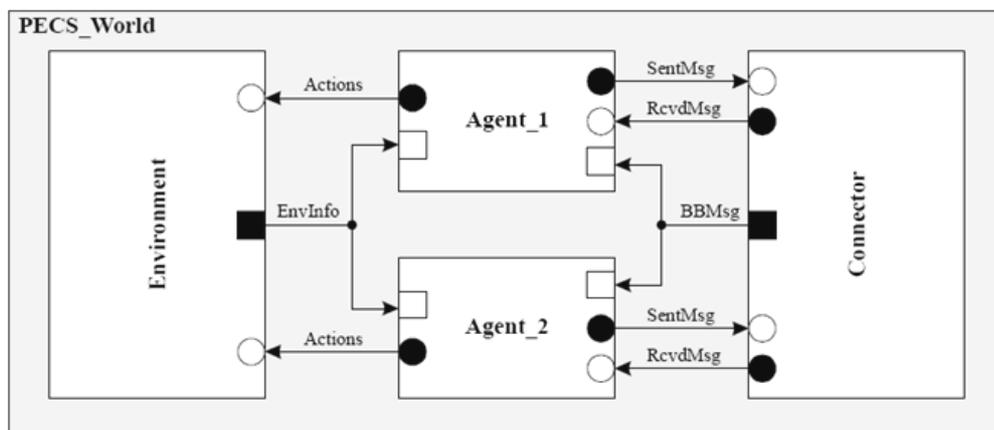


Abbildung 2.6 - Übersicht über die Komponenten des PECS-Referenzmodells und deren Interaktion

PECS\_World ist ein geschlossenes System, da es nicht von externen Faktoren abhängig ist. Die enthaltenen Komponenten hingegen sind offene Systeme, da sie mit anderen Komponenten interagieren. So besteht eine kausale Abhängigkeit zwischen den Agenten und Environment, dargestellt durch die Verbindung EnvInfo, über die jeder Agent Informationen über die aktuellen Zustände der Umwelt erhält. Eine weitere Abhängigkeit besteht zwischen den Agenten und der Komponente Connector durch die Verbindung BBMsg, über die Informationen abgefragt werden können, die auf dort definierten Blackboards hinterlegt sind. Diese für den Agenten externen Informationen werden in der Komponente Agent verarbeitet und bilden die Grundlage für die Berechnung auszuführender Aktionen. Diese Aktionen werden entweder als Datenobjekte Actions an Environment übermittelt, wenn die Umwelt vom Agenten modifiziert werden soll. Oder sie werden an Connector weitergeleitet, wenn der Agent Nachrichten mit anderen Agenten austauschen will. Dabei handelt es sich um Nachrichten, die nicht auf einem Blackboard und damit öffentlich allen Agenten zur Verfügung gestellt werden sollen, sondern solche, die direkt adressiert sind. Wenn ein Agent eine solche Nachricht übermitteln will, wird ein entsprechendes Datenobjekt über die Schnittstelle SentMsg an Connector gesendet, wo die Weiterleitung gesteuert wird. Der Empfang einer direkt adressierten Nachricht erfolgt über RcvdMsg. Die Komponente Agent steht dabei im Mittelpunkt der Architektur und dient der vollständigen Abbildung menschlicher Zustände und Entscheidungsprozesse. Aus diesem Grund ist ihr Aufbau entsprechend umfangreich und bedarf einer detaillierten Betrachtung.

Um komplexe Verhaltenssteuerungen der Agenten modellieren zu können, müssen alle Einfluss nehmenden psychosomatischen Größen des Menschen in die Berechnung der internen und externen Aktionen mit einbezogen werden. Dabei spielen physische und emotionale Zustände ebenso eine Rolle wie kognitive Fähigkeiten und die soziale Positionierung in der Gruppe. Aufgrund dessen betrachtet das PECS-Referenzmodell genau physische, emotionale, kognitive und soziale Zustände und Prozesse des Agenten. Daraus ergeben sich der Name des Referenzmodells und die vier großen Komponenten im Agenten (Abbildung 2.7 [Urb04]).

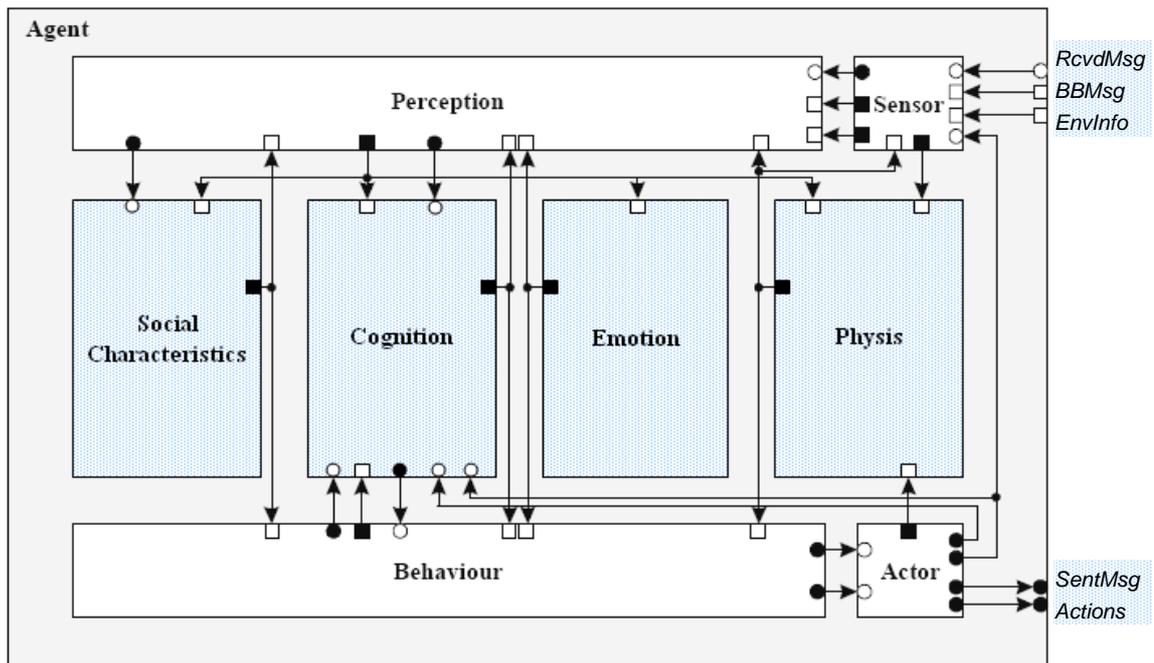


Abbildung 2.7 - Aufbau eines PECS-Agenten

Alle acht Komponenten des Agenten können in drei Kategorien eingeteilt werden. Die erste Kategorie umfasst die Komponenten Sensor und Perception, die für die Weiterleitung und Verarbeitung sensorischer Informationen verantwortlich sind. Dazu liest Sensor die bereits beschriebenen kausalen Abhängigkeiten EnvInfo und BBMsg und empfängt Nachrichten über die Schnittstelle RcvdMsg. Die Informationen werden je nach Bedeutung entweder an Physis oder Perception weitergeleitet. Dass nicht nur eine kausale Abhängigkeit von Sensor nach Physis, sondern auch umgekehrt besteht, soll das sensorische Aufnahmevermögen des Agenten in Abhängigkeit seines physischen Zustandes, zum Beispiel bei vermindertem Sehvermögen, vereinfacht abbilden. Die Komponente Perception ist für die Bereitstellung und Verteilung der Informationen – intern auch als Perzepte bezeichnet – an die verantwortlichen Komponenten zuständig und kann dabei gewünschte Wahrnehmungsfunktionen modellieren. So ist es zum Beispiel denkbar, dass aufgrund einer Analyse der externen und internen Situation nur die für die Situation als relevant erachteten Perzepte weitergeleitet werden, so dass nicht alle Informationen bewusst wahrgenommen werden. Des Weiteren lässt sich hier auch die Unschärfe von Informationen modellieren, indem die Inputdaten durch Störgrößen modifiziert werden.

In der zweiten Kategorie, gebildet von *Physis*, *Emotion*, *Cognition* und *Social Characteristics*, werden der interne Zustand sowie die zugehörigen Zustandsdynamiken des Agenten beschrieben. Sie sind für die Namensgebung des Referenzmodells verantwortlich und stehen im Zentrum der Architektur. Diese vier Komponenten werden im weiteren Verlauf der Arbeit auch zusammengefasst als PECS-Klassen bezeichnet. Dort sollten all jene Größen integriert werden, die für die gegebene Fragestellung von Bedeutung sind und den internen Zustand oder auch die Outputs des Agenten beeinflussen. Da interne Zustände häufig in enger Wechselwirkung mit anderen internen oder externen Zuständen stehen, besteht für alle vier Komponenten eine kausale Abhängigkeit von *Perception*, um Zugriff auf diese zu erhalten. Ein Beispiel: Befinden sich Menschen in einem geschlossenen Raum, der sich mit Rauch füllt, wird sich wahrscheinlich ein Gefühl von Angst oder Panik als emotionale Reaktion einstellen. Sie könnte wiederum bewirken, dass soziale Distanzen, definiert in *Social Characteristics*, verringert werden und sich Gruppen bilden, ebenso könnten Menschen nervös reagieren und vermehrt im Raum hin und her laufen. Der Rauch selbst als externe Umweltgröße beeinträchtigt das Sehvermögen und die Orientierung und hat damit Einfluss auf die Komponente *Physis*. Des Weiteren kann aufkommende Toxizität durch Sauerstoffmangel die Wahrnehmung stark beeinträchtigen, was in *Perception* modelliert werden könnte. Eine besondere Stellung nimmt die Komponente *Cognition* ein, die nicht nur die kognitiven Zustände und Fähigkeiten, wie zum Beispiel Intelligenz oder Wissensstand des Agenten abbilden soll, sondern sowohl Umwelt- als auch Selbstbildnis verwaltet und für die Auswahl von Handlungszielen und Erstellung entsprechender Handlungspläne verantwortlich ist.

In Abhängigkeit von der Problemstellung bei der Simulation müssen nicht immer alle vier PECS-Klassen im Modell berücksichtigt werden. Sie können stattdessen beliebig kombiniert werden. Entscheidend ist jedoch, dass es möglich sein muss, auch komplexe Modelle aufzubauen, die alle vier Komponenten enthalten und deren Wechselwirkungen nicht außer Acht lassen [Sch01].

Die dritte Kategorie in der Agenten-Komponente wird von *Behaviour* und *Actor* gebildet und ist für die Modellierung der Verhaltenssteuerung und die Realisierung der Aktionen des Agenten verantwortlich. In *Behaviour* wird die Ausführung der in *Cognition* erstellten Handlungspläne gesteuert. Des Weiteren werden dort die Regeln verwaltet, die angeben, welche Verhaltensweisen der Agent in Abhängigkeit seiner internen Zustände zur Ausführung auswählt. In der Komponente *Actor* werden dann

die erstellten Ausführungsanordnungen für interne und externe Aktionen abgearbeitet. Nach der Bearbeitung werden Nachrichten an diejenigen Komponenten verschickt – sowohl interne als auch externe, deren Zustand durch die Ausführung der Aktion beeinflusst wurde. Bei externen Aktionen, also bei Nachrichtenaustausch mit anderen Agenten oder Modifizierung der Umwelt, werden die entsprechenden Informationen an Connector und Environment gesendet. Bei internen Aktionen erfolgt eine Rückmeldung an Sensor, sodass von dort aus eine Verteilung der neuen Informationen erfolgt und in allen betroffenen Komponenten die entsprechenden Konsequenzen berechnet werden können. Auf diese Weise wird eine kontinuierliche Berechnung der Zustände von Agent und Umwelt ausgeführt.

## 3 Adaption des Referenzmodells an SimWalk

Das beschriebene PECS-Referenzmodell ist als Konstruktionsschema für ein Gesamtsystem entwickelt, in dem alle Vorgänge – Veränderungen in der Umwelt sowie Zustände und Aktionen des Agenten – abgebildet werden. Die Herausforderung besteht nun darin, die Vorgaben aus dem Modell im Hinblick auf einen bereits vorhandenen Simulator anzupassen. Dabei soll soweit möglich nur zu einem geringen Maße in den bestehenden Programmcode von SimWalk eingegriffen werden. Das bedeutet, dass vornehmlich von Seiten des Referenzmodells Anpassungen am vorgegebenen Rahmen erforderlich sind. Trotzdem sollte das Referenzmodell noch als Grundlage des umgesetzten Entwurfes erkenntlich sein, um den entwickelten Programmcode nachvollziehbar zu machen und zukünftige interne Erweiterungen in einem fest definierten, vollständigen Rahmen ermöglichen zu können. Die Weiterentwicklungen, sowohl von SimWalk als auch der PECS-Komponenten, sollen dabei flexibel und unabhängig voneinander erfolgen können. Das heißt, bei einer Veränderung der Implementierung von SimWalk soll der Anpassungsaufwand für die PECS-Berechnungen so gering wie möglich sein und umgekehrt.

Da die Simulation in SimWalk mithilfe der Agententechnologie von je her rechenintensiv ist, soll weiterhin der zusätzliche Zeitaufwand für einen Simulationslauf minimal gehalten werden. Der Umfang der zusätzlichen Berechnungen muss sich auf die wichtigsten, für die Personenströme relevanten Einflüsse beschränken.

### 3.1 Anforderungen an einen Agenten in SimWalk

Das PECS-Referenzmodell ist entwickelt worden, um den Rahmen für komplexe Wechselwirkungen und Mechanismen nicht nur in der Umwelt sondern vor allem im Agenten vorzugeben. Der grundsätzliche Umfang der Komponente Agent ermöglicht es, einen intelligenten Agenten mit kognitiven Fähigkeiten abzubilden. In Abhängigkeit der Problemstellung bei der Simulation sollte jedoch entschieden werden, ob der

volle Umfang des Konzeptes erforderlich ist, oder ob einzelne Bestandteile keinen weiteren messbaren Nutzen bringen und stattdessen zusätzliche Rechenkapazitäten erfordern. Als »messbarer Nutzen« soll in diesem Fall in Anlehnung an die Einsatzmöglichkeiten von SimWalk eine deutliche Veränderung in den zu beobachtenden Personenströmen bezeichnet werden.

Für die Simulation eines Solidarnetzwerkes zum Beispiel ist das Interagieren der Agenten auf emotionaler und sozialer Ebene einer der Hauptaspekte, der prognostiziert werden soll. Derartige Simulationen dienen der Abbildung des »Zusammenlebens« der Agenten. So wird der Inhalt der Komponenten Emotion und Social Characteristics entsprechend detailliert menschliche Parameter in diesen Bereichen umsetzen. Auch die Erstellung von Handlungszielen und -plänen wird sich an gemeinnützigen Zielen und, neben den eigenen Bedürfnissen, auch an den Bedürfnissen anderer Agenten ausrichten, die das Zusammenleben beeinflussen. Im Gegensatz dazu spielen physische Aspekte wie zum Beispiel das Sehvermögen oder die körperliche Ermüdung eine stark untergeordnete, wenn nicht sogar vernachlässigbare Rolle. In einer Simulation, in der jedoch die tatsächliche Bewegung der Agenten im Vordergrund steht – wie es bei SimWalk der Fall ist, muss das Bewegungsvermögen und die damit in Zusammenhang stehenden Parameter und Prozesse detaillierter abgebildet werden. Die weiteren erforderlichen Inhalte der PECS-Klassen – vor allem der mentalen Einheiten Emotion, Cognition und Social Characteristics – variieren situationsabhängig und bedürfen einer getrennten Betrachtung. Die physischen Faktoren werden als prinzipiell identisch angesehen, nehmen jedoch in Abhängigkeit der Situation mehr oder weniger Einfluss und werden ihrerseits auch mehr oder weniger von extern beeinflusst.

Zunächst soll bei der Beurteilung der mentalen Komponenten auf Situationen ohne besondere externe Einflussfaktoren eingegangen werden. Emotional unterscheiden sich Menschen in diesen Situation nur geringfügig voneinander. Sicher verfügen sie über unterschiedlich starke Ausprägungen verschiedenster innerer Zustände wie Angst oder Freude, doch werden diese in normalen »Fortbewegungssituationen« nicht signifikant auffallen. Ähnlich verhält es sich mit sozialen Bedürfnissen. Menschen nehmen in Abhängigkeit ihrer Umgebung verschiedene soziale Rollen an und entwickeln ebenso verschiedene Bedürfnisse. So definiert sich ein Mensch zum Beispiel im Kreis der Familie anders als am Arbeitsplatz. Diese verschiedenen Ausprägungen der Persönlichkeit in Abhängigkeit der Umgebung werden in der Psychologie als ökologi-

sche Selbst-Konzepte definiert [PSY]. Doch als Fußgänger verhalten sich die meisten Menschen eher passiv und soziale Faktoren stehen im Hintergrund, da die Bewegung lediglich dem Wechsel von einem Ort zum anderen dient. Ein weiterer Aspekt ist, dass Menschen, die sich an öffentlichen Plätzen begegnen, häufig einander unbekannt sind und damit ein geringeres Zusammengehörigkeits- und dementsprechendes Verantwortungsgefühl füreinander haben. Aus diesem Grund verfolgen Menschen beinahe ausschließlich die eigenen Interessen – auch in Gefahrensituationen.

Das bedeutet jedoch nicht, dass dort keine sozialen Bedürfnisse oder Eigenschaften eine Rolle spielen. Physisch oder psychisch starke Menschen nehmen in derartigen Situationen oftmals eine Art Führungs- oder Heldenrolle ein. Verängstigte oder orientierungslose Menschen tendieren dazu, sich solchen Personen unterzuordnen. So kann eine gewisse Gruppenbildung beobachtet werden. Auch die Abstände, die Personen zu Hindernissen und anderen Menschen einhalten – die Größen »Object Range« und »Pedestrian Interaction Range« in SimWalk – nehmen einen weniger hohen Stellenwert bei der Wahl des zu laufenden Weges ein und werden für gewöhnlich verringert. Die emotionalen Parameter können in derartigen Situationen stark schwanken und weitere interne Zustände beeinflussen sowie zur Ausführung verschiedener Verhaltensweisen führen. So ist das Gesichtsfeld eines in Panik geratenen Menschen oftmals verengt – der Zustand wird dementsprechend häufig als Tunnelblick bezeichnet – und kann sich unter anderem in Orientierungslosigkeit und häufigem Richtungswechsel in der Bewegung auswirken. Dies ist ein Faktor, der bei der Simulation eines Evakuierungsszenarios nicht zu vernachlässigen ist, da Menschen in diesem Zustand mehr Zeit benötigen, um Ausgänge zu finden und dabei gleichzeitig andere Menschen behindern.

Zusammenfassend sollten also emotionale und soziale Eigenschaften und Fähigkeiten genau wie die physischen Aspekte berücksichtigt werden. Dies kann aufgrund der Komplexität derartiger Wechselwirkungen jedoch nur im Rahmen einer groben Annäherung geschehen und muss auf solche Faktoren und Prozesse beschränkt werden, die Einfluss auf die Bewegung und die damit in Zusammenhang stehenden Parameter nehmen.

Um die Anforderungen an die kognitiven Fähigkeiten eines Agenten in SimWalk beurteilen zu können, muss abgewogen werden, in welchem Ausmaß die Erstellung eines Selbstbildnisses und Gesamteindrucks der Umwelt für die dem Agenten gegebene Aufgabenstellung von Bedeutung ist. Ferner sollen im Rahmen kognitiver Fähig-

keiten für gewöhnlich nicht nur Entscheidungen getroffen und Ziele entwickelt sondern auch Gedächtnisleistungen ausgeprägt werden [Fer01]. Inwiefern die Abbildung dieser Fähigkeiten möglich ist, hängt unter anderem davon ab, wie detailliert dem Agenten Informationen über interne und externe Zustände zur Verfügung stehen. Bei der Entscheidung über die Notwendigkeit spielen zahlreiche weitere Faktoren eine Rolle.

In Anbetracht der möglichen Aufgaben, die sich einem Agenten in SimWalk stellen – als »Ziel finden« und »Objekten ausweichen« zusammenfassbar – stellt sich die Frage, welche Art von Handlungszielen und -plänen ein solcher Agent aufstellen können muss. Es ist nach dem momentanen Entwicklungsstand des Tools nicht vorgesehen, dass der Agent in der Lage sein soll, seine Umwelt aktiv zu gestalten, denn die gesamte Umgebung wird nur durch eine geringe Zahl von Parametern repräsentiert und das jeweilige zugrunde gelegte CAD-Modell ist nicht veränderbar. Wenn die Umwelt nicht modifizierbar ist, benötigt der Agent dementsprechend keine Fähigkeiten, um Handlungsziele und -pläne zur Einwirkung auf örtliche Gegebenheiten erstellen zu können. Wenn man Handlungsziele jedoch als Bewegungsziele auslegt, also das eigentliche Ziel der Handlung das Ankommen an einem räumlichen Zielpunkt ist, dann muss ein Agent, bei dem die Fähigkeit zur physischen Fortbewegung im Vordergrund steht, in der Lage sein, solche Ziele autonom zu definieren. Die Notwendigkeit für ein solches Verhalten tritt immer dann auf, wenn der Agent aufgrund interner oder externer Bedingungen gezwungen ist, seinen anzusteuernden Zielpunkt – zum Beispiel vom Anwender als Teil des Pfades deklariert – abzuändern. Die Erstellung eines expliziten Handlungsplanes ist dabei nicht erforderlich, denn die Agenten besitzen bereits alle Fähigkeiten, um sich von einem Start- zu einem Zielpunkt zu bewegen. Es ändert sich nur die Position des Zieles. Und da das Selbst- und Umweltbildnis als Grundlage für die Erstellung von Zielen und Plänen dient, können die Umfänge dementsprechend dort stark eingeschränkt werden.

Wenn eine Handlung lediglich das Bestimmen eines Zielpunktes zu einem gegebenen Zeitpunkt an einer gegebenen Position im Raum ist, dann sind die auszuführenden mentalen Prozesse zur Erstellung des Planes in jeder Situation zwar durch dieselben Regeln vorgegeben, es ist jedoch nicht möglich, anhand vorangegangener Situationen einen Vorteil durch das Erinnerungsvermögen zu ziehen. Die Gedächtnisleistung sollte es vor allem ermöglichen, anhand einer gegebenen Situation, definiert durch eine Zahl innerer und äußerer Zustände, zu überprüfen, ob sich der Agent schon einmal in

einer solchen Situation befunden hat und welche Handlung dort ausgeführt wurde. Wenn die Situation jedoch nur durch Raum und Zeit definiert ist, dann ist es kaum möglich, Parallelen zur Vergangenheit zu ziehen. An jeder Position muss operativ entschieden werden, wohin sich der Agent als nächstes bewegen soll. Ein Protokollgedächtnis soll für das Agentenmodell aus diesem Grund nicht umgesetzt werden.

Nach der Betrachtung interner Aktionen im Rahmen psychosomatischer und kognitiver Prozesse bleibt die Beurteilung der Rolle der kommunikativen Fähigkeiten eines Agenten als externe Aktionsmöglichkeit. Grundsätzlich kann ein Agent ausschließlich die ihm selbst bewussten Informationen weitergeben. Dazu gehören der eigene Zustand und der seiner aktuellen Umgebung sowie die von ihm geplanten Handlungen. Nachrichten über Eigenschaften der Umwelt zu verschicken ist dann sinnvoll, wenn dort lokal beschränkte Ereignisse oder Zustände eingetreten sind. Dies kann zum Beispiel der Fall sein, wenn in einer Gefahrensituation keine globale Benachrichtigung per Alarm ausgelöst wird, sondern die Gefahr nur in der unmittelbaren Umgebung der Gefahrenquelle ersichtlich ist. Die Information wird sich dann von dort über den Raum verteilen, indem jeder Agent, der bereits eine Nachricht über die Situation erhalten hat, diese an alle innerhalb einer Kommunikationsdistanz von ihm aus erreichbaren Agenten weitergibt.

Aufgrund der bereits erwähnten »persönlichen Distanz« zwischen einander unbekanntem Menschen sollen Informationen über den eigenen Zustand nicht weitergegeben werden, jedoch spielen die entwickelten Handlungsziele eine Rolle. Ein Handlungsziel, also ein definierter Zielpunkt, bietet dabei hingegen weder Anlass, diese Information auf einem Blackboard öffentlich bereitzustellen, noch einen anderen Agenten darüber zu informieren. Die Notwendigkeit besteht nur dann, wenn ein Agent orientierungslos ist und einen Zielpunkt »erfragen« möchte. Das heißt, Kommunikation als solche wird, solange es sich nicht um relevante Umweltinformationen handelt, ausschließlich auf Anforderung eines Agenten betrieben. Dieser Agent kann, wenn die Anfrage an einen sich in der Nähe befindlichen Agenten weitergeleitet werden konnte und dieser sich auf ein definiertes Ziel zu bewegt, den entsprechenden Zielpunkt übernehmen und sich ebenfalls dorthin bewegen.

Auf diese Art und Weise bleibt die Idee des Simulators SimWalk, Personenströme abzubilden, immer noch deutlich im Vordergrund. Trotzdem werden Faktoren betrachtet, die den Verlauf der Ströme beeinflussen können. Ziel sollte es sein, dem Agenten

ein deliberatives Verhalten zu ermöglichen. Das heißt, dass Handlungen im Hinblick auf ein Ziel bewusst ausgeführt werden. Eine Beschränkung auf die wichtigsten Aspekte menschlichen Verhaltens soll zunächst den zusätzlichen Rechenaufwand möglichst gering halten. Des Weiteren kann so auch die Entwicklungszeit für ein vollständiges Modul verringert und dem zu erwartenden Nutzen gerecht werden.

### 3.2 Anpassung der Komponente Agent

Für die Architektur der Komponente Agent bedeuten die beschriebenen Anforderungen, dass Bestandteile aus dem Modell entfernt oder in ihrer Bedeutung verändert werden müssen. Der im Folgenden vorgestellte Entwurf zeigt noch deutliche Verbindungen zum Referenzmodell, auch wenn die Funktionalität stark eingeschränkt wurde.

Beginnend mit dem Einlesen der externen Informationen (1) müssen im Agenten die folgenden Berechnungen ausgeführt werden:

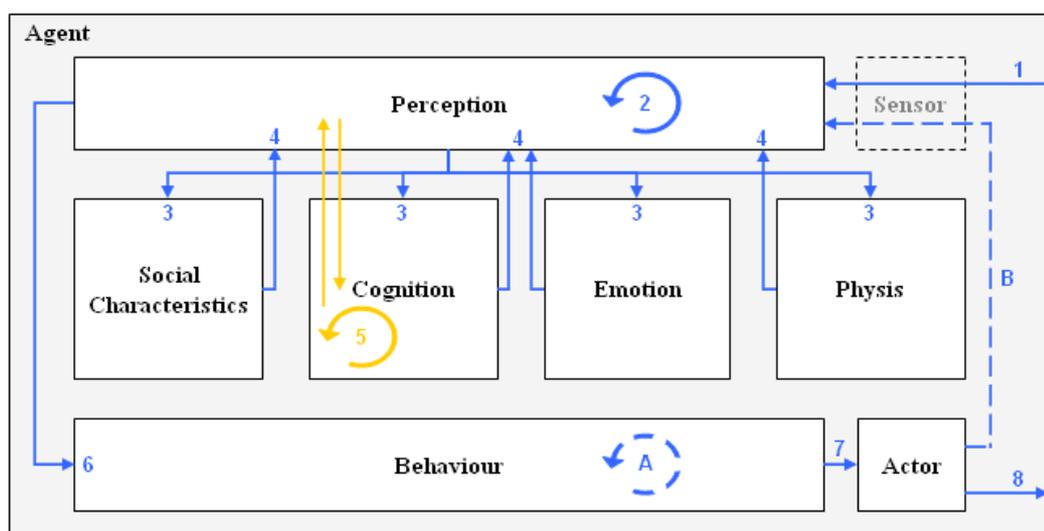


Abbildung 3.1 - Berechnungsabläufe im Agenten

Zunächst werden die Informationen aus der Umwelt sowie eingehende Nachrichten aus dem Connector gefiltert, um verschiedene Wahrnehmungsfunktionen modellieren zu können (2). Dies erfolgt nun ausschließlich gesteuert durch die Komponente Per-

ception, um die angedachte Verteilung der Filterung auf zwei Komponenten zu vermeiden und stattdessen zentralisiert erfolgen zu lassen. Sensor ist nicht mehr beteiligt und muss dementsprechend nicht im Modul implementiert werden. Die Filterung anhand physischer Einflussfaktoren erfolgt also ebenfalls durch Perception. Dabei muss jedoch festgelegt werden, dass diese Filterung zeitlich grundsätzlich vor der Abbildung weiterer Wahrnehmungsfunktionen ausgelöst wird, um die Idee der Komponente Sensor beizubehalten, dass Parameter, die aufgrund physischer Zustände nicht wahrnehmbar sind, nicht in die weiteren Berechnungen eingehen. Nach der Aufbereitung der Daten können sie je nach Zugehörigkeit auf die vier PECS-Klassen verteilt werden (3). Zu diesem Zeitpunkt wird der Agent also initialisiert. Das bedeutet, dass er durch das Setzen der internen Zustände in den vier Klassen entsprechend durch diese Eigenschaften charakterisiert wird. Dort erfolgt anschließend aufgrund der definierten Dynamiken die Neuberechnung der aktuellen psychosomatischen Zustände einschließlich solcher kognitiver Parameter, die nicht als Informationen an das Modul übergeben wurden, sondern in jeder Situation intern berechnet werden müssen. Dies können zum Beispiel Parameter zur Zufriedenheit oder dem Sicherheitsgefühl des Agenten sein. Die Berechnungsergebnisse werden wieder zurück an Perception gegeben (4). Nach dieser einmaligen Aktualisierung der Zustände wird überprüft, ob ein neues Handlungs- bzw. Bewegungsziel berechnet werden muss. Ist dies der Fall, wird die entsprechende Methodik – definiert in Cognition, ausgeführt (5). Sowohl das entsprechende Ziel, als auch die neuen Zustände aus Perception werden an Behaviour weitergegeben (6). Die Möglichkeit, Verhaltensregeln ohne explizites Handlungsziel ausschließlich aufgrund der internen Zustände zur Ausführung zu bringen, soll zunächst beibehalten werden (A), auch wenn folgend kein Prozessbeispiel dazu implementiert wird. Nach den Vorgaben des Referenzmodells soll jedoch Behaviour genau diese Aufgabe wahrnehmen können. Im Gegensatz zu den aktiven Denkprozessen in Cognition könnten hier einfache Reiz-Reaktions-Muster, also rein reaktives Verhalten des Menschen umgesetzt werden. Aufgrund dessen ist die Komponente zunächst im PECS-Modul vorgesehen, auch wenn sie dort in der momentanen Umsetzung keine echte Funktionalität besitzt, sondern lediglich die Daten an Actor weiterleitet (7). Es bleibt zu entscheiden, ob derartig detaillierte Reaktionen auf interne Zustände tatsächlich für die Personenstromsimulation abgebildet werden sollen. Dies müsste wahrscheinlich für die einzelnen in den speziellen Fällen möglichen Prozesse individuell entschieden werden und ist stark abhängig von den zukünftig tatsächlich umgesetzten

Zuständen. Von Actor aus werden die Werte anschließend zurück an den Simulator gegeben (8) und die neuen Eigenschaften und Ziele des Agenten können in SimWalk gespeichert und angewendet werden. Auch die Komponente Actor nimmt hier zunächst lediglich die Aufgaben der Parameterrückgabe wahr, kann jedoch in ihrer Funktionalität noch erweitert werden. So könnte von dort aus gesteuert werden, dass die neu berechneten Zustände und Ziele noch einmal an Perception gegeben werden (B), um eventuelle, daraus direkt resultierende Zustandsänderungen berechnen zu können. Dies könnte in Frage kommen, wenn die zeitlichen Abstände zwischen den Berechnung für einen Agenten sehr hoch liegen und Reaktionen damit zeitlich stark verzögert erfolgen würden.

Die Berechnungsabläufe sind derartig stark strukturell gegliedert, um mögliche Fehlerquellen auszuschließen. Nur wenn ein Teil der Berechnung komplett abgeschlossen ist, zum Beispiel die Aktualisierung der Zustände, kann der nächste Schritt ausgeführt werden – in diesem Fall dann die Berechnung eines Zieles, falls erforderlich. So ist gesichert, dass stets mit den vollständigen und korrekten Daten gerechnet wird. Es ist weiterhin zu beachten, dass die Berechnung der neuen Zustände auf der Grundlage der Gesamtmenge der übergebenen Zustände erfolgt. Es soll nicht möglich sein, eine Reihenfolge bei der Aktualisierung der PECS-Klassen vorzugeben, das heißt eine Komponente vor der anderen zu berechnen, um dann gleich die neuen Werte weiter zu verwenden. Alle Berechnungen eines Durchlaufes müssen auf denselben Daten basieren, da sonst Informationen verloren gehen könnten. Des Weiteren ist die Festlegung der »richtigen« Reihenfolge ein vermutlich nichttriviales Problem. Lediglich die Bestimmung der aktiv auszuführenden Handlungen erfolgt direkt auf der Grundlage der neuen Zustände und nicht erst im folgenden Berechnungsdurchlauf anhand der übergebenen Daten. Das bedeutet, die Aktualisierung der Parameter in den PECS-Klassen dient neben dem Wirken auf andere interne Parameter vornehmlich der Vorbereitung für die Erstellung von Handlungszielen. Ein geeignetes Ziel wird stets unter Berücksichtigung aller eventuell Einfluss nehmenden, aktuellen Zustände des Agenten ermittelt.

### 3.3 Zur Implementierung des PECS-Moduls

Zur Erstellung eines geeigneten Klassendesigns, welches in C++ implementiert werden kann, müssen weitere Analysen durchgeführt werden. Ergebnis ist ein objektorientiertes Designmodell (Abbildung 3.2), welches als Vorlage für die Programmierung dient. Algorithmen sind stets in unterschiedlichen Formen darstell- und interpretierbar. Das OO-Paradigma beschreibt eine Algorithmenform, die auf der Grundlage miteinander kommunizierender Objekte beruht. Aufgrund dieses Paradigmas ergeben sich weitere Änderungen am bisherigen Konzept. Jede Komponente im Entwurf kann dabei als eine Klasse interpretiert werden, die Daten und Funktionen enthält und so auch als Klasse in C++ definiert wird. Bei der objektorientierten Programmierung werden Daten und darauf auszuführende Logiken und Methoden zu Objekten zusammengefasst. Das heißt, dass bei der Berechnung im Modul nicht die Klassen selbst, sondern Instanzen von ihnen als Objekte verwendet werden. Diese müssen als solche instanziiert werden. Die Implementierung einer Klasse bildet damit den Konstruktionsrahmen, die Klasseninstanzen – die Objekte – werden während der Laufzeit des Programms erzeugt, mit den entsprechenden Daten versehen und für die Berechnungen verwendet. Sind die Berechnungen abgeschlossen geben die Objekte die neuen Werte an die Klasse bzw. deren Objekt zurück, in der sie erzeugt wurden und können anschließend entfernt werden. Aufgrund dessen ist es sinnvoll, genau diejenige Klasse von SimWalk aus anzusprechen, die als letzte Daten wieder an den Simulator zurückgibt und das ist Actor.

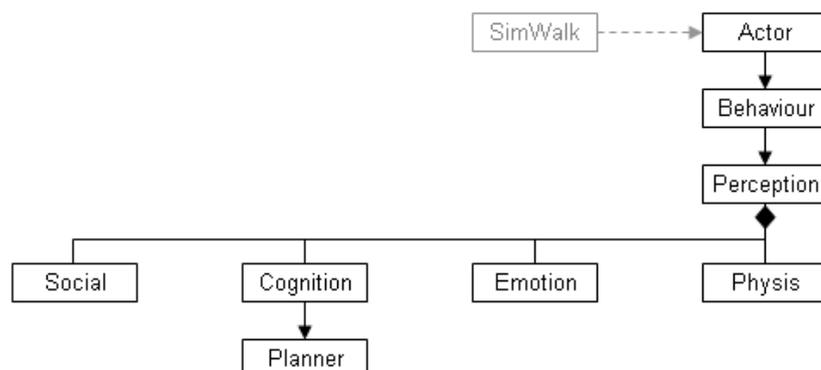


Abbildung 3.2 - Klassendiagramm des PECS-Moduls

In Actor wird ein Objekt der Klasse Behaviour erzeugt, in welchem wiederum ein Objekt der Klasse Perception definiert ist. Bei der Initialisierung von Perception werden die vier PECS-Klassen als Objekte instanziiert und die entsprechenden internen Zustände dort gesetzt. Anschließend können diese im Rahmen der Informationsfilterung noch einmal modifiziert werden. Dann beginnt die Berechnung der in den PECS-Klassen definierten Zustandsdynamiken. Nach der Aktualisierung aller Zustände und Rückgabe an Perception werden die Methoden im Objekt Cognition angestoßen, die für die Überprüfung der Notwendigkeit einer Anpassung des Zielpunktes verantwortlich sind. Muss ein neues Ziel berechnet werden, erfolgt dies über die im Objekt Planner definierte Methodik. Der neue Zielpunkt wird ebenfalls an Perception zurückgegeben.

Im Klassendesign für das umzusetzende Modul ist festgelegt, dass die Erzeugung des Objektes von Perception stets mit einer Initialisierung der vier Objekte der PECS-Klassen einhergeht, was im Diagramm als Komposition durch das Rautensymbol dargestellt ist. Diese Komposition ist eine starke Beziehung zwischen den beteiligten Komponenten und bedeutet, dass die vier PECS-Objekte nur in Verbindung mit Perception existieren können und mit der Destruktion des Objektes nach Rückgabe der Werte an Behaviour ebenfalls zerstört werden.

Nach der Weiterleitung der Daten an den Simulator kann das Objekt Actor und somit das PECS-Modul für den aktuell berechneten Agent gelöscht und eine neue Instanz für den nächsten Agenten erzeugt werden.

Es wird bereits hier deutlich, dass die Komponente Perception im Modul eine stärkere Verwaltungsrolle einnimmt als im ursprünglichen Konzept des Referenzmodells. Nur von dort aus können und sollen die Informationen kontrolliert werden. Eine direkte Verbindung zwischen Behaviour und den PECS-Klassen besteht nicht mehr, denn die neu berechneten Werte liegen in Perception und sollten erst bei der Destruktion zugänglich gemacht werden, um eine mögliche Fehlerquelle auszuschließen und die strukturell gegliederten Berechnungsabläufe nicht zu durchbrechen.

Unterstützt wird diese Anforderung durch die so genannte Kapselung, welche Teil des OO-Paradigmas ist [Mey06]. Mithilfe der Kapselung kann der Zugriff auf definierte Objekte gesteuert werden. Dies ist sinnvoll, wenn ein Objekt nicht in der Lage sein soll, den Inhalt eines anderen Objektes zu manipulieren. So hat zum Beispiel Behaviour keinen direkten Zugriff auf die Parameter in Perception, sondern kann diese nur

über eine Schnittstelle, die in Perception definiert ist, anfordern. Die Bereitstellung der Information wird dementsprechend durch Perception und nicht durch Behaviour gesteuert, wodurch eine Manipulation der Daten durch die Komponente vermieden werden kann.

Und auch die PECS-Klassen haben keinen direkten Zugriff aufeinander. Das führt dazu, dass Perception auch hier für die Bereitstellung der Daten verantwortlich ist, wenn für die Berechnung von Zustandsdynamiken eines PECS-Objektes Informationen aus den jeweils anderen drei Objekten erforderlich sind (Abbildung 3.3). Um die Anforderung erfüllen zu können, dass ausschließlich mit den übergebenen und nicht mit bereits aktualisierten Zuständen gerechnet wird, müssen alte und neue Zustände getrennt verwaltet werden. Die Sicherung der übergebenen Parameter übernehmen die zugehörigen Objekte mithilfe definierter Zugriffsrechte; die Daten werden für externe Objekte gekapselt.

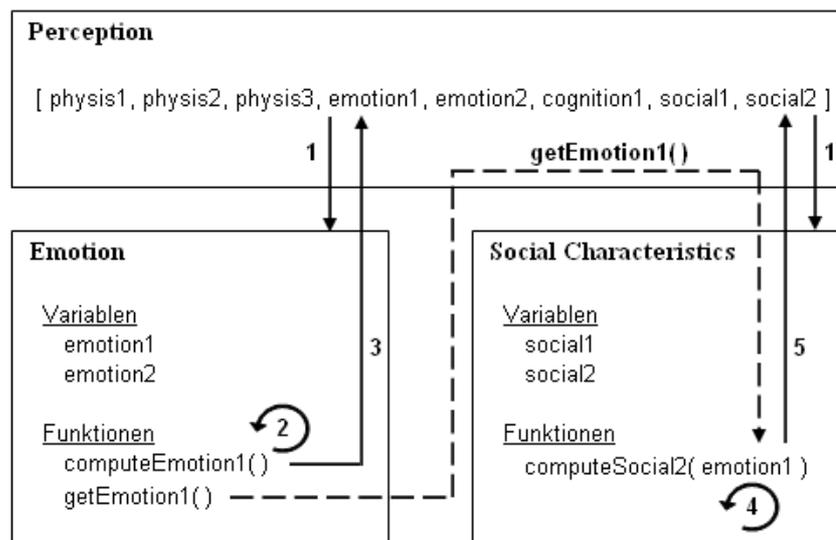


Abbildung 3.3 - Kapselung der Daten im Agenten

Die entsprechenden Variablen in den vier Objekten werden dazu mit den übergebenen Zuständen von Perception aus initialisiert (1) und sind dann als geschützt deklariert. Das bedeutet, dass lediglich das Objekt selbst Zugriff darauf hat. So kann die Dynamik eines Zustandes, der nicht von weiteren Einflussfaktoren anderer PECS-Objekte abhängig ist, direkt anhand der Variablen im Objekt berechnet werden (2). Dabei darf der neue Wert jedoch nicht im Objekt überschrieben, sondern muss an Perception zurückgegeben und dort abgelegt werden (3). Wenn nun ein Zustand berech-

net werden soll, der genau von diesem berechneten Parameter abhängt, darf nicht der aktualisierte sondern nur der originale Wert verwendet werden (4). Im abgebildeten Beispiel hat Social Characteristics keinen Zugriff auf den Wert emotion1 im Objekt Emotion. Es muss also Perception diesen über die von Emotion definierte Schnittstelle getEmotion1( ) bereitstellen. Der neu berechnete Parameter social2 kann anschließend in Perception überschrieben werden (5).

Es wäre ebenso möglich, die Originalwerte in Perception zu behalten und die Variablen in den PECS-Objekten zu aktualisieren. Dann wäre es nach der Berechnung aller Zustandsdynamiken jedoch erforderlich, alle Variablen aus den Objekten über eine get( )-Schnittstelle zurück nach Perception zu kopieren, unabhängig davon, ob diese verändert wurden oder nicht. Im aktuellen Entwurf müssen diese Schnittstellen nur auf Anforderung genutzt werden und unnötige Abfragen können vermieden werden. Eine weitere Fehlerquelle kann umgangen werden, wenn sich die Implementierung des Moduls am Singleton Entwurfsmuster orientiert. Dies ist ein spezielles Konzeptionsschema für die Software-Entwicklung, welches sichert, dass jedes Objekt nur genau einmal innerhalb eines anderen Objektes instanziiert werden kann [Gam04]. Wenn es in Perception möglich wäre, eine der vier PECS-Klassen zweimal als Objekt zu instanzieren – zum Beispiel zwei Objekte der Klasse Physis, so würde dies zu Unstimmigkeiten in den internen Zuständen führen. Das erste Objekt würde die Aktualisierung der physischen Zustände übernehmen und diese an Perception zurückgeben. Wenn nun ein zweites Objekt der Klasse Physis erzeugt wird, bekommt dieses die bereits berechneten Parameter zugewiesen und nicht die aus dem Simulator übergebenen Werte. Objekte der Klassen Emotion, Cognition oder Social Characteristics würden nun mit manipulierten Daten rechnen, wenn sie physische Parameter aus diesem Objekt für ihre Dynamikberechnungen benötigten. Ist es jedoch nur erlaubt eine Instanz eines Objektes zu erzeugen, würde bei der Initialisierung des zweiten Objektes ein Fehler zurückgegeben werden, so dass es zu keiner Berechnung der Parameter kommt. Der Programmcode muss dann korrigiert werden.

Unter Einbeziehung der Möglichkeiten, die die objektorientierte Modellierung und Programmierung bietet, kann bereits in den ersten Entwicklungsstufen festgelegt werden, nach welchen Prinzipien das PECS-Modul arbeitet, was auf der einen Seite möglich und was auf der anderen Seite dabei nicht erlaubt sein soll. Diese Bedingungen müssen bei der Weiterentwicklung beachtet werden, da auf diese Weise die korrekte Behandlung der Daten sichergestellt werden kann.

### 3.4 Überlegungen zur Integration in den Simulator

Zur Integration des gesamten Modells müssen zunächst Überlegungen angestellt werden, worin sich die Ideen des PECS-Referenzmodells von der bisherigen Implementierung SimWalks unterscheiden und auf welche Weise das Konzept in den Simulator integriert werden soll. Diese werden die Gesamtarchitektur des Entwurfes bestimmen.

Um das zu entwickelnde Modell möglichst flexibel handhaben und unabhängig entwickeln zu können, wäre eine vollständige Trennung vom Simulator ideal. Jedoch ist eine reine Ergänzung von SimWalk über ein komplett extern realisiertes Referenzmodell nicht sinnvoll, da dies die Anforderungen für erforderlichen Speicher und Rechenkapazität enorm erhöhen würde. Denn SimWalk verfügt bereits über ein Konzept zur Verwaltung der bereits vorhandenen Informationen über Umweltlayout und Agenten – sie werden vom Anwender bei der Erstellung des Simulationsmodells eingegeben und in XML-Dateien getrennt abgelegt. Diese Daten werden in der implementierten Methodik in SimWalk verwendet, sie wären aber auch für die Berechnungen im externen Modell erforderlich und müssten dementsprechend dort ebenfalls vorliegen. Nach jeder Modifikation von extern müssten dann wiederum die Daten im Simulator synchronisiert werden. Das bedeutet aufgrund der Fülle der Daten einen hohen zusätzlichen Zeit- und Rechenaufwand. Der bisherige Berechnungsaufwand in SimWalk ohne zusätzliches Modul kann für mehrere tausend Agenten selbst für eine kurze Simulationszeit schon im Stundenbereich liegen. Der hinzukommende Aufwand durch die Abbildung mehrerer Parameterdynamiken und weiterer Prozesse im Agenten sollte so gering wie möglich sein.

Aus diesem Grund wird das Referenzmodell soweit erforderlich in SimWalk integriert – die Umsetzung dessen wird von Seiten Savannah Simulations erfolgen. Vorschläge für die Vorgehensweise und den Umfang ergeben sich aus dem Entwurf der Komponente Agent und werden im weiteren Verlauf der Arbeit noch erläutert. Da für die Berechnung des Agenten neben seinen Eigenschaften nur eine begrenzte Anzahl zusätzlicher Informationen erforderlich sind, kann sie in einem eigenständigen Modul, dem PECS-Modul, realisiert werden. Dies bietet zusätzlich die Möglichkeit, die Umsetzung des Referenzmodells und den Einsatz im Simulator von Entwicklerseite aus kontrollieren zu können. Dieses Vorgehen bedeutet jedoch, dass der Agent und damit die vollständige Verarbeitung seiner Zustände und Dynamiken aus PECS\_World ausge-

lagert werden. Kausale Abhängigkeiten wie sie im Referenzmodell zum Beispiel mit EnvInfo vorgesehen sind, sind damit in dieser Form nicht mehr möglich. Beim Aufruf des Moduls über eine in SimWalk integrierte Schnittstelle müssen nun alle Informationen als Datenobjekte übermittelt werden.

Die Berechnung der Agentenbewegung in SimWalk erfolgt in sogenannten Berechnungsrunden für die Agenten nacheinander in diskreten Zeitschritten. Es ist damit nicht erforderlich, für jeden Agenten ein PECS-Modul zur Verfügung zu stellen. Stattdessen verbleiben die Daten im Simulator und dienen, wenn die Berechnung für den Agenten erfolgen soll, als Initialisierungsgrundlage für das Modul, welches dann den Agenten für die Berechnung repräsentiert. Kontinuierliche Zustandsänderungen im Agenten können damit nicht abgebildet werden, zumal fraglich ist, ob die kontinuierliche Betrachtung von Zuständen, in Anbetracht des resultierenden Rechenaufwandes, einen Nutzen für die Personenstromsimulation mit sich bringt. Allein unter dem Gesichtspunkt des Speicher- und Rechenaufwandes für das Lösen von Differentialgleichungen für mehrere tausend Agenten pro Simulation mit je einem Dutzend internen Zuständen scheidet diese Art der Umsetzung aus, da nicht die Reaktion Einzelner im Mittelpunkt steht, sondern das Verhalten von Menschenmassen.

Nach der Aktualisierung des Agenten können die neuen Daten in SimWalk gespeichert, die Instanz gelöscht und für den nächsten Agenten das Modul erneut instanziiert werden. Das gesamte PECS-Modul kann so als eine Klasse betrachtet werden, von der im Sinne des OO-Paradigmas für jeden Agenten ein entsprechendes Objekt instanziiert wird, wenn dessen Zustände und Prozesse berechnet werden sollen. Ferner ist im Hinblick auf eine verteilte Lösung der Simulation die Umsetzung im Rahmen einer externen Komponente ebenfalls sinnvoll. Es könnten auf diese Weise mehrere Instanzen des Moduls gleichzeitig durch verschiedene Prozessoren berechnet werden.

Die Umsetzung und Integration des Referenzmodell soll zunächst alle Möglichkeiten offenlassen, wann und in welchen Abständen das Modul aufgerufen werden soll. Prinzipiell könnte dies in jeder Berechnungsrunde geschehen. Inwiefern das sinnvoll ist, hängt davon ab, wie viele Agenten in der Simulation berechnet werden müssen und wie lang dementsprechend eine Runde dauert. Wenn die Dauer des Simulationslaufes aufgrund einer hohen Zahl von Agenten oder im Modul umgesetzter Prozesse durch die zusätzliche Berechnung stark erhöht wird, ist es möglich, die Berechnung für mehrere Runden auszusetzen. Das hat zur Folge, dass die Agenten verzögert auf

veränderte interne oder externe Bedingungen reagieren. Es muss also ein Kompromiss zwischen zusätzlichem Rechen- und Zeitaufwand sowie der Genauigkeit der Personenströme gefunden werden. Da der Anwender Simulationsmodelle mit einer beliebigen Agentenzahl erstellen kann, ist es sinnvoll, auch ihm selbst diese Entscheidung zu überlassen. Neben der Aktualisierung der Agenten in diskreten Zeitschritten sollte zusätzlich die Möglichkeit bestehen, eine Berechnungsrunde über das Modul ereignisbasiert, also aufgrund eines Ereignisses in der Umwelt, anzustoßen. So kann selbst bei einem vom Anwender hoch gewählten Berechnungsintervall auf besondere Situationen, zum Beispiel bei Gefahr oder einer anderweitigen globalen Veränderung, direkt in der nächsten Berechnungsrunde reagiert werden.

Weiterhin muss das Modul dem Agenten keine aktive Bewegung ermöglichen, da der bisher implementierte Bewegungsalgorithmus in SimWalk in seiner Funktion unverändert bleiben soll. Es ist jedoch möglich, die Berechnungsergebnisse aus dem PECS-Modul in Form einer Kraft in die Bewegung mit einfließen zu lassen, so wie es bereits mit den Kräften Object Pressure und Pedestrian Pressure realisiert ist. Wird für einen Agenten berechnet, dass er sich aufgrund eines inneren Antriebes auf einen bestimmten Punkt in der Umwelt zubewegen soll, kann für diese Koordinaten eine Kraft berechnet werden, die dem Potenzialfeld überlagert wird und den Agenten in die entsprechende Richtung zieht. Wird keine Kraft berechnet, bleibt die Kraft des Potenzialfeldes im Vordergrund.

Der Umfang und die Art und Weise der Integration der Komponenten Environment und Connector sind stark abhängig von den für den Agenten definierten Aufgaben. Die beiden Komponenten müssen also nur genau den Umfang haben, mit dem sie den Agenten bei den zu berechnenden Prozessen unterstützen können. Die Aufgabe des Connectors beschränkt sich von Seiten des Agenten auf die Bearbeitung einer Kommunikationsanfrage. Dazu hat die Komponente Zugriff auf alle Agentendaten und kann so die Anfrage für einen suchenden Agenten bearbeiten. Im Referenzmodell ist weiterhin vorgesehen, dass Connector globale, öffentliche Informationen auf Blackboards bereitstellt, die von allen Agenten mit Daten gefüllt werden. Wenn der Kommunikationsumfang eines Agenten jedoch im Rahmen der Personenstromsimulation beschränkt wird, ist die Idee der Blackboards so nicht mehr erforderlich. Trotzdem soll sie eine Verwendung finden. Diese steht nun eng im Zusammenhang mit den verfügbaren Umweltinformationen.

Wenn man vernachlässigt, dass die bisherige Methodik in SimWalk andere Agenten als Hindernisse in der Umgebung betrachtet, und sich die Umwelt durch die Bewegung der Agenten nach dieser Sichtweise dynamisch ändert, ist das Umweltmodell statisch. Ein Agent soll nicht in der Lage sein, die definierten Gebäudewände oder festen Abgrenzungen von Anlagen zu verändern. Wenn die Umwelt nicht durch Aktionen beeinflusst werden soll, ist keine Komponente erforderlich, die derartige Veränderungen berechnet oder verwaltet. Dies hat maßgeblichen Einfluss auf die Architektur des Entwurfes und erfordert einige Veränderungen an den Vorgaben des PECS-Referenzmodells (Abbildung 3.4).

Alle Informationen, die der Agent über seine Umwelt benötigt sind statisch oder werden von SimWalk verwaltet und aktualisiert. Das umfasst Hindernisse, anzusteuernde Punkte und ihre Eigenschaften sowie globale Zustände. Die Hindernisse werden direkt im Bewegungsalgorithmus verarbeitet, so dass nur noch Punkte und Zustände zur Berechnung an das PECS-Modul übergeben werden müssen. Diese Daten müssen zur Übergabe in geeigneter Form zusammengestellt sein. Bisher liegen die verfügbaren Informationen in einer XML-Datei vor und sind so nicht veränderbar. Nun sollen sie in Blackboards der Komponente Connector direkt in SimWalk verwaltet werden. Environment kann damit aus dem Referenzmodell entfernt werden.

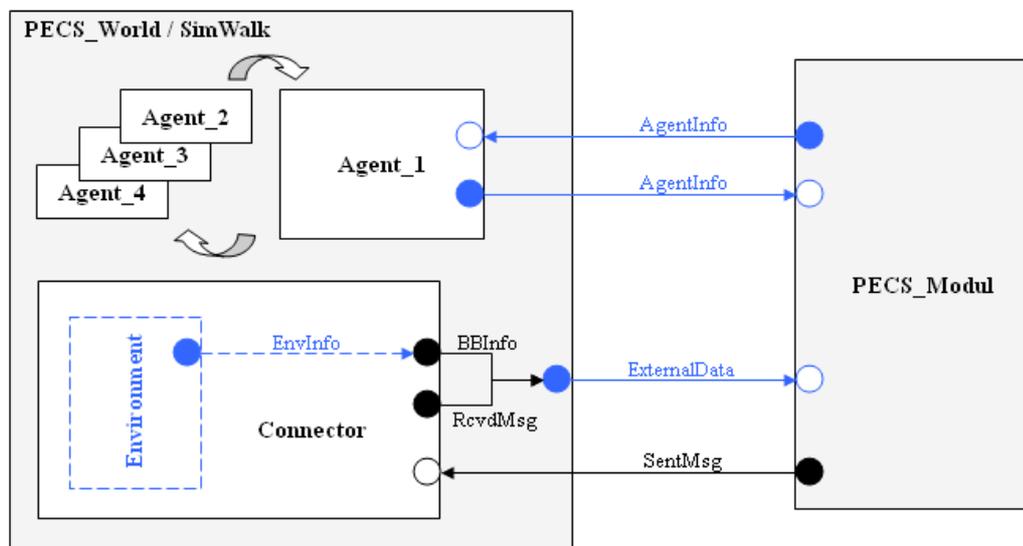


Abbildung 3.4 - Umstrukturierung von PECS\_World

Die benötigten externen Informationen werden bei der bisherigen Implementierung direkt von SimWalk verwaltet und können von dort an den Connector weitergegeben werden. Die Blackboards speichern also weiterhin globale, öffentliche Informationen, jedoch nun über die gesamte Simulationsumgebung. Die Komponente nimmt damit eine noch stärkere Verwaltungsrolle ein, als im PECS-Referenzmodell ursprünglich angedacht. Dies eröffnet die Möglichkeit, alle Daten, die der Agent neben seinen internen Informationen benötigt, in einem Datenpaket zusammenzufassen. Die Schnittstelle `SentMsg` vom Modul zum Connector dient der Übermittlung der beschriebenen Kommunikationsanforderung, wenn ein Agent kein räumliches Ziel findet, welches für die Erfüllung seines Handlungszieles geeignet wäre. Im Connector werden dann alle Agenten im Modell überprüft, die sich zum suchenden Agenten in einer vorgegebenen Kommunikationsreichweite befinden. Die Antwort wird in `RcvdMsg` im Datenobjekt `ExternalData` zurück an das Modul versandt.

Das bedeutet, die in SimWalk zu implementierende Schnittstelle umfasst nur wenige Objekte. Es müssen lediglich zwei Datenpakete an das PECS-Modul zur Berechnung übergeben werden: die Agentendaten in `AgentInfo` sowie Umweltdaten und Nachrichten in `ExternalData`. Aus dem Modul werden nach der Berechnung die aktualisierten Agentendaten sowie eventuelle Nachrichten zurückgegeben. Soll der Umfang oder die Art der Implementierung der Informationen verändert werden, muss damit lediglich die Methodik zur Zusammenstellung der Daten zur Übergabe, nicht aber die Schnittstelle selbst und damit auch nicht das Modul angepasst werden.

## 4 Parameter- und Prozessbeispiele

Eine wichtige Anforderung an die Konzeption des Moduls ist die einfache Erweiterbarkeit um Parameter und Prozesse, da es zunächst nur einen Rahmen bildet, der zu einem späteren Zeitpunkt mit Inhalt gefüllt werden soll. Aus diesem Grund wurden erste exemplarische Werte und Handlungsstränge in das Modul integriert. So kann aufgezeigt werden, auf welche Art und Weise diese definiert werden können. Des Weiteren sollen sie die Möglichkeit eröffnen, abschätzen zu können, wie viel Zeit die Berechnung für einen Agenten in Anspruch nimmt, um eine erste Prognose zu stellen, wie hoch der zusätzliche Berechnungsaufwand beim Simulationslauf ist.

Dazu wurden für jede der PECS-Klassen in der Komponente Agent Parameter festgelegt, die an das Modul beim Aufruf übergeben und anhand ebenso exemplarisch definierter Dynamiken neu berechnet werden. Falls erforderlich, bilden die veränderten Parameter dann die Grundlage für die Festlegung eines Handlungszieles.

### 4.1 Externe und interne Zustände

Um die Funktionsweise des implementierten PECS-Moduls testen zu können, wurde ein Rahmen programmiert, der die späteren Aufgaben von SimWalk abbildet. Das heißt, dort sind die Agentendaten abgelegt, es werden Veränderungen in der Umwelt angezeigt und entsprechend die Verwaltung der Informationen in der Komponente Connector gesteuert. Es werden jedoch keine tatsächlichen Bewegungen der Agenten abgebildet, da der Entwicklungsaufwand für eine selbst vereinfachte Nachbildung des bereits in SimWalk implementierten Algorithmus im Hinblick auf die Aussagekraft der Ergebnisse nicht gerechtfertigt ist.

Da Connector die Grundlage für die Zusammenstellung externer Informationen zur Übergabe an das PECS-Modul bildet, wurden dort verschiedene Blackboards definiert, die sowohl bereits in SimWalk vorhandene Informationen als auch solche beinhalten, die noch zukünftig integriert werden müssten (Abbildung 4.1). Das Blackboard

BB\_Air speichert Informationen über die Luftqualität und -temperatur der Umwelt, was vor allem in Gefahrensituationen an Bedeutung gewinnt. Entsprechend wurden solche Parameter für das Blackboard gewählt, die in derartigen Situationen eine Rolle spielen könnten. In Anlehnung daran existiert das Blackboard BB\_Alarm zu dem Zweck, Gefahrensituationen, die global in der Umwelt bekannt sind, anzuzeigen. Wenn Gefahr besteht, dann nimmt die Variable `is_dangerous` den Wert eins also »wahr« an und die Position der Gefahrenquelle in x,y-Koordinaten wird abgespeichert. Sonst hat `is_dangerous` den Wert null beziehungsweise »falsch«.

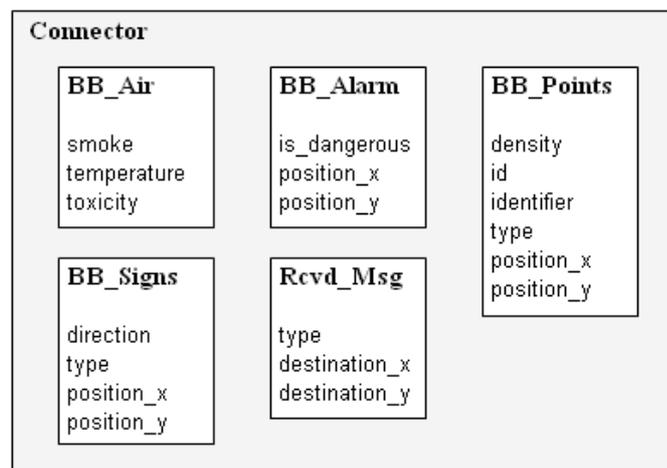


Abbildung 4.1 - Verwaltung der externen Informationen im Connector

Für alle vom User in SimWalk definierten Punkte, die ein Agent ansteuern kann, wurde das Blackboard `BB_Points` eingerichtet, welches diese auf geeignete Weise zur Verfügung stellt. Neben der vom System gesetzten ID, einem Bezeichner »identifier« und der Position in x,y-Koordinaten – alles so bereits im Simulator vorgesehen – werden die Punkte zusätzlich durch einen Dichtewert und Typ charakterisiert. Die Dichte gibt dabei die momentane Anzahl von Agenten am Punkt an. Der Typ dient der Klassifizierung des Punktes zum Beispiel als »Ausgang« oder »Nahrungsquelle«, und sollte stark generalisiert angegeben werden, um große Klassen von Punkten zu bilden und dem Agenten damit bei der späteren Auswahl mehr Möglichkeiten offen zu halten. Ähnlich verhält es sich bei der Angabe der Beschilderung. Sie werden auf `BB_Signs` verwaltet und der dort definierte Typ gibt die Klasse des verwiesenen Zieles an. Weiterhin wird ein Schild durch seine Position und die Richtung charakterisiert, in der sich das Ziel befindet. Und letztlich soll auch in den folgenden Prozessbeispielen die

Annahme weiterverfolgt werden, dass Kommunikationsaustausch in der Personenstromsimulation vornehmlich dem Austausch von Zielkoordinaten dient. Aus diesem Grund beinhaltet das Blackboard Rcvd\_Msg für die zu versendenden Nachrichten über die gleichnamige Schnittstelle lediglich die Angaben zum Typ des Zielpunktes und seine Koordinaten. Sollen zukünftig noch weitere Informationen zur Verfügung stehen und andere Arten von Nachrichten versandt werden, so ist die Komponente Connector beliebig durch die entsprechenden Strukturen erweiterbar. Das von SimWalk zusammengestellte Datenobjekt ext\_data enthält also die Informationen der vier Blackboards sowie eventuell angeforderte Kommunikation zur Orientierung. Zusammen mit diesem Objekt und den im Simulator gespeicherten internen Informationen eines Agenten, kann anschließend für diesen das PECS-Modul instanziiert werden, so dass beide Datensätze in Actor vorliegen und weiterverarbeitet werden können (Abbildung 4.2).

Im implementierten Modul umfassen die Informationen über den Agenten seine eventuell in SimWalk definierte Zugehörigkeit zu einer speziellen Gruppe von Personen sowie die Position im Raum und die ID des Punktes, den er zum aktuellen Zeitpunkt ansteuert. Hinzu kommen die internen Zustände, die auf die PECS-Klassen aufgeteilt werden können.

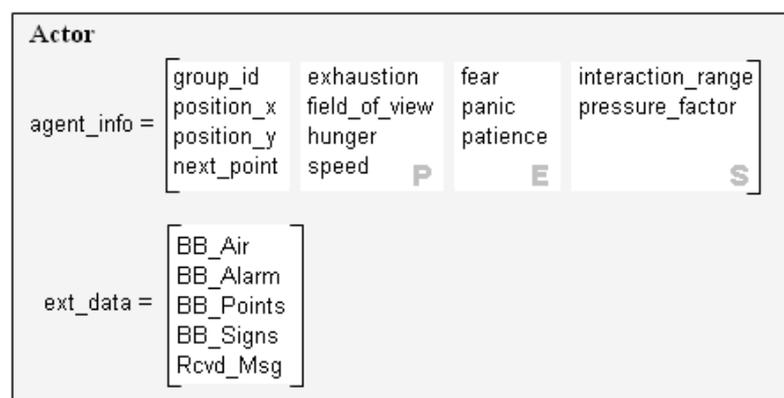


Abbildung 4.2 - an Actor übergebene Daten

Zu beachten ist, dass der momentane Umfang keine kognitiven Parameter zur Übergabe aufweist. Die Komponente Cognition enthält lediglich solche Zustände, die bei jedem Aufruf neu berechnet und aus diesem Grund nicht übergeben werden müssen. Denkbar wäre jedoch die Integration von Informationen, die die kognitiven Fähigkei-

ten des Agenten charakterisieren und nicht von externen Faktoren abhängig sind. Dies könnte zum Beispiel ein auffällig niedriger oder hoher Intelligenzquotient sein, der Entscheidungen des Agenten stärker beeinflusst.

Der Vektor `agent_info` wird unverändert an Perception weitergeleitet, wo die Initialisierung der PECS-Objekte mit den zugehörigen Werten erfolgt. Die allgemeinen Agenteninformationen über Position, anzusteuern Punkt und Gruppenzugehörigkeit werden bei Bedarf an die Komponenten übergeben, wenn sie für eine Berechnung erforderlich sind. Da für die Bestimmung eines Zielpunktes im Planner stets die aktuelle Position des Agenten benötigt wird, wird das Objekt Planner bereits mit `position_x` und `position_y` von Cognition aus initialisiert (Abbildung 4.3).

In Cognition selbst sind die Zustände `safety` und `attention` definiert. So kann berechnet werden, wie sicher sich der Agent in einer Situation fühlt und wie aufmerksam er sich in dieser bewegt. Eine Rückgabe dieser beiden Parameter und Speicherung in SimWalk ist nicht erforderlich, da sie in den gewählten Experimenten als rein situationsabhängig gesehen werden und damit keiner weiteren Dynamik unterliegen.

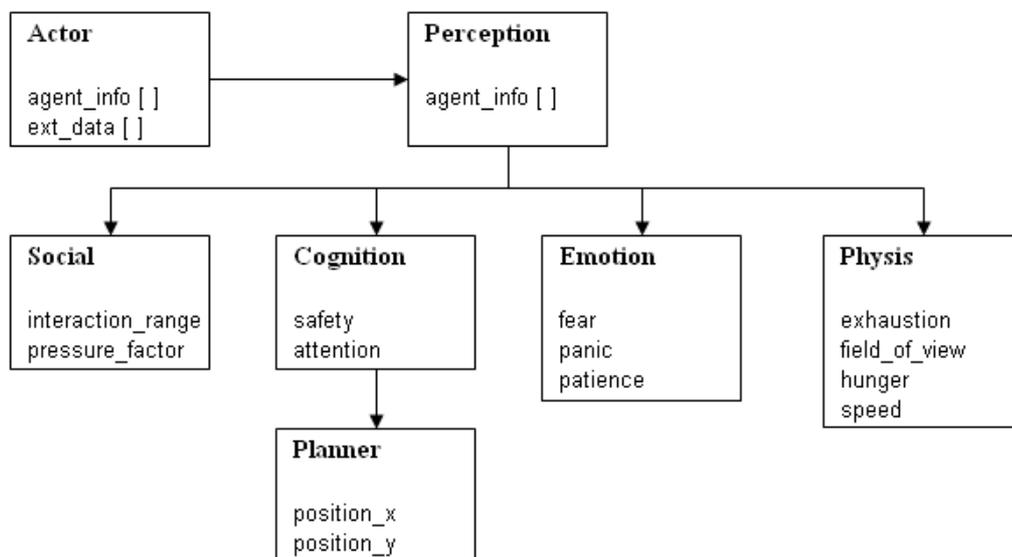


Abbildung 4.3 - interne Parameter des Agenten

Alle Parameter, die in keinem definierten Wertebereich liegen, wie zum Beispiel die emotionalen oder kognitiven Größen, sind normalisiert angegeben. Über eine geeignete Quantifizierung derartiger Faktoren müsste vor der Integration in den Simulator entschieden werden.

## 4.2 Das Zusammenspiel psychosomatischer Parameter und Prozesse

Im Folgenden soll näher auf die in den PECS-Klassen umgesetzten Zustände und ihre Dynamiken eingegangen werden. Die Daten, die in das Modul geladen werden, sind für alle durchgeführten Experimente zufällig innerhalb definierter Wertebereiche erzeugt. Des Weiteren sind alle im Modul umgesetzten proportionalen Zusammenhänge und Quantitäten lediglich als Versuchswerte zu sehen und bedürfen einer separaten, detaillierten Beurteilung, die nicht Bestandteil der vorliegenden Diplomarbeit sein soll.

Beginnend mit der Initialisierung der PECS-Objekte wird im Rahmen der Informationsfilterung in Perception die Ausdehnung des Gesichtsfeldes vom Agenten, charakterisiert durch den physischen Parameter `field_of_view`, dem Umweltfaktor `smoke` angepasst. Dies geschieht in Anlehnung an die bereits beschriebene Verkleinerung des Gesichtsfeldes, die bei eventuellem Rauchaufkommen auftreten kann. Anschließend können, von Perception aus gesteuert, weitere Zustandsdynamiken und Prozesse ausgeführt werden. Aufgrund der Anforderung, dass alle Funktionen mit denselben Daten rechnen, ist die Reihenfolge der Funktionsaufrufe irrelevant, was den Ausbau des Moduls um weitere Prozesse vereinfacht.

In der Komponente Physis wird zunächst `exhaustion` – der Ermüdungszustand des Agenten – neu berechnet. Dieser unterliegt in den erstellten Prozessbeispielen sowohl einer konstanten Eigendynamik als auch einer durch die Umwelteigenschaften `smoke`, `toxicity` und `temperature` beeinflussbaren Dynamik. Sind diese drei externen Faktoren außerhalb der exemplarisch definierten Normgrenzen, ermüdet der Agent schneller. Das bedeutet, dass `exhaustion` um einen größeren Faktor erhöht werden muss als unter Eigendynamik. Auch für das Hungergefühl des Agenten, umgesetzt durch den internen Zustand `hunger`, wird eine konstante, rein von der Zeit abhängige Zunahme des Wertes angenommen. Eine Veränderung durch externe Faktoren soll nur dann erfolgen, wenn der Agent eine Nahrungsquelle aufgesucht hat. Nach Passieren des entsprechenden Punktes in der Umgebung, kann der angenommene Hunger auf null gesetzt werden. Der letzte physische Parameter, der aktualisiert wird, ist die Laufgeschwindigkeit des Agenten. Sie unterliegt keiner Eigendynamik, wird jedoch sowohl von internen als auch von externen Größen beeinflusst. Es ist zunächst umgesetzt, dass der Ermüdungszustand in der Art auf die Geschwindigkeit einwirkt, dass ein

höherer Wert von exhaustion eine Verringerung des Parameters speed zur Folge hat. Eine weitere Modifikation wird als erforderlich gesehen, wenn erhöhte emotionale Zustände vorliegen. Für die noch folgenden Experimente wird angenommen, dass ein verängstigter Mensch in seinem Handeln häufig sehr vorsichtig ist und sich dementsprechend langsam bewegen wird. Ein Mensch in Panik hingegen könnte vor der Ursache, die diese Emotion ausgelöst hat, davonlaufen. Der Wert von speed müsste in diesem Fall deutlich erhöht werden.

In der Komponente Emotion werden die Änderungen der bisher definierten Zustände vornehmlich aufgrund externer Einflussfaktoren erfolgen. Inwiefern sich Angst und Panik, also die Parameter fear und panic, tatsächlich voneinander unterscheiden, bleibt ungeklärt. Zunächst treten sie beide in Abhängigkeit des Vorhandenseins einer bekannten Gefahr auf Grundlage der Informationen im Blackboard BB\_Alarm auf und werden auf eins gesetzt. Denkbar wäre hier, nur einen Parameter umzusetzen oder in Abhängigkeit der gegebenen Situation die Möglichkeit zu geben, dass ein Parameter den anderen hemmen kann. In etwa um abzubilden, dass in einer Paniksituation die Wahrnehmung vermindert ist und damit zum Beispiel auch das Empfinden von starker Angst in gewisser Weise gehemmt werden kann. Des Weiteren ist Angst ein Faktor, der ohne Panik auftreten kann und generell bewirkt, dass sich ein Mensch verändert bewegt. Für beide Parameter wurde ferner eine Eigendynamik definiert, die das Abklingen der Emotionen nach ihrer Aktivierung modellieren soll. Die dritte Größe, die der Komponente zugeordnet ist, ist die Geduld des Agenten. Sie kann in Fortbewegungssituationen insofern eine Rolle spielen, da ein ungeduldiger Mensch häufig die Richtung seiner Bewegung ändert und ein soeben gewähltes Ziel wieder aufgibt, um ein neues zu suchen. Auch wenn Kompromisse bei der Pfadwahl eingegangen werden, kann die Geduld eine wichtige Größe sein.

Eine Änderung des Agentenverhaltens und damit eine auch für den Anwender deutlich sichtbare Veränderung der Bewegung wird durch die Größe interaction\_range der Komponente Social Characteristics bewirkt. Sie ist in dieser Form bereits in SimWalk umgesetzt und die angesprochenen Veränderungen können beobachtet werden, wenn der Parameter vor dem Simulationslauf modifiziert wird. Mithilfe des PECS-Moduls kann interaction\_range nun auch aufgrund externer oder interner Gegebenheiten autonom vom Agenten selbst während der Simulation angepasst werden. Eine hohe Distanz zu anderen Agenten kann bewirken, dass eine signifikante Änderung des gelaufenen Weges eintritt, da das reaktive Ausweichverhalten bei Unterschreiten der Dis-

tanz etwa durch eine Wand oder weitere Agenten behindert wird. In so einem Fall kann es passieren, dass der Agent förmlich vom Strom abgedrängt wird und daraufhin einem anderen Pfad folgt. In Gefahrensituationen wiederum könnte `interaction_range` verringert werden, um eine eintretende Gruppenbildung und die Verringerung der persönlichen Distanz zueinander aufgrund von Angst zu modellieren.

Die Gefahr spielt auch bei dem im Modul umgesetzten kognitiven Parameter `safety` eine Rolle. Wenn der Wert von `is_dangerous` aus `BB_Alarm` wahr ist, dann wird `safety` auf null gesetzt. Eine quantitative Einteilung ist schwierig und wird in den aktuellen Beispielen vermieden. So kann `safety` ebenfalls nur null oder eins sein, das heißt, der Agent fühlt sich in der Umgebung nicht sicher oder die Situation wird als unbedenklich beurteilt. Der zweite Parameter `attention` ist zunächst ausschließlich von internen Zuständen der PECS-Klassen abhängig und ohne Eigendynamik umgesetzt. Er kann aufgrund dessen für jeden Berechnungsdurchlauf des Agenten rein intern neu ermittelt werden. Möglich wäre die Abbildung einer Wechselwirkung mit dem Ermüdungszustand des Agenten. Je höher der Wert von `exhaustion` in `Physis` ist, desto geringer könnte die Aufmerksamkeit sein. Da beide Werte normalisiert zwischen null und eins vorliegen, wird zunächst vereinfacht eine indirekte Proportionalität angenommen. Mit einer verringerten Aufmerksamkeit wäre es zum Beispiel denkbar, dass ein oder mehrere zur Verfügung stehende Zielpunkte übersehen werden. Dies könnte mithilfe einer zufälligen Einflussgröße modelliert werden, die Punkte aus den Blackboards für den Agenten ausblendet.

Nachdem alle internen Zustände in den PECS-Klassen neu berechnet und in `Perception` aktualisiert wurden, können von dort aus nun zwei Dinge veranlasst werden. Erstens könnte der Vektor mit den neuen Daten an `Behaviour` zurückgegeben werden, wo rein aufgrund der internen Zustände über eine Reaktion des Agenten entschieden wird. Jedoch soll im folgenden Verlauf lediglich näher auf die aktive und bewusste Entscheidung für ein Handlungsziel, also ein Bewegungsziel, eingegangen werden.

### 4.3 Autonome Optimierung des Pfades

Der folgende Prozess ist ein elementarer Vorgang bei der Fortbewegung in einem Raum, in dem verschiedene Punkte anzusteuern sind: Während der Bewegung soll der Pfad mithilfe einer kognitiven Leistung optimiert werden. Das heißt, es soll auf-

grund möglicher Gegebenheiten in der Umwelt eine bewusste Entscheidung bezüglich des nachfolgend anzusteuernenden Punktes getroffen werden. Erforderlich kann dies in Situationen sein, in denen der Agent nicht nur einen Punkt, sondern aus einer Menge von Punkten frei wählen kann, die alle gleichwertig sind. Die »Gleichwertigkeit« kann in SimWalk vom Anwender definiert werden, indem mehrere Punkte mit dem gleichen Bezeichner versehen werden. Muss der abzubildende Fußgänger zum Beispiel beim Betreten eines Gebäudes durch eine Schleuse, und im Eingangsbereich des Gebäudes stehen drei Schleusen zur Verfügung, dann kann der Anwender in SimWalk alle drei Schleusen mit dem gleichen Namen versehen und muss dann nur einmal diesen Bezeichner in den Agentenpfad einfügen. Im folgenden Szenario wird zusätzlich für die Schleusen ein gewisser Zeitverbrauch festgelegt, zum Beispiel für eine Sicherheitskontrolle an dieser Stelle. Das dazu aufgebaute Modell in SimWalk wird mit den von Savannah Simulations definierten Standardparametern für die Agenten durchgeführt. Daraufhin ergibt sich folgendes Bild während der Simulation (Abbildung 4.4): Die Agenten – hier blau dargestellt – wählen die erste Schleuse rechts und aufgrund der definierten Wartezeit dort ergibt sich ein Stau. Nur eine sehr geringe Zahl von Agenten wählt die mittlere Schleuse.

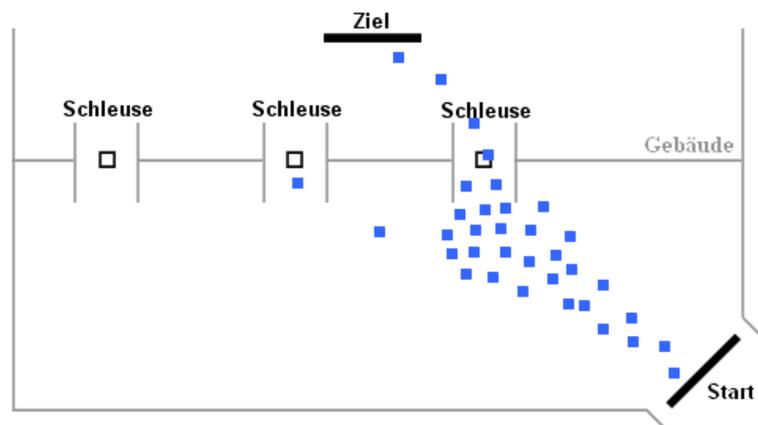


Abbildung 4.4 - Problemstellung bei der Pfadwahl

Um eine andere Schleuse als die rechte zu wählen, müssten sich die Agenten ein Stückweit gegen die Kraft des Potenzialfeldes, von derer sie gezogen werden, bewegen. Dieser Vorgang gleicht dem Bergaufgehen, wenn das Potenzialfeld als Höhenfeld gesehen wird. Das ist jedoch von Seiten des Bewegungsalgorithmus, der lediglich Bewegungen in Richtung eines geringeren Potenzials vorsieht, nicht möglich. Die Agen-

ten, die doch einen anderen Pfad gewählt haben, haben dies nicht aufgrund eines Denkprozesses getan, sondern weil sie vom kürzesten Pfad aufgrund der hohen Agentenzahl an der ersten Schleuse abgedrängt wurden. Sie wurden durch die anderen Agenten, also durch die Kraft Pedestrian Pressure, buchstäblich bergauf geschoben und sind aufgrund dessen in den Wirkungsbereich der mittleren Schleuse gelangt.

Ein Fußgänger würde in dieser Situation bewusst erkennen, dass zwei weitere passierbare Schleusen zur Verfügung stehen und daraufhin Überlegungen anstellen, den nächstgelegenen Durchgang zu wählen, auch wenn dies einen Umweg bedeutet. Es wird abgeschätzt, welche Variante – an der ersten Schleuse warten oder den Umweg über eine andere nehmen – weniger zeitintensiv ist. Hinzukommen Faktoren wie der zu erwartende körperliche Aufwand für den Weg, die Geduld, die ein Mensch in dieser Situation aufbringt, oder wie erschöpft er ist. Ein ungeduldiger Mensch würde eher den Umweg wählen, als jemand der bereits stärker erschöpft ist und sich aus diesem Grund für die Wartezeit entscheiden würde.

Eine Abbildung dieses Verhaltens kann über die kognitive Wahl eines Bewegungsziels erfolgen. Die beschriebenen Einflussfaktoren können zum Beispiel anhand der Werte *exhaustion* in *Physis* und *patience* in *Emotion* mit in die Berechnung eingehen. Die Umsetzung dessen erfolgt in der Klasse *Planner*. Dazu wird zunächst nach jeder Aktualisierung der Parameter in den PECS-Klassen in *Cognition* berechnet, ob ein spezielles Handlungsziel aufgrund interner Zustände verfolgt werden muss. Ist dies nicht der Fall, wird im *Planner* überprüft, ob der Pfad optimiert werden kann. Benötigte Informationen sind die aktuellen internen Zustände, der aktuell angesteuerte Punkt des Agenten, sowie alle zur Verfügung stehenden Punkte, die in *Actor* innerhalb von *BB\_Points* vorliegen.

Die Daten im *Blackboard* werden dabei nach solchen Punkten durchsucht, die den gleichen Bezeichner wie der aktuelle Punkt besitzen. Ist ein solcher gefunden wird die euklidische Distanz zwischen ihm und dem Agenten berechnet. Nur wenn diese innerhalb der Größe des Gesichtsfeldes des Agenten liegt, also vom Agentenstandpunkt aus gesehen werden kann, wird der Punkt weiter betrachtet. Um neuer Zielpunkt zu werden, müssen jedoch zwei weitere Bedingungen erfüllt sein. Erstens muss er der zum Agenten nächstgelegene der bisher betrachteten Punkte sein. Zweitens sollte die aktuelle Agentendichte an diesem Punkt kleiner sein als die am momentan angesteuerten Punkt, da sich die Situation des Agenten andernfalls nicht verbessern würde. Wenn eine hohe Anzahl von Punkten derselben Art vorliegt, wäre es möglich, auch

hier das Kriterium einer minimalen Dichte einzuführen, um den Durchgang mit dem geringsten Zeitverbrauch zu finden. Um den Kompromiss bezüglich der zusätzlich zu laufende Wegstrecke und der eventuellen Wartezeit am Punkt zu modellieren, können Entfernung des Punktes und seine Agentendichte unterschiedlich gewichtet werden. Die Gewichtung könnte dabei in Abhängigkeit von den bereits erwähnten Einfluss nehmenden Parametern aus den PECS-Klassen geschehen, um individuelle Präferenzen des Agenten abzubilden.

Zur Visualisierung der Berechnungsergebnisse richtet sich das PECS-Modul nach dem Standard in SimWalk (Abbildung 4.5): Die einzelnen Zielpunkte sind farblich gekennzeichnet und jeder Agent wird entsprechend seines von ihm gewählten Zieles farblich kodiert. In der Ausgangssituation – links dargestellt – bewegen sich alle 50 für dieses Szenario definierten Agenten auf die rechte Schleuse zu. Ihre internen Zustände und ihre Position wurden mithilfe von Zufallszahlen innerhalb vorgegebener Grenzen erzeugt, so dass sie eine Menge von Personen mit denselben Fähigkeiten aber individuellen Eigenschaften darstellen. Auch die Berechnungsreihenfolge der Agenten ist zufällig.

Nach der Berechnung – rechts im Bild – haben 13 Agenten ihren Pfad verändert und die mittlere Schleuse gewählt, 6 Agenten haben das linke, blau dargestellte Ziel gewählt und 31 Agenten würden sich weiterhin auf den nächstgelegenen Punkt zu bewegen.

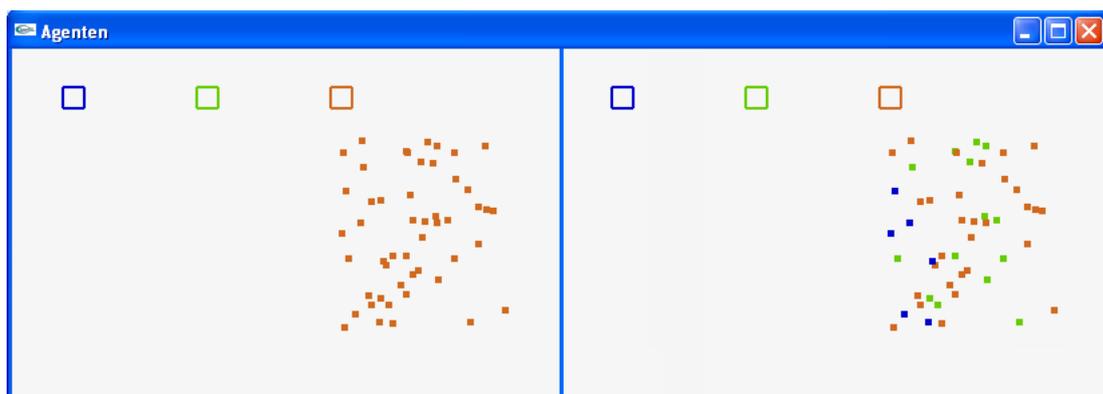


Abbildung 4.5 - im PECS-Modul berechnete Änderung des Pfades

Da das PECS-Modul lediglich der Berechnung der internen Parameter dient und der für Testzwecke entwickelte Rahmen keinerlei tatsächliche Fortbewegung der Agenten berechnet, wird der Dichtewert des entsprechenden Zieles direkt nach der Berechnung

des jeweiligen Agenten angepasst, um eine Veränderung der Situation, sprich das Ankommen des Agenten an diesem Punkt zu simulieren. So kann den nachfolgend zu bearbeitenden Agenten eine neue Datenbasis zur Berechnung geboten werden, auch wenn dies im Simulator mehrere Durchläufe dauern würde. Die Dichtewerte dort stehen aufgrund der voraus gelaufenen Agenten zur Verfügung, was ohne eine echte Bewegungsabbildung auf diese Weise nicht nachzubilden ist.

Das Ergebnis zeigt auf, inwiefern die Entfernung des Zielpunktes eine Rolle bei der Entscheidungsfindung spielt. Lediglich solche Agenten, die sich im Vergleich zu den übrigen auffällig nah an der blauen Schleuse befinden, haben diese als Zielpunkt gewählt. Agenten, die von vornherein weit von der mittleren oder linken entfernt sind, behalten die rechte Schleuse als Ziel bei. Die Größe des Gesichtsfeldes wurde für das Experiment bewusst hoch gewählt, damit dieser Parameter das Ergebnis nicht verfälscht. Aufgrund der zufällig festgelegten Berechnungsreihenfolge der Agenten ist jedoch keine klare Grenze bei der Pfadwahl zu ziehen, da es möglich ist, dass ein Agent, obwohl er der blauen Schleuse deutlich näher ist als andere, diese nicht als Ziel wählt, weil dort bereits eine erhöhte Agentendichte vorherrscht. Aufgrund der Gewichtung von Wegstrecke und Dichte fällt dann die Entscheidung gegen den zusätzlich zu laufenden Weg aus.

Eine bewusste Wahl des zu laufenden Pfades kann nicht nur die beobachtbaren und messbaren Simulationsergebnisse, sondern auch die Akzeptanz des Simulators beim Anwender steigern. Denn die rein reaktive Pfadverfolgung wird von diesem bereits auf den ersten Blick als unrealistisch eingeschätzt und wirft Fragen und Zweifel bezüglich der umgesetzten Agententechnologie auf.

#### 4.4 Berechnete Parameter als Handlungsauslöser: Das Beispiel Nahrungssuche

Zur Erläuterung eines Handlungsbeispiels aufgrund eines internen Zustandes wird das Verhalten zur Nahrungssuche verwendet, auch wenn dies bei den momentanen Einsatzgebieten von SimWalk nicht im Vordergrund steht. Denkbar wäre jedoch, dass ein Agent aufgrund eines aufkommenden Hungergefühls autonom ein Restaurant oder eine ähnliche Lokalität in der Umwelt aufsucht. Auch hier ist es sinnvoll, die definierten Punkte, die der Agent dabei ansteuern kann, zu einer großen Klasse »Nah-

rungsquelle« zusammenzufassen, um eine flexible Auswahl zu ermöglichen. Nur wenn dies nicht erwünscht ist und der Agent auf einen bestimmten Typ beschränkt sein soll, müssen die Nahrungsquellen weiter klassifiziert werden.

Zur Berechnung des Zieles erhält Cognition über die Komponente Perception Informationen über den momentanen Nahrungsbedarf des Agenten. Hier kann direkt der aktualisierte Wert aus dem Vektor übergeben werden. Liegt dieser oberhalb einer im Modul exemplarisch definierten Grenze, wird im Planner die Funktion `satisfyHunger()` aufgerufen, die dem Finden einer geeigneten Nahrungsquelle dient. Dazu werden alle vom Anwender definierten Punkte im Simulationsmodell überprüft, die im Blackboard `BB_Points` Teil der übermittelten externen Informationen sind. Die ID des Punktes, auf den sich der Agent zum Zeitpunkt der Berechnung zubewegt, ist Teil der Datenstruktur `agent_info`. Dementsprechend kann der Typ dieses Punktes in `BB_Points` ermittelt werden. Für jeden anderen zur Verfügung stehenden Punkt, der vom gleichen Typ ist wie der aktuell angesteuerte, kann anschließend die euklidische Distanz zum Agenten berechnet werden. Ist der Punkt für den Agenten sichtbar, ist also die berechnete Distanz kleiner als die Ausdehnung des Gesichtsfeldes, und bisher keine andere Nahrungsquelle gefunden worden, die näher am Agenten liegt, so kann die Distanz zu diesem Punkt zunächst gespeichert und bei den folgend zu überprüfenden Punkten als Maß genommen werden. Nach Abfrage aller in Frage kommenden Elemente von `BB_Points` werden die Koordinaten der nächstgelegenen Nahrungsquelle zum neuen Zielpunkt des Agenten.

Ist keine Nahrungsquelle in Blickweite hat der Agent zwei Möglichkeiten. Zuerst wird überprüft, ob stattdessen ein Schild im Bereich des Gesichtsfeldes liegt, welches auf eine Nahrungsquelle zeigt. Aus diesem Grund ist es erforderlich, Schilder auf dem Blackboard `BB_Signs` mit einem Typ zu versehen. Dieser sollte den klassifizierten Punkten entsprechen. Soll ein Agent eine Nahrungsquelle vom Typ »Restaurant« finden, werden auch nur Schilder desselben Typs den Agenten bei der Suche unterstützen. Ein Mensch, der sich anhand von Beschilderung orientiert und ein Schild ausgewählt hat, welches ihm die gewünschten Informationen anzeigt, wird sich jedoch nicht direkt auf das Schild zu bewegen, um dann – dort angekommen – die Richtung zu wechseln. Stattdessen wird er abschätzen, an welcher Position sich der Punkt befindet, auf den das Schild verweist und dann den direkten Weg eingeschlagen (Abbildung 4.6). Unbekannt ist jedoch dabei die von einem Menschen subjektiv erwartete Entfernung zwischen Schild und Ziel. Wenn als Versuchswert für die Experimente 200 Me-

ter angenommen wird, kann der anzusteuende Punkt über die Translation der Schildposition um den Versuchswert in die angegebene Richtung ermittelt werden.

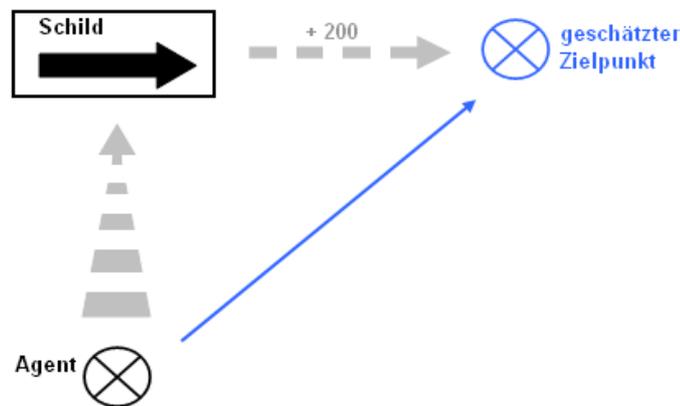


Abbildung 4.6 - Abschätzen eines Zieles anhand von Beschilderung

Ist auch kein Schild für den Agenten sichtbar, bleibt die Möglichkeit, ein Ziel über Kommunikation mit anderen Agenten zu finden. Das heißt die Berechnung wird hier beendet und ein Datenobjekt mit der entsprechenden Information, welche Art von Ziel gesucht wird, wird aus dem PECS-Modul an SimWalk zurückgegeben. Dort kann über die Komponente Connector ein Agent ermittelt werden, der sich innerhalb einer vorgegebenen Kommunikationsdistanz von 50 Metern zum aktuell suchenden Agenten befindet. Wird ein Agent gefunden, der sich auf ein Ziel, welches die erforderlichen Kriterien erfüllt, zu bewegt, dann kann dieses Bewegungsziel vom suchenden Agenten übernommen werden. Werden mehrere Agenten gefunden, wird derjenige ausgewählt, welcher dem suchenden Agenten am nächsten ist. Hat ein Agent auch durch Kommunikation keine passende Nahrungsquelle gefunden, behält er sein aktuelles Ziel bei.

Für das entsprechende mit dem PECS-Modul durchgeführte Experiment wurden die Ausgangsparameter der Agenten in der Art erzeugt, dass allen Agenten, bei denen schon vor der Berechnung der zufällig erzeugte Faktor hunger über dem Grenzwert liegt, bereits eine Nahrungsquelle als aktueller Zielpunkt zugewiesen wurde. Die Zuordnung vom Agenten zum Punkt erfolgte dabei ebenfalls zufällig und unabhängig von anderen Zuständen oder der Entfernung. Es ergibt sich folgendes Bild (Abbildung 4.7):

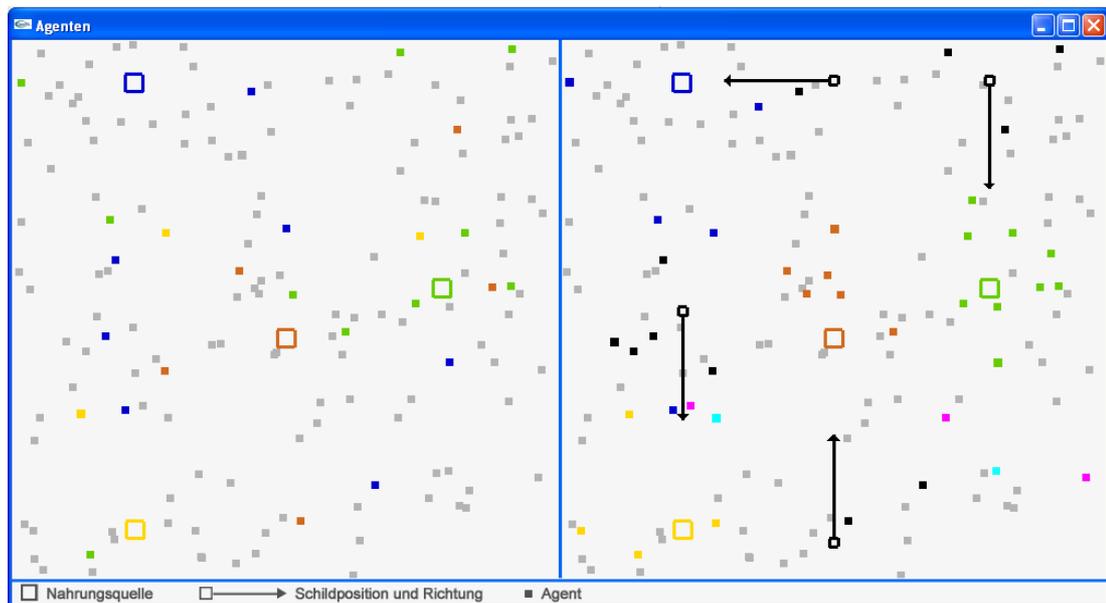


Abbildung 4.7 - Auswahl einer Nahrungsquelle als Zielpunkt

Links in der Abbildung ist die Ausgangssituation vor der Berechnung im Modul dargestellt. Es werden die für das Experiment erzeugten Informationen farbkodiert angezeigt. All jene Agenten, bei denen das Nahrungsbedürfnis unterhalb des Grenzwertes liegt, sind grau dargestellt. Ihre Parameter und Ziele sollen nicht weiter betrachtet werden, um die Übersichtlichkeit zu gewährleisten. Jene Agenten denen eine beliebige der vier Nahrungsquellen zugewiesen wurde, sind in derselben Farbe wie die entsprechende Nahrungsquelle kenntlich gemacht.

Die Rechte Hälfte der Abbildung zeigt die Situation nach der Berechnung. Dort werden auch die in der Umwelt definierten Schilder dargestellt. Der Algorithmus im Planner wird für alle Agenten ausgeführt, für die der Parameter hunger den Grenzwert überschreitet. Das heißt, sowohl die Agenten, die bereits vor der Berechnung einer Nahrungsquelle zugewiesen waren, als auch diejenigen, für die das Handlungsziel erst neu im Modul festgelegt wurde, berechnen ein neues Ziel. Für die erste Gruppe von Agenten wird damit praktisch die erste Zuweisung optimiert, die zweite Gruppe wählt sofort den optimalen anzusteuernenden Punkt. Diejenigen Agenten, deren Suche erfolglos war, bewegen sich weiterhin auf den ihnen zugewiesenen Punkt zu. Sie sind im Bild daran zu erkennen, dass sie in einer Farbe dargestellt werden, die keine der Nahrungsquellen symbolisiert.

Bewegt sich ein Agent nun auf den anhand der Beschilderung geschätzten Punkt zu, wird er in schwarz dargestellt. Die Pfeilspitze des Schildes gibt den Punkt an, an wel-

chem das Ziel vermutet wird. Die Schildpositionen im durchgeführten Experiment sind so gewählt, dass die Nahrungsquelle nach Ankunft des Agenten am geschätzten Punkt in seinem Gesichtsfeld liegt und folglich angesteuert werden kann. Wäre die Distanz größer, entstünde eine Art Informationslücke in der Umwelt. Dies lässt Szenarien sinnvoll erscheinen, in denen mithilfe des umgesetzten Algorithmus zum Beispiel die Beschilderungen in Gebäuden oder öffentlichen Anlagen optimiert werden können.

Die berechneten Zielkoordinaten können anschließend im Simulator in die Berechnung einer Kraft eingehen, die den Agenten gegen die Kraft des Potenzialfeldes in die gewünschte Richtung zur Nahrungsquelle zieht.

## 4.5 Die Bestimmung eines geeigneten Entfluchtungs-punktes

Die Berechnungen der euklidischen Distanzen und das Bestimmen des nächstgelegenen Punktes, Schildes oder Agenten stimmen in allen Fällen von der Vorgehensweise her überein. Wenn diese Methoden stark modularisiert werden – das heißt jede einzelne Teilberechnung durch eine eigene Funktion repräsentiert wird – können neue Handlungsstrategien innerhalb kürzester Zeit erstellt werden, indem man die bereits zur Verfügung stehenden Methoden entsprechend zusammenschaltet. So kann das folgende Handlungsbeispiel, welches der Suche eines Entfluchtungs-punktes in einer Gefahrensituation dient, auf Grundlage des soeben Beschriebenen erstellt werden. Handlungsauslöser ist das Vorhandensein einer Gefahrenquelle, charakterisiert im Blackboard BB\_Alarm. Der kognitive Parameter safety wird auf null gesetzt und aufgrund dessen die Methode findExit( ) in der Klasse Planner aufgerufen. Die zu treffenden Entscheidungen sind ähnlich denen der Nahrungssuche, jedoch ist das Regelsystem komplexer, da auch die Position der Gefahrenquelle für die Entscheidungsfindung eine bedeutende Rolle spielt. In die Berechnung gehen alle Punkte aus BB\_Points ein, die vom Typ »Ausgang« sind. Die Art der Klassifikation der Punkte könnte auch hier wieder das Simulationsergebnis verändern.

Um den Einfluss der Position der Gefahrenquelle abzubilden, wird anhand der Zweipunkteform die Gerade zwischen Agent und momentan betrachtetem Ausgang berechnet. In die so erstellte Geradengleichung können die Koordinaten der Gefahren-

quelle eingesetzt werden. Ergibt die Lösung der Gleichung dann den Wert null, liegt die Gefahrenquelle direkt auf der Verbindung zwischen Agent und Ausgang und ein Agent müsste durch die Gefahr hindurch, um zum Ausgang zu gelangen. Aufgrund dessen ergibt sich für jeden Ausgang eine Art kegelförmiger Einflussbereich der Gefahrenquelle (Abbildung 4.8). Agenten, die sich in diesem befinden, werden sich in Richtung eines anderen Ausganges bewegen.

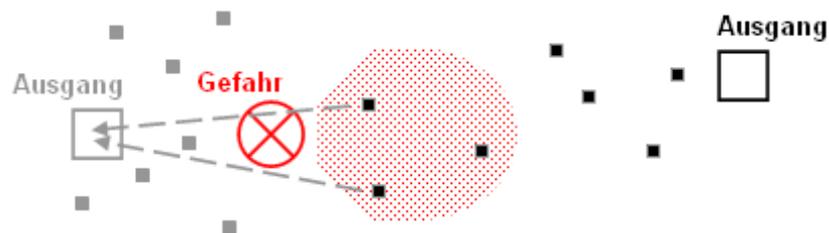


Abbildung 4.8 - Einflussnahme der Position der Gefahrenquelle auf die Zielwahl

Je höher der Wert ist, der nach Einsetzen der Position in die ermittelte Geradengleichung errechnet wird, desto weiter liegt die Gefahrenquelle von der Geraden entfernt. Nur wenn die Gefahrenquelle einen definierten Mindestabstand zur Geraden hat, kann sich der Agent dem Ausgang sicher nähern. In den Experimenten wurde dieser Abstand auf 100 Meter gesetzt.

Doch auch wenn sich die Gefahrenquelle innerhalb dieser Distanz befindet, bedeutet das noch nicht, dass der Ausgang nicht vom Agenten angesteuert werden kann. Es sind die folgenden vier Fälle möglich:

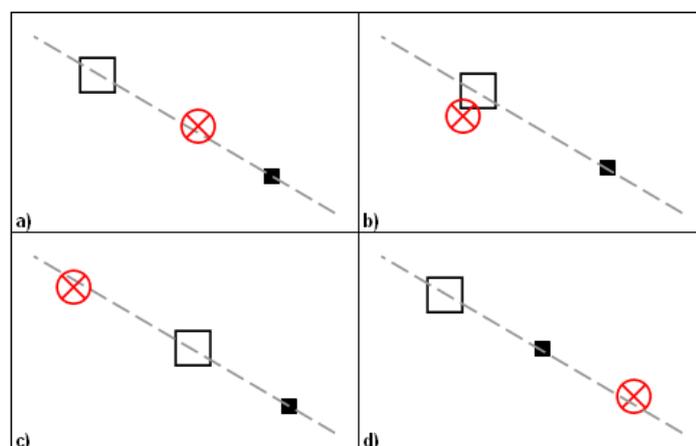


Abbildung 4.9 - Möglichkeiten der Lage von Ausgang, Gefahrenquelle und Agent zueinander

Im Fallbeispiel (a) liegt die Gefahrenquelle auf der direkten Verbindung und innerhalb der beschriebenen Sicherheitsdistanz. In diesem Fall soll der Agent einen anderen Ausgang wählen, auch wenn es möglich wäre, die Gefahrenquelle zu umlaufen. Dazu müssten jedoch die Werte des Potenzialfeldes an den betroffenen Stellen erhöht werden, um den Agenten korrekt zu leiten. Um zusätzlich abzubilden, dass ein Mensch tendenziell dazu neigt, sich von der Gefahr zu entfernen, wenn die Möglichkeit dazu besteht, wird der entsprechende Punkt für die Berechnung im Algorithmus nicht weiter betrachtet und der nächste Ausgang geprüft. Dies gilt auch für eine Situation, wie sie sich in Teil (b) der Abbildung darstellt. Hier liegt die Gefahrenquelle sehr nah am Ausgang. Ein solcher als unsicher einzuschätzender Punkt soll ebenfalls nicht angesteuert werden. Der Abstand, bei dem ein Punkt durch Positionierung der Gefahr als unsicher definiert wird, liegt in den Experimenten bei 50 Metern.

In den Fällen (c) und (d) kann der Ausgang jedoch als Zielpunkt ausgewählt werden, obwohl sich die Gefahrenquelle auf der Verbindung von Agent und Punkt befindet. Denn die Geradengleichung liefert keine Strecke mit einem durch Ausgang und Agent definierten Start und Endpunkt, sondern eine Berechnungsvorschrift für eine Gerade mit unendlicher Länge. Würde die Position der Gefahrenquelle in die Gleichung eingesetzt ohne die Koordinaten aller drei Objekte im Algorithmus zu vergleichen, würde der Ausgang abgelehnt werden. Da die Gefahrenquelle in beiden Fällen jedoch nicht zwischen Agent und Ausgang liegt und die ebenso geforderte Mindestdistanz zum Ausgang nicht unterschritten ist, kann der betrachtete Punkt als Ziel übernommen werden.

Wenn kein Punkt direkt gefunden werden kann, besteht auch bei diesem Handlungsstrang die Möglichkeit, dass die Orientierung anhand der definierten Beschilderung erfolgt. Voraussetzung ist, dass Schilder mit dem Typ »Ausgang« vom Anwender angegeben worden sind. Die Berechnung erfolgt in der auch schon für die Nahrungssuche definierten Methode zur Schildsuche und Abschätzung der Lage des Zielpunktes. Zu beachten ist, dass bei der Suche eines Entfluchtungspunktes anhand von Schildern auch für den geschätzten Zielpunkt überprüft werden muss, ob die Gefahrenquelle zwischen diesem und dem Agenten liegt. Es sind auch hier die beschriebenen vier Fälle möglich, die auf die Annahme des Punktes als Ziel Einfluss haben.

Als dritte Möglichkeit zur Bestimmung eines Ausganges zur Evakuierung bietet sich letztlich die Kommunikation an. Wird vom Connector ein anderer Agent innerhalb der angegebenen Kommunikationsdistanz mit einem passenden Zielpunkt gefunden,

wird sich der suchende Agent zum gleichen Ausgang bewegen. Der Nachrichtenaustausch wird in den beschriebenen Handlungsstrategien stets als letzte Alternative genutzt, da angenommen wird, dass sich ein Mensch eher zu einem selbst gefundenen Ziel als zu einem von einer fremden Person vorgeschlagenem Ziel bewegen wird.

Die internen Zustände der Agenten sind für die durchgeführten Experimente mit Zufallszahlen initialisiert und es sind acht Entfluchtungspunkte definiert sowie vier Schilder, die auf die äußeren Ausgänge der Umwelt verweisen (Abbildung 4.10). Allen 150 Agenten sind zunächst zufällige Zielpunkte zugeordnet und die Zuordnung ist erneut farblich kodiert. Die Gefahrenquelle kann im PECS-Modul für die Experimente über eine Eingabefunktion an einer beliebigen Position platziert werden und wird in der grafischen Ausgabe durch ein rotes Kreuz markiert.

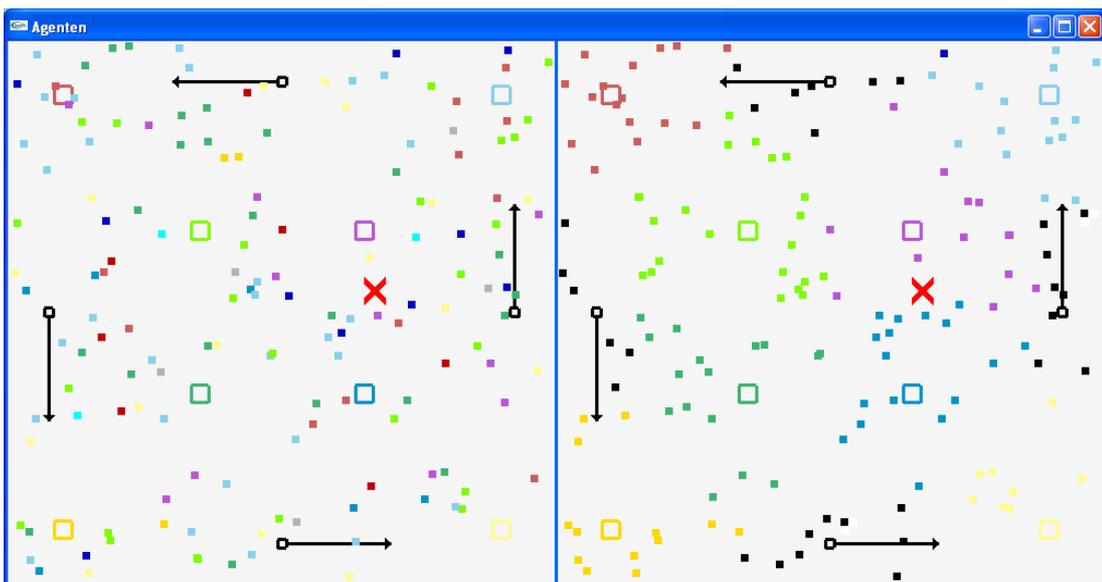


Abbildung 4.10 - im Modul berechnete Entfluchtungspunkte

Die Übersicht zeigt gut, dass die Agenten wenn möglich den jeweils nächstgelegenen Ausgang als anzusteuern den Zielpunkt berechnen. Jeder Punkt hat ein Einzugsgebiet um sich herum, welches in der Ausgabe durch die Farbkodierung der Agenten kenntlich gemacht wird. Wird die Position der Gefahrenquelle verändert, kann das Einfluss auf diese Gebiete haben (Abbildung 4.11). Drei der Agenten im folgend durchgeführten Szenario müssen aufgrund der Verschiebung einen anderen Ausgang wählen.

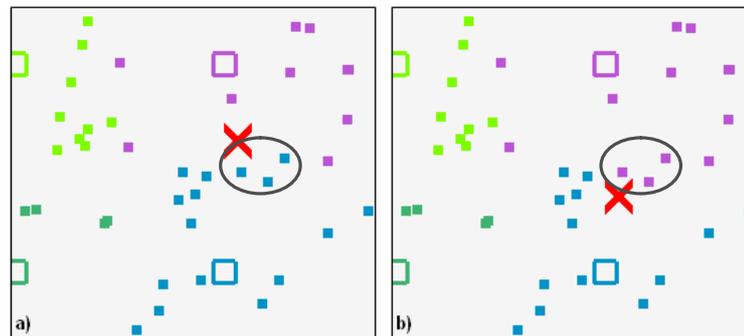


Abbildung 4.11 - Ergebnisse bei verschiedenen Positionen: (a) Originalszenario (b) südlich verschobene Gefahrenquelle

In zwei weiteren Szenarien wird die Entscheidungsfindung der Agenten bei beeinträchtigten Sichtverhältnissen in der Umwelt untersucht. Dazu wird der Faktor smoke aus dem Blackboard BB\_Air auf 0,5 und 0,6 gesetzt und die Ergebnisse mit dem Originalszenario verglichen. Die Rauchdichte wird normalisiert verwendet, wobei 0 unbeeinträchtigte Sicht und 1 dichter Rauch bei unter 5 Metern Sicht bedeutet. Die Größe des Gesichtsfeldes des Agenten wird mit  $(1 - \text{smoke})$  multipliziert um eine indirekte Beziehung zwischen zunehmender Rauchdichte und abnehmender Sichtweite zu modellieren. Dies hat zur Folge, dass in den durchgeführten Experimenten (Abbildung 4.12) weniger Agenten einen Ausgang finden können und daraufhin ihr aktuelles Ziel beibehalten.

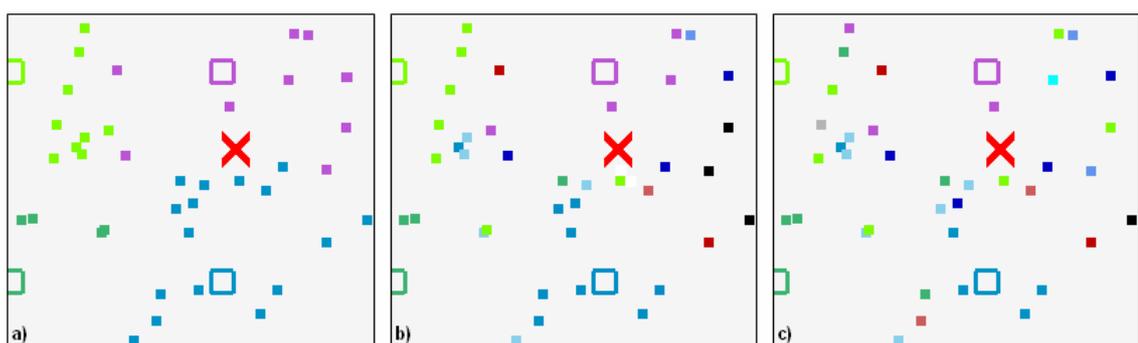


Abbildung 4.12 - Ergebnisse bei verschiedenen Umweltzuständen: (a) unbeeinträchtigte Sicht (b) Rauchdichte = 0,5 (c) Rauchdichte = 0,6

Die Breite der Zone zwischen den definierten Punkten, in denen die Agenten keinen geeigneten Ziel finden können, nimmt mit der Höhe der Rauchdichte zu. Von den in diesem Ausschnitt der Umwelt dargestellten 38 Agenten steuern bei mittlerer Rauch-

dichte nur noch 19 den Punkt an, der unter normalen Sichtbedingungen berechnet wurde. Bei einer weiteren Erhöhung des Wertes von smoke sind es nur noch 9 Agenten. Auch die Sicht auf die Beschilderung rechts in der Umwelt wird in diesem Fall erschwert, so dass für 2 der Agenten am rechten Rand des Abbildungsausschnittes im Experiment (c) das Schild nicht im stark verkleinerten Gesichtsfeld liegt.

Um eine derartige Entfluchtungssituation im momentanen Entwicklungsstand von SimWalk zu modellieren, müssten dort zahlreiche Startpunkte für die Agenten definiert werden, um deren zufällige Verteilung im Raum während einer Bewegungssituation nachzubilden. Für die Simulation der Evakuierung eines Kinos zum Beispiel müsste bei sehr genauer Modellierung beinahe jeder Sitzplatz und Verkaufsschalter als Startpunkt angegeben werden und für jeden der Punkte nur eine sehr geringe Zahl von dort startenden Agenten. Die Aufwand dafür ist enorm. Die Evakuierung würde anschließend direkt beim Simulationsstart beginnen, da es nicht möglich ist, eine Gefahr im Modelllauf zu erzeugen. Panik, Orientierungslosigkeit, Gruppenbildung und weitere wichtige Faktoren finden keine Berücksichtigung.

Mit dem PECS-Modul könnte mit einer normalen Bewegung gestartet werden. Es müssten also lediglich die für die Personenströme relevanten Punkte angegeben werden und die Agenten verteilen sich zunächst normal im Raum. Die Evakuierung beginnt zu einem vom User anzugebenden Zeitpunkt. Dies bedeutet einen geringen Aufwand zur Erstellung des Simulationsmodells. Weiterhin wäre es nicht mehr erforderlich, zwei Modelle zu erstellen – eins für die Analyse der Personenströme in einer normalen Situation und eins für die Entfluchtung, so wie es bisher erforderlich ist.

Der Zeitaufwand für die Berechnung eines derartigen Szenarios für die dargestellten 150 Agenten liegt im Bereich weniger zehntel oder hundertstel Millisekunden, in Abhängigkeit der zur Verfügung stehenden Rechentechnik. Die Berechnungsdauer für die vorangegangenen beschriebenen Handlungsstränge ist ähnlich, da die auszuführende Methodik dieselbe ist. Mit diesem Wert können verschiedene Prognosen angestellt werden, wie hoch der zusätzliche Zeitaufwand für die Berechnungen im PECS-Modul bei einem Simulationslauf ist (Abbildung 4.13). Folgend sollen einige wenige Beispiele berechnet werden, um einen Eindruck zu vermitteln, in welchem zeitlichen Rahmen sich der Aufwand bewegen könnte.

Das Ergebnis wird neben der Rechenleistung bestimmt durch die Anzahl der definierten Agenten, die zu simulierende Zeit im Modell und der Größe des Intervalls in welchem das Modul aufgerufen werden soll.

Zahl Agenten	Berechnungszeit [ s ]	Simulationszeit [ h ]	Aufrufintervall [ s ]	Anzahl Aufrufe	Gesamtzeit zusätzlich		
					[ s ]	[ min ]	[ h ]
150	0,05	5	10	1800,00	90,00	1,50	0,03
5000	1,67	5	10	1800,00	3000,00	50,00	0,83
5000	1,67	5	3	6000,00	10000,00	166,67	2,78
10000	3,33	12	5	8640,00	28800,00	480,00	8,00

Abbildung 4.13 - Prognose über den zusätzlichen Berechnungsaufwand

Als Arbeitsgrundlage diente ein Computer mit Intel Pentium 4 Prozessor mit 3 GHz Takt und einem Arbeitsspeicherumfang von 1 GB. Bei den in den Experimenten definierten 150 Agenten und einer Berechnungszeit von 0,05 Sekunden pro Berechnungsrunde würde sich bei einer Simulationszeit von 5 Stunden, in denen das Modul alle 10 Sekunden aufgerufen werden würde, eine zusätzliche Rechenzeit von 1,5 Minuten für den gesamten Simulationslauf ergeben. Bei denselben Bedingungen und 5.000 Agenten würden sich der Aufwand auf 50 Minuten erhöhen. Eine deutliche Steigerung der Rechenzeit ergibt sich, wenn die Anzahl der Aufrufe pro Simulation erhöht wird, um die Genauigkeit der Ergebnisse zu steigern. Proportional erhöht sich die Gesamtzeit auch bei der Steigerung der zu simulierenden Zeit. Ein Szenario mit 10.000 Agenten simuliert über 12 Stunden ist bei großen Test zur Analyse der Personenströme, die an einem Tag auftreten können, denkbar. Häufig werden jedoch nur kurze Experimente durchgeführt, die das Verhalten in speziellen Situationen testen sollen.

In Anbetracht der aktuell bereits sehr hohen Rechenzeit des Simulators wird der zusätzliche Aufwand als akzeptabel im Vergleich zur erwarteten Verbesserung der Simulationsergebnisse gesehen. Durch eine verteilte Lösung, bei der mehrere Agenten gleichzeitig auf verschiedenen Recheneinheiten bearbeitet werden, könnte der Aufwand noch verringert werden.

## 5 Resümee

### 5.1 Zusammenfassung der Arbeit

Die computergestützte Simulation dient der Abbildung eines Ausschnittes der Realität. Für gewöhnlich handelt es sich dabei um Systeme, in denen zahlreiche Größen und deren komplexe Wechselwirkungen vorhanden sind. Mit einem geeigneten Simulationsmodell ist es möglich, das System vollständig zu analysieren und Komplexitäten aufzuschlüsseln. Dadurch kann für jede beliebige Ausgangssituation prognostiziert werden, wie sich das abgebildete System in der Zukunft verhalten wird. Weiterhin kann, sofern erforderlich, eine Optimierung vorgeschlagen und im Modell getestet werden. Dies alles setzt voraus, dass das Simulationsmodell die Realität hinreichend genau approximiert. Das bedeutet, dass bei möglichst geringem Modellierungsaufwand die Ausgabedaten der Simulation und das abgebildete Verhalten mit der Realität übereinstimmen. Aufgrund dessen gilt es vor der Simulation herauszufinden, welche Parameter tatsächlich das Systemverhalten beeinflussen und welche aufgrund eines geringen Einflusses vernachlässigt werden können. Eine hohe Anzahl umgesetzter Parameter und Zusammenhänge bedeutet dabei stets einen beträchtlichen Entwicklungsaufwand für die Erstellung des Modells und ebenso hohe Anforderungen an Speicherplatz und Rechenzeit bei der Simulation. Eine zu geringe Anzahl kann dazu führen, dass die Ergebnisse nicht die erforderliche Genauigkeit aufweisen.

So verhält es sich auch bei der Simulation von Personenströmen. Der bisherige Umfang der in SimWalk integrierten Größen und Fähigkeiten, die die Personen im Modell charakterisieren, approximiert die Realität nicht hinreichend genau. Dies ist schon während der Simulation für den Anwender ersichtlich, wenn er auf dem Bildschirm verfolgt, wie sich die Agenten in ihrer definierten Umwelt bewegen und welche Pfade sie dabei einschlagen. Auch die gespeicherten Durchsatzzahlen und Dichtewerte für die verschiedenen im Modell definierten Punkte können einem Vergleich mit der Realität nicht Stand halten, da einige von ihnen auffällig hoch oder gering frequentiert sind. Die Ursache dafür liegt im definierten Agentenmodell. Dieses beschreibt ledig-

lich das passive Reagieren der Agenten auf veränderte Umweltbedingungen in Form von Hindernissen und ermöglicht dabei keinerlei Abbildung aktiver Entscheidungen. Menschen treffen jedoch täglich zahlreiche Entscheidungen, selbst in einfachsten Bewegungssituationen – wenn auch dabei häufig unbewusst. Derartiges aktives Handeln und die damit in Zusammenhang stehenden kognitiven Fähigkeiten ermöglichen es einem Menschen, intelligentes Verhalten zu zeigen. Intelligenz kann jedoch in keinem Fall mit reiner Reaktivität nachgebildet werden. Ein Agentenmodell muss Aspekte kognitiven Handelns berücksichtigen. Dabei kann und sollte der Umfang im Hinblick auf die zusätzliche Rechenzeit und den erforderlichen Entwicklungsaufwand so gering wie möglich sein. Denn es steht nicht die Intelligenzleistung eines Menschen bei der Personenstromsimulation im Vordergrund, sondern stets die Bewegung in der Umgebung. Ausgewählte kognitive Fähigkeiten sollen den Agenten lediglich bei der Pfadwahl unterstützen. Neben den Fertigkeiten nehmen verschiedene psychosomatische Zustände eines Menschen Einfluss auf seine Bewegungen. Aus diesem Grund muss auch hier eine geeignete und für den Nutzen angemessene Auswahl getroffen werden.

Die vorliegende Diplomarbeit beschäftigt sich jedoch nicht mit der Bestimmung der relevanten Größen und Funktionen, da dies nur in enger Zusammenarbeit mit Experten aus den Bereichen der Psychologie und Soziologie erfolgen kann. Vielmehr sollte zunächst aufgezeigt werden, auf welche Art und Weise derartige Parameter und Prozesse in der Simulation verwendet werden können und dass eine Integration die Ergebnisse der Simulation deutlich beeinflussen kann. Die Herausforderung bestand also zum einen in der Argumentation anhand geschätzter Größen und zum anderen in der Tatsache, dass nicht ein neues Simulationstool entwickelt, sondern das bestehende Tool SimWalk um die entsprechenden Aspekte erweitert werden sollte. Ungeachtet dessen wurde bei der Auswahl der Zustände von Agenten und Umwelt auf Plausibilität und Relevanz für die Personenstromsimulation geachtet. Die Integration erfolgte im Rahmen des PECS-Referenzmodells, da dieses auch die Modellierung komplexer Zusammenhänge ermöglicht.

Um entscheiden zu können, in welchem Ausmaß menschliche Verhaltensweisen approximiert werden sollen, wurden Anforderungen definiert, die ein Agent in SimWalk erfüllen muss. In den Vordergrund wurden dabei solche Fähigkeiten gestellt, die die Orientierung und Optimierung des Weges in der Umwelt unterstützen. Das bedeutet,

dass zunächst alle aktiv ausführbaren Handlungen ausschließlich der Pfadwahl dienen sollen, um den Schwerpunkt der Funktionalität des Simulators weiterhin auf Personenströmen zu belassen. Als psychosomatische Parameter müssen diejenigen selektiert werden, die einen Einfluss auf das Simulationsergebnis haben, indem sie Weglängen, Durchsätze an den Punkten oder benötigte Wegzeiten der Agenten beeinflussen. Unter der Annahme, dass es in jedem der Bereiche Physis, Emotion, Cognition und Social Characteristics derartige Faktoren gibt, wurde der Aufbau der Komponente Agent des Referenzmodells nur geringfügig verändert. Jedoch konnten sowohl der Umfang der Aufgaben einzelner Bestandteile sowie die Vernetzung untereinander reduziert werden.

Die Komponente Connector, die im Referenzmodell dem Nachrichtenaustausch der Agenten untereinander dienen soll, wurde ähnlich vereinfacht. Denn Kommunikation soll auch hier nur noch zum Zwecke der Orientierung in der Umwelt dienen, wenn dies erforderlich ist. Der Austausch öffentlicher Nachrichten, das heißt solcher, die für alle Agenten bestimmt sind, fällt damit aus dem Entwurf heraus. Doch der Connector bekommt eine zusätzliche Aufgabe, die auf der Idee der Blackboards aufbaut. Da die Umgebung nicht modifizierbar ist und ihre Zustände dementsprechend nur definiert sind, um vom Agenten mental verarbeitet zu werden, würde die Komponente Environment lediglich eine Verwaltungsaufgabe wahrnehmen. Diese wird stattdessen an Connector übertragen, indem dort Blackboards für die Umweltinformationen definiert wurden. So können alle externen Daten, die dem Agenten zur Berechnung übermittelt werden sollen, in dieser einen Komponente zusammengestellt werden.

Die vorgestellten Parameter- und Prozessbeispiele gehen auf Situationen ein, in denen die in SimWalk definierten Agenten kein angepasstes Verhalten zeigen können, da menschliche Entscheidungen bisher nicht abgebildet werden. Die Fähigkeit dazu ist jedoch Voraussetzung für eine optimale Lösung in den beschriebenen Situationen. Die durchgeführten Experimente zeigen auf, welches Potenzial in der Entwicklung steckt und wie sie genutzt werden kann.

## 5.2 Bewertung der Entwicklung

Ziel der vorliegenden Arbeit war es, zu zeigen, wie das in SimWalk umgesetzte Agentenmodell mithilfe des PECS-Referenzmodells weiterentwickelt werden kann. Um

dieses Ziel zu erreichen, wurden verschiedene Anforderungen an das Gesamtkonzept und die Fähigkeiten des Agenten gestellt. Neben der Funktionalität stand dabei die Modularität des Konzeptes im Vordergrund. Es sollte der Aufwand für Anpassungen und Weiterentwicklungen am Simulator so gering wie möglich gehalten werden. Aufgrund der zu verarbeitenden Datenmengen wurde jedoch entschieden, dass eine vollständige Trennung nicht sinnvoll ist. Daher ist der erstellte Entwurf für das Gesamtmodell bewusst so gestaltet, dass dennoch eine zentrierte und größtenteils von SimWalk losgelöste Weiterentwicklung erfolgen kann. Dies wird gerade im Hinblick auf die Lösung zur Integration der Komponenten Environment und Connector deutlich. Die Umsetzung kann innerhalb einer Klasse erfolgen und erfordert wenige Eingriffe in die bisherige Funktionalität des Simulators. Des Weiteren nehmen künftige Veränderungen im Konzept dort wenig Einfluss auf das zur Agentenberechnung aufzurufende PECS-Modul. Die Schnittstelle ist so angelegt, dass sie flexibel nutzbar ist, selbst bei einer Erweiterung der zu übergebenden externen oder internen Daten.

Bei der Entwicklung wurde zusätzlich bewusst auf die Ähnlichkeit zum PECS-Referenzmodell geachtet, obwohl nicht für alle Komponenten sicher ist, ob sie in einer endgültigen Umsetzung Verwendung finden werden. Die Parallelen zwischen Referenzmodell und Entwurf ermöglichen die Orientierung an der Originaldokumentation zum Modell. So bleibt nachvollziehbar, welche Ideen hinter den verschiedenen Bestandteilen des Entwurfes stehen. Mit dieser Vorgehensweise sollen mögliche Fehler bei der Weiterentwicklung vermieden werden, da das Hinzufügen einer neuen Komponente mehr Risiken für die Details des Gesamtentwurfes birgt, als das Entfernen einer bestehenden. Wenn Funktionalitäten hinzugefügt werden sollen, kann sich in der Originaldokumentation informiert werden, in welchem Rahmen dies erfolgen könnte. Die beschriebenen Parameter aus den verschiedenen psychosomatischen Bereichen und ihre Dynamiken vermitteln einen ersten Eindruck, auf welche Art und Weise das implementierte PECS-Modul und die direkt in SimWalk integrierten Komponenten zusammenarbeiten können, obwohl sie lediglich aufgrund von psychologischem und biologischem Basiswissen definiert wurden und die Korrektheit der umgesetzten Größen und Abläufe nicht beweisbar ist. Alle Zusammenhänge zwischen den Parametern sind aus diesem Grund einfach gehalten. Und doch lassen sie erkennen, inwiefern sich die Fähigkeiten eines Agenten verbessern lassen.

Für einen Agenten in SimWalk ist das Vorhandensein mehrerer gleichwertiger Punkte, die angesteuert werden können, ein unlösbares Problem. Genauer gesagt erfolgt tatsächlich keine bewusste Wahrnehmung des Problems, was zur Folge hat, dass keine Lösung entwickelt werden kann. Die Bewegung wird aufgrund dessen vom Anwender in der Beobachterposition als unrealistisch, weil nicht menschlich genug, eingeschätzt. Denn die aktive Optimierung des Weges, der zurückgelegt werden muss, ist ständiger Bestandteil von Bewegung – auch wenn die dazu erforderlichen Prozesse im menschlichen Gehirn häufig unbewusst ablaufen. Innerhalb eines Bruchteils einer Sekunde hat ein Mensch eine Entscheidung getroffen. Wenige Schritte weiter müssen neue Alternativen abgewogen werden. Die im PECS-Modul implementierte Methodik zur autonomen Anpassung des Pfades durch den Agenten bildet genau dieses Verhalten nach. In die Entscheidungsfindung gehen wie in der Realität verschiedenste Faktoren ein, die Einfluss auf die getroffene Wahl haben könnten. Das durchgeführte Experiment zeigt eine deutliche Veränderung der Pfade, die die Agenten laufen würden. Und auch die Szenarien zur Suche einer Nahrungsquelle oder eines Entfluchtungspunktes sind mit aktiv auszuführenden Handlungen verbunden. Sie ermöglichen es, Situationen zu simulieren, die mit dem bisherigen Entwicklungsstand des Simulators nicht möglich sind. Dabei spielen die allgemeinen Parameter eine weniger entscheidende Rolle als die kognitiven Größen und Funktionen. Mit diesen jedoch kann menschliches Verhalten tatsächlich genauer abgebildet werden als bisher.

Auch wenn die Fähigkeiten sicherlich stark beschränkt sind, so kann doch festgestellt werden, dass sich die Eigenschaften des zur Verfügung stehenden Agentenmodells verändert haben. Aufgrund der internen und auch externen Zustände ist der Agent nun in der Lage eine Vorstellung von seiner Situation entwickeln. Auf Basis dieser kann durch Eigeninitiative – also proaktiv – eine Konsequenz etwa in Form eines Handlungszieles entwickelt werden. Der Agent arbeitet dadurch in einem höheren Maße autonom. In Anlehnung an die erwähnten Eigenschaften, die einen Agenten zu einem intelligenten Software-Programm machen, sind auch die Fähigkeiten zur Kommunikation mit anderen Agenten mithilfe des PECS-Moduls geschaffen worden. In Anbetracht der angestrebten und zu erwartenden Einsatzmöglichkeiten des Simulators können sie jedoch auf ein sinnvolles Minimum beschränkt werden. Auch die sozialen Fähigkeiten sind nun, wenn auch nicht vordergründig, genauer als bisher abgebildet und könnten die Simulationsergebnisse in verschiedenen Situationen beeinflussen. Als letztes Kriterium für einen intelligenten Agenten wurde die Lernfähigkeit ge-

nannt. Doch eine Umsetzung dieser Eigenschaft wird in der reinen Simulation von Personenströmen als nicht gerechtfertigt gesehen und findet aus diesem Grund keine Anwendung im Entwurf.

Bei der Weiterentwicklung ist stets genau abzuwägen, welche Prozesse nicht nur etwas zur Intelligenz des Agenten beitragen, sondern tatsächlich für die Simulationsergebnisse von Bedeutung sind. Und auch nur solche sollten letztendlich umgesetzt werden, da das Ziel des zusätzlichen PECS-Moduls die hinreichende Annäherung der Ergebnisse an die Realität ist. Es ist für eine Simulation in SimWalk nicht erforderlich, alle Facetten menschlichen Handelns und Entscheidens zu berücksichtigen.

### 5.3 Ausblick

Die vorliegende Diplomarbeit beschreibt die Motivation, die Hintergründe und letztlich auch das Potenzial des vorgestellten Konzeptes. Wie die Integration von Seiten der Entwickler tatsächlich ausfallen wird, bleibt dabei offen. Jedoch ist eine Verbesserung des Agentenmodells in jedem Fall erforderlich, um auf dem Markt Akzeptanz zu finden. Mit den entwickelten Vorschlägen zur Umsetzung des Referenzmodells wird es möglich sein, auch solche Szenarien durchzuführen, die bisher nicht realisierbar sind. Dazu gehört zum Beispiel das Testen der Positionierung von Beschilderung in der Umgebung oder die Durchführung von Evakuierungsszenarien. Letztere sind zwar zurzeit bereits denkbar, jedoch nur unter einem enormen Aufwand für die Entwicklung des Simulationsmodell und wenig Autonomie bei der Pfadwahl der Agenten. Auch die gerade in solchen Extremsituationen relevanten psychosomatischen Parameter können zukünftig in die Berechnungen eingehen. Bis es soweit ist, sind jedoch weitere Entwicklungen und Untersuchungen, sowie Anpassungen am Simulator erforderlich.

Die bisher rein reaktiv funktionierenden Agenten nehmen ihre Umwelt nur stark eingeschränkt wahr und sind in ihrem Handlungsvermögen beschränkt. Um ihnen deliberatives Verhalten zu ermöglichen, muss nicht nur die Architektur des Agenten, sondern auch die der Umwelt erweitert werden. Dabei ist letzteres in vielen Fällen nur eine Frage der Strukturierung und Weiterleitung der Information. Die möglichen anzusteuern Punkte, die vom Anwender definiert werden können, existieren bereits in SimWalk, auch wenn ihnen zusätzliche Eigenschaften zugeordnet werden müssten.

Auch die Definition von Schildern ist von Seiten des Simulators ohne größeren Entwicklungsaufwand zu realisieren, da sie genau wie Punkte angegeben werden können. Sie werden sich lediglich in der Zahl und Art der charakterisierenden Parameter unterscheiden. Der eigentliche Unterschied ihrer Bedeutung wird ausschließlich im PECS-Modul verarbeitet. Entsprechend muss auch die grafische Oberfläche des Simulators um passende Eingabemöglichkeiten erweitert werden.

Die größte Herausforderung ist und bleibt jedoch die Bestimmung, welche Parameter für die Simulation von Bedeutung sind und wie sie am Rechner abgebildet werden können. Das umfasst sowohl eine geeignete Quantifizierung als auch die Modellierung von eigenen und von extern induzierten Dynamiken. Doch die Frage, wie zum Beispiel Emotionen in einen Wertebereich eingeordnet werden können, beschäftigt Forscher seit Jahren und es bleibt zu überlegen, ob eine stark vereinfachte Abbildung für die Simulation ausreichend wäre. Derartige Fragen können nur in Zusammenarbeit mit Experten aus den entsprechenden Fachgebieten und in zahlreichen Studien und Experimenten geklärt werden. Auch die Komplexität menschlicher Entscheidungsprozesse wird dazu beitragen, dass umgesetzte Funktionen entweder stark vereinfacht dargestellt werden müssen oder provisorisch ohne wissenschaftliche Grundlage, bis die offenen Fragen geklärt werden konnten.

Unabhängig davon, welche Zustände und Dynamiken letztlich umgesetzt werden, können die vorliegende Arbeit und die dazu entwickelten Programmcodes als Grundlage für die vollständige Integration in den Simulator SimWalk dienen. Das entwickelte PECS-Modul kann die Entwicklung vereinfachen und das Grundgerüst für die spätere Implementierung weiterer Funktionen bilden, ohne dass die Struktur des Moduls aufwendig angepasst werden muss. Mit der Integration psychosomatischer Parameter und Prozesse werden zukünftige Simulationen menschliches Verhalten genauer abbilden können als bisher. Die Entwicklungen werden damit zu einer höheren Akzeptanz des Tools beitragen und zahlreiche neue Einsatzmöglichkeiten eröffnen.

## Referenzen

- [Fer01] FERBER, Jacques: Multiagentensysteme – Eine Einführung in die Verteilte Künstliche Intelligenz. Addison-Wesley Verlag, 2001, ISBN 3-8273-1697-0.
- [Gam04] GAMMA, Erich / HELM Richard / JOHNSON, Ralph / VLISSIDES, John: Entwurfsmuster – Elemente wiederverwendbarer objektorientierter Software. Addison-Wesley Verlag, 2004, ISBN 3-8273-2199-9
- [ISE] KRUSE, Rudolf: Intelligente Systeme: Einführung. Vorlesungsmaterialien. <http://fuzzy.cs.uni-magdeburg.de/studium/ise/>
- [Mey06] MEYERS, Scott: Effektiv C++ programmieren. Addison-Wesley Verlag, 3. Auflage, 2006, ISBN 3-8273-2297-9.
- [PSY] LINNEWEBER, Volker: Differentielle und Persönlichkeitspsychologie. Vorlesungsmaterialien. <http://www-e.uni-magdeburg.de/linneweb/>
- [Sch01] SCHMIDT, Bernd: The Modelling Of Human Behaviour. SCS Publication, 2003.
- [Stu03] STUCKI, Pascal: Obstacles in Pedestrian Simulations. ETH Zürich, Department of Computer Science, 2003.
- [Urb04] URBAN, Christian: Das Referenzmodell PECS – Agentenbasierte Modellierung menschlichen Handelns, Entscheidens und Verhaltens. Universität Passau, Fakultät für Mathematik und Informatik, Dissertation, 2004.



# Selbstständigkeitserklärung

Hiermit erkläre ich, dass ich die vorliegende Arbeit ohne unzulässige Hilfe Dritter und ohne Benutzung anderer als der angegebenen Hilfsmittel angefertigt habe. Die aus fremden Quellen direkt oder indirekt übernommenen Gedanken sind als solche kenntlich gemacht.

München, den 13. April 2006

---