

# Ansätze für selbstlernendes Systemverhalten in der Heimautomation

DIPLOMARBEIT

zur Erlangung des akademischen Grades

**Diplom-Ingenieur**

im Rahmen des Studiums

**Medizinische Informatik**

eingereicht von

**Thomas Abinger**

Matrikelnummer 0325143

an der  
Fakultät für Informatik der Technischen Universität Wien

Betreuung  
Betreuer/in: Ao. Univ. Prof. Dr. Wolfgang Kastner  
Mitwirkung: Dipl.-Ing. Georg Neugschwandtner

Wien, 19. Jänner 2011

\_\_\_\_\_

(Unterschrift Verfasser/in)

\_\_\_\_\_

(Unterschrift Betreuer/in)



Thomas Abinger

Markomannenstraße 15/9/24

1220 Wien

Hiermit erkläre ich, dass ich diese Arbeit selbständig verfasst habe, dass ich die verwendeten Quellen und Hilfsmittel vollständig angegeben habe und dass ich die Stellen der Arbeit – einschließlich Tabellen, Karten und Abbildungen –, die anderen Werken oder dem Internet im Wortlaut oder dem Sinn nach entnommen sind, auf jeden Fall unter Angabe der Quelle als Entlehnung kenntlich gemacht habe.

Wien, am 19. Jänner 2011

## Kurzfassung

---

Diese Arbeit verfolgt die Idee, ein Haus oder Gebäude zu schaffen, das sich automatisch an die Bedürfnisse und Wünsche der Bewohner anpasst. Damit sich ein Bewohner bei einer bestimmten Tätigkeit wohlfühlt, sind oft mehrere Betriebsparameter aus technischen Systemen unterschiedlicher Gewerke aufeinander abzustimmen. Weiters sollen individuelle Eigenschaften des Gebäudes in die Steuerung und Regelung mit einbezogen werden. Stehen unterschiedliche Alternativen für eine Steuerung/Regelung zur Verfügung, soll die kostengünstigste Option gewählt werden, ohne dabei den Komfort der Bewohner zu schmälern. In dieser Arbeit werden in zwei Schwerpunkten Mittel untersucht, selbstlernendes Systemverhalten in der Heimautomation umzusetzen. Der erste Schwerpunkt liegt auf der Frage, wie mit Hilfe künstlich neuronaler Netze die komplexen Einflussfaktoren Mensch, Gebäude, Gebäudetechnik und Umgebung in einem intelligenten Heizungsregelkreis gefasst werden können. Die für diesen Zweck erstellten künstlich neuronalen Netze sind vom Typ der Feed Forward-Netze, Jordan-Netze und Simple Recurrent Networks und verändern nach einem zeitlichen Raumbelungsplan, den Raum- und Außentemperaturen sowie der aktuellen Uhrzeit den Sollwert eines herkömmlichen Heizungsreglers. Die so gegenüber einer herkömmlichen, rein uhrzeitabhängigen Strategie erzielbare Effizienzsteigerung wird in einer physikbasierten Simulation bestimmt. Der zweite Schwerpunkt liegt auf der Strukturierung der komplexen Steuerungsaufgaben im Rahmen eines selbstlernenden Softwaresystems. Die Lern- sowie Steuerungs- und Regelungsaufgaben werden dazu als Multiagentensystem abgebildet. Verschiedene, an diesen Aufgaben orientierte Agententypen werden modelliert und deren dynamisches Zusammenspiel betrachtet.

## Abstract

---

This thesis pursues the idea of creating a home or building that automatically adapts to the requirements and desires of its occupants. To ensure that occupants are comfortable during various activities, it is often necessary to coordinate multiple operational parameters from systems belonging to different building disciplines. Furthermore, individual structural properties or the building are to be included into the control strategy. If there are multiple options available to realise the desired conditions, the cheapest method shall be chosen, but occupant comfort shall not be compromised. This work focuses on two points of research regarding the implementation of self-learning system behaviour for home automation systems. The first focus is on the question how artificial neural networks can be used to create an intelligent heating control loop that better considers the complex parameters and inputs coming from humans, buildings, building services, and the natural environment. The artificial neural networks created for this purpose are of the Feed Forward Network, Jordan Network, and Simple Recurrent Network type. According to a presence schedule, outdoor and indoor temperatures and the current time, they modify the set value of a conventional heating controller. The resulting increase in efficiency over a conventional, purely time based strategy is evaluated using physics-based simulation. The second focus is on the structuring of complex control tasks and self-learning capabilities within the framework of a software application. To this end, a multi-agent system is implemented. In it, various types of agents reflect the various learning and control tasks. Besides the agent types themselves, their dynamic interplay is also modelled.

## Danksagung

---

Ich möchte meiner Frau Bettina für die benötigte Zeit und ihre Geduld danken, die ich mit dem Erstellen dieser Diplomarbeit beschäftigt war. Ihre Motivation hat mich dazu gebracht, dieses Thema nach einer langen Pause, bedingt durch meine Arbeit, wieder aufzunehmen. Danke für die tatkräftige Unterstützung, die es mir ermöglicht hat, mich auf das Wesentliche zu konzentrieren und diese Diplomarbeit zu einem finalen Ende zu bringen.

Meinen Eltern danke ich dafür, dass sie mir eine technische Ausbildung ermöglicht haben und mich von Beginn an - in der Schule und nun auch bei meiner universitären Ausbildung - unterstützt haben. Danke für die Zeit und die Mittel, die ihr mir gegeben habt.

Meinen Freunden, besonders Gernot Kurz und Franz Hartl, möchte ich für die guten und schlechten Stunden während unseres Studiums danken.

Ich möchte meinem Diplomarbeitsbetreuer Wolfgang Kastner und meinem Kollegen Georg Neugschwandtner für die Hilfe und Unterstützung danken, die ich bei der Erstellung dieser Arbeit erfahren habe - auch für die Erfahrung auf einer wissenschaftlichen Konferenz selbst mit einem Thema antreten zu dürfen.

Letztlich will ich all jenen hier ungenannten Freunden und Gönnern danken, die mich in irgendeiner Weise in meinem Studium unterstützt und weitergebracht haben.

## Inhalt

<b>1</b>	<b>Einführung</b>	<b>3</b>
1.1	Motivation .....	3
1.2	Absteckung der Fragestellung .....	5
1.3	Aufbau der Arbeit .....	5
<b>2</b>	<b>Stand der Wissenschaft und Technik</b>	<b>6</b>
2.1	Einleitung .....	6
2.2	Wissenschaftliche Forschungen .....	7
2.2.1	Optimum Start-Stop .....	7
2.2.2	Szenarienerkennung .....	10
2.2.3	Softwarearchitekturmodelle .....	16
2.3	Technische Implementierungen .....	24
2.3.1	Kommerziell verfügbare Produkte .....	24
2.3.2	Feldstudien .....	30
<b>3</b>	<b>Anwendungsszenarien</b>	<b>37</b>
3.1	Automatische energiesparende Raumtemperaturregelung für eine Person nach einem gelernten Belegungsplan .....	37
3.1.1	Sprachliche Beschreibung .....	38
3.1.2	Modellanalyse mit UML .....	38
3.2	Automatische Raumtemperaturregelung für eine Mehrpersonenbelegung mit Konfliktlösungseinheit .....	45
3.2.1	Sprachliche Beschreibung .....	45
3.2.2	Modellanalyse mit UML .....	46
3.3	Singlehaushalt: Aufstehen am Morgen während der Arbeitswoche .....	49
3.3.1	Sprachliche Beschreibung .....	49
3.3.2	Modellanalyse mit UML .....	51
<b>4</b>	<b>Optimierte Kontrollstrategien zur Heizungsansteuerung</b>	<b>57</b>
4.1	Multiple Regression als Vorheizzeitschätzer .....	59
4.1.1	Einführung .....	59
4.1.2	Mathematik der Stichprobenregressionsanalyse .....	59
4.1.3	Fazit für die Schätzung der Vorheizzeit mit Hilfe der multiplen Regression .....	68
4.2	Künstliche neuronale Netze .....	68
4.2.1	Einführung .....	68
4.2.2	Trainings- und Testphase von künstlich neuronalen Netzen .....	73

4.2.3	Auswahl des geeigneten Netztyps für die Vorhersage der Vorheizzeit	74
4.2.4	Lernregeln	79
4.2.5	Neuronale Netze Simulator - MemBrain	81
<b>5</b>	<b>Einsparungspotential von - Ein Vergleich zwischen künstlich neuronalen Netzen zu konventioneller Zeitsteuerung</b>	<b>89</b>
5.1	Gebäudemodell	89
5.2	Belegungsplan	91
5.3	Verwendete Neuronale Netze	94
5.3.1	Feed Forward Network	95
5.3.2	Simple Recurrent Network	95
5.3.3	Jordan Network	96
5.4	Simulation	97
5.4.1	Der Simulationsablauf	97
5.4.2	Auswertung der Simulationsergebnisse	103
5.5	Schlussfolgerungen über die Simulationsergebnisse	109
<b>6</b>	<b>Multiagentensystem für die Heim- und Gebäudeautomation</b>	<b>110</b>
6.1	Einführung	110
6.1.1	Softwareagenten	110
6.1.2	Multiagentensysteme	113
6.2	Design - und Modellierungswerkzeuge für Multiagentensysteme	113
6.2.1	Design von Multiagentensystemen mit Prometheus	113
6.2.2	Modellierung eines Multiagentensystems mit JADDEX	114
6.3	Vorgeschlagene Multiagentenplattform	114
6.3.1	Beschreibung von Softwareagenten	116
6.3.2	Interface Agent	118
6.3.3	Trending Agent	119
6.3.4	Presence Agent	120
6.3.5	User Agent	121
6.3.6	Conflict Resolution Agent	123
6.3.7	Control Agent	124
6.3.8	Weather Forecast Agent	125
6.3.9	Building Analysis Agent	126
<b>7</b>	<b>Schlussfolgerungen und Ausblick</b>	<b>127</b>
<b>8</b>	<b>Anhang</b>	<b>129</b>

# 1 Einführung

## 1.1 Motivation

Das Leben von Menschen wird immer dynamischer und die täglichen Aufgaben und Anforderungen nehmen laufend zu. Technische Hilfsmittel helfen uns diese Aufgaben zu meistern. Angefangen dabei, dass Menschen mit Hilfe von Telekommunikationstechnik über weite Strecken hinweg in Kontakt stehen können, Flugzeuge in engen Zeitslots sicher mit Hilfe von Radar gestützter Luftraumüberwachung und Flugleitsystemen Passagiere und Fracht transportieren können, Menschen mittels Satellitennavigationsgeräten in fremden Gegenden den richtigen Weg zu ihrem gewünschten Ziel finden und viele weitere nicht weniger wichtige Anwendungen mehr. Auch in der Heim- und Gebäudetechnik macht dieser rasanter werdende Fortschritt nicht halt.

Betrachtet man Elektroinstallationen in Wohn- und Nutzbauten, dann kann man den Schritt von der klassischen Einzelverkabelung hin zu aktuell verbauten Heim- und Gebäudeautomationssystemen auf Basis von modernen Kommunikationsnetzwerken beobachten. Diese Maßnahme reduziert den Verkabelungsaufwand, Wartungs- und Instandhaltungskosten sowie die Freiheit nach der Fertigstellung der Installation Änderungen nach ganz speziellen Wünschen der Anwender an der Schaltungslogik vorzunehmen. Beispielsweise kann ein Lichtschalter, der in der ursprünglichen Variante eines Gebäudeautomationssystems nur einen Beleuchtungskörper schalten konnte, so umkonfiguriert werden, dass genau dieser Lichtschalter eine ganze Gruppe von Leuchten schaltet, ohne nur eine Kabel- oder Steckverbindung der ursprünglichen physikalischen elektrischen Installation zu ändern.

Eine weitere Errungenschaft solcher Heim- und Gebäudeautomationssysteme ist, dass komplexe Szenarien einprogrammiert werden können. Am einfachsten lässt sich das an Hand von 2 Szenarien illustrieren. Ein Szenario soll „Fernsehen“ lauten, ein anderes eine Zentralfunktion „alles aus“ übernehmen. Bei dem Szenario „Fernsehen“ sollen die Fenster mit Hilfe von Rollos verdunkelt und die elektrische Beleuchtung gedimmt werden. Im zweiten Fall werden sämtliche Beleuchtungskörper ausgeschaltet und die Rollos nach oben gefahren. Beide Szenarien erfordern den Eingriff in die Steuerung der Rollläden und Beleuchtungskörper, zwei in der klassischen Elektroinstallation üblicherweise getrennten Systemen. Doch mit nur einem Tastendruck lassen sich Beleuchtung und Verschattung in einem modernen Heim- und Gebäudeautomationssystem perfekt steuern, um den Wünschen der Anwender rasch und bequem gerecht zu werden.

Mit heutigen Heim- und Gebäudeautomationssystemen lassen sich einfache Änderungen, wie zum Beispiel die Raumtemperatur, anpassen und einfach durchführen, so dass diese den Vorstellungen der Benutzer entspricht. Komplizierte Änderungen des Steuerungsverhaltens, wie in den beiden vorangegangenen Szenarien illustriert sind, lassen sich allerdings oft nur durch teures qualifiziertes Fachpersonal konfigurieren und abändern.

Für Anwender wäre es einfacher, wenn die Heim- und Gebäudeautomationsanlagen von den Vorlieben und Gewohnheiten des täglichen Ablaufs ihrer Bewohner durch Beobachtung lernen könnten, um die Steuerungsaufgaben für sie geschickt versteckt im Hintergrund automatisch ausführen zu können. Dabei erstreckt sich das Spektrum der Möglichkeiten auf Raumheizung, Lüftung, Kühlung, Brauchwasseraufbereitung, Lichtregelung, Verschattung,

Multimediaangebote, Alarm- und Sicherheitssysteme, mobile und stationäre Roboter und auf viele weitere Anwendungen. Mit einem selbstlernenden Heim- und Gebäudeautomationssystem ist es möglich, die kontextspezifischen Komfortansprüche der Bewohner eines Gebäudes zu erfüllen.

Ferner können, da die Gewohnheiten der Bewohner bekannt sind, Energiekosten bei Heizung, Kühlung, Beleuchtung und anderen elektrischen Verbrauchern gespart werden. Sind zum Beispiel keine Bewohner in einem Raum, dann sollte die Heizung, Kühlung oder Lüftung reduziert und die Beleuchtung oder andere elektrische Verbraucher komplett abgeschaltet werden. In Zeiten von immer weiter steigenden Energiepreisen [1] und dem stärker in den Köpfen der Bevölkerung vordringenden Umweltgedanken [2] können solche Maßnahmen die Energiekosten reduzieren und letztlich helfen Kosten einzusparen.

Nicht nur von den Bewohnern kann ein Heim- und Gebäudeautomationssystem lernen, sondern auch vom Gebäude selbst. Für Licht und Heizung stellt die Gebäudeausrichtung nach den Himmelsrichtungen, dem Gebäudestandort und der Raumeinteilung und Raumnutzung sowie Kopplungseffekte einen wesentlichen Faktor dar. Auch das aktuelle Wetter, Trends und Wettervorhersagen können bei energieintensiven Aufgaben wie Heizen oder Kühlen eines Raumes oder Gebäudes wichtige Einflussgrößen für Regelentscheidungen sein.

Ein weiterer Vorteil eines solchen Systems ist, dass Sicherheitsmaßnahmen automatisch beim Verlassen des Objekts aktiviert werden können und auf eventuelle Gefahrenpotentiale, wie zum Beispiel ein offen stehendes Fenster, in geeigneter Weise hingewiesen werden kann. Auch die medizinische Sicherheit der Bewohner kann mit einem selbstlernenden Heim- und Gebäudeautomationssystem gewährleistet werden.

Die medizinischen Standards sind heute bereits so hoch, dass die ältere Bevölkerung im Vergleich zu früher immer mehr zunimmt [3]. Damit wird auch mehr und mehr in der Altenbetreuung qualifiziertes medizinisch ausgebildetes Pflege- und Fachpersonal benötigt, das in der Mehrzahl der heutigen Organisationsstrukturen in Krankenhäusern oder Altenpflegeheimen anzutreffen ist. Nicht nur der Betrieb dieser Anstalten kostet Ländern und Bund einen großen Anteil an Geld, sondern auch der zukünftige Mangel an Pflege- und Betreuungsplätzen wird mit dem aktuellen Betreuungsmodell ein Problem.

Abhilfe schafft das Modell für betreutes Wohnen - zu Hause im Eigenheim. Bei diesem Betreuungsmodell kümmert sich mobiles medizinisches Personal um mehrere Patienten. Doch besteht in herkömmlichen Wohnungen die Gefahr, dass ein Patient zum Beispiel stürzt und keine Chance hat rechtzeitig Hilfe heran zu rufen. Das ließe sich mit einer Unterstützung durch die Gebäudeinfrastruktur bewerkstelligen. In dem aufgezeigten Fall müsste das Gebäude so intelligent sein, um den Sturz zu erkennen und dann mit geeigneten Maßnahmen entsprechend zu reagieren. Angemessen heißt in diesem Fall, dass das Gebäude einen Sturz erkennt, versucht mit der betroffenen Person Kontakt über zweckmäßige Mittel herzustellen und falls kein Kontakt hergestellt werden konnte, den nächsten für diesen Fall geeigneten Kontakt zu alarmieren.

## 1.2 Absteckung der Fragestellung

Ziel dieser Arbeit ist eine Untersuchung, mit welchen geeigneten Mitteln aktuellen Heim- und Gebäudeautomationssystemen intelligentes Handeln „beigebracht“ werden kann. Dabei werden zwei Fragen verfolgt:

- a) Mit welchen Mitteln können komplexe Einflussfaktoren von Mensch, Gebäude, Technik und Umwelt in intelligenten Regelkreisalgorithmen berücksichtigt werden?
- b) Wie können komplexe Softwaresysteme aufgebaut werden, die das Verhalten von Mensch, Gebäude, Technik und Umwelt berücksichtigen?

## 1.3 Aufbau der Arbeit

Im nachfolgenden Kapitel *2 Stand der Wissenschaft und Technik* werden aktuelle Trends in Wissenschaft und Technik dargestellt.

Mögliche Anwendungsszenarien, die in einem intelligenten Haus anzutreffen sind, werden in Kapitel *3 Anwendungsszenarien* vorgestellt.

In Kapitel *4 Optimierte Kontrollstrategien* wird der Frage in Kapitel 1.2 unter Punkt a) nachgegangen. Dabei beschäftigt sich diese Arbeit mit der optimalen Nutzung der vorhandenen technischen automatisierbaren Ressourcen unter dem ökologischen und ökonomischen Gesichtspunkt. Die größten Potentiale findet man<sup>1</sup> bei Heizungs-, Kühlungs-, Lüftungs-, Beleuchtungstechnik und Standby-Geräten. Deswegen werden Algorithmen zur optimalen Heizungssteuerung untersucht, da hier das größte monetäre Einsparungspotential zu finden ist.

Der erarbeitete Regelalgorithmus wird in Kapitel 5 in mehreren Varianten in einem Gebäudesimulationsmodell getestet und anhand von Energieverbrauchsdaten analysiert. Es wird gezeigt, wie groß die Energieeinsparungen durch intelligentes Heizen sind.

In Kapitel 6 wird eine Antwort auf die Frage unter Punkt (b) gegeben. Dabei wird eine Softwareengineeringmethode in allen Einzelheiten vorgestellt, um eine intelligente Heim- und Gebäudetechnik zu ermöglichen, die es erlaubt Komfort- und Energiesparansprüche zu erfüllen.

Im letzten Kapitel *7 Schlussfolgerungen und Ausblick* werden die gesammelten Ergebnisse diskutiert und ein Ausblick gegeben, wie diese in der Zukunft genutzt werden können.

---

<sup>1</sup> der Reihenfolge nach geordnet.

## 2 Stand der Wissenschaft und Technik

In diesem Kapitel werden wichtige, mit dieser Arbeit assoziierte Ansätze aus der aktuellen Forschung und kommerziell erhältliche Produkte sowie Feldstudien vorgestellt. Besonders wird an geeigneter Stelle betont, welche Teile der vorgestellten Produkte und Feldstudien mit den in dieser Arbeit erarbeiteten Ergebnissen in Verbindung stehen.

### 2.1 Einleitung

An dieser Stelle wird zuerst das Thema der Möglichkeiten eines optimalen Heizungsstartstopps vorgestellt. Begonnen wird der Abschnitt mit der Vorstellung des klassischen Fuzzy Logic Reglerkonzepts. Diese Methode hat sich bereits in der Heim- und Gebäudeautomation durchgesetzt und wird in vielen Heizungssteuerungen eingesetzt. Artikel [4] behandelt das Thema „Optimum Start-Stop“ mit künstlich neuronalen Netzen<sup>2</sup>. Passend zu ANNs wird anschließend das Projekt ACHE [5] im Adaptive House vorgestellt. Bei diesem wissenschaftlichen Projekt wurden Beleuchtung, Lufttemperatur, Ventilation und Warmwasseraufbereitung mit ANNs geregelt.

Als nächstes werden wissenschaftliche Ansätze gezeigt, wie man die Komplexität eines intelligenten Systems in einer Softwarearchitektur in den Griff bekommen kann. Eine komplexe Aufgabe einer Softwarearchitektur für Heim- und Gebäudeautomationssysteme muss erkennen, welcher Bewohner welchen Tätigkeiten aktuell nachgeht und in welchem aktuellen Kontext diese stehen. Dieses Thema wird in dem Artikel [6] behandelt. Anschließend folgt das Forschungsprojekt AMIGO [7], ein Middlewareansatz eines Industriekonsortiums unter der Federführung der Firma Philips, die sich zum Ziel gesetzt hat, Automationstechnik, braune und weiße Ware sowie IT-Komponenten miteinander zu verbinden. Ein ganz anderer Ansatz wird mit Agentensystemen verfolgt. Es wird gezeigt, wie Smart Homes verbessert [8] und wie Multiagentensysteme für Heim- und Gebäudeautomation modelliert [9] werden können.

Verfügbare Produkte bilden den ersten Teil der praktischen Anwendungen. Hier wird Heim- und Gebäudeautomationsproduktlinie Siemens Synco Living [10] für den Heimgebrauch vorgestellt. MyPersonalGekko [11], von der Firma Ekon in Südtirol, zeigt eine drahtlose Gebäudeautomation für die jüngere Generation als Zielgruppe. Ein Gebäudeautomationssystem der Schweizer Firma Adhoco [12] kann ebenfalls in diese Kategorie Technologie für junge Menschen eingeordnet werden, aber auch für betreutes Wohnen für ältere Menschen Verwendung finden.

Zum Abschluss des Kapitels Stand der Wissenschaft und Technik werden Feldstudien vorgestellt. Beginnend mit dem Fraunhofer inHaus-Zentrum in Duisburg in Deutschland, das als Werkstatt für die Umsetzung neuer Ideen gilt. Ein weiteres interessantes Projekt ist Living Tomorrow in Vilvoorde in Belgien. Hier beschäftigt man sich mit dem Thema, wie Haushaltsgeräte und Wohnen in der nahen Zukunft für die Bewohner einfacher und intelligenter gemacht werden können. Das Projekt ist ein Zusammenschluss von mehr als 50 Firmen. Den Abschluss dieses Abschnittes bildet das Projekt Adaptive House von M. Mozer in Boulder in Colorado.

---

<sup>2</sup> Im nachfolgenden Text kurz KNN genannt oder synonym engl. Artificial Neural Network, kurz ANN.

## 2.2 Wissenschaftliche Forschungen

### 2.2.1 Optimum Start-Stop

Bei Optimum-Start-Stop wird unter gewissen Optimierungsregeln das primäre Heizsystem für die Raumheizung oder die Warmwasseraufbereitung erst dann mit voller Leistung gestartet, wenn Wärmeenergie benötigt wird. Dabei wird vorrangig geringer Energieverbrauch als Zielfunktion betrachtet.

Beispielsweise soll im Bad die Raumtemperatur um 7h morgens 24°C betragen. Um das zu gewährleisten, muss die Heizung eine gewisse Zeit im Vorhinein starten, damit die Temperatur im Raum zum gewünschten Zeitpunkt erreicht wird. Natürlich darf das nicht zu früh passieren, damit nicht unnötig Energie verschwendet wird. Dabei müssen eine Vielzahl von Parametern zur Berechnung des Startzeitpunkts, wie etwa gemessene Raumtemperatur vor 7 Uhr morgens, Raumgröße, wie schnell heizt der Raum auf<sup>3</sup>, Außentemperatur, Raumausrichtung und viele andere Größen berücksichtigt werden. Häufig wird bei Heizsystemen im Wohn- oder Nutzbau ein Warmwasserboiler verwendet. Mit zunehmenden Maß zu einem herkömmlichen primären Heizungskesselsystem werden auch sekundäre Warmwasseraufbereitungsanlagen mit Energie aus alternativen Energiequellen installiert. Diese leisten aufgrund ihrer Natur keine konstante Energieabdeckung für den durchgehenden Betrieb der Heizungsanlage. Daher muss die überschüssig erzeugte Energie gespeichert werden.

#### 2.2.1.1 Applications of artificial neural network to predict optimal start time for heating system in building [4]

In dieser Arbeit werden künstlich neuronale Netze zur Vorhersage des optimalen Startzeitpunktes für die Heizung eines Zweckbaus beschrieben, die einen fixen Belegungsplan haben. In den Belegungszeiträumen muss Komforttemperatur in den Räumen vorliegen, damit sich die Nutzer nicht unwohl fühlen. Jedoch darf die Heizung nicht zu früh gestartet werden, damit nicht unnötig Energie für die Heizung in jener Zeit verschwendet wird, wo die Räume noch gar nicht belegt sind.

Der optimale Startzeitpunkt für die Heizung hängt von mehreren Faktoren ab, beispielsweise von den Klimabedingungen am Gebäudestandort, Kapazitäten des Heizsystems, Gebäudestruktur. Will man diese Faktoren für die exakte Bestimmung des optimalen Startzeitpunktes einfließen lassen, so müsste man ein rekursives Gleichungssystem für eine stark nichtlineare Regelstrecke aufstellen. Diese Gleichungen müssten zu jeder Jahreszeit adaptiert werden, um den geänderten Witterungsverhältnissen zu entsprechen. Zusätzlich verhält sich das Wetter nicht jedes Jahr gleich. Das bedeutet, dass das mögliche rekursive Gleichungssystem jedes Jahr neu angepasst werden müsste. Aus diesem Grund wird in der vorgestellten Arbeit die Regelung des Heizungssystems eines Zweckbaus mit künstlich neuronalen Netzen untersucht.

---

<sup>3</sup> Gebäudephysik

### 2.2.1.2 Künstlich neuronale Netze

Künstlich neuronale Netze sind natürlichen neuronalen Netzen<sup>4</sup> nachempfunden und bestehen aus einem oder mehreren Eingabeneuronen<sup>5</sup>, ein oder mehrere versteckten Schichten mit einem oder mehreren versteckten Knoten und einem oder mehreren Ausgabeknoten. In natürlichen neuronalen Netzen sind die Neuronen über das Soma, einem Nervenkanal verbunden, bei künstlich neuronalen Netzen spricht man häufig auch von Kanten. Künstliche Neuronen besitzen wie ihr natürliches Vorbild einen oder mehrere Eingänge. Wird an den Eingängen eines Neurons eine Eingabe angelegt, die eine gewisse Schwelle überschreitet, „feuert“ das Neuron. In künstlich neuronale Netze können verschiedenste Aktivierungsfunktionen<sup>6</sup> für unterschiedlichste Anwendungszwecke eingesetzt werden.

Künstlich neuronale Netze müssen mit Trainingsdaten ihr Verhalten lernen. Dafür werden alle Eingabe- und Ausgabetrainingsdaten normiert, sodass diese in einem Wertebereich zwischen 0 und 1 liegen.

In der vorgestellten Arbeit werden Feedforward Netze<sup>7</sup> verwendet, die mit dem Backpropagation-Algorithmus, einer Variante des überwachten Lernens, trainiert wurden.

### 2.2.1.3 Das Simulationsmodell

Das Modell basiert auf einem Raum, modelliert mit einer finite Elemente Toolbox, wie sie im Buch [13] vorgestellt wurde. Der Raum besteht aus 4 Wänden, Fußboden, Decke, 1 Fenster, 1 Heizkörper, 1 Person, elektrischen Geräten (PC mit Monitor), Einrichtungsgegenständen, Beleuchtungskörper und einer Kopplung zu benachbarten Räumen. Das Verhalten des Raumes ist abhängig von der Raumhöhe, der Raumfläche, der Fensterfläche, internem Wärmegewinn, Beiwert der Verschattung und Einflussgrößen aus benachbarten Räumen. Bei letzteren werden 3 Fälle unterschieden:

- a) 1 Seite ist eine Außenwand und die 3 übrigen Seitenwände sind Innenwände
- b) 2 Seiten sind Außenwände und die beiden übrigen Seitenwände sind Innenwände
- c) Alle 4 Seiten sind Außenwände

Für die Simulation wurde ein experimentelles Systemdesign angewendet, um Lerndaten zu erstellen, da für die thermische Analyse zu viele Simulationsschritte notwendig und nicht handhabbar gewesen wären.

### 2.2.1.4 Das Modell eines künstlich neuronalen Netzes

Wichtig ist, dass das neuronale Netz so wenig Fehler wie nur möglich macht, damit keine unnötige Energie verschwendet wird und sich die Nutzer in den Räumen wohl fühlen. Die gestellten Anforderungen können mit einem Feedforward Netz gelöst werden. Das Verhalten des künstlich neuronalen Netzes wird mit Backpropagation-Lernen trainiert. Die

---

<sup>4</sup> Beispielsweise dem menschlichen Gehirn.

<sup>5</sup> Neuronen oder auch Knoten, wie aus der mathematischen Graphentheorie bekannt, bezeichnet.

<sup>6</sup> Beispielsweise ein Schwellenwert, eine lineare Funktion oder eine sigmoide Funktion. Weitere Details findet man im Abschnitt 4.2

<sup>7</sup> Die Kanten entsprechen einem gerichteten Graphen, dessen Durchlaufrichtung auf neuronale Netze angewendet von der Eingabeschicht über eventuell vorhandene versteckte Schichten zur Ausgabeschicht und auch ohne Schleifen ist.

Autoren des Artikels evaluieren die optimale Lernrate, das optimale Moment, die optimale Anzahl der versteckten Schichten, die optimale Anzahl der versteckten Knoten in den versteckten Schichten und den Einfluss von externen Störgrößen.

Zuerst werden die Eingabedaten mit den dazugehörigen Ausgabedaten mit dem Simulationsmodell erarbeitet. Die Eingabevariablen sind die Anzahl der Eingabeknoten in der Eingabeschicht, vgl. dazu Abbildung 1. Als Eingabevariablen gelten die aktuelle Außenlufttemperatur, die Änderungsrate der Außenlufttemperatur, die aktuelle Innenlufttemperatur und die Änderungsrate der Innenlufttemperatur.

Die Ausgabe des künstlich neuronalen Netzes ist der optimale Heizungsstartzeitpunkt. Zwischen der Eingabe- und Ausgabeschicht können mehrere Schichten mit versteckten Knoten liegen, deren Knotenanzahl unterschiedlich sein kann.

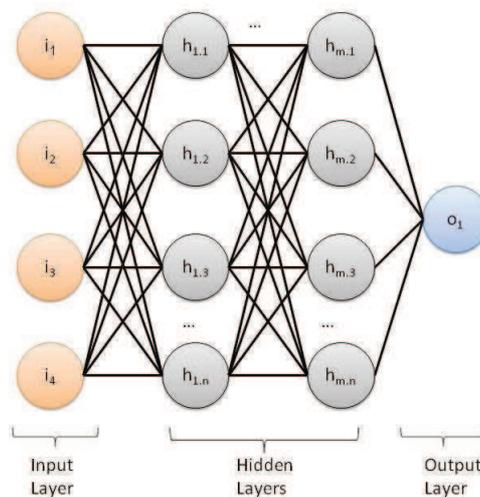


Abbildung 1: Generisches Feedforward neuronales Netz

Die Datensätze werden normalisiert, bevor sie an das künstlich neuronale Netz angelegt werden. Damit wird erreicht, dass die Ausgabe durch die Eingabedaten nicht in Sättigung geht. Daher wurde jeder einzelne Eingabedatensatz mit der Formel 1 normalisiert.

$$X = \frac{X_i - X_{\min}}{X_{\max} - X_{\min}} \cdot 0,8 + 0,1$$

Formel 1: Normalisierung der Eingabewerte

### 2.2.1.5 Optimierung des künstlich neuronalen Netzes

Zuerst wurde die optimale Lernrate ermittelt. Dafür wurde ein künstlich neuronales Netz, wie in Abbildung 1 gezeigt, mit 4 Eingabeknoten, einer versteckten Schicht mit 8 versteckten Knoten, einem Ausgabeknoten und einem variablen Bias, der auf die Ein- und Ausgabedaten wirkt, verwendet. Der absolute Fehler für die optimalen Lernergebnisse weist keine signifikanten Abweichungen für eine Lernrate zwischen 0,2 und 1,0 auf. Die optimale Lernrate mit dem kleinsten Fehler liegt bei 0,75.

Das Variationsmoment stellt die Messgröße dar, die die steigende Lerneffizienz und die sinkende Lernzeit beschreibt. Für diesen Zweck wurde das künstlich neuronale Netz wie für

die Messung der Lernrate verwendet. Das optimale Moment, wo der totale Fehler minimal ist, liegt bei 0,9.

Für die Ermittlung der optimalen Anzahl der versteckten Schichten wurde ein neuronales Netz, wie schon vorhin beschrieben, mit 4 Eingabeknoten, 8 Knoten pro versteckter Schicht, einem Ausgabeknoten und einem variablen Bias verwendet. Es wurde iterativ getestet, ob 1, 2 oder 3 versteckte Schichten unter der Bedingung, dass der totale Fehler minimal ist, ermittelt. Es stellte sich heraus, dass eine versteckte Schicht optimal ist, aber der totale Fehler mit 2 Schichten minimal größer ist als mit einer Schicht.

Die Anzahl der Knoten pro versteckten Layer wurde in ähnlicher Weise, wie in den Messungen zuvor ermittelt. Dabei wurden 4 Eingabeknoten, 1 versteckte Schicht mit 6 bis 10 Knoten, einem Ausgabeknoten und einem variablen Fehlerterm ermittelt. Als optimal wurden 8 Knoten für die versteckte Schicht gefunden.

In den Tests wurde festgestellt, dass sich ein variabler Bias auf das künstlich neuronale Netz besser auswirkt als mit keinem.

Variable	Optimaler Wert
Lernrate	0,75
Moment	0,90
Anzahl der versteckten Schichten	1
Anzahl der Knoten einer versteckten Schicht	8
Bias	variabel

Tabelle 1: Zusammenfassung der ermittelten optimalen Werte für ein Feedforward KNN

In Tabelle 1 sind die optimalen Werte für ein Feed-Forward Netz zusammengefasst.

Evaluiert wurde das ermittelte künstlich neuronale Netz für 27 Räume, wo jeweils ein künstlich neuronales Netz zum Einsatz kam. Für jeden Raum wurden 100 Lerndaten zur Verfügung gestellt, die insgesamt bis zu 3000-mal gelernt wurden. Es zeigte sich, dass der totale Fehler ab 100 Lerndurchgängen unter 2 und bei 3000 Durchgängen nicht wesentlich weniger ist. Die Autoren verglichen auch das Verhalten des künstlich neuronalen Netzes mit Lerndaten im Vergleich zu historisch eingelernten Daten durch das künstlich neuronale Netz. Der Koeffizient lag bei 0,988. Das zeigt, dass künstlich neuronale Netze sehr gut die Ein- und Ausgabebeziehungen lernen können, um dann im Anwendungsfall von den Eingabedaten auf die Ausgabedaten schließen zu können.

### 2.2.1.6 Verbindung mit dieser Arbeit

In dieser Arbeit wird auch das Thema optimale Heizungsstart mit künstlich neuronalen Netzen behandelt. Im Vergleich zu der vorgestellten Arbeit wird das Anwendungsszenario in dieser Diplomarbeit auf ein Wohnhaus verlagert. Es werden verschiedene Netztopologien evaluiert.

### 2.2.2 Szenarienerkennung

Durch Identifizieren von einzelnen Personen und deren Handlungen, aber auch aus dem Wissen über das Verhalten des Hauses und dessen Umgebung, können Voraussagen über zukünftige Zustände und Ereignisse gemacht werden.

### 2.2.2.1 Scenario Recognition in Modern Building Automation

Ein intelligentes Heim- und Gebäudeautomationssystem muss Ereignisse, die bereits eingegangen sind, erkennen können, um auf zukünftiges Verhalten zu schließen und daraus Regel- und Steuerparameter zu errechnen.

Ein Gebäudeautomationssystem besteht aus verschiedensten Sensoren und Aktuatoren. Einzelne Sensorwerte oder Gruppen von Sensorwerten sind Momentaufnahmen, die zu einem gewissen Zeitpunkt aufgenommen wurden. Betrachtet man eine Momentaufnahme, dann fällt es schwer zu beurteilen, was in der Vergangenheit geschehen ist, um daraus mögliche zukünftige Vorgänge abzuleiten. Aus diesem Grund wird eine gewisse Anzahl an Momentaufnahmen zu einem Szenario assoziiert, das in der Vergangenheit abgelaufen ist. Mit dem Wissen, welches Szenario gerade abläuft, lässt sich mit ziemlich großer Wahrscheinlichkeit eine richtige Aussage darüber treffen, was in naher Zukunft passieren wird und damit Steuerungswerte für Aktuatoren zu errechnen.

Die Projektgruppe um R. Lang hat sich dabei mit 2 verschiedenen Ansätzen beschäftigt, wie Szenarien erkannt werden können. Zum einen mit einem strukturierten Szenarienerkennungssystem und zum anderen mit statistischen Methoden. Nachfolgend werden beide Ansätze vorgestellt und im Anschluss die Vor- beziehungsweise Nachteile gegenüber gestellt [6].

#### 2.2.2.1.1 Strukturiertes Szenarienerkennungssystem

Diese Methode erfasst und interpretiert verschiedene Sensorwerte semantisch mit vordefinierten Wahrnehmungsmustern, die Image Templates genannt werden. Ein Image Template stellt einen Schnappschuss von semantisch interpretierten Sensorwerten zu einem bestimmten Zeitpunkt dar. Aus den Image Templates über die Zeit hinweg betrachtet werden dann Szenarien erkannt. Das bedeutet, hat das System ein Szenario erkannt, dann wird über Image Templates auf ein nächstes Szenario geschlossen.

Diese Technik wird in 3 Phasen aufgeteilt: in die Design Phase, die Template Definition Phase und die Operational Phase, wie in Abbildung 2 dargestellt. In den ersten beiden Phasen interagiert das System mit einem Menschen. In der Operational Phase läuft das System automatisch ohne einen menschlichen Eingriff.

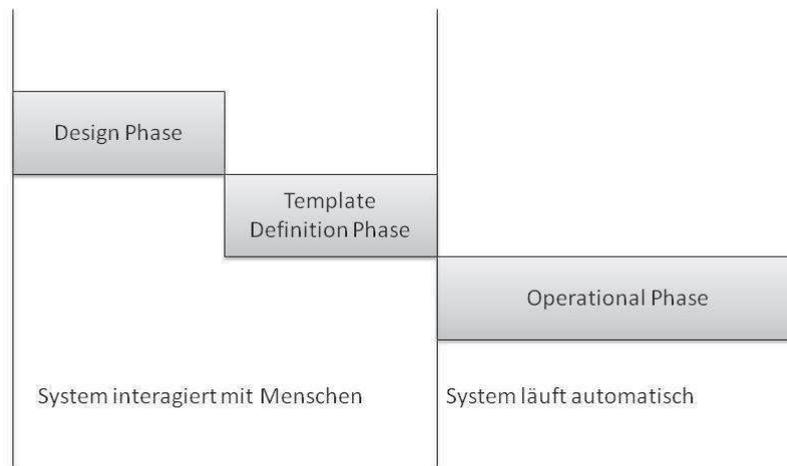


Abbildung 2: 3 Phasen des strukturierten Szenarienerkennungssystems

In der Design Phase wird festgelegt, welche Sensorentitäten im System vorhanden sind. In der Template Definition Phase werden Sensordaten Symbole<sup>8</sup> zugeordnet, die dann wiederum Szenarien zugehörig sind. Man unterscheidet zwischen Image Templates<sup>9</sup>, die die wahrgenommenen Sensordaten zu einem definierten Zeitpunkt darstellen und Scenario Templates, die Sequenzen von Image Templates darstellen. Die Operational Phase startet dann, wenn alle Image Templates und Scenario Templates definiert und wenn ein oder mehrere Szenarien erkannt wurden.

Mathematisch lässt sich ein Image Template  $IT_n$ , wie nachfolgend beschrieben, definieren. Es gibt eine Menge  $S$  an möglichen Symbolen (z.B.  $S_1, S_2, \dots, S_i$ ). Diese Menge  $S$  beinhaltet die tatsächlich wahrgenommene Symbolmenge  $P$  (z.B.  $S_3, S_5, \dots, S_j$ ). Ein Image Template  $IT_n$  ergibt sich aus Konjunktion der gewichteten einzelnen wahrgenommenen Symbole  $S$  der Menge  $P$ .

Um nun zu berechnen, mit welcher Wahrscheinlichkeit ein Image Template  $IT_n$  auf die gegebenen Sensorwerte zutrifft, wird ein Baum konstruiert, um die gewichteten einzelnen Sensorwerte zu evaluieren. Dafür sind 3 Basiselemente, wie in Tabelle 2 zu sehen, definiert.

	Symbol	Basis
Image Element iE	○	-
Image Node iN	◇	iE
Image Leaf iL	☆	iE

Tabelle 2: Image Template Basiselemente

Eine Entität vom Typ Image Element enthält die Daten: Name, Beschreibung, optionaler oder zwingender Knoten. Der Typ Image Node hält dieselben Daten wie Image Element und zusätzlich noch, ob der Knoten Kindknoten hat und welchen Typ, Image Node oder Image Leaf der Kindknoten hat. Zusätzlich beinhaltet der Image Node den Datentyp und die binären Verknüpfungsoperatoren UND und ODER sowie den unären Operator NEG für jeden Kindknoten. Ein Image Leaf besitzt alle Elemente von Image Element und zusätzlich den

<sup>8</sup> Zum Beispiel die Anzahl der Personen in einem Raum, Lufttemperatur oder Helligkeit.

<sup>9</sup> Satz von Regeln, die die Wahrnehmung bestimmter Symboltypen erlauben.

Datentyp des Symbols und einen Operator ( $=, \neq, <, \leq, >, \geq$ ), um den Wert der Symbole und den definierten Wert des Image Leafs zu vergleichen. Abbildung 3 skizziert das in einem Beispiel.

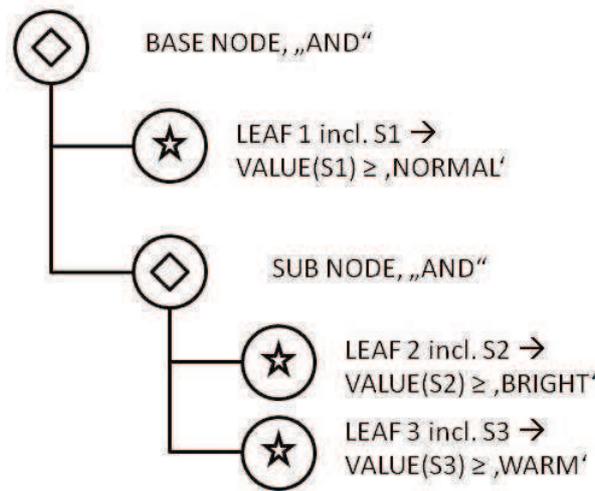


Abbildung 3: Beispiel eines Image Templates

Anschließend wird berechnet, welches Image Template am besten passt. Dazu werden die Wahrscheinlichkeiten berechnet, mit der ein Image Template zutrifft. Jenes mit der höchsten Wertung wird dann als zutreffendes Image Template ausgewählt. Mit welcher Wahrscheinlichkeit ein Image Template vorliegt, wird in der Tabelle 3 für das Beispiel in Abbildung 3 exemplarisch gezeigt.

Wahr-scheinlichkeit	L1	L2	L3
1,0	X	X	X
1,0	X	X	-
1,0	X	-	X
0,5	X	-	-
0,5	-	X	X
0,5	-	X	-
0,5	-	-	X
0,0	-	-	-

Tabelle 3: Wahrscheinlichkeit mit der ein Image Templates zutrifft

Steht fest, welches Image Template zutrifft, wird durch die zeitliche Abfolge der Symbole ein Szenario festgelegt. Dies wird durch einen Graphen<sup>10</sup> gezeigt, wie zum Beispiel in Abbildung 4 dargestellt.

<sup>10</sup> Als Darstellungsform wurde die Automatendarstellung gewählt.

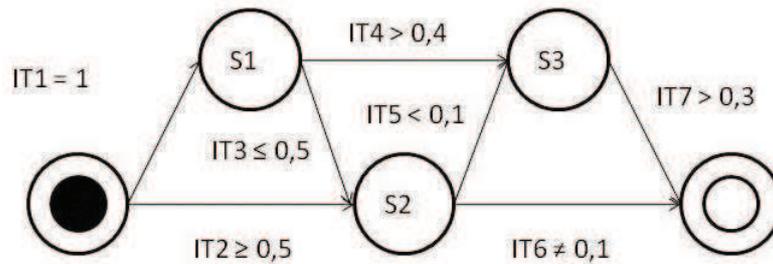


Abbildung 4: Beispiel Szenario Template

Zwischen den Knoten, welche die geltenden Symbole darstellen, ist die Übergangswahrscheinlichkeit eingezeichnet. Kann ein Graph vom Start bis zum Ziel durchlaufen werden, dann wurde ein Szenario erkannt. Treffen Übergangswahrscheinlichkeiten von einem Knoten zu einem der Nachfolgerknoten nicht zu, wird das Szenario verworfen.

### 2.2.2.2 Szenarien erkennen mit stochastischen Methoden

Die zweite Methode verwendet stochastische Methoden zur Szenarienerkennung. Dabei bedient man sich einem Hidden-Markov-Modell (HMM), das schon in anderen informationsverarbeitenden Disziplinen zum Suchen von Sequenzen eingesetzt wird. Beispielsweise kann dieses Verfahren als Mensch-Maschine-Schnittstelle für die Computerspracherkennung von natürlich gesprochener Sprache oder Gestenerkennung eingesetzt werden.

Im konkreten Anwendungsfall „Erkennen von Szenarien“ geht es darum, aus einer großen Anzahl von Sensordaten wiederkehrende Verhaltensmuster zu erkennen, diese zu klassifizieren und daraus auf ein Szenario zu schließen. Mit der gewonnenen Erkenntnis lassen sich Aktuatoren so steuern wie von den Bewohnern des Hauses gewünscht. In Abbildung 5 wird der Workflow gezeigt, wie das Hidden-Markov-Modell für den konkreten Anwendungsfall modelliert wird.

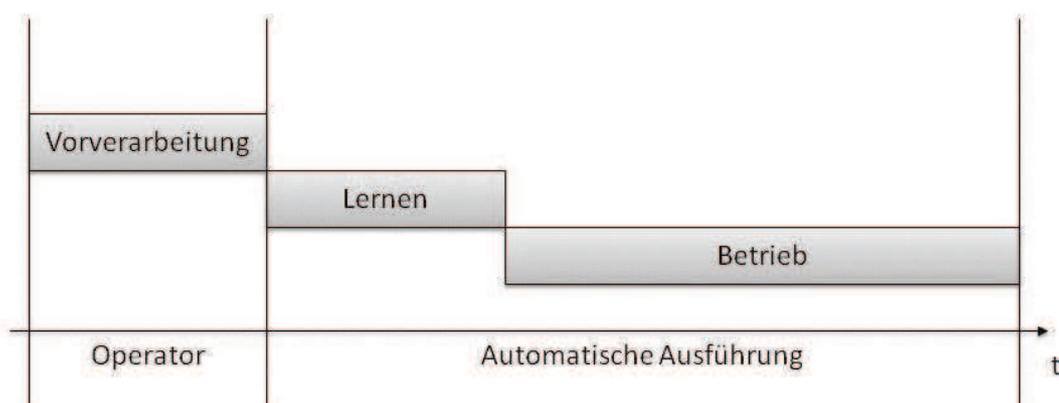


Abbildung 5: Implementierungsphasen des Hidden-Markov-Modells

Der Modellierungsprozess besteht aus einer Vorverarbeitungs-, Lern- und Betriebsphase. In der Vorverarbeitungsphase wird dem System bekannt gemacht, welche Sensoren zur Datenverarbeitung zur Verfügung stehen und welche Daten davon wichtig sind. Daraus wird die Menge aller Zustände, die Grundsteine der Mustererkennung, modelliert. Mit diesen Erkenntnissen, die in einer Datenbank gespeichert sind, kann auf das Verhalten von

Systemen oder Personen geschlossen werden. Diese Aufgabe wird vorwiegend von einem menschlichen Operator durchgeführt.

In der darauffolgenden Lernphase wird durch den Ablauf verschiedener Sensorzustände, deren Muster in der Datenbank gespeichert sind, auf ein Szenario geschlossen. Das wird mit Emissionsübergangswahrscheinlichkeiten ausgedrückt. Die Wahrscheinlichkeit, dass ein bestimmtes Szenario auf einen Ablauf von Zuständen folgt, wird mit Übergangswahrscheinlichkeiten beschrieben. Ähnliche oder gleiche Szenarien werden grob von denselben Sensorwertereihen eingeleitet. Die Lernphase wird als abgeschlossen angesehen, wenn sich keine großen Änderungen an den Übergangswahrscheinlichkeiten feststellen lassen.

In der Betriebsphase wird anhand der gelernten Emissions- und Übergangswahrscheinlichkeiten der Sensorwerte auf Steuer- und Regelwerte für ein Szenario geschlossen.

Für diese Szenarienerkennung mit Hidden-Markov-Modellen wurden folgende Algorithmen verwendet:

- *Forward Algorithmus*: evaluiert mit welcher Wahrscheinlichkeit eine beobachtet Wertsequenz für das Modell zutrifft (Evaluierung)
- *Viterbi Algorithmus*: berechnet den wahrscheinlichsten Pfad durch das Modell (Dekodierung)
- *Forward-Backward* und *Baum-Welsh Algorithmus*: diese Algorithmen werden in der Lernphase dazu verwendet, um die Modellparameter an neue beobachtete Beobachtungssequenzen anzupassen (Lernen von Parametern)

Ein Beispiel eines möglichen Modells ist in Abbildung 6 dargestellt.

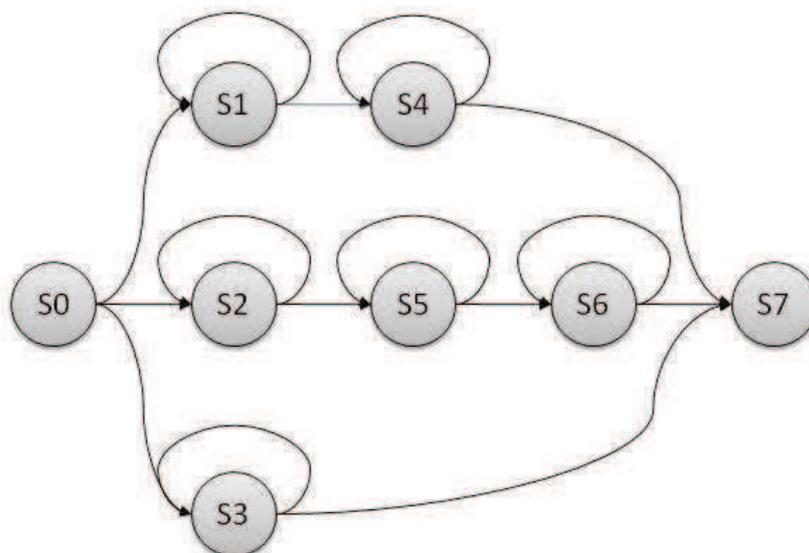


Abbildung 6: Beispiel eines Hidden-Markov-Modell

Die Kreise stellen die möglichen Zustände  $S_0$  bis  $S_7$  dar, wobei  $S_0$  den Startzustand symbolisiert und  $S_7$  den Endzustand. Die Pfeile zwischen den Zuständen symbolisieren Pfade durch das Modell. In dem gezeigten Modell gibt es 3 verschiedene Pfade, die tägliche Abläufe darstellen. Zum Beispiel ergibt sich ein Pfad  $S_0 \rightarrow S_2 \rightarrow S_5 \rightarrow S_6 \rightarrow S_7$ . Die Zustände

stellen tägliche Aufgaben dar. Beispielsweise kann  $S_2$  Kaffee kochen,  $S_5$  normale Arbeit und  $S_6$  eine Besprechung darstellen.

### 2.2.2.3 Vergleich der beiden statistischen Methoden

Beim strukturierten Szenarienerkennungssystem wird die Lernphase in der Designphase und die Templatedefinitionsphase von einem menschlichen Operator ausgeführt. Im Vergleich dazu wird bei der Methode Szenarien erkennen mit stochastischen Methoden nur in der Vorverarbeitungsphase ein menschlicher Operator benötigt. Der Ansatz des strukturierten Szenarienerkennungssystems geht davon aus, dass alle möglichen Szenarien vor der Betriebsphase dem System bekannt gemacht werden. Das hat den Vorteil, dass nur Sensorwertreihen mit genügend wichtiger Bedeutung als Template gespeichert werden. Auf der anderen Seite können neu auftretende Zustände mit der Methode, die sich das Hidden-Markov-Modell zu Nutze macht, im laufenden Betrieb in die Regelbasis dynamisch integriert werden.

Beide Methoden stellen gute Ansätze dar, wie ein Heim- und Gebäudeautomationssystem Szenarien erkennen kann. Kombiniert man die Vorteile beider Ansätze, dann werden die meisten Nachteile der jeweilig anderen Methode kompensiert. Auf der einen Seite hat man die Präzision der Sensorwertreihen in der Templatedatenbank, auf der anderen Seite die Dynamik, die ein Heim- und Gebäudeautomationssystem haben sollte, um das Verhalten von Bewohnern oder Gebäuden zu erkennen.

### 2.2.2.4 Verbindung mit dieser Arbeit

Eine Erkennungskomponente wird auch in der vorgestellten Softwarearchitektur<sup>11</sup> dieser Arbeit, im Presence Agent für das Personenverhalten und im Trend Agent für das Gebäudeverhalten verwendet.

## 2.2.3 Softwarearchitekturmodelle

### 2.2.3.1 AMIGO

Das europäische Forschungsprojekt Amigo [14] [7]<sup>12</sup> hat das Ziel, eine quelloffene Middleware-Plattform für die Vernetzung von unterschiedlichen Haustechnikkomponenten verschiedenster Hersteller zu erforschen und zu entwickeln. Das Projekt wurde im Herbst 2004 gestartet und endete 42 Monate später im Ende Februar 2008. Unter den Projektpartner waren unter anderem der VTT<sup>13</sup>, die französische Telekom, einige Fraunhofer Institute<sup>14</sup>, Microsoft Deutschland, Philips und universitären Einrichtungen<sup>15</sup> zu finden.

Die wichtigsten Ziele von Amigo sind die Erkennung des Kontextes, indem eine Aktion stattfindet oder stattgefunden hat, die Anpassung an Benutzerwünsche, eine einfache Benutzerschnittstelle zwischen dem System und den Benutzern über Sprachsteuerung oder Gesten, eine im Hintergrund agierende Technik. Bei AMIGO stehen die Benutzer im Zentrum

---

<sup>11</sup> Siehe Kapitel 6

<sup>12</sup> Ambient Intelligence for the networked home environment

<sup>13</sup> Technische Forschungsstätte in Finnland

<sup>14</sup> SIT (Deutschland), IPSI (Deutschland) und IMS (Deutschland)

<sup>15</sup> Universität Paderborn (Deutschland), Nationale Technische Universität Athen (Griechenland)

der Betrachtung, die Sicherheit der Benutzer, die Betriebssicherheit und die Geheimhaltung der gesammelten Benutzerinformationen. An die eingesetzten intelligenten Geräte und Anwendungen werden die Ansprüche einfache Konfigurierbarkeit und Verwaltbarkeit sowie eine hohe Zuverlässigkeit gestellt. Alle eingebundenen Geräte und Netzwerke sollen miteinander kommunizieren können.

Die Software Amigo folgt einem serviceorientierten Paradigma und ist für die Programmierplattform .NET entwickelt. Einige Komponenten sind nach Java/OSGi portiert worden. Die Beschreibung der Servicefunktionen, der Kontext sowie des Benutzermodell und zugehörige Profile werden mit Hilfe einer semantischen Wissensrepräsentation dargestellt.

In Abbildung 7 ist die Architektur der Amigo Middleware abgebildet. Als grundlegende Plattform können beliebige Kommunikationsbusse mit Feldgeräten angeschlossen werden. Es ist auch möglich, Geräte über verschiedene Kommunikationsbusse hinweg zu kontrollieren und zu steuern.

In der Base Middleware werden die Aufgaben Auffindung, Aufbau und Ausführung von Diensten<sup>16</sup> abgewickelt. Die 4 verschiedenen Auffindungstypen beinhalten die semantische und kontextsensitive Auffindung<sup>17</sup> von Diensten sowie Quality of Service. Für den Aufbau von Diensten wird die XML-basierte Sprache WS-Business Process Execution Language BPEL zur Modellierung verwendet. Zur Ausführung kommen einfache Remote Procedure Calls oder asynchrone Übertragung von Ereignissen<sup>18</sup>.

---

<sup>16</sup> Die Dienste sind als Webservices modelliert.

<sup>17</sup> Webservice-Discovery und Webservice-Adressing

<sup>18</sup> z.B. mit Webservice-Eventing

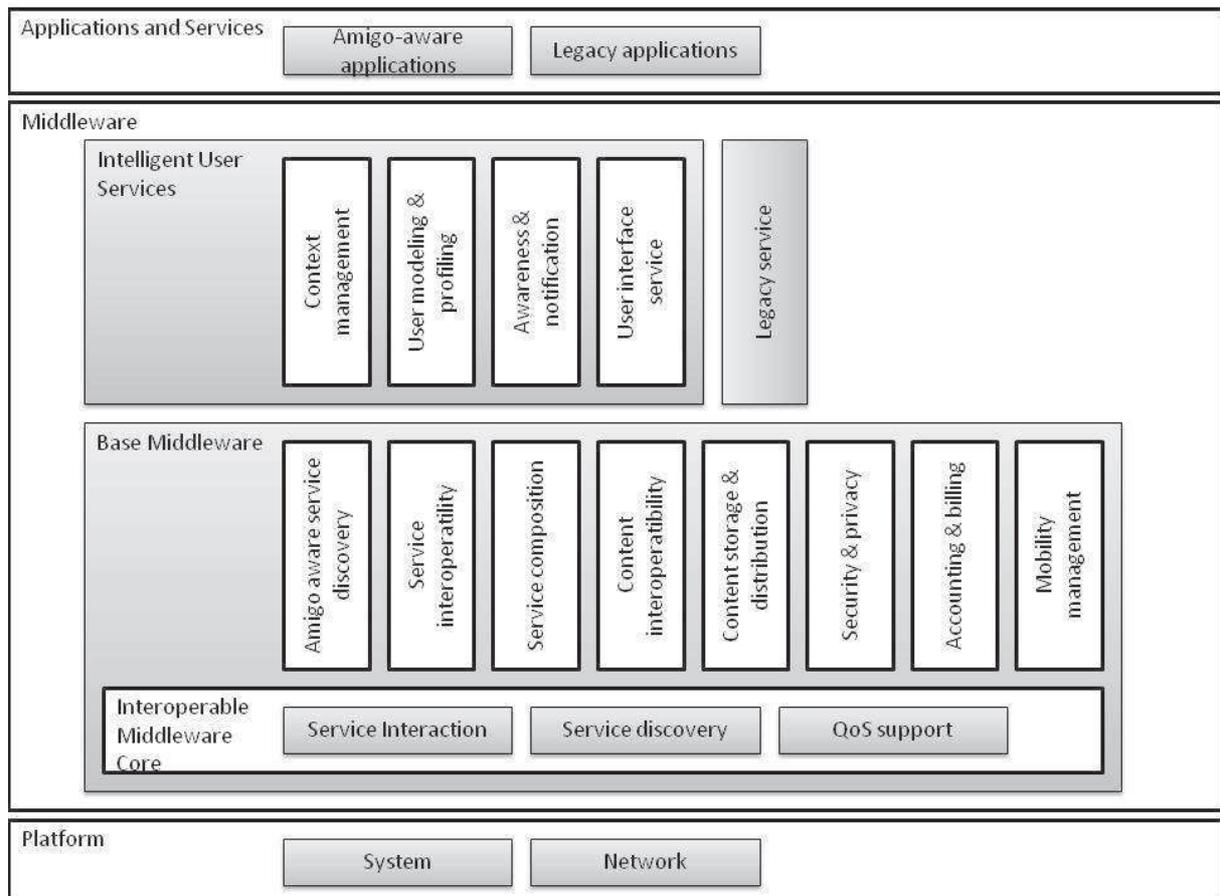


Abbildung 7: Amigo Middleware Architektur

Die Intelligent User Services der Amigo Middleware werden in die 4 Kategorien Kontextverwaltung, Modellieren von Benutzerprofilen, Erkennen und Benachrichtigen von Ereignissen und der Benutzerschnittstelle eingeteilt. Die Kontextverwaltung interpretiert die Sensordaten aufgrund historischer Daten und teilt die Sensordaten einer bestimmten vorliegenden Situation zu. Daraus kann auf zukünftiges Verhalten geschlossen werden. Das Modellieren von Benutzerprofilen wird mit Hilfe von Stereotypen durchgeführt. Dabei wird das Benutzerprofil zuerst statisch in einer grafischen Oberfläche modelliert, anschließend werden aus der Analyse des Feedbacks die Benutzervorzüge dynamisch zur Laufzeit modelliert. In dieser Einheit findet auch die Vereinigung der Benutzervorzüge und -wünsche von mehreren Benutzern statt, die gleichzeitig in einem Raum sind statt, um daraus eine Kompromisslösung abzuleiten. Die Erkennung und Benachrichtigung aufgrund vorliegender Ereignisse basieren auf einen regelbasierten Ansatz, indem Kontextinformation verarbeitet wird. Die Benutzerschnittstelle interpretiert 2 und 3 dimensionale Gesten, gesprochene Worte und erkennt die Sprecher. Grafische Ausgaben werden, je nach Benutzer und Anwendungskomponenten individuell angepasst, ausgegeben.

Auf die Amigo Middleware aufbauend können beliebige Amigo-Anwendungen und Anwendungen von Dritten bereitgestellt werden.

### 2.2.3.1.1 Verbindung mit dieser Arbeit

Die hier vorgestellte Softwarearchitektur abstrahiert die unterliegende hardwarenahe Kommunikationsarbeit. Amigo kümmert sich um die Verarbeitung von Sensordaten und

ordnet diese verschiedenen Kontexte zu, interpretiert aufgrund des Kontexts die Ereignisse und leitet zukünftiges Bewohnerverhalten ab.

In dieser Diplomarbeit wird hiergegen das Gebäudeverhalten, dynamisches Lernen von den Bewohnern und vom Gebäude und dessen Umgebung, in einer agentenbasierten Architektur behandelt.

### 2.2.3.2 A Multi-Agent System for Building Control

Diese Arbeit [15] beschreibt das Multiagentensystem MASBO<sup>19</sup>, einem Vorschlag für ein Multiagentensystem in dieser Kategorie, und einem frühen noch nicht funktionsfertigen Prototypen des Subprojekts CIMPS<sup>20</sup>. Das MASBO Projekt wird von der Universität Reading in England und von diversen Partnern aus der Wirtschaft vorangetrieben. Die Projektteilgebiete erschließen sich auf die Themen: Agentensysteme, künstliche Intelligenz, Kontrollnetzwerke und intelligente Gebäude.

MASBO hat als Ziele, Bewohnerpräferenzen zu lernen, Entscheidungen auf der Basis von Gebäudeinformationen zu treffen, drahtlose Sensornetzwerke zu verwenden und eventuell vorhandene oder neue Gebäudeautomationssysteme zu nutzen.

Aus den Anstrengungen des Projekts verspricht man sich intelligentere Gebäude, die autonom auf der Basis von Messdaten und nach gewissen Regeln die Heim- und Gebäudeautomationskomponenten steuern, um damit Energie zu sparen und den Bewohnern mehr Komfort zu bieten.

Beispielsweise könnte mit intelligenter Beleuchtungs- oder Heizungsregelung in Abhängigkeit der Anwesenheit der Bewohner deren Präferenzen individuell berücksichtigt werden. Damit MASBO auf die Präferenzen einer bestimmten Person in einer gegebenen Situation eingehen kann, benötigt das System Wissen darüber, welche Personen anwesend sind und welchen Tätigkeiten ein Bewohner gerade nachgeht und in Zukunft nachgehen könnte. Die Erarbeitung einer Lösung für diese komplexe Teilaufgabe wird nicht direkt im MASBO Projekt nachgegangen, sondern man stützt sich auf die Ergebnisse aus [16] und [17]. Das Agentensystem lernt die Benutzervorzüge, wie in Abbildung 8 dargestellt, aus Benutzerrückmeldungen.

---

<sup>19</sup> Multi-Agent System for Building cOntrol

<sup>20</sup> Coordinated Management of Intelligent Pervasive Space

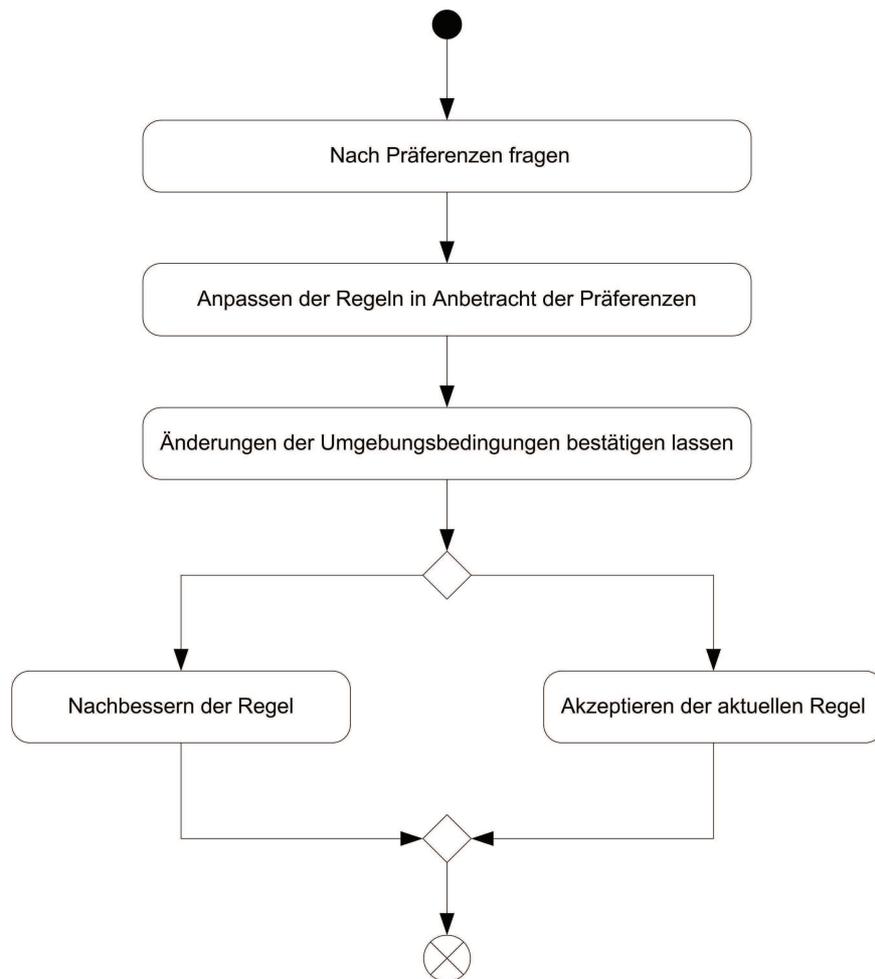


Abbildung 8: Lernen mit Benutzerrückmeldung

Die Optimierungsvorgänge zum Energiesparen ohne Komfortverlust stehen in einem starken Gegensatz, der in einer Konfliktlösungseinheit aufgelöst werden muss. Genauso wie das auch bei auftauchenden Konflikten der Bewohnervorzüge mehrerer anwesenden Bewohner geschehen muss, die gleichzeitig in einem Raum sind.

Prinzipiell hilft MASBO, den Bewohnern den Aufenthalt in einem Gebäude so angenehm wie möglich zu machen und Kommandoeingriffe auf Aktuatoren möglichst selbstständig aus den gelernten Erfahrungen vorzunehmen.

### 2.2.3.2.1 Architektur

Die Systemarchitektur von MASBO, wie sie in Abbildung 9 zu sehen ist, besteht aus 3 Schichten. Die unterste der 3 Schichten ist die Geräteschicht. Hier erfolgt die Kommunikation mit den Sensoren und Aktuatoren. In der vorgestellten Arbeit wird dafür BACnet, LON oder KNX/EIB angedacht. Die darüber liegende Schicht, die Kommunikationsschicht, interagiert zwischen der Geräte- und der darüber liegenden Software-Agentenschicht. Als Middleware wird BACnet/WS, OSGi oder oBIX verwendet. Das eigentliche intelligente Agentensystem von MASBO liegt in der Software-Agentenschicht.

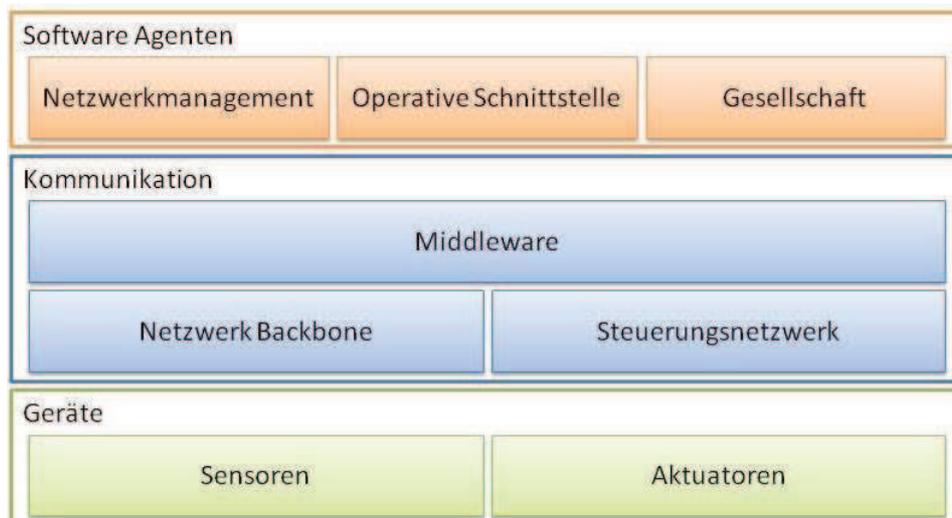


Abbildung 9: Systemarchitektur von MASBO

Ein Softwareagent agiert stellvertretend im Interesse eines anderen Programms oder eines Benutzers. Ein Agent besitzt nachfolgende Eigenschaften:

- **Reaktiv** Ein Agent erkennt den Kontext, in dem er arbeitet und reagiert dementsprechend.
- **Proaktiv** Ein Agent macht keine einfache Reaktion auf Umwelteinflüsse, sondern bietet vielmehr zielgerichtetes Verhalten und setzt aufgrund dessen in angemessener Weise Aktionen.
- **Soziale Fähigkeiten** Agenten können untereinander kommunizieren und koordinieren dadurch ihre gemeinsamen Aufgaben.
- **Fortwährend** Agenten handhaben mindestens einen Thread und planen ihre zukünftigen Aktivitäten.

Um Regeln für die Präferenzen der Bewohner festlegen zu können, interagiert das System mit den Benutzern. Dabei werden in Echtzeit, aufgrund von Sensordaten und Aktuatoreinstellungen, Entscheidungen getroffen. MASBO besteht aus mindestens einem Präsenzmelder, jeweils einem Temperatursensor für die Innen- und die Außentemperatur, einem Heizaktor, einem Helligkeitssensor, je einem Verschattungs- und einem Beleuchtungsaktor und einer Informationsquelle, die die Bewohner identifiziert.

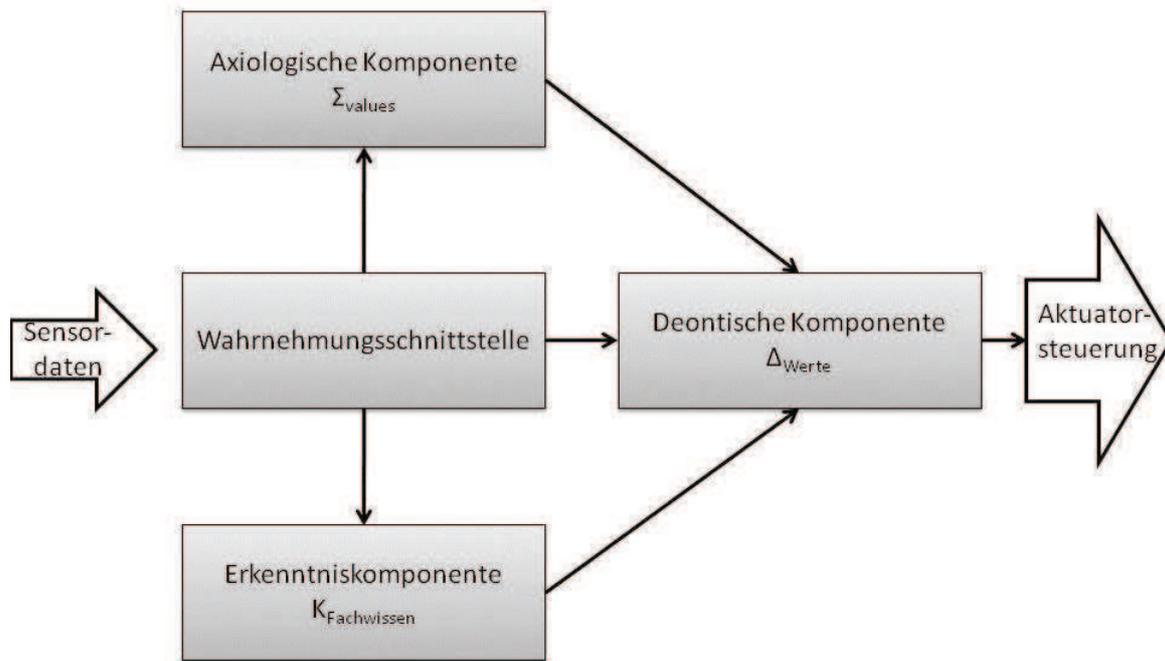


Abbildung 10: Verarbeitung von Sensordaten in MASBO

Das EDA-Modell [18] von MASBO ist in Abbildung 10 zu sehen. Die Wahrnehmungsschnittstelle nimmt alle Sensorwerte entgegen und filtert diese entsprechend der Wahrnehmungseigenschaften der Agenten und axiomatischen Normen. Die axiologische Komponente entscheidet, welche wahrgenommenen Zeichen beachtet werden sollen und was ausgeführt werden soll. Die Erkenntniskomponente enthält die Vorstellungen der Agenten in impliziter und expliziter Form, um daraus per Deduktion neue Schlussfolgerungen abzuleiten. Die deontische Komponente enthält Pläne in impliziter und expliziter Form. Jeweils ein Plan wird je nach Vorbedingungen ausgeführt. Aus den Plänen entstehen Stellgröße für die Aktuatoren.

MASBO besteht aus 3 verschiedenen Agententypen, wie diese in Abbildung 11 zu sehen sind.

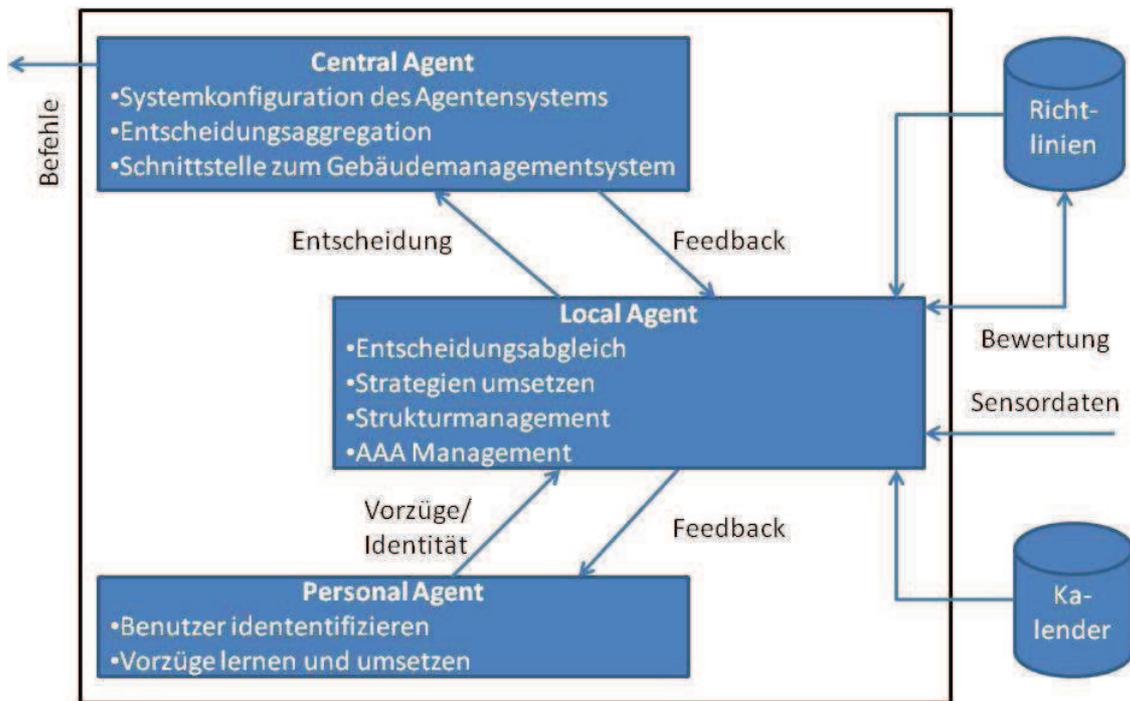


Abbildung 11: Funktionelle Architektur von MASBO

Die Personal Agents agieren für ihre Benutzer und kommunizieren mit anderen Agenten und Benutzern. Eine ihrer Aufgaben ist die Identifikation ihrer zugeordneten Benutzer auf Basis von gespeicherten Merkmalen. Sie lernen, nach welchen Merkmalen ihre zugeordneten Benutzer identifiziert werden können und deren Vorzüge für z.B. Temperatur oder Beleuchtung. Personal Agents verzeichnen, an welchem Ort sich ihr zugeordneter Bewohner zu welcher Zeit und für welche Zeitdauer aufhält. Sie beobachten, in welchem Aktivitätskontext welche Bewohnervorzüge zum Tragen kommen. Die Eingabedaten für Personal Agents kommen vom Local Agent. Die Ausgaben werden an die Benutzer und an die Umgebung über den Local Agent geleitet. Die Personal Agents können somit die Aktuatoren ihrer Umgebung nicht direkt steuern.

Der Local Agent ist der Mediator in dem Multiagentensystem. Er leitet Informationen an andere Agenten weiter und setzt Strategien um. Er löst Konflikte auf, wenn zum Beispiel 2 Personal Agents für ihre jeweiligen gemeinsam in einem Raum befindlichen Benutzer verschiedene Vorzüge für die Raumtemperatur an den Local Agent melden. In der Regel stehen Personal Agents mit dem Local Agent im Konflikt, da die gemeldeten Vorzüge der Bewohner nicht mit der Energiesparpolitik übereinstimmt. Zukünftige Ereignisse können durch Einsicht der Bewohnerkalender geplant werden. Die Verwaltung der Struktur des Agenten Clusters gehört ebenfalls zu den Kernaufgaben des Local Agent. Die Authentifizierung, Authorisierung und Buchhaltung über Personal Agents zu den jeweiligen Bewohnern erfolgt ebenfalls im Local Agent. Als Eingabedaten erhält der Local Agent Sensordaten und Aktuatorwerte aus der physikalischen Umgebung in der MASBO arbeitet. Diese werden in einer aufgearbeiteten Form an die Personal Agents weitergeleitet. Über die Vorzüge der jeweiligen Bewohner wird der Local Agent von den Personal Agents informiert. Der Local Agent sendet die Entscheidungen basierend auf einer Abstimmung zwischen den Vorzügen der Bewohner und den Richtlinien an den Central Agent und erhält von diesem

eine Rückmeldung über die Ergebnisse der durchgeführten Aktionen basierend auf den Entscheidungen des Local Agent.

Der Central Agent aggregiert Entscheidungen des Local Agent und dient als Schnittstelle zu internen und externen Diensten des Local Agent. Über den Central Agent können Agenten des Agentensystems von Operatoren gestartet, gestoppt, eingesetzt und gelöscht werden sowie die Clusterinformation des Local Agent modifiziert werden. Bevor Kommandos an das Gebäudemanagementsystem gesendet werden, müssen alle empfangenen Vorzüge aggregiert werden, um einen Steuerbefehl abzusetzen. Der Central Agent isoliert also die Agenten des Multiagentensystems vom Gebäudemanagementsystem. Als Eingaben erhält der Central Agent Entscheidungen vom Local Agent. Als Ausgabe an das Multiagentensystem wird eine Rückantwort über die gemachte Entscheidung an den Local Agent gesendet. Ebenso können vom Central Agent Befehle von Operatoren zur Systemkonfiguration entgegen genommen und in umgekehrter Weise Informationen über das Laufzeitsystem an die Operatoren übermittelt werden.

#### **2.2.3.2 Verbindung mit dieser Arbeit**

Das MAS in der vorgestellten Arbeit bietet einen Rahmen für Agenten, um dynamisch auf Umweltvorkommnisse in geeigneter Art und Weise zu reagieren. Dabei wird sowohl auf Bewohnerpräferenzen als auch auf Energiesparregeln Wert gelegt. MASBO modelliert ebenfalls, wie in dieser Arbeit erarbeitet, für Benutzer stellvertretende Agenten. Die Möglichkeit Merkmale und Vorzüge zu lernen, sind sowohl in der vorgestellten Arbeit wichtig als auch in dieser Diplomarbeit. Die Aufgaben werden in der vorgestellten Arbeit in sehr grobe Stücke aufgeteilt. In dem Agentenmodell dieser Diplomarbeit werden mehrere Agententypen vorgestellt, um die Aufgaben besser handhaben zu können.

### **2.3 Technische Implementierungen**

An dieser Stelle werden kommerziell verfügbare Produkte und wissenschaftliche Projekte präsentiert, die sich mit dem speziellen Thema Heimautomation befassen.

#### **2.3.1 Kommerziell verfügbare Produkte**

Bei dem ersten Produkt Siemens Synco Living, des größten hier vorgestellten Herstellers von Heimautomationsprodukten, handelt es sich um ein sehr ausgereiftes System, das über vordefinierte Szenarien verfügt, wie beispielsweise für die Beleuchtungssteuerung. Die Bewohner eines Hauses werden in erster Linie mit Komfortfunktionen über Zeitpläne unterstützt. Das System basiert auf dem offenen Heimautomationsprotokollstandard KNX/EIB.

Das zweite Heimautomationssystem My Personal Gekko, ein System der Südtiroler Firma Ekon, arbeitet in ähnlicher Weise mit vordefinierten zeitgesteuerten Szenarien wie Siemens Synco Living, jedoch mit den Vorteilen, dass es über ein Bedienpanel mit einer grafischen Touchoberfläche, über Internet und über Apple iPhone gesteuert werden kann. Das Heimautomationssystem kann zudem über Software-Plug-Ins, die von der Herstellerseite bezogen werden können, mit weiteren Funktionen erweitert werden. Außerdem ist es

möglich, nicht nur das eigene proprietäre Busprotokoll zu nutzen, sondern auch andere Busprotokolle der Branche, wie zum Beispiel KNX/EIB.

Das Heimautomationssystem der Schweizer Firma Adhoco steuert Aktuatoren nach Gebäude- und Bewohnerverhalten, das zur Laufzeit mit Hilfe genetischer Algorithmen dynamisch gelernt werden kann. Die Benutzerinteraktion erfolgt über das zentrale Steuerelement adhoco.H1, über Funkfernbedienungen oder über das Internet. Die Kommunikation der Geräte, das jüngste und innovativste der hier vorgestellten Produkt, basiert auf dem Funkstandard ZigBee. Das erlaubt den Einsatz in vorhandenen Gebäuden, wo die Verlegung von zusätzlichen Kommunikations- und Versorgungskabel schwer oder gar nicht möglich ist.

### 2.3.1.1 Siemens Synco Living

#### 2.3.1.1.1 Produktbeschreibung

Bei dieser Produktfamilie [10] handelt es sich um ein Heimautomationssystem das über lokal installierte Bedienpaneele, Smartphones oder einen PC gesteuert werden kann. Die letzteren beiden Endgeräteklassen können auch über das Internet auf die Funktionen von Siemens Synco Living zugreifen.

Das System steuert den Heizungskessel, der die Wärmeenergie für die Raumheizung und Brauchwasseraufbereitung liefert. Die Heizkörper oder Fußbodenheizungen können in bis zu 12 Einzelräumen über einen fix vorgegebenen Zeitschaltplan geregelt werden. Mit Hilfe der festgelegten Zeitpläne, einer Regeltoleranz von  $\pm 0,2^\circ\text{C}$  Raumtemperatur und Erkennen von geöffneten Fenstern, wo die Raumheizung deaktiviert wird, kann Heizenergie eingespart und somit die Energiekosten minimiert werden. Im Vergleich zu Thermostatventilen, die die Raumtemperatur mit einer typischen Regeltoleranz von  $\pm 1^\circ\text{C}$  regeln, erfolgt somit nach Herstellerangaben mit Siemens Synco Living eine Kosteneinsparung von bis zu 30% bei den Wärmeenergiekosten.

An heißen Tagen im Jahr können die Räume unter Einhaltung des Belegungsplans gekühlt werden. Um das Haus auch richtig mit Frischluft versorgen zu können, wird der Luftvolumenstrom je nach Raumbelagung und  $\text{CO}_2$  Gehalt in der Raumluft geregelt, ohne Verlust von Kühl- oder Heizenergie.

Desweiteren können Jalousien und elektrische Verbraucher gesteuert werden. Die Jalousien können vollkommen automatisch in Abhängigkeit der Tageszeit hoch- und niedergefahren werden. Die elektrischen Verbraucher, wie zum Beispiel Haushaltsgeräte<sup>21</sup>, können mit binären Funkzwischensteckern ein- und ausgeschaltet werden. Für Lampen gibt es einen Funkzwischenstecker, mit dem die Leuchtmittel<sup>22</sup> gedimmt werden können.

Für die Sicherheit im Eigenheim kommen Rauchmelder sowie Fenster- und Türkontakte zum Einsatz. Die Rauchmelder lösen im Brandfall einen lauten Signalton aus. Zusätzlich werden Alarmmeldungen an vordefinierte Empfänger per SMS, Pager oder Email gesendet. Die Fenster- und Türkontakte dienen zur Einbrechererkennung, wobei die Fensterkontakte auch zur Raumtemperaturregelung genutzt werden, um bei geöffneten Fenstern die Raumheizung oder -kühlung zu deaktivieren.

---

<sup>21</sup> Zum Beispiel Kaffeemaschinen.

<sup>22</sup> Dimmbare Leuchtmittel sind Voraussetzung.

Siemens Synco Living arbeitet mit Szenarien und Zeitplänen. Unter Zeitplänen versteht man die vorgegebenen Aktionen, die das Heimautomationssystem mit Hilfe von Aktuatoren in einem Raum zu einem bestimmten fest vorgegebenen Zeitpunkt ausführt. Unter Szenarien wird in diesem Kontext die gleichzeitige Steuerung von mehreren Aktuatoren verstanden. Ist das Szenario „Fernsehen“ für den Raum Wohnzimmer definiert worden, dann werden zum Beispiel die Jalousien heruntergefahren, die Raumbelichtung gedimmt und der Fernseher eingeschaltet. Ein Szenario wird manuell von einem Benutzer ausgeführt und unterbricht die durch den Zeitplan vorgegebene Aktion.

Gesteuert wird Siemens Synco Living über Wohnungszentralen, Raumzentralen, Taster, Funkfernbedienungen oder einem Webservermodul. Die Gruppe der Sensoren besteht aus Raumtemperaturfühler, Außentemperaturfühler, Helligkeitssensoren, Regensensoren, Luftdruckfühler, Rauchmelder sowie Tür- und Fensterkontakte. Als Aktuatoren kommen Heizkörperregelantriebe, Heizkreisregler, Multikontroller, schalt- oder dimmbare Funksteckdosen und Jalousieaktuatoren zum Einsatz.

Siemens Synco Living eignet sich nicht nur zur Installation in Neubauten, sondern kann auch in bestehenden Objekten nachgerüstet werden. Technisch basiert Siemens Synco Living auf KNX [19].

#### 2.3.1.1.2 Verbindung zu dieser Arbeit

Dem Konzept des Produkts Siemens Synco Living zur allumfassenden Steuerung von Heim- und Gebäudeautomationstechnik wird in dieser Arbeit in ähnlicher Weise nachgegangen. Siemens Synco Living arbeitet mit zuvor festgelegten zeitgesteuerten Ereignisplänen. Die einzelnen Ereignisse beruhen auf vordefinierte zusammenschaltete Steuerung von verschiedenen Aktuatoren. Zusammengesetzte Ereignisse werden in ein Szenario zusammengefasst. Das in dieser Arbeit vorgestellte Heim- und Gebäudeautomationssystem lernt dynamisch die Aktivitäten der Benutzer und steuert somit (fast) ohne Steuereingriffe durch die Benutzer die Aktuatoren im Gebäude im Kontext von Zeit, Raum- und Umgebungsmesswerten sowie Vorhersagedaten externer Informationsquellen. Bei Siemens Synco Living können technisch versierte Benutzer das Heim- und Gebäudeautomationssystem programmieren, jedoch nur statisch. Ebenfalls ist die Programmierung für Benutzer ohne PC Kenntnisse ein Hindernis, wohingehend in dieser Diplomarbeit automatisches Lernen von Benutzerverhalten im Vordergrund steht und somit die Benutzerfreundlichkeit und zugleich die Akzeptanz des Systems beim Benutzer gesteigert wird. Genauso wie bei Siemens Synco Living wird bei dem in dieser Diplomarbeit vorgestellten Heim- und Gebäudeautomationssystem versucht so sparsam wie nur möglich mit dem Energieverbrauch umzugehen.

#### 2.3.1.2 myPersonalGEKKO

Das Heimautomationssystem myPersonalGEKKO<sup>23</sup> [11], von der Firma Ekon GmbH in Bruneck in Südtirol in Italien, steuert und regelt die klassischen Gewerke Licht, Heizung,

---

<sup>23</sup> Es gibt auch noch die 2 Produktlinien myBusinessGEKKO und myNetworkGEKKO. myBusinessGEKKO beinhaltet im Prinzip die Funktionen von myBusinessGEKKO und ist zusätzlich um die Funktionen Zeiterfassung und Zutrittskontrolle erweitert. Bei myNetworkGEKKO handelt es sich um ein System für Verbrauchsdatenerfassung und Fakturierung, Lastmanagement, Netzüberwachung, Service- und Dienstleistungsgewerbe, wie zum Beispiel Fernwärmekraftwerke, Internet Service Provider und Telefonanbieter.

Warmwasseraufbereitung, Lüftung, Kühlung und Verschattung. Zusätzlich können elektrische Geräte wie zum Beispiel Wäschetrockner, Waschmaschine, Fernseher, Audioanlagen über schaltbare Steckdosen gesteuert werden.

Das System operiert in den 4 Modi: Hand-, Ferien-, Automatikbetrieb und komplett ausgeschaltet. Im Handbetrieb kann jeder Aktuator mit einem geeigneten Bedienelement gesteuert werden. Im Ferienbetrieb werden alle elektrischen Verbraucher, soweit möglich, deaktiviert und die Heizung, Lüftung und Kühlung auf Maß reduziert, um ein Minimum an Energie zu verbrauchen. In der häufigsten Betriebsart Automatikbetrieb wird mit Zeitplänen gearbeitet. In den Zeitplänen wird definiert, wann welches Gerät mit welcher Stellwertvorgabe angesteuert wird. Es ist zusätzlich auch möglich, auf Ereignisse, wie etwa sehr starker Wind, zu reagieren. Die meisten Steuer- und Regelungsaufgaben sind von komplexer Natur, wie zum Beispiel bei starkem Regen die geöffneten Fenster zu schließen und ausgefahrene Markisen einzufahren. Für komplexe Steueraufgaben können bis zu 16 Szenarien definiert werden, die nicht nur auf Fensteröffner und Markisenantriebe beschränkt sind, sondern alle möglichen steuerbaren Geräte umfassen.

Die Steuerung und Regelung der Heizung, Lüftung und Klimatechnik, im weiteren Textverlauf HVAC-Anlage genannt, wird mit einer Wetterwarte ergänzt, deren Messwerte für die Stellgröße der HVAC-Anlage benützt wird. Die Wetterstation misst dabei die Außentemperatur, die Luftfeuchtigkeit, die Windgeschwindigkeit und die Sonneneinstrahlung.

Neben der Überwachung der Umgebungsparameter mit Hilfe der Wetterstation werden alle Räume und Betriebsanlagen auf Betriebssicherheit überwacht. Das System myPersonalGEKKO meldet zum Beispiel Rohrbrüche, Leistungsüberbeanspruchung von Stromleitungen oder einen leeren Brennmaterialbunker für die Zentralheizung. Über Tür- und Fensterkontakte sowie Bewegungsmelder implementiert myPersonalGEKKO die Funktionen einer privaten Alarmanlage, die die Alarmmeldungen per SMS an vordefinierte Empfänger, wie Feuerwehr, Polizei, den Bewohnern oder auch Nachbarn schicken kann.

Ebenfalls erfasst myPersonalGEKKO die Verbrauchsdaten für Strom, Wasser und Brennmaterial, die später für Energieoptimierungsmaßnahmen durch die Bewohner des Hauses verwendet werden können.

Mit myPersonalGEKKO wird Videoüberwachung, etwa bei Türöffnern, und eine Einbindung von entfernten Videostreams angeboten. Die Multimediasteuerung von myPersonalGEKKO umfasst auch Audioquellen, wie MP3-Player, Webradio, CD-Player und Apple iPod, die bequem über vordefinierte Szenarien als Weckmusik verwendet und sogar im ganzen Haus über alle Audioausgabegeräte gleichzeitig abgespielt werden können.

Da myPersonalGEKKO den Anspruch eines allumfassenden Heimautomatisierungssystems stellt, haben die Entwickler auch an die Grünflächen im Garten gedacht und eine individuelle Bewässerungssteuerung integriert.

Der Benutzerzugriff auf das System erfolgt über Wandtaster, an der Wand eingebaute Multifunktionsdisplays mit Touchscreensteuerung, vordefinierte SMS Nachrichten, die man an myPersonalGEKKO von einer vordefinierten Handynummer schicken kann oder über eine proprietäre Remotecontrol-Software, die auf Microsoft Windows, Linux-Betriebssystem, Apple Mac OS X und Apple iPhone installiert werden kann. Mit der zuletzt genannten Kontrollvariante lässt sich auf alle für die Bewohner eines Hauses wichtigen Funktionen

zugreifen. Damit ist es aus der Ferne möglich, alle Geräte abzuschalten oder die Anwesenheitssimulation zu aktivieren.

Das zentrale Element von myPersonalGEKKO ist das Bedienpanel myGEKKO Touch. Mit diesem lassen sich alle Steuerungs- und Regelfunktionen bedienen. In das Touchpanel ist ein Lautsprecher und ein Mikrofon für eine Gegensprechanlage integriert sowie ein RFID Lesesystem, eine LAN- und eine USB-Schnittstelle. Über die LAN-Schnittstelle lässt sich myPersonalGEKKO mit dem lokalen Netzwerk und mit dem Internet verbinden.

Zusätzliche Softwarefunktionen können über Softwaredownloads vom Hersteller nachgerüstet werden. Wird das System erweitert, dann sind dank der Plug&Play-Technik für den Anwender keine zusätzlichen manuellen Systemanpassungsschritte nötig.

Das verfügbare Hardwareangebot des Heimautomationssystems myPersonalGEKKO durch den Hersteller lässt sich mit Schnittstellenkomponenten der Firma Wago erweitern. Mit diesen Schnittstellen zu Fremdsystemen ist es möglich Geräte, die über DALI, KNX/EIB, TELENOT, SCHRACK und EnOcean angebunden sind, zu steuern und regeln.

#### **2.3.1.2.1 Verbindung zu dieser Arbeit**

Das Konzept von myPersonalGEKKO, ein allumfassendes Heimautomationssystem zum Steuern und Regeln aller im Haus verbauten Geräte aller Gewerke zu sein, passt sehr gut zu dem Themenschwerpunkt dieser Arbeit. Zielgruppen sind sowohl jüngere, als auch pflegebedürftige Menschen. Es können dynamisch eigene Gerätekomponenten und Komponenten anderer Systeme mit Schnittstellenkomponenten mit myPersonalGEKKO gesteuert oder geregelt werden. Die Programmierung der Szenarien erfolgt über eine übersichtlich gestaltet Bedienoberfläche. Das Heimautomationssystem besitzt allerdings nicht die Möglichkeit die Gewohnheiten der Bewohner sowie die komplexen Eigenschaften des Gebäudes und der Umgebung für die automatische Regelung und Steuerung von Aktuatoren zu lernen, die Thema dieser Diplomarbeit sind.

#### **2.3.1.3 adhoc**

Das Heimautomationssystem adhoc [12] der gleichnamigen Schweizer Firma ermöglicht intelligentes Wohnen. Das System besteht typischerweise aus den folgenden Komponenten:

- einem selbstlernenden und selbstkonfigurierenden zentralen Steuerungsmodul
- mehrere Präsenz- und Bewegungsmeldern mit integrierten Beleuchtungssensoren
- einem Wetterstationsmodul, das autark und ohne Verkabelung Außentemperatur, Luftfeuchtigkeit, Windverhältnisse und Sonneneinstrahlung misst
- binären Schaltaktuatoren
- Dimmaktuatoren für Licht- und Ventilatorenregelung
- Steuergeräten für Verschattungsaktuatoren
- funkgesteuerten Heizkörperventile
- Funkfernbedienungen

Die Geräte von adhoc basieren auf dem ZigBee [20] Standard und sind somit drahtlos auf der Frequenz von 2.4 GHz über maximal 75 Meter miteinander verbunden. Durch Routing der Nachrichten über andere ZigBee Knoten lassen sich in einem ZigBee Funknetzwerk größere Strecken über weitaus weitere Nachrichtenübertragungstrecken überwinden, als es

die maximale Funkdistanz von nur einem Knoten zum Nächsten ermöglicht. Die Feldgeräte, wie zum Beispiel Präsenz- und Bewegungsmelder, lassen sich in existierende Gebäude leicht und in der Regel ohne Stemmarbeiten verbauen. Sensoren beziehen ihre Energie von Batterien oder Solarzellen. Das zentrale Steuerungsmodul bezieht seine Energie aus PoE<sup>24</sup>.

Die von der Firma Adhoco angebotenen ZigBee Feldgeräte lassen sich ohne Zentrale wie normale Sensoren und Aktuatoren verwenden, doch das wahre Herz des Heimautomationssystems der Firma Adhoco ist das mit einem Farbdisplay versehene zentrale Steuerungsmodul adhoco.h1.

Dieses ermöglicht die Automatisierung von Beleuchtung, Verschattung, Heizung und Lüftung in energieoptimierter Weise. Im Bereich Sicherheit deckt das zentrale Steuermodul die Aufgaben Anwesenheitssimulation, Einbruchserkennung über Bewegungsmelder, Überwachung elektrischer Großverbraucher im Haus und Unfallerkennung ab. Auch Monitoringaufgaben, wie Energieverbrauch Aufzeichnen und die grafische Aufbereitung gehören zu den Aufgaben des zentralen Steuermoduls. Auf die Funktionen des zentralen Steuerungsmoduls adhoco.H1 lässt sich über eine Funkfernbedienung und über Netzwerk, über die Adhoco-Webseite, mit einem beliebigen Browser zugreifen. Bei letzterer Zugriffsmöglichkeit baut das adhoco.H1 Steuerungsmodul einen gesicherten VPN-Tunnel zu einem Server des Herstellers auf. Mit diesem Verfahren können im Vergleich zu einer direkten Anbindung der zentralen Steuereinheit an das Internet Einbruchsversuche auf das adhoco.H1 Modul über das Internet minimiert werden.

Die automatischen Steueraufgaben lernt adhoco.H1 mit Hilfe genetischer Algorithmen vom Verhalten des Gebäudes und dessen Bewohner. Ein Bewohner kann das automatische Verhalten der Aktuatoren, die ihre automatischen Steuer- und Regelbefehle vom zentralen Steuerungsmodul erhalten, mit Hilfe von Funkfernbedienungen übersteuern. Die gemachten Änderungen erfasst das zentrale Steuerungsmodul adhoco.H1 und lernt dadurch das Verhalten der Bewohner.

### 2.3.1.3.1 Verbindung zu dieser Arbeit

Die Ansätze des Heimautomationssystems der Firma Adhoco sind im Bereich selbstständiges Lernen des Gebäude- und Benutzerverhaltens für kommerziell erhältliche Systeme dieses Anwendungsbereichs bahnbrechend. Das System ist momentan das einzige kommerziell erhältliche System, das selbstständig vom Gebäude- und von den Bewohnern lernt, ohne vor der Benutzung kompliziert Steuerungs- und Regelungskurven sowie Anwendungsszenarien zu konfigurieren. Dies wird mit genetischen Algorithmen gerechnet, die von der Seite der Firma Adhoco unter Verschluss gehalten werden. Genau dieser Ansatz, von Benutzern zu lernen, wird auch in dieser Arbeit verfolgt. Das Heimautomationssystem von Adhoco verfolgt diesen Ansatz in einer sehr vereinfachten Weise. Das System unterscheidet zum Beispiel nicht zwischen einem und mehreren Bewohnern und erlaubt nur eine Erkennung der aktuell vorgefallenen Ereignisse. Außerdem baut das Heimautomationssystem nur auf eigene oder vom Hersteller empfohlene ZigBee Komponenten auf. Besser wäre es, wenn das zentrale Steuerungsmodul adhoco.H1 über ein generisches Interface verfügen würde, mit dem auch eine Anbindung etwa zu KNX/EIB oder LON Feldgeräten möglich wäre.

---

<sup>24</sup> Power over Ethernet, kurz PoE

Das Heimautomationssystem der Firma Adhoco wurde zum ersten Mal 2008 in einem Pilotprojekt mit älteren Menschen getestet. Die Ergebnisse waren ernüchternd [21]. Das lag zum einen an der noch unausgereiften jungen Technik, zum Anderen an der mangelnden Schulung der Benutzer und der Auswahl einer repräsentativen Testgruppe.

### 2.3.2 Feldstudien

Die meisten Feldstudien, die man zum Thema Smart Home findet, sind im Bereich Multimediaanwendungen. Nur einige wenige der gefundenen Feldversuche beschäftigen sich mit der ganzheitlichen Regelung und Steuerung der Gebäudeautomationskomponenten. An dieser Stelle werden 3 verschiedene Projekte vorgestellt. Bei den Betreibern der Projekte handelt sich um Firmen sowie private und öffentliche Forschungseinrichtungen.

#### 2.3.2.1 Fraunhofer inHaus-Zentrum Duisburg

Das Fraunhofer inHaus-Zentrum in Duisburg wird von 7 Fraunhoferinstituten und etwa 100 Partnern aus der Wirtschaft betrieben. Das inHaus-Zentrum, das nicht nur technische Entwicklungen forciert, sondern auch Markt- und Akzeptanzforschung betreibt, besteht aus 2 Gebäuden: inHaus1<sup>25</sup> als Testumgebung für Anwendungen in Wohnimmobilien und inHaus2<sup>26</sup> als entsprechendes Pendant für Nutzzimmobilien.

##### 2.3.2.1.1 inHaus1

inHaus1 wurde Anfang April 2001 eröffnet und verfolgt drei primäre Ziele. Das erste Ziel ist den Energieverbrauch zu senken und damit Kosten bei Brennstoffen zu sparen sowie die Umwelt zu schonen. Als zweites Ziel will man den Komfort für Haustechnik und Multimediaeinrichtungen steigern. Das wohl am schwierigsten umzusetzende Ziel ist die Betreuung von älteren Personen oder Personen mit einer Behinderung in ihrer eigenen Wohnumgebung. inHaus1 besteht aus 2 getrennten Gebäudeteilen, einem Teil mit einem typischen Einfamilienhaus-Charakter, indem zeitweise für einen Zeitraum von ein bis drei Monaten Probanden wohnen und einen Teil, indem Techniker arbeiten.

Das Haus hat die Bereiche Wohnen, Schlafbereich, Bad und Badlabor, Home-Office, Servicefahrzeug sowie Außenbereich und Garten. Die Anlage in diesem Haus beherbergt eine Technikzentrale für die Erfassung von Gas-, Wasser- und Stromverbrauch. Die Temperatur und die Beleuchtung wird über busvernetzte Komponenten geregelt und gesteuert.

Der bewohnbare Teil von inHaus1 hat folgende Räumlichkeiten: Bad, Küche, Toilette, Schlafzimmer, Wohnzimmer, Kinderzimmer, Gartenterrasse und eine Garage. Hier werden verschiedene neue Technik- und Anwendungslösungen mit Testpersonen oder Familien in Workshops oder Testwohnphasen evaluiert. Die Testwohnphase wird von Experten aus den Bereichen Soziologie, Psychologie und Marktforschung begleitet.

---

<sup>25</sup> Der Fokus in dieser Arbeit wird auf das inHaus1 gelegt, da in diesem Technikinnovationen im Heimbereich behandelt werden.

<sup>26</sup> Bei diesem Versuchshaus handelt es sich um größtenteils um Demonstrationsräume für den Zweckbau.

Im Schlafbereich werden telemedizinische Studien erprobt. Zum Beispiel werden Vitalwerte vom Gebäudeautomationssystem überwacht. Bei kritischen Werten wird ein Arzt oder eine Pflegekraft alarmiert. Es ist auch denkbar, dass die aufgezeichneten medizinischen Parameter von einem Arzt in periodischen Abständen aus dem Heimautomationssystem über einen gesicherten Zugang heruntergeladen werden können, um Diagnosen zu validieren und Therapiepläne an neu gemachte medizinische Erkenntnisse anzupassen. Dadurch verringern sich teure Kosten für Visiten und ein Arzt kann in der Zukunft mehr Patienten betreuen, als das heute möglich ist.

Das Bad im inHaus1, in dem gehobener Komfort eine wesentliche Rolle spielt, lüftet sich ganz automatisch nach gegebener Luftfeuchtigkeit und -qualität. Diese Maßnahme reduziert den Heizenergiebedarf und Schimmelbildung. Außerdem wird der rationelle Umgang mit Wasser mit Hilfe einer vernetzten intelligenten Wasserinstallation forciert, mit der gleichzeitig ältere Menschen und Menschen mit spezifischen Bedürfnissen selbsttätig tägliche Körperpflege durchführen können.

Der Bürobereich beherbergt das lokale Management der Heimautomationsanlage, die inHaus Bibliothek und ein Datenträgerlager. In der Technikwerkstatt findet man alle Komponenten für Smart Metering, wie Zähler jeder Art und Sicherungen. Die Servicewerkstatt dient als Entwicklungsmöglichkeit für Hard- und Softwarekomponenten, die dann später im inHaus erprobt werden.

Im Außenbereich und Garten findet man Glasbruchsensoren, Bewegungsmelder und Kameras, die einen Einbrecher erkennen können und im Notfall Alarm mit geeigneten Maßnahmen Alarm schlagen. Außerdem findet man im Außenbereich Warnsysteme, die dabei helfen Lecks in Wasser- oder Rohrleitungen zu finden. Auf dem Dach ist eine Wetterstation und für die Warmwasseraufbereitung eine thermische Solarenergieanlage installiert. Der Garten wird von einem Rasenmäroboter, einem Bewässerungssystem und verteilten Feuchtigkeitssensoren gepflegt.

Zu einem Haus gehört natürlich auch ein Auto. Das Auto ist mit einem Bord-PC, einem Media-Center und einem intelligenten Kofferraum mit einem RFID Scanner ausgestattet. Damit lassen sich von unterwegs Hausparameter abrufen und auch Teile der Gebäudetechnik steuern. Zusätzlich lassen sich auch bestimmte Parameter wie Treibstoffstand und Reifendruck vom Haus aus ermitteln.

Als Automationssoftware kommt die Software Middleware Amigo<sup>27</sup> zum Einsatz.

#### 2.3.2.1.2 inHaus2

Das zweite, etwas neuere inHaus2, das seit November 2008 zur Verfügung steht, verfolgt Ziele die im Zweckbau beheimatet sind. Es gibt vier Labore:

- **nextHotel&Event-Lab**, für Versuche und Entwicklungen zum Thema Hotel- und Veranstaltungstechnik
- **nextHealth&Care-Lab**, für Krankenhaus- und Pflegeheimanwendungen
- **nextOffice-Lab**, als Testumgebung im Büro- und Servicebereich
- **nextFM-Lab**, für Gebäudemanagementanwendungen.

---

<sup>27</sup> Mehr zu dem Thema findet man im Kapitel 2.2.3.1.

### 2.3.2.1.3 Verbindung zu dieser Arbeit

Ähnliche Ziele wie diese Arbeit werden im inHaus1 verfolgt. Der Schwerpunkt der Forschungs- und Entwicklungsarbeit liegt bei Produkten, die sich kurzfristig oder spätestens mittelfristig auf den Markt bringen lassen. Daher entstehen sehr viele innovative Insellösungen. Eine ganzheitliche Gebäudetechniksteuerung soll mit Hilfe der Amigo-Middleware umgesetzt werden, doch Publikationen zu diesem Thema sind rar.

Die Idee des inHaus1 den Komfort zu erhöhen, Energie zu sparen und die Pflege von alten Menschen oder Menschen mit einer Behinderung zu ermöglichen, findet man auch in dieser Arbeit wieder. Jedoch wird im Vergleich zum inHaus1 in dieser Arbeit der Fokus auf ganzheitliche, intelligente und selbstlernende Gebäudeautomation gelegt.

### 2.3.2.2 Living Tomorrow in Vilvoorde

Die Ausstellung Living Tomorrow<sup>28</sup> [22], die im Jahr 2003 ihre Pforten öffnete, ist in einem eigens für den Zweck gebauten Haus in Vilvoorde in einem Vorort von Brüssel untergebracht, in dem neue innovative Technologien ausgestellt sind. Das Konzept von Living Tomorrow ist eine Gesamtlösung für Technik, die das private Leben im Heimumfeld in naher Zukunft erahnen lässt.

Die Themenschwerpunkte sind Architektur, neue Materialien für Möbel, Küche, Wohnzimmer und weitere Wohnräume, Automationstechnik für Heizung, Lüftung, Kühlung, Beleuchtung, Verschattung, Alarmanlagentechnik, Zutrittskontrolle für Bewohner und Lieferdienste, Wohnen im Alter, (Kranken-) Pflege im Eigenheim, Multimediaanwendungen für Kinder und Erwachsene, allgegenwärtige Computerunterstützung und automatisierte Gartenpflege. Zusätzlich zu den Kernschwerpunkten sind mit Wohnen assoziierte Themen ausgestellt, wie zum Beispiel das Auto der Zukunft, das Büro zu Hause, die Bank und der Supermarkt von Morgen.

Die Ausstellung wird von einem Konsortium bestehend aus Architekten, Gebäudetechnikern, Multimedia- und Kommunikationsproduktherstellern, Autoherstellern, Banken und wissenschaftlichen Beratern, um nur einige zu nennen, unterstützt.

Das Haus von Living Tomorrow ist eine Testumgebung für Geräte und Techniken der teilnehmenden Hersteller, in dem innovative Produkte, die in naher Zukunft auf den Markt kommen werden, für Besucher ausgestellt sind.

#### 2.3.2.2.1 Verbindung zu dieser Arbeit

Bei dieser Ausstellung wird weniger auf technische Feinheiten Wert gelegt, dafür aber viel mehr auf die Ideen, wie man sich zum Beispiel eine Wohnung für ältere Menschen vorstellt. Dabei sind auch Themen wie beispielsweise Telemedizin<sup>29</sup>, Sturzkontrolle, Bedienen von Badearmaturen hervorgehoben. Die Kenntnis von normalen Tätigkeiten und die Präferenzen einer Person sind für eine Hausautomatisierungsanlage zum Erkennen von Notfällen von großer Bedeutung. Bei den Zielpersonen für eine betreute Wohnung handelt es sich um ältere Menschen, die oft mit neuer Technik überfordert sind. Daher muss die Technik im Hintergrund ablaufen und die Heimautomationsanlage den Tagesablauf einer bestimmten

---

<sup>28</sup> Vorgängerausstellungen in Amsterdam, Niederlande: House of the Future (1995-2000), House and Offices of the Future (2000-2005)

<sup>29</sup> Messen der Vitalwerte, des Blutzuckerspiegels und weiterer wichtige medizinische Parameter.

Person und deren Wunschbedingungen automatisch lernen. Das Konzept des maschinellen Lernens findet man auch in dieser Arbeit wieder<sup>30</sup>.

### 2.3.2.3 The Adaptive House

Das Praxisprojekt eines lernfähigen Hauses [5] [23] in Marshall in Colorado wird von mehreren Instituten der Universität Colorado betrieben. Die Größe und Struktur des Hauses ist an ein typisches Einfamilienhaus angelehnt. Es besteht aus einem Vorraum, einer Garage, 3 Schlafzimmern, 3 Bädern, einer Küche, einem Wohn- und Esszimmer und einem Technikräumen. Das Haus ist mit 75 Sensoren ausgestattet. Die Sensoren können Helligkeit, Temperatur, Schallpegel, Bewegung und den Status, ob eine Tür- oder Fenster geöffnet oder geschlossen ist, messen. Es kann die Raumtemperatur über elektrische Heizkörper oder Zentralheizung, Boilerladung, Decken- und die Wandlampen sowie Deckenventilatoren geregelt werden. Ziel dieser Feldstudie ist, eine Hausautomationsanlage zu schaffen, die sich an die Bedürfnisse und Wünsche der Bewohner anpasst. Dabei achtet das Haus auf Umgebungszustände und auf die Handlungen der Bewohner und lernt die Vorzüge und Zeitpläne, wann welche Aktionen erfolgen und zu welcher Zeit ein Raum belegt ist. Ziel ist, dass das Haus die Wünsche der Bewohner vorhersagen kann und Energie spart.

Der Regelalgorithmus arbeitet mit einer Kostenfunktion. Die Kosten teilen sich in Kosten bei Vernachlässigung der Wünsche der Bewohner und in Energiekosten auf. Formel 2 wird für die optimale Steuerungs- und Regelung unter der Berücksichtigung der vorhin genannten Bedingungen verwendet.

$$J(t_0) = E \left[ \lim_{\kappa \rightarrow \infty} \frac{1}{\kappa} \sum_{t=t_0+1}^{t_0+\kappa} d(x_t) + e(u_t) \right]$$

Formel 2: Methode zur optimalen Steuerung

Die Formelzeichen in Formel 2 bedeuten

t ..... Index über nichtüberlappende Zeitintervalle

t<sub>0</sub> ..... aktuelles Zeitintervall

u<sub>t</sub> ..... Regelungsentscheidung im Intervall t

x<sub>t</sub> ..... Umgebungszustand im Intervall t

d(x<sub>t</sub>) ..... Jene Kosten, die auftreten, sobald sich ein Bewohner in einem Raum nicht wohl ..... fühlt

e(u<sub>t</sub>) ..... Energiekosten, je nach Gerät und Einstellung

Das Regelungssystem mit dem Namen Adaptive Control of Home Environments (ACHE) betrachtet jede Aufgabe<sup>31</sup>, die geregelt oder gesteuert werden soll, separat. Im Feldversuch wurden die Bereiche Lufttemperaturregelung und Beleuchtungssteuerung evaluiert. Zum Regeln der Lufttemperatur kann ACHE auf die Zentralheizung, Raumheizung und

<sup>30</sup> Siehe Kapitel 6.

<sup>31</sup> Raumtemperaturregelung, Beleuchtungssteuerung und Warmwasserbereitung

Ventilatoren zugreifen<sup>32</sup>. Wandleuchten und Deckenbeleuchtung wurden für die Beleuchtungssteuerung verwendet.

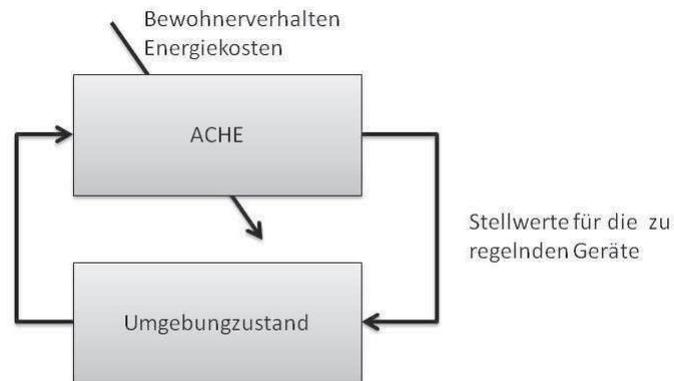


Abbildung 12: ACHE Regelschleife

Der Regelungskreis von ACHE sieht, wie in Abbildung 12 gezeigt, aus. ACHE entspricht dem Regler, der aufgrund von Bedingungen, wie das aktuelle Verhalten der Bewohner und Energiekosten, die Gerätestellwerte vorgibt und damit den Umgebungszustand beeinflusst. Feedbacks über die aktuell vorliegenden Umgebungszustände fließen in die Regelung mit ein. Damit ist der Regelkreis geschlossen.

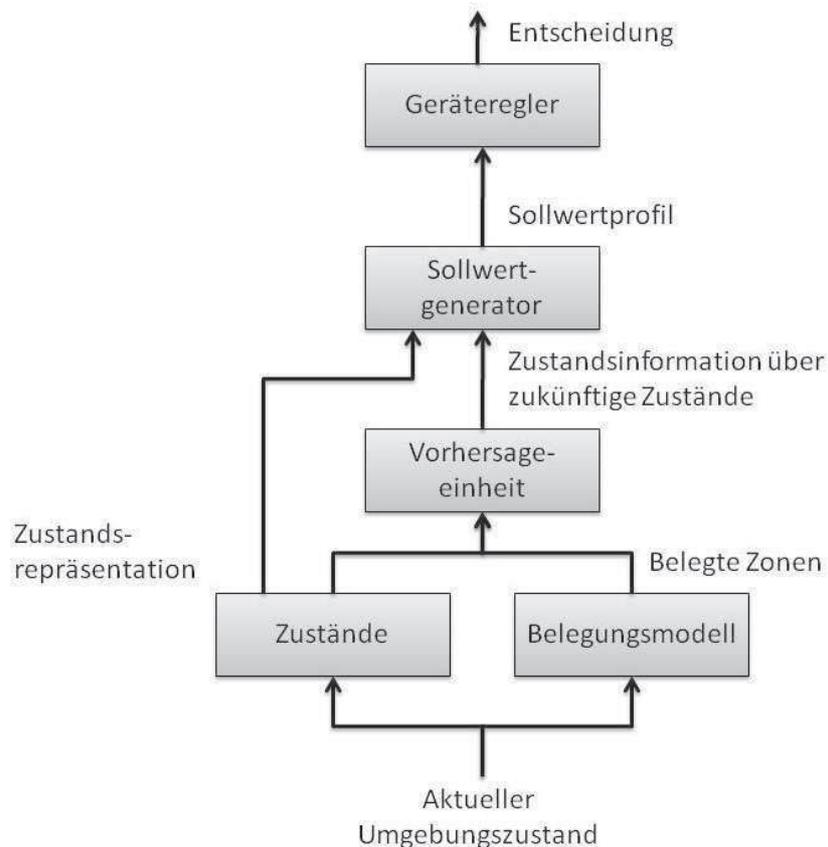


Abbildung 13: ACHE Architekturübersicht

In Abbildung 13 ist die Architektur von ACHE abgebildet. ACHE misst den aktuellen Umgebungszustand. Aus erkannten Zuständen und dem Belegungsmodell wird in der Vorhersageeinheit auf zukünftige Zustände geschlossen. Aus den getroffenen Vorhersagen

<sup>32</sup> Luftklappen und Jalousien waren auch angedacht, wurden aber nicht im Feldversuch angesteuert.

und aus den aktuellen Zuständen wird im Sollwertgenerator ein Sollwertprofil abgeleitet und dem Geräteregele weitergegeben, der schließlich die Stellwerte zur Regelung der Geräte weiterleitet. Die Vorhersageeinheit, der Sollwertgenerator und der Geräteregele werden mit Hilfe von künstlich neuronalen Netzen implementiert.

Die Vorhersageeinheit besitzt Wissen über die Zeitpläne der Bewohner. Im Sollwertgenerator wird Wissen über die Komfortansprüche und die Vorzüge der Bewohner abgeleitet. Im Geräteregele wird Wissen über die Gebäudestruktur und die darin verbauten Geräte angewendet.

Trainiert wird die Vorhersageeinheit mit einem selbstlernenden Netz. Der Sollwertgenerator wird durch bestärkendes Lernen<sup>33</sup> und Nachschlagetabellen oder speicherbasierten Ansätzen trainiert. Der Geräteregele verwendet neuronale Netze zur Identifikation der Feldgeräte.

In der Lernphase muss evaluiert werden, welche Sequenzen von Entscheidungen für die beobachteten Kosten verantwortlich sind. Zu dem Zweck wird das Zeitintervall auf 200ms gelegt. Kürzere Beobachtungszeiten haben sich in dem Feldversuch als nicht praktikabel herausgestellt.

Für das Training der Beleuchtungsregelungseinheit wurden die künstlich neuronale Netze mit mehr als 70.000 Datensätzen trainiert. Das System wurde über 3 Monate hinweg getestet. Über die Zeit gesehen, wurden die Energiekosten und die Kosten für Unbehagen<sup>34</sup> um etwa 25% gesenkt. Ähnliche Ergebnisse wurden auch bei der Raumtemperaturregelung erzielt.

Die Feldstudie mit dem Prototyp ACHE hat gezeigt, dass mit künstlich neuronalen Netzen der Komfort für die Bewohner gesteigert und die Energiekosten gesenkt werden können. Die Fragen, wie man das System ACHE designen kann, sodass es in einem beliebigen Haus ohne vorherige Konfiguration läuft und inwieweit man historischen Daten für die Vorhersage Bedeutung schenkt, wurden aufgeworfen. Wichtig ist, dass die Bewohner so wenig wie möglich mit der Komplexität des Systems in Kontakt kommen, um nicht überfordert zu werden. Damit so ein System wie ACHE arbeiten kann, werden viele Sensoren benötigt, je mehr desto besser. Außerdem wurde festgestellt, dass eine Aktivitätsklassifizierung von hohem Wert für die Vorhersage ist.

### 2.3.2.3.1 Verbindung zu dieser Arbeit

Die vorgestellte Feldstudie zeigt, wie mit Hilfe von künstlich neuronalen Netzen die automatische Beleuchtungs- und die Raumtemperaturregelung implementiert werden kann. Dabei hat sich bewiesen, dass für jede Anwendung unterschiedliche künstlich neuronale Netze verwendet werden müssen. Das Projekt hat auch die Frage an den Tag gebracht, wie weit die Vergangenheit für eine Vorhersage berücksichtigt werden muss. In dieser Arbeit werden verschiedene künstliche neuronale Netze für eine Einzelraumheizung evaluiert.

Das ACHE System generalisiert die Bewohner. Das heißt, es unterscheidet nicht zwischen verschiedenen Bewohnern und daher kann es vorkommen, dass Bewohner für gleiche

---

<sup>33</sup> Engl. Reinforcement Learning. Wird für maschinelles Lernen eingesetzt. Das zugrunde liegende Konzept sind Markow-Entscheidungsprobleme. Weitere Literatur siehe [62].

<sup>34</sup> Mit dieser Größe wird fiktiv errechnet welche monetären Kosten entstehen, falls sich ein Bewohner bei den eingestellten Umgebungsbedingungen unwohl fühlt.

Tätigkeiten verschiedene Ansprüche an ihre Umgebung stellen. Zum Beispiel kann die eine Person beim Fernsehen gerne die Deckenbeleuchtung und einen Deckenfluter gedimmt haben wollen und die andere Person will die Deckenbeleuchtung aus haben, dafür den Deckenfluter etwas heller eingestellt haben. Diese Diplomarbeit beschäftigt sich daher auch mit dem Problem, wie mit den Komfortansprüchen mehrerer Bewohner individuell umgegangen werden kann.

## 3 Anwendungsszenarien

Die gestellten Anforderungen an ein intelligentes Heim- und Gebäudeautomationssystem sind sehr hoch. Aus diesem Grund werden an dieser Stelle einige ausgewählte Anwendungsszenarien vorgestellt, die diese illustrieren. Die Anwendungsszenarien werden in natürlicher Sprache und anschließend mit objektorientierten Analysemethoden von UML [24] [25] beschrieben.

Als graphische Darstellungsformen kommen die folgenden Verhaltensdiagramme zur Verwendung:

- **Anwendungsfalldiagramm** (Use Case Diagram): Als Abkürzung in den nachfolgenden Zeichnungen wird das Kürzel „*uc*“ verwendet. Mit diesem Diagrammtyp wird gezeigt, was das geplante System leisten kann, ohne auf Details der Realisierung einzugehen. Dabei werden zu den jeweiligen Anwendungsfällen die dazugehörigen Akteure identifiziert. Akteure können Personen oder aber auch eine Hard- oder Softwareeinheit sein.
- **Aktivitätsdiagramm** (Activity Diagram): Als Abkürzung in den nachfolgenden Zeichnungen wird das Kürzel „*ad*“ verwendet. Mit Hilfe dieses Diagrammtyps wird das kommunikative Verhalten von Systemen untersucht. Dabei wird zwischen Daten- und Kontrollfluss unterschieden. Die Aufgaben werden in einzelne Arbeitsschritte zerteilt. Mit Aktivitätsdiagrammen lassen sich die Reihenfolge der Arbeitsschritte, parallele Arbeitsschritte und alternative Entscheidungswege identifizieren. Ein Anwendungsfall besteht aus Knoten und Kanten. Die Knotenmenge besteht aus Aktionsknoten<sup>35</sup>, Objektknoten<sup>36</sup> und Kontrollknoten<sup>37</sup>. Die Kanten zwischen den Knoten sind gerichtet und zeigen die Kontroll- bzw. Datenflussrichtung.
- **Sequenzdiagramm** (Sequence Diagram): In den Zeichnungen findet man das Kürzel „*sd*“ zur Identifikation dieses Diagrammtyps. Mit dem Sequenzdiagramm ist es möglich, den Nachrichtenfluss zwischen Objekten darzustellen. Dabei ist der zeitliche Aspekt der Nachrichten von besonderer Bedeutung.

In den Anwendungsfällen und auch im weiteren Verlauf dieser Arbeit wird in manchen Szenarien zwischen Eco- und Komforttemperatur unterschieden. Die Eco-Temperatur ist mit 16°C<sup>38</sup> festgelegt, die Komforttemperatur liegt bei 22°C<sup>39</sup>.

### 3.1 Automatische energiesparende Raumtemperaturregelung für eine Person nach einem gelernten Belegungsplan

In diesem Szenario kombiniert die Raumheizungsregelung die Energie aus einem Heizkessel und der Sonneneinstrahlung. Die Räume Küche, Esszimmer, Wohnzimmer, Schlafzimmer, WC, Badezimmer und Arbeitszimmer werden einzeln geheizt. Das Heim- und Gebäudeautomationssystem kennt den Gebäudeplan und die Ausrichtung der Räume nach den Himmelsrichtungen.

---

<sup>35</sup> Sind die elementaren Verhaltensbausteine.

<sup>36</sup> Hilfsknoten, die den Fluss von Objekten durch das Netz spezifizieren.

<sup>37</sup> Steuern den Kontroll- und Datenfluss.

<sup>38</sup> Um Schimmel zu vermeiden.

<sup>39</sup> Durchschnittliche Raumtemperatur, bei der sich die meisten Menschen wohl fühlen.

### 3.1.1 Sprachliche Beschreibung

Es hat bei wenig Wind 15°C Außentemperatur um 16:00 Uhr an einem sonnigen Mittwochnachmittag im April. Die Raumtemperatur im Singlehaushalt beträgt in allen Räumen Eco-Temperatur. Um 17 Uhr kommt der Bewohner von der Arbeit nach Hause und bevorzugt ab diesem Zeitpunkt Komforttemperatur in den südwestseitig gelegenen Räumen Küche, Esszimmer und Wohnzimmer. Das Schlafzimmer, WC, Badezimmer und Arbeitszimmer, also die restlichen Räume des Hauses, werden in diesem Szenario nicht betrachtet.

Das Heim- und Gebäudeautomationssystem misst jede Minute periodisch die jeweilige Raumtemperatur, Außentemperatur, Windstärke und Sonneneinstrahlung. Außerdem werden zur Berechnung des optimalen Startzeitpunkts für die Raumheizung die Raumausrichtung und Onlinewetterdaten für den kommenden Abend berücksichtigt. Aus diesem Mix an Informationen errechnet das Heim- und Gebäudeautomationssystem den optimalen Startzeitpunkt für die Raumheizung. Damit das Heim- und Gebäudeautomationssystem rechtzeitig die Maßnahmen für die Raumheizung einleiten kann, wird der angelernte Raumebelegungsplan betrachtet. Alle Belegungen, die ab dem aktuellen Zeitpunkt in weniger oder gleich 60 Minuten starten, werden für die Berechnung des optimalen Heizungsstarts in Betracht gezogen. Aus dem Grund werden in diesem Beispiel die Jalousien bei günstiger Sonneneinstrahlung hochgezogen, um solare Thermoenergie für die Raumheizung<sup>40</sup> zu nutzen.

Die Berechnung des Heim- und Gebäudeautomationssystems wird in diesem Szenario um 16:00 Uhr das erste Mal durchgeführt. Dabei berechnet das Heim- und Gebäudeautomationssystem, dass die Aufheizzeit der benutzten Räume von 18°C Eco-Temperatur auf 22°C Komforttemperatur 33 Minuten mit dem Heizungskessel dauern wird. Wenn die Jalousien bei den aktuellen Umgebungsverhältnissen hochgezogen werden, dann sind nur 28 Minuten notwendig, aber ohne Zufuhr von Energie durch den Heizungskessel kann die Komforttemperatur nicht erreicht werden. Damit die solare thermische Energie ihre maximale Wirkung erzielen kann, werden die Jalousien sofort in den betroffenen Räumen geöffnet. Die nächste Berechnung findet um 16:01 Uhr statt. Die Aufheizzeit mit aktuell geöffneten Jalousien und dem später zugeschaltetem Heizungskessel beträgt zu diesem Zeitpunkt 33 Minuten. Um 16:20 Uhr beträgt die Aufheizzeit 30 Minuten und um 16:30 Uhr 29 Minuten. Der tatsächliche Startzeitpunkt für den Heizungskessel wird durch das Heim- und Gebäudeautomationssystem auf 16:32 Uhr festgelegt.

Um 17:00 Uhr, wenn der Bewohner das Haus betritt, haben alle gewünschten Räume Komforttemperatur erreicht.

### 3.1.2 Modellanalyse mit UML

Dieses Szenario wird zuerst mit einem Anwendungsfalldiagramm beschrieben. Anschließend wird die Raumtemperaturregelung in einem Anwendungsfalldiagrammen genauer betrachtet. Die übergeordnete Aktivität „In einem Raum wohl fühlen“ wird in einem Sequenzdiagramm dargestellt.

---

<sup>40</sup> Glashauseffekt

3.1.2.1 Anwendungsfall

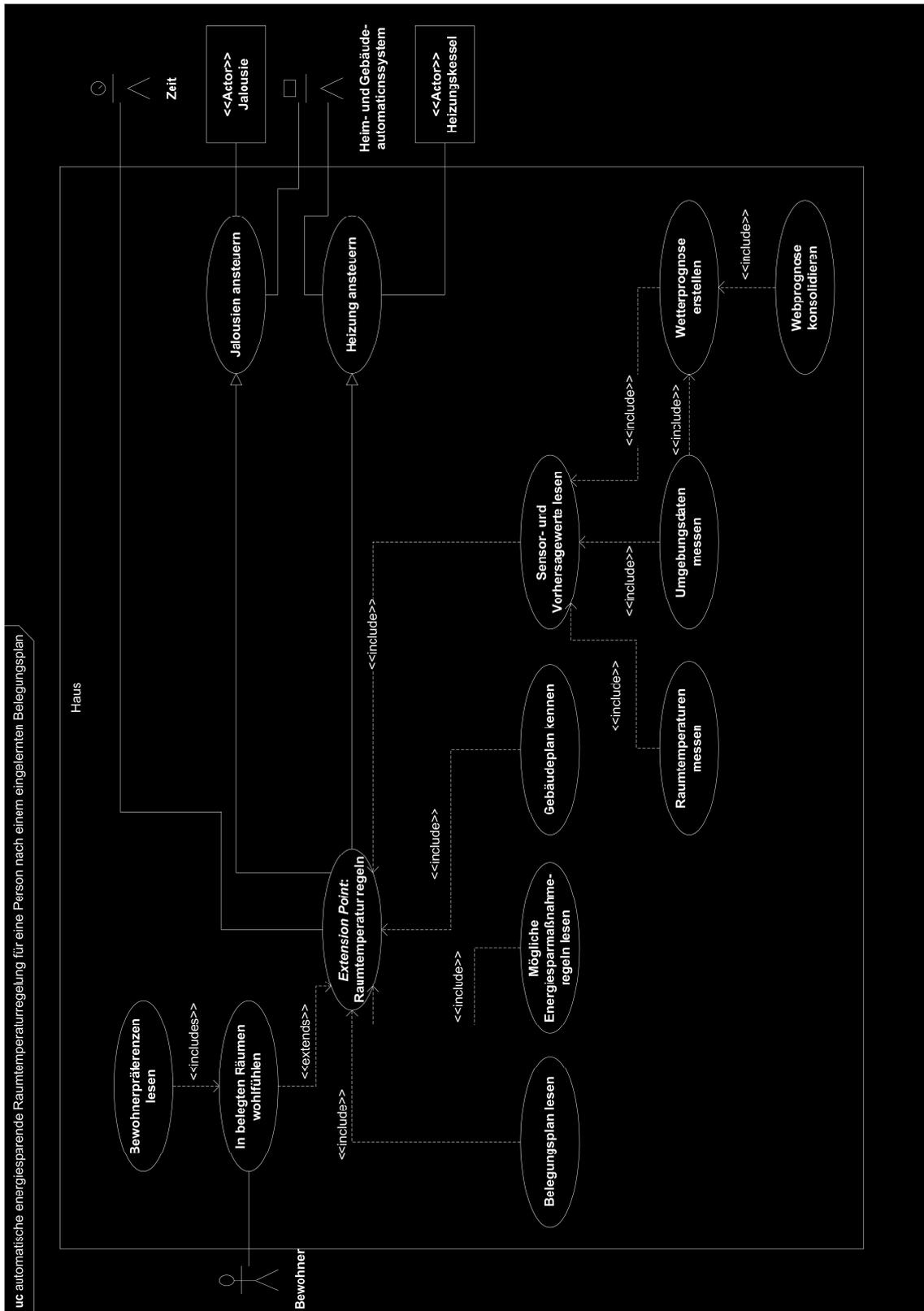


Abbildung 14: Anwendungsfall: Automatische Raumtemperaturregelung nach einem eingelernten Belegungsplan

## 3.1.2.1.1 Beschreibung

<b>Name des Anwendungsfalls</b>	automatische energiesparende Raumtemperaturregelung für eine Person nach einem eingelernten Belegungsplan
<b>Ziel</b>	Energiesparendes Einstellen der Raumtemperatur nach einem Raumbelungsplan
<b>Kategorie</b>	Primärer Anwendungsfall
<b>Vorbedingung</b>	Bekannter Raumbelungsplan, Gebäudemodell, Raumtemperatur, Außentemperatur, Windgeschwindigkeit, Sonneneinstrahlung, Wettervorhersagedaten aus dem Internet
<b>Nachbedingung</b>	Eingestellte Raumtemperatur nach dem Belegungsplan mit möglichst geringem Heizungsenergieverbrauch
<b>Akteure</b>	Bewohner, Heim- und Gebäudeautomationssystem
<b>Auslösendes Ereignis</b>	Uhrzeit
<b>Beschreibung</b>	Das Heim- und Gebäudeautomationssystem misst jede Minute Raumtemperatur, Außentemperatur, Windgeschwindigkeit, Luftfeuchtigkeit, Sonneneinstrahlung und Wettervorhersagedaten. Außerdem kennt das Heim- und Gebäudeautomationssystem die Gebäudestruktur aus dem Gebäudeinformationsmodell. In Verbindung mit dem bereits gelernten Raumbelungsplan weiß das Heim- und Gebäudeautomationssystem, wann ein Raum auf Eco-Temperatur oder auf Komforttemperatur geheizt werden muss. Das Heim- und Gebäudeautomationssystem kann mit Hilfe der Verschattungsanlage solare Wärmeenergie in die Räume lassen und bei Bedarf fehlende Wärmeenergie mit dem Heizungskessel erzeugen.

Tabelle 4: Beschreibung des Anwendungsfalls - automatische energiesparende Raumtemperaturregelung für eine Person nach einem eingelernten Belegungsplan

## 3.1.2.2 Aktivitätsdiagramme

## 3.1.2.2.1 Raumtemperatur regeln

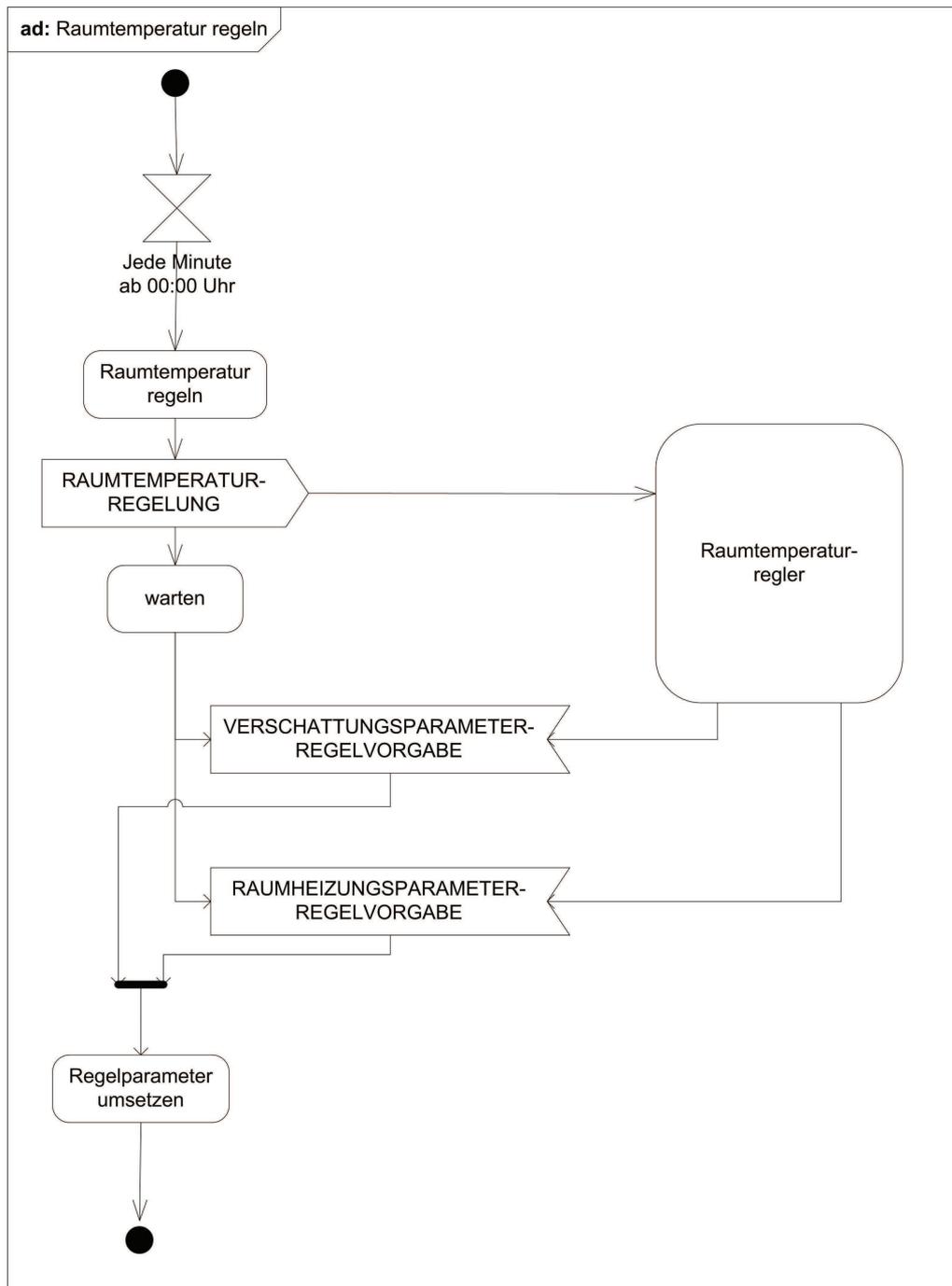


Abbildung 15: Aktivitätsdiagramm zur Raumtemperaturregelung

Das Heim- und Gebäudeautomationssystem führt jede Minute eine Raumtemperaturregelung durch. In diesem Aktivitätsdiagramm erfolgt eine getrennte Darstellung zwischen kontinuierlichen zeitlichen Ereignissen und der eigentlichen Raumtemperaturregelung, die in Abbildung 16 genauer betrachtet wird.

3.1.2.2.2 Raumtemperaturregler

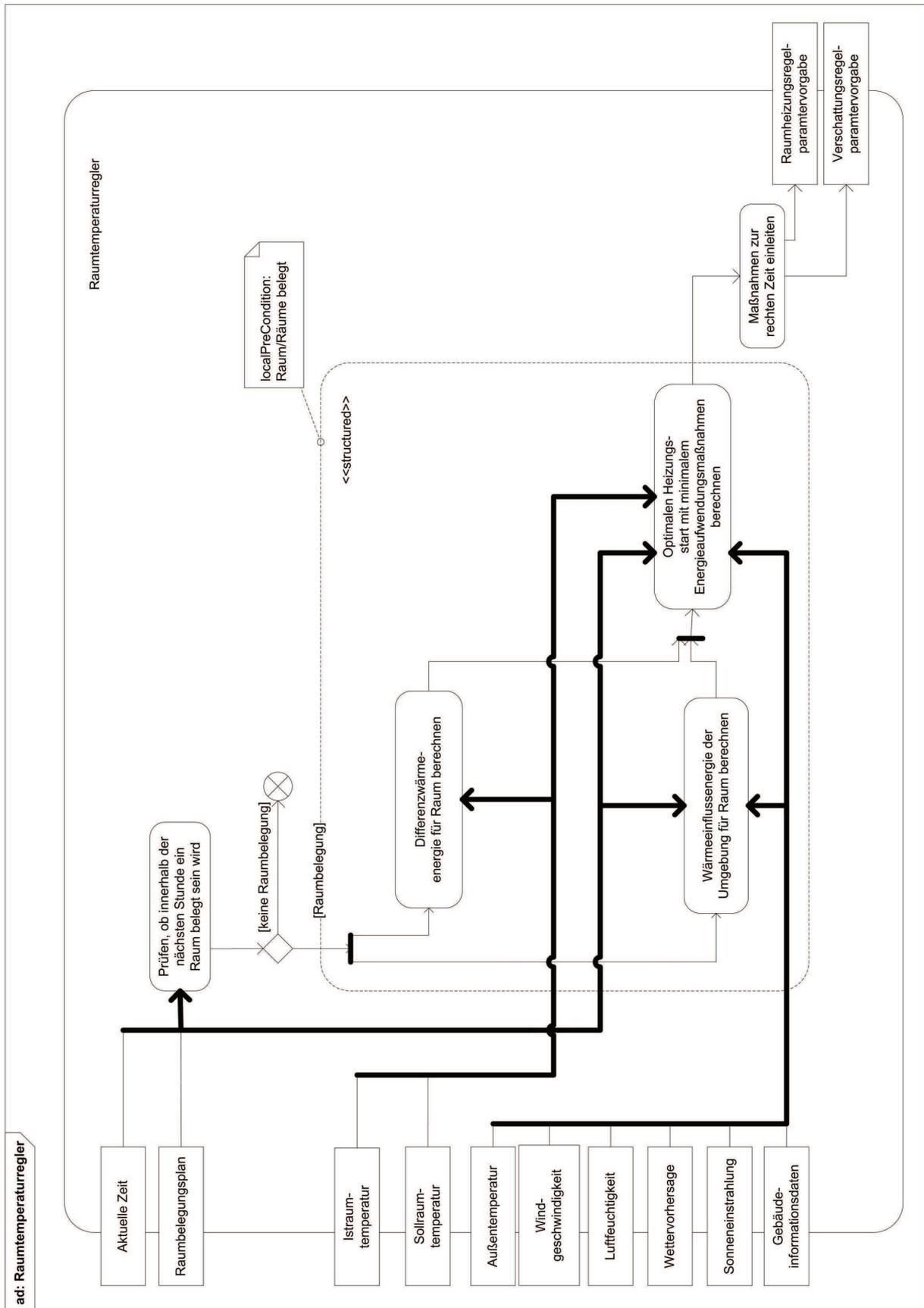


Abbildung 16: Aktivitätsdiagramm für den Raumtemperaturregler

Das Diagramm wird von links nach rechts durchlaufen. Zuerst wird geprüft, ob aktuell oder innerhalb der nächsten Stunde ein Raum belegt sein wird.

Anschließend wird im positiven Fall, dass ein Raum belegt sein wird, die benötigte Differenzwärmeenergie für diesen Raum aus der Isttemperatur und der Solltemperatur errechnet. Gleichzeitig wird die Beeinflussung durch Umgebungsparameter wie Außentemperatur, Windgeschwindigkeit, Luftfeuchtigkeit, Sonneneinstrahlung, Wettervorhersage und Informationen aus dem Gebäudemodell mit berücksichtigt.

Aus diesen beiden vorangegangenen Berechnungen werden die optimalen Maßnahmen zur Erreichung der Raumtemperatur bestimmt und der optimale Zeitpunkt für den Start des Heizkessels ermittelt. Die geplanten Maßnahmen für die Raumtemperierung werden dann über Verschattung oder Heizungskesselstart mit entsprechender Brenndauer umgesetzt.

## 3.1.2.3 Sequenzdiagramm

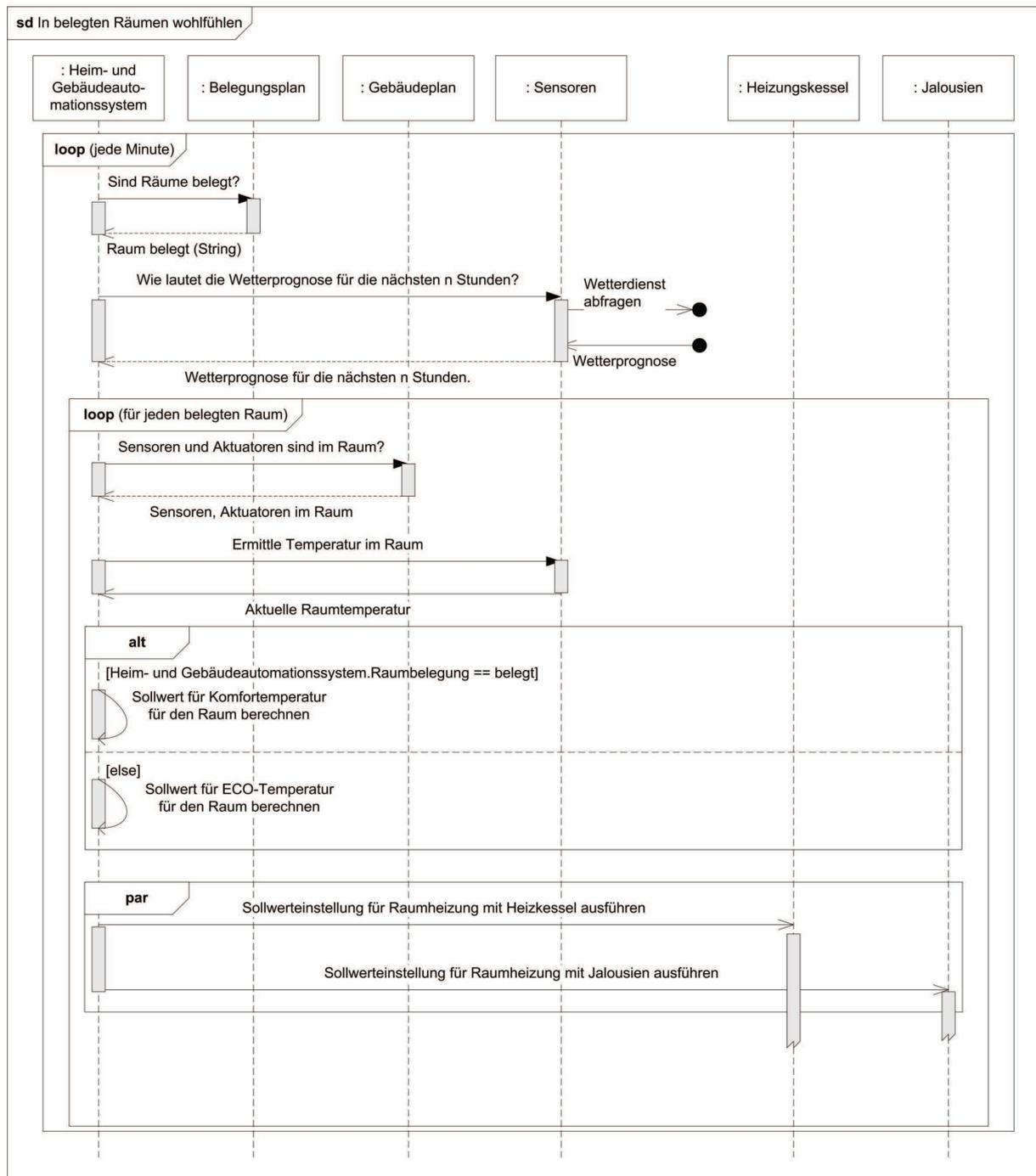


Abbildung 17: Sequenzdiagramm für das thermische Wohlbefinden eines Bewohners in einem Raum

Im Sequenzdiagramm in Abbildung 17 ist der Kommunikationsablauf für den Anwendungsfall „In belegten Räumen wohl fühlen“, der als Erweiterung Raumtemperatur regeln gilt, dargestellt.

Jede Minute sieht das Heim- und Gebäudeautomationssystem nach, ob ein Raum belegt ist oder innerhalb einer Stunde belegt sein wird. Das Heim- und Gebäudeautomationssystem konsolidiert die Wetterinformationen, diese bestehen aus den aktuellen Umgebungsdaten in Verbindung mit Internetwettervorhersagedaten. Anschließend werden für jeden Raum die Anzahl und die Identifikationsnummern der verbauten Temperatursensoren abgefragt.

Mit diesen Informationen kann das Heim- und Gebäudeautomationssystem die Temperatur in einem Raum ermitteln.

Je nachdem, ob ein Raum belegt ist oder nicht, ergeben sich unterschiedliche Temperatursollwertvorgaben und Regelungsabläufe.

Aus den gesammelten Informationen berechnet das Heim- und Gebäudeautomationssystem optimale Startzeitpunkte und Sollwertvorgaben für die Heizung und Jalousien, damit Komforttemperatur oder ECO-Temperatur in den Räumen hergestellt werden kann. Diese Vorgaben werden dann vom Heiz- oder Verschattungssystem ausgeführt.

## **3.2 Automatische Raumtemperaturregelung für eine Mehrpersonenbelegung mit Konfliktlösungseinheit**

### **3.2.1 Sprachliche Beschreibung**

In einem Haus wohnt ein Ehepaar. Für beide Personen hat das Heim- und Gebäudeautomationssystem zwei unterschiedliche Wohlfühltemperaturen gelernt. Die Frau hat eine um 2°C höhere Wohlfühltemperatur als die Komforttemperatur, also hat das Heim- und Gebäudeautomationssystem die Komforttemperatur für die Frau auf 24°C festgelegt. Die Wohlfühltemperatur des Mannes entspricht der voreingestellten Komforttemperatur. Sind beide Personen in unterschiedlichen Räumen, dann erkennt dies das Heim- und Gebäudeautomationssystem und stellt daher für die Frau 24°C und 22°C in den durch den Mann belegten Räumen ein. Typischerweise befindet sich die Frau an einem Abend bis um 20:00 Uhr im Arbeitszimmer. Dort ist die Raumtemperatur auf 24°C eingestellt. Für den Mann, der zur selben Zeit in der Küche für beide kocht, ist 22°C Raumtemperatur eingestellt. Aus dem beobachteten Zeitplan kann das Heim- und Gebäudeautomationssystem schließen, dass typischerweise 30 Minuten nachdem der Mann zu kochen begonnen hat, das Essen fertig sein wird und der Essbereich warm sein muss. Um 20:00 Uhr berechnet das Heim- und Gebäudeautomationssystem zum ersten Mal die optimale Startzeit für die Heizung für den Essbereich. Dort ist zu diesem Zeitpunkt noch Eco-Temperatur eingestellt.

Zum Auflösen eines Wohlfühltemperaturkonflikts betrachtet das Heim- und Gebäudeautomationssystem die Wohlfühltemperaturen jedes einzelnen Bewohners und stellt bei unterschiedlichen Wohlfühltemperaturen jene persönliche Wohlfühltemperatur ein, die am wenigsten Energie für die Raumtemperaturnachführung benötigt. Somit würde in diesem Beispiel die Wohlfühltemperatur des Mannes eingestellt werden. Die Bewohner haben dazu noch die Möglichkeit über eine Benutzerschnittelle die Temperatur abzuändern. Das Heim- und Gebäudeautomationssystem lernt diese Änderung und wendet das gelernte Wissen bei einer kommenden Situation an.

Der bestehende Temperaturkonflikt zwischen Frau und Mann wurde zuvor durch das Heim- und Gebäudeautomationssystem aufgelöst. Das Heim- und Gebäudeautomationssystem stellt für die Belegung des Essbereichs die Temperatur auf 23°C, ein Kompromiss, der durch manuelle Raumkontrollelemente zuvor automatisch eingelernt wurde. Damit es um 20:30 im Essbereich 23°C hat, muss die Raumheizung 27 Minuten heizen. Daher beginnt die Raumheizung den Essbereich um 20:03 Uhr zu heizen.

### 3.2.2 Modellanalyse mit UML

Zuerst wird das Szenario anhand eines Anwendungsfalldiagramms gezeigt. Später wird auf die Konfliktlösung eingegangen.

#### 3.2.2.1 Anwendungsfall

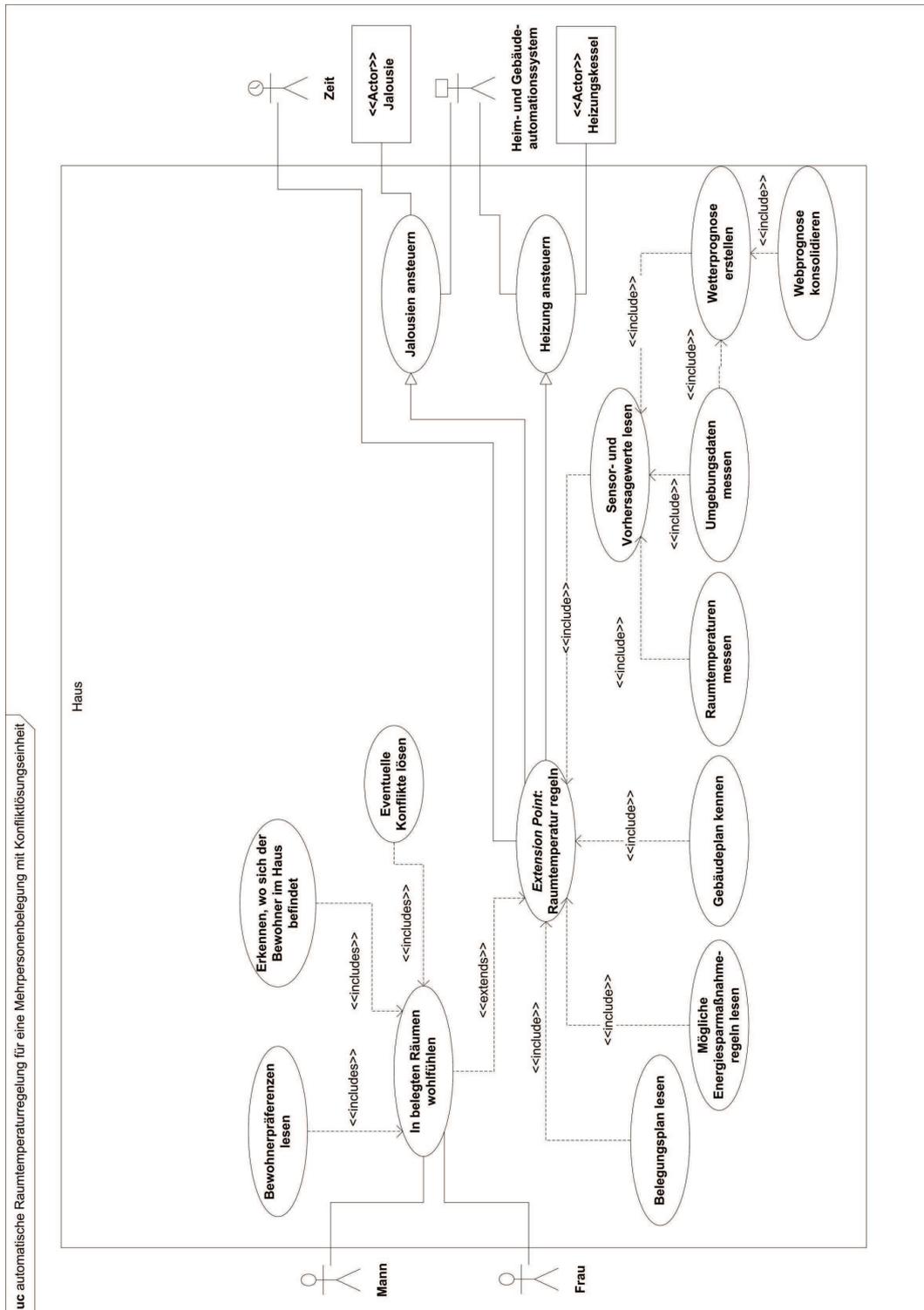


Abbildung 18: Anwendungsfall zur automatischen Raumtemperaturregelung für einen Mehrpersonen Haushalt mit Konfliktlösung

## 3.2.2.1.1 Beschreibung

Name des Anwendungsfalls	automatische Raumtemperaturregelung für eine Mehrpersonenbelegung mit Konfliktlösungseinheit
<b>Ziel</b>	Regeln der Raumtemperatur für jeden Bewohner individuell nach einem Belegungsplan und persönlicher Wohlfühltemperaturen. Bei einer Belegung eines Raumes durch mehrere Bewohner gleichzeitig wird eine Kompromissraumtemperatur gebildet.
<b>Kategorie</b>	Primärer Anwendungsfall
<b>Vorbedingung</b>	Bekannter Raumbellegungsplan, bekannte persönliche Wohlfühltemperaturen, Konfliktlösungswissen, Gebäudemodell, Raumtemperatur, Außentemperatur, Windgeschwindigkeit, Sonneneinstrahlung, Wettervorhersagedaten aus dem Internet.
<b>Nachbedingung</b>	Eingestellte Raumtemperatur nach dem Belegungsplan mit möglichst geringem Heizungsenergieverbrauch für jeden Bewohner individuell, falls mehrere Bewohner einen Raum belegen, dann eine Kompromissraumtemperatur.
<b>Akteure</b>	Bewohner, Heim- und Gebäudeautomationssystem
<b>Auslösendes Ereignis</b>	Uhrzeit, Belegung eines Raumes durch die Bewohner
<b>Beschreibung</b>	<p>Das Heim- und Gebäudeautomationssystem misst jede Minute die Raumtemperatur, die Außentemperatur, die Windgeschwindigkeit, die Sonneneinstrahlung und die Wettervorhersagedaten. Außerdem kennt das Heim- und Gebäudeautomationssystem die Gebäudestruktur aus dem Gebäudeinformationsmodell. In Verbindung mit dem bereits gelernten Raumbellegungsplan weiß das Heim- und Gebäudeautomationssystem, wann ein Raum auf Eco-Temperatur oder auf Wohlfühltemperatur geheizt werden muss. Das Heim- und Gebäudeautomationssystem kann mit Hilfe der Verschattungsanlage solare Wärmeenergie in die verwendeten Räume lassen und bei Bedarf fehlende Wärmeenergie mit dem Heizungskessel aus Brennstoffen erzeugen.</p> <p>Bei einer Mehrfachraumbelegung und bei Konflikten der persönlichen Wohlfühltemperaturen der jeweiligen Bewohner wird die eingelernte Kompromiss-raumtemperatur durch das Heim- und Gebäude-automationssystem eingestellt.</p>

Tabelle 5: Beschreibung des Anwendungsfalls - automatische Raumtemperaturregelung für eine Mehrpersonenbelegung mit Konfliktlösungseinheit

## 3.2.2.2 Sequenzdiagramm

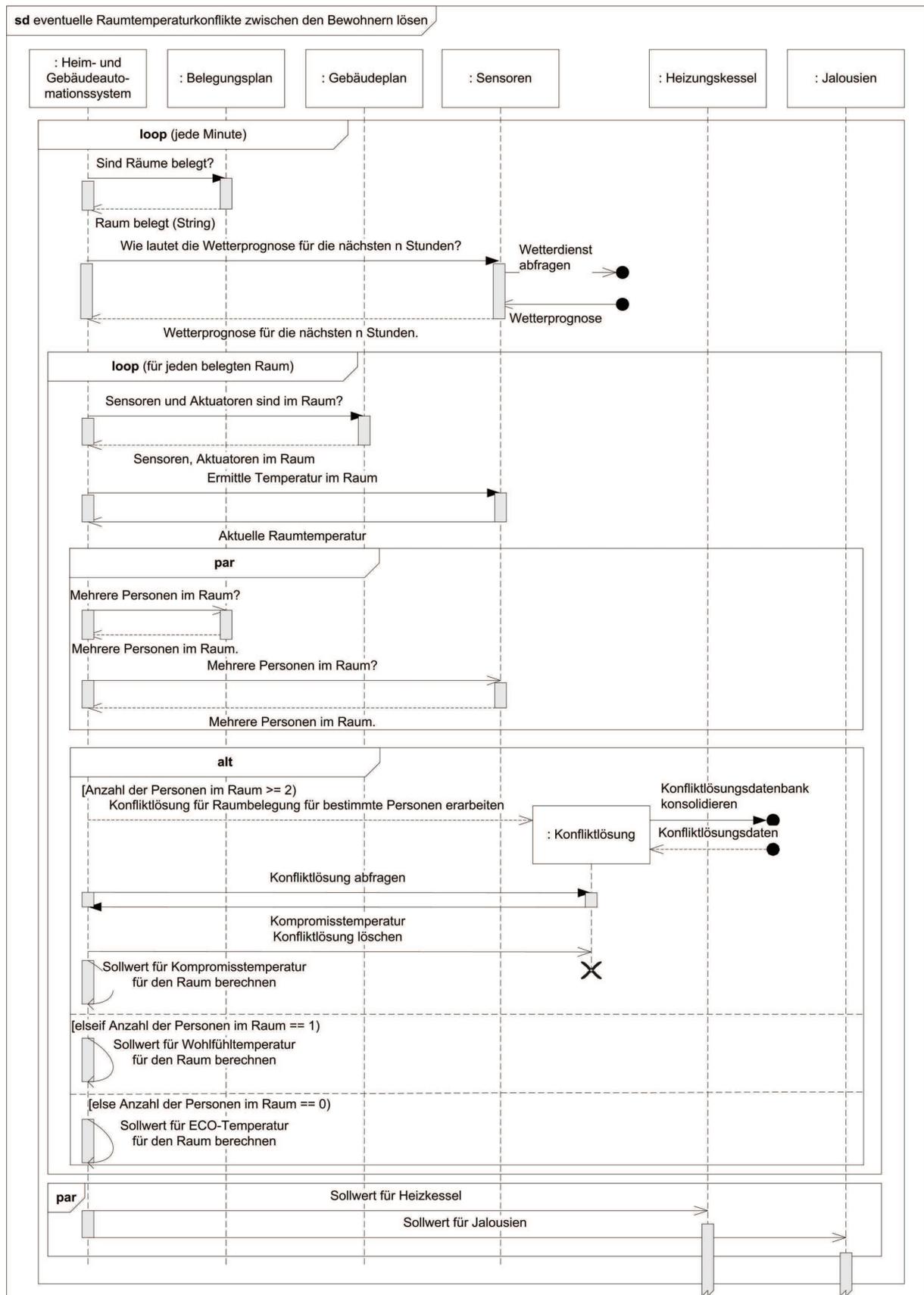


Abbildung 19: Sequenzdiagramm zur Auflösung eventuell auftretender Raumtemperaturkonflikte zwischen den Bewohnern

Das Heim- und Gebäudeautomationssystem sieht jede Minute im Belegungsplan nach, ob ein Raum aktuell belegt ist oder ob in der nächsten Stunde mit einer Raumbellegung gerechnet werden muss. Anschließend wird eine Wetterprognose vom Heim- und Gebäudeautomationssystem angefordert. Dazu werden lokale Sensoren, wie Außenthermometer, Hydrometer, Anemometer sowie ein externer Wettervorhersagedienst konsultiert.

Daraufhin wird für jeden Raum nachgesehen, welche Sensoren sich in diesem befinden. Diese Information erhält das Heim- und Gebäudeautomationssystem aus einem digitalen Gebäudeplan, in dem architektonische Informationen sowie Informationen über Heizung, Lüftung, Kühlung, Verschattung, sonstige Gebäudeautomationssysteme, elektrische Geräte und weitere Geräte, die über ein Kommunikationsnetzwerk mit dem Heim- und Gebäudeautomationssystem kommunizieren können. Mit diesen Informationen kann das Heim- und Gebäudeautomationssystem die gemessenen Temperaturwerte einem bestimmten Raum zuordnen.

Die Anzahl der Personen, die sich aktuell im Raum befinden, wird über Präsenzdetectoren in den jeweiligen Räumen festgestellt. Die Anzahl der Personen, die sich typischerweise in dem Raum befinden oder sich in Zukunft darin befinden könnten, wird dem Belegungsplan entnommen.

Je nachdem, welche Personen sich in einem Raum befinden, wird eine andere Wohlfühltemperatur eingestellt. Befinden sich mehr als 2 Personen in einem Raum, dann wird aufgrund sehr wahrscheinlich unterschiedlicher Wohlfühltemperaturen der Bewohner eine Kompromisstemperatur eingestellt. Der Kompromiss wird in der Konfliktlösungseinheit bearbeitet, die auf Aufzeichnungen von vergangenen Ereignissen und Ausgängen von Konfliktlösungen zugreifen kann, wo die Personen in einem Raum waren. Die durch die Konfliktlösung resultierende Wohlfühltemperatur wird dem Heim- und Gebäudeautomationssystem mitgeteilt, das dann die Sollwertberechnung für den Heizungsregler errechnet.

Befindet sich nur eine Person in einem Raum, dann wird die Wohlfühltemperatur der jeweiligen Person für die Sollwertberechnung verwendet.

Ist aktuell und in der nächsten Zeit keine Person in dem betrachteten Raum, wird ECO-Temperatur für die Sollwertberechnung verwendet.

### **3.3 Singlehaushalt: Aufstehen am Morgen während der Arbeitswoche**

#### **3.3.1 Sprachliche Beschreibung**

In diesem Szenario wird davon ausgegangen, dass eine Person in einem Haus lebt und einer Arbeit mit geregelter Normalarbeitszeit nachgeht, die gewöhnlich zwischen 8:00 Uhr morgens und 17:00 Uhr abends liegt. Außerdem wird angenommen, dass das beschriebene Szenario während der Heizperiode im Februar stattfindet.

Geweckt wird die Person täglich von Montag bis Freitag um 6:30 Uhr über einen Radiowecker. Nachdem die Person aufgewacht und aufgestanden ist, erledigt die Person die tägliche Morgentoilette im Bad. Anschließend wird um circa 7:00 Uhr mit Kaffee und

Brötchen bei Radiomusik und der frischen Onlinetageszeitung gefrühstückt. Dann geht es um ca. 7:30 Uhr mit dem Auto zur Arbeit, um pünktlich um 8:00 Uhr ohne größere Verkehrsverzögerungen das Büro zu betreten.

Nachfolgend werden die internen Abläufe des Heim- und Gebäudeautomationssystems beschrieben. Zuerst wird nur die Raumheizung betrachtet. Um Kosten zu sparen, ist in der Nacht die Temperatur des Wohnhauses auf Eco-Temperatur abgesenkt worden. Das Heim- und Gebäudeautomationssystem weiß aus Beobachtungen, dass die Person um 6:30 Uhr geweckt werden muss. Noch bevor der Wecker läutet, wird das Heizungssystem rechtzeitig im Vorhinein beginnen die Raumtemperatur von 18°C auf 22°C zu heben. Um den optimalen Startzeitpunkt für die Heizung zu berechnen, müssen einige statische und dynamische Parameter neben der aktuellen Uhrzeit, wie zum Beispiel die maximale Heizleistung des Heizungssystems, thermisches Raumverhalten, aktuelle Raumtemperatur, aktuelle und gewünschte Brauchwassertemperatur im Boiler, Außenumweltbedingungen<sup>41</sup> und die Wettervorhersage mitberücksichtigt werden.

Die aktuelle Raumtemperatur in diesem Anwendungsszenario beträgt um 5:30 Uhr morgens 18°C, denn einige Stunden zuvor wurde die Temperatur im Haus auf Eco-Temperatur eingestellt. Die Außentemperatur beträgt aktuell 5°C bei absoluter Windstille. Das Wetter für die nächsten 3 Stunden ist gleichbleibend bewölkt mit einer von einem Internetwetterdienst prognostizierten Umgebungstemperatur von 7°C um 6:30 Uhr. Das Heim- und Gebäudeautomationssystem berechnet um 5:30 Uhr eine Aufheizzeit von 20 Minuten bei den aktuellen Bedingungen. Diese Berechnungen werden jede Minute neu durchgeführt. Schließlich um 6:10 Uhr beginnt die Heizung die Räume auf 22°C zu heizen, damit um 6:30 die Komforttemperatur hergestellt ist. Die Heizung bleibt solange an, bis der Bewohner das Haus mit dem Auto über die Garage verlassen hat. Sofort danach wird die Raumtemperatur auf Eco-Temperatur abgesenkt.

Nun wird die Brauchwassertemperatur und -menge betrachtet, die für die Morgentoilette zur Verfügung gestellt werden muss. Das Heim- und Gebäudeautomationssystem hat aus Beobachtungen gelernt, dass der Bewohner um 6:45 Uhr eine Dusche für circa 3 Minuten mit einer Wassermenge von circa 18 Liter bei einer Duschwassertemperatur von 38°C benötigt und circa 2 Liter Wasser für sonstiges Brauchwasser bei der gleichen Temperatur entnimmt. Somit müssen in Summe 20 Liter Brauchwasser zur Verfügung gestellt werden. Wenn möglich, wird der Boiler für Brauchwasser vom Heizungskessel nicht zur selben Zeit geladen, wo die Raumheizung aktiv ist. Das heißt, das Brauchwasser im Boiler wird schon auf Temperatur sein, noch bevor der Heizkessel die Räume heizen muss. Im Haus ist ein 300 Liter Warmwasserboiler verbaut, dessen Temperatur auf unter 30°C gefallen ist. Daher muss der Heizungskessel den Boiler aufheizen. Aus Beobachtungen weiß das Heim- und Gebäudeautomationssystem, dass der Heizungskessel circa 90 Minuten mit der Boilerladung beschäftigt sein wird, um die Wassertemperatur auf des Boilers auf 60°C zu heizen. Mit der Boilerladung wird daher um 4:50 Uhr begonnen. Das sind 90 Minuten vor dem Beginn der Raumheizung, wo die komplette Kesselleistung für Raumheizung verwendet wird.

Um 6:30 wird der vom Heim- und Gebäudeautomationssystem automatisch gestellte Wecker für den Bewohner aktiviert. Aus den Lautsprechern ertönt der Lieblingsradiosender in sanft ansteigender Lautstärke, die Jalousien im Haus werden hochgezogen und das Licht auf eine

---

<sup>41</sup> Zum Beispiel: Temperatur, Luftfeuchtigkeit, Windstärke und -richtung

für den Bewohner angenehme Aufwachhelligkeit eingestellt. Nun muss das Heim- und Gebäudeautomationssystem darauf achten, dass der Bewohner nicht verschläft. Im Bett sind Drucksensoren eingearbeitet, die eine Bettbelegung feststellen. Falls die Person nicht wach werden will und aus dem Bett steigt, greift das Heim- und Gebäudeautomationssystem nach circa 10 Minuten ab der Weckuhrzeit zu drastischeren Mitteln. Mit lauterer Musik und Ein- und Ausschalten von Licht wird der Bewohner auch bei starker Müdigkeit geweckt. Er steht schließlich auf und begibt sich ins Bad, um mit der Morgentoilette zu beginnen.

Während der Bewohner im Bad ist, werden ab circa 6:50 Uhr in der Küche im Heißluftrohr schon die Brötchen aufgebacken und die espressomaschine aktiviert. So wird sichergestellt, dass um 7:00 Uhr, wenn der Bewohner zum Frühstück kommt, die Brötchen schön knusprig sind, genauso wie sie der Bewohner gerne hat, und die Kaffeemaschine aufgeheizt ist.

Hat sich der Bewohner um 7 Uhr in die beleuchtete Küche begeben, schaltet das Heim- und Gebäudeautomationssystem das Radio vollautomatisch auf den gewünschten Sender und auf die bevorzugte Lautstärke. Gleichzeitig wird das Licht im Schlafzimmer und im Bad abgeschaltet. Zwischen 7:00 Uhr und 7:30 Uhr versorgt das Heim- und Gebäudeautomationssystem den Bewohner über sein iPad mit den aktuellen Nachrichten. Gleichzeitig wird der Bewohner über Verkehrsnachrichten informiert, die den Weg mit dem Auto von zu Hause zur Arbeit betreffen. Um 7:25 Uhr macht das Heim- und Gebäudeautomationssystem den Bewohner über eine Nachricht auf dem iPad darauf aufmerksam, dass es Zeit zur Abfahrt ist und schaltet die Beleuchtung zur Garderobe sowie zur Garage ein.

Der Bewohner zieht sich an und begibt sich in die Garage und dann ins Auto. Noch während der Bewohner einsteigt, öffnet das Heim- und Gebäudeautomationssystem automatisch das Garagentor. Nachdem das Auto aus der Garage gefahren ist, wird das Garagentor geschlossen, die Alarmanlage scharf gestellt, alle Lichter ausgeschaltet und die Raumtemperatur auf Eco-Level gebracht.

### 3.3.2 Modellanalyse mit UML

Das sprachlich beschriebene Szenario wird nun mit Hilfe eines Anwendungsfalldiagramms modelliert. Auf den Teilaspekt Personenerkennung wird im Sequenzdiagramm eingegangen.

3.3.2.1 Anwendungsfall

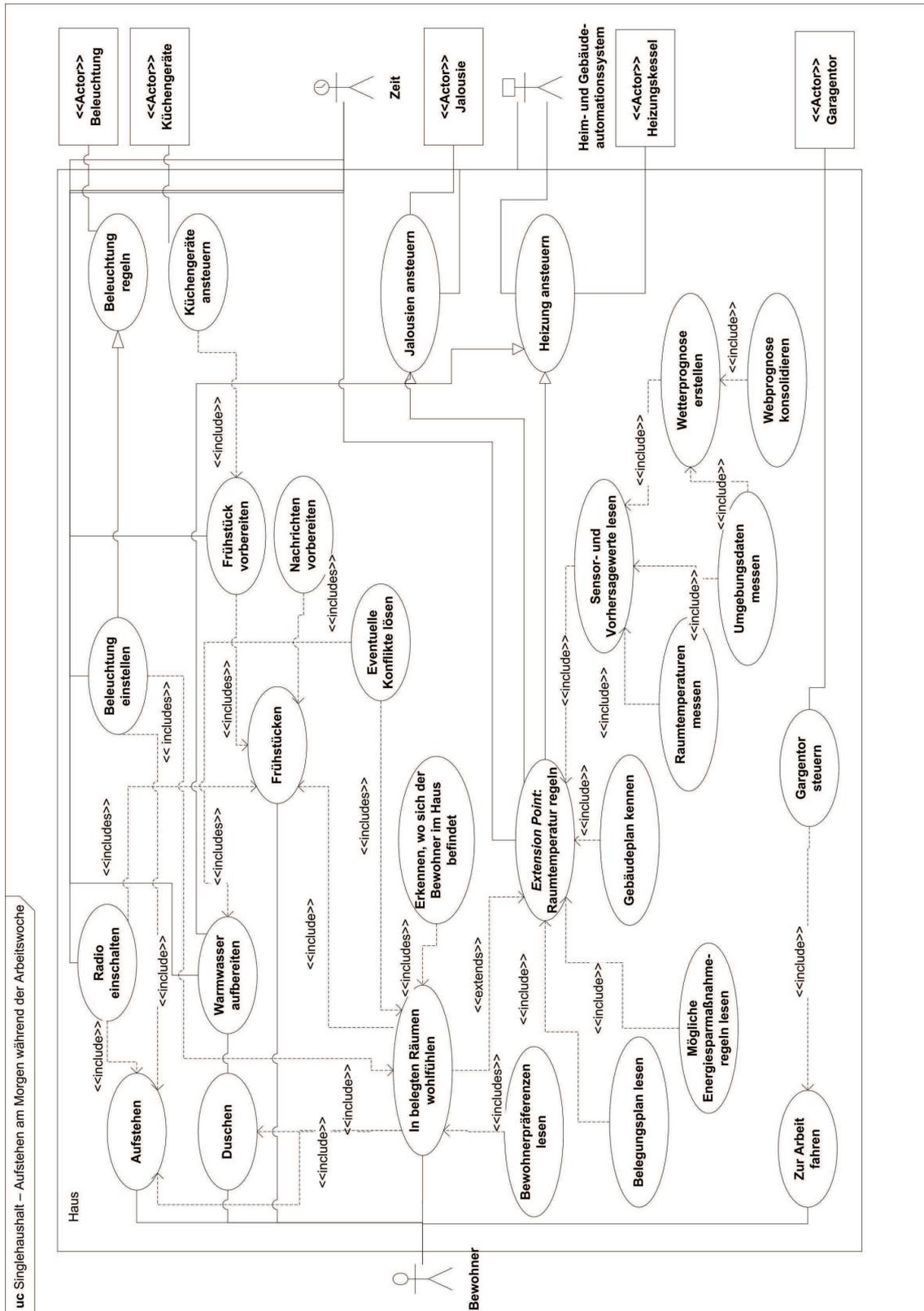


Abbildung 20: Anwendungsfall zu Aufstehen am Morgen während der Arbeitswoche

## 3.3.2.1.1 Beschreibung

<b>Name des Anwendungsfalls</b>	<b>Singlehaushalt: Aufstehen am Morgen während der Arbeitswoche</b>
<b>Ziel</b>	Energiesparsames Regeln der Raumtemperatur nach einem Belegungsplan, energiesparsames Bereitstellen des Warmwassers, Wecken des Bewohners inklusive Aufstehkontrolle, Verschattungsregelung nach einem Belegungsplan und Umgebungsbedingungen, Beleuchtungsregelung nach Belegungsplan und Umgebungsbedingungen, Steuern von Audiogeräten, Informationsbereitstellung, Steuern von Küchengeräten für die Frühstückszubereitung, Steuern des automatischen Garagentors.
<b>Kategorie</b>	Primärer Anwendungsfall
<b>Vorbedingung</b>	Bekannter Raumbelegungsplan, bekanntes Gebäudemodell, Raumtemperatur, Außentemperatur, Windgeschwindigkeit, Sonneneinstrahlung, Wettervorhersagedaten aus dem Internet, bekannte Duschwassertemperatur und Brauchwassermenge, bekannter Tagesplan des Bewohners, Essensvorzüge, bekannte Informationsinteressen, bekannte Fahrroute vom Haus zum Arbeitsplatz,
<b>Nachbedingung</b>	Für jede Aktivität richtig eingestellte Raumtemperatur nach dem Belegungsplan mit möglichst geringem Heizungsenergieverbrauch, Warmwasser mit bevorzugter Temperatur und nötiger Wassermenge aufgeheizt, Bewohner wach, Räume nach Belegungsplan energiesparsam beleuchtet, Frühstück zubereitet, auf iPad Tageszeitung und Verkehrsnachrichten für den Weg von zu Hause zum Arbeitsplatz geladen, Audioanlage nach Wohnervorstellungen gesteuert, Garagentor nach Ausfahrt aus der Garage geschlossen, Bewohner auf dem Weg zur Arbeit
<b>Akteure</b>	Bewohner, Heim- und Gebäudeautomationssystem
<b>Auslösendes Ereignis</b>	Uhrzeit, Aktivitäten des Bewohners
<b>Beschreibung</b>	Das Heim- und Gebäudeautomationssystem misst jede Minute Raumtemperatur, Außentemperatur, Windgeschwindigkeit, Sonneneinstrahlung und Wettervorhersagedaten. Außerdem kennt das Heim- und Gebäudeautomationssystem die Gebäudestruktur aus dem Gebäudemodell. In Verbindung mit dem bereits gelernten Raumbelegungsplan weiß das Heim- und Gebäudeautomationssystem, wann ein Raum auf Eco-Temperatur oder auf Komforttemperatur geheizt sein

muss. Das Heim- und Gebäudeautomationssystem kann mit Hilfe der Verschattungsanlage solare Wärmeenergie in die verwendeten Räume lassen und bei Bedarf fehlende Wärmeenergie mit dem Heizungskessel aus Brennstoffen erzeugen.

Außerdem weiß das Heim- und Gebäudeautomationssystem, wann Warmwasser benötigt wird und wie groß die Heizleistung des Heizungskessels ist und wann die Heizleistung für Raumheizung benötigt wird. Das Heim- und Gebäudeautomationssystem heizt Brauchwasser erst dann, wenn es unter einer bestimmten Boilertemperatur gesunken ist und nur dann, wenn der Heizkessel nicht für die Raumheizung benötigt wird.

Durch den Raumbelegungsplan und den Tagesablauf des Bewohners weiß das Heim- und Gebäudeautomationssystem, wann welcher Raum wie beleuchtet werden muss.

Durch die Kenntnis der Vorzüge des Bewohners, für Weckzeit, Audiopräferenzen, Strategie für effektives Wecken, Aktivitäten des Bewohners, Warmwassertemperatur, Speisen und Getränke, Nachrichtenpräferenzen und die Kenntnisse über den Arbeitsweg kann das Heim- und Gebäudeautomationssystem gezielt die Bedürfnisse des Bewohners erfüllen.

Tabelle 6: Beschreibung des Anwendungsfalls - Singlehaushalt: Aufstehen am Morgen während der Arbeitswoche

## 3.3.2.2 Sequenzdiagramm

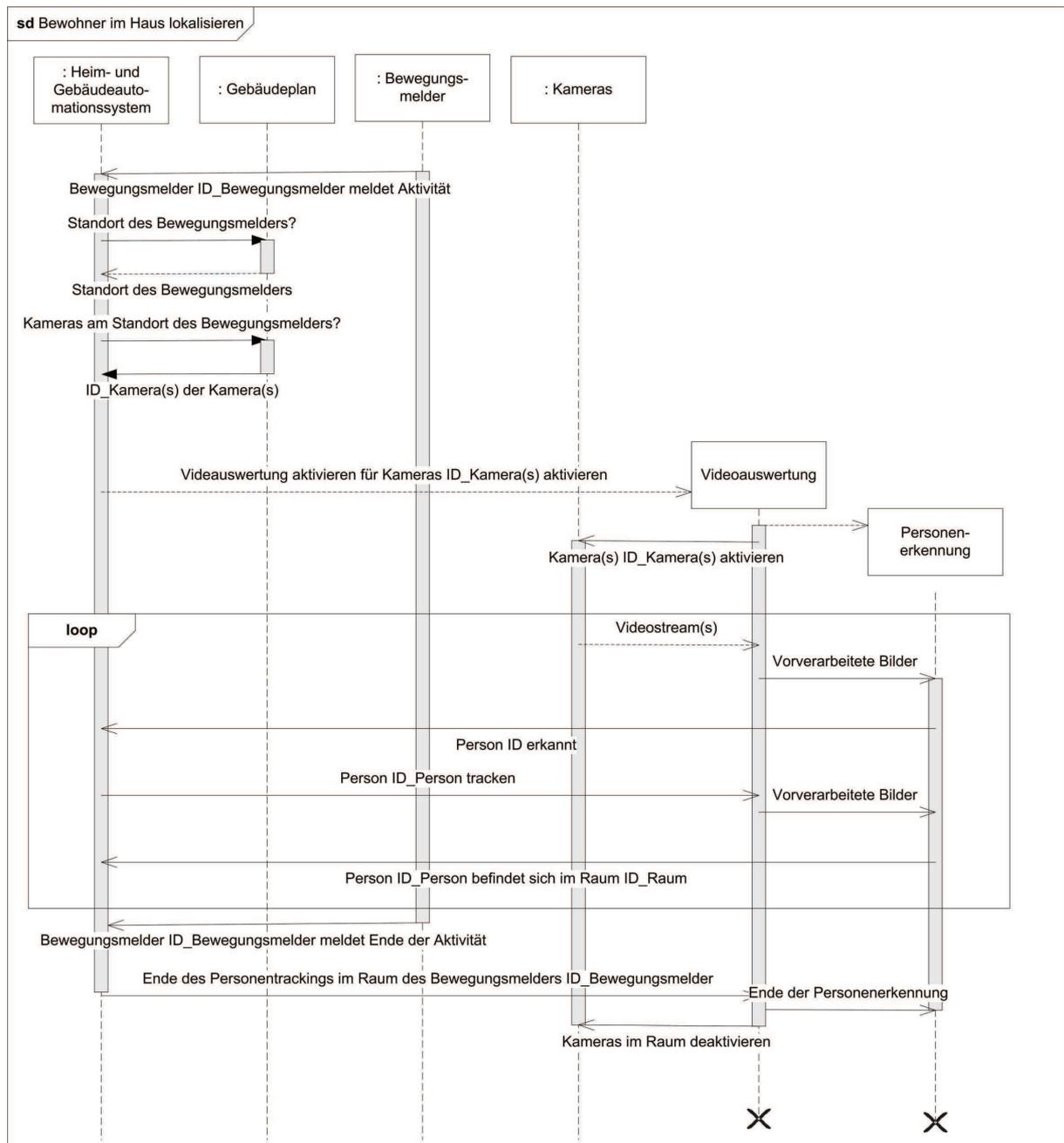


Abbildung 21: Sequenzdiagramm zur Lokalisierung der Bewohner im Haus

Ausgelöst wird das Sequenzdiagramm durch eine Nachricht eines Bewegungsmelders, der ein Ereignis detektiert hat. In der Nachricht befindet sich nur die eindeutige Erkennungsnummer des Bewegungsmelders. Durch eine Abfrage des Gebäudeplans durch das Heim- und Gebäudeautomationssystem wird dem Bewegungsmelder ein Standort zugeordnet. Welche Kameras für die Objekterkennung und Objekttracking aktiviert werden müssen, wird ebenfalls dem Gebäudeplan entnommen.

Nun wird die Videoüberwachung für den jeweiligen Raum eingeschaltet. Wenig später wird die Personenerkennung zugeschaltet. Für Datenmaterial sorgen die Kameras, die von der Videoauswertung aktiviert werden. Die Videoverarbeitung stellt nun grob fest, um welchen Typ von Objekt es sich handelt. Wird dabei ein Mensch erkannt, dann werden die

gewonnen vorverarbeiteten Daten der Personenerkennung übergeben. Diese erkennt bekannte Personen mit Hilfe gespeicherten biometrischer Daten. In dem gezeigten Fall erfolgt das mit Gesichtserkennungsmerkmalen.

Wurde eine Person erfolgreich erkannt, wird dem Heim- und Gebäudeautomationssystem mitgeteilt, welche Person sich im besagten Raum und wo genau im Raum befindet. Die Person wird weiter beobachtet und jede Standortänderung dem Heim- und Gebäudeautomationssystem mitgeteilt. Verlässt eine Person einen Raum, dann meldet der Präsenzmelder keine Aktivität mehr, wodurch die Videoauswertung gestoppt wird, die Kameras deaktiviert werden und somit auch die Personenerkennung und -tracking ausgeschaltet werden.

## 4 Optimierte Kontrollstrategien zur Heizungsansteuerung

In diesem Kapitel werden ausgewählte algorithmische Ansätze präsentiert, die eine energiesparende Ansteuerung der Heizung ermöglichen und verschiedene Umgebungsvariablen berücksichtigen.

Will man einen Raum von einer vorliegenden niedrigeren Raumtemperatur auf eine gewünschte höhere Raumtemperatur heizen, dauert das in Abhängigkeit einer Vielzahl von Einflussfaktoren<sup>42</sup> eine bestimmte Zeitspanne. Zu diesen Einflussfaktoren zählen:

- Leistung der Heizung,
- verwendete Baumaterialien des Hauses,
- aktuelle Kerntemperaturen der Baumaterialien des Hauses,
- Ausrichtung des Hauses und somit auch gleichzeitig die Ausrichtung der Räume nach den Himmelsrichtungen in Abhängigkeit der Anordnung und Größe im Haus,
- Möbeln im Raum,
- Personen im Raum,
- aktiven elektrische Geräten in einem Raum,
- Temperaturübertragung der Räume untereinander,
- aktuelle Außenlufttemperatur,
- aktuelle Windverhältnisse,
- aktuelle Sonneneinstrahlung,
- Wetterprognose für die kommenden Stunden des aktuellen Tages,
- Wettertrend für die kommenden Tage.

Diese Zeit wird im nachfolgenden Text als Vorheizzeit bezeichnet. Die Vorheizzeit ist jene Zeit, die ein installiertes Heizungssystem benötigt, einen bestimmten Raum von einem vorliegenden niedrigeren Temperaturniveau unter der Einwirkung verschiedener Einflussfaktoren auf ein gewünschtes höheres Temperaturniveau zu bringen.

Zusätzlich zu den Bemühungen durch die Automationstechnik sei an dieser Stelle erwähnt, dass durch die europäischen Gesetzgeber - ebenfalls um die Energiekosten bei der Heizung, Lüftung und Kühlung in Wohn- und Nutzbauten zu senken - bei allen Neubauten und sanierten Objekten sowie bei Zu- und Umbauten ein Energieausweis [26] beim behördlichen Bauverfahren gefordert wird. Seit 2009 gibt es auch ein zusätzliches Gesetz in Österreich, dass bei Verkauf, Verpachtung oder Vermietung von Häusern, Wohnungen, Büros oder Betriebsobjekten ein Energieausweis vom Bauherrn, Vermieter oder Verkäufer erforderlich ist. Der Energieausweis ist ein weiterer Ansatz, der schon beim Bau der Wohn- oder Nutzbauten durch den Gesetzgeber vorgeschrieben wird, um Energiekosten im Betrieb der Heizungs-, Lüftungs- oder Kälteanlage zu sparen<sup>43</sup>.

In dieser Arbeit werden nur jene Einflussfaktoren verwendet, die bei einem vorhandenen Einfamilienwohnhaus anfallen und bei denen die größte Systembeeinflussung stattfindet. Eine große Anzahl an Einflussfaktoren, wie sie in der vorangegangenen Auflistung gezeigt wurden, wie zum Beispiel die aktuelle Kerntemperatur der Hülle und der Zwischenwände

---

<sup>42</sup> Die Auflistung der Einflussfaktoren für die Aufheizzeit konzentriert sich, ohne Anspruch auf Vollständigkeit, auf die augenscheinlichsten Größen.

<sup>43</sup> Einer der Hintergründe solcher gesetzlichen Vorschriften sind die CO<sub>2</sub>-Emissionen, deren Erreichen sich ein Land im Kyoto-Protokoll [63] verpflichtend bis zum Jahr 2012 vorgenommen hat.

des Hauses oder die Abgabe der Wärmeenergie durch elektrische Verbraucher in einem Raum, können nicht direkt und auch nur schwer gemessen werden. Aus diesem Grund werden in dieser Arbeit nur jene physikalisch mit einfachen technischen Mitteln erfassbaren Einflussgrößen, wie in Tabelle 7 dargestellt, in Betracht gezogen. Es wird aus Gründen der Vereinfachung des Modells angenommen, dass sich die Räume in einem Wohnhaus gleich verhalten. Die Leistung der Heizungsanlage, die Raumgrößen und die Bausubstanz werden in dieser Arbeit als gegeben angesehen und wirken sich indirekt auf die Dauer der Vorheizzeit aus.

$T_{ist R_i}$	Aktuelle vorliegende Lufttemperatur im Raum i
$T_{soll R_i t_j}$	Solltemperatur im Raum i zum Zeitpunkt j
$T_{ist außen}$	Aktuell gemessene Lufttemperatur im Außenbereich
$T_{außen t_j}$	Prognostizierte Außentemperatur zum Zeitpunkt j
$t_{Vorheizzeit}$	Zeitdauer, um einen Raum von einem Ist-Temperaturniveau mit geeigneten technischen Mitteln auf ein gewünschtes Soll-Temperaturniveau zu bringen

Tabelle 7: Betrachtete physikalische Messgrößen

Die Temperaturregelung eines Hauses ist in Abhängigkeit der Einflussfaktoren ein stochastischer Prozess. Das heißt, dass die beeinflussenden Faktoren zwar exakt gemessen werden können, jedoch mit einem durch den Zufall hervorgerufenen Rauschen behaftet sind. Die Vorgehensweise der hier vorgestellten Methoden teilt sich in 2 voneinander unterscheidbaren Phasen. Zum einen wird in der ersten Phase das Verhalten der Temperaturregelung unter Einbehaltung der verschiedenen Einflussfaktoren „gelernt“. In der zweiten Phase wird das in der ersten Phase „gelernte“ Wissen über die Physik angewendet. Die betrachteten physikalischen Messgrößen verhalten sich stark nichtlinear im Verhältnis zur prognostizierenden Aufheizzeit, wie im Kapitel 5 zu sehen sein wird.

Zur Schätzung der Vorheizzeit bieten sich verschiedene mathematische Werkzeuge an. Ein klassisches mathematisches Modell, das an dieser Stelle für die Schätzung der Vorheizzeit zur Hand genommen wird, kommt aus dem Bereich der Statistik. Es wird gezeigt, wie die Einflussgrößen mit Hilfe der multiplen Regression auf die Vorheizzeit schließen lassen. In dieser Arbeit wird nur ein kurzer Überblick geboten, bevor die Schwerpunkte dieser Arbeit - künstlich neuronale Netze - beschrieben werden.

In dem Abschnitt über künstlich neuronale Netze wird zunächst darauf eingegangen, warum diese Methode für die Schätzung der Vorheizzeit verwendet werden kann und zwar speziell in Hinblick auf Einflussgrößen. Anschließend wird erklärt, wie künstlich neuronale Netze für diese Anwendung topologisch aufgebaut sein können und welche Lernalgorithmen in Frage kommen. Zum Schluss des Kapitels wird die in dieser Arbeit verwendete Modellierungs- und Simulationssoftware MemBrain kurz vorgestellt.

## 4.1 Multiple Regression als Vorheizzeitschätzer

### 4.1.1 Einführung

Die Regression [27] schafft einen mathematischen Zusammenhang zwischen verschiedenen quantitativ erfassbaren Merkmalen, zum Beispiel in der Domäne der Physik, Chemie oder Sozialwissenschaften herzustellen. Aus bekannten stochastischen x-Werten, die die Eingabe des Modells darstellen, wird als Ausgabe des Modells ein prognostizierter y-Wert ermittelt. Die Regressionsanalyse basiert auf den Arbeiten von Francis Galton [28] und Karl Pearson [29].

Die x-Merkmale für die Regression bilden die in Tabelle 7 abgebildeten Einflussmerkmale. Das abhängige y-Merkmal ist die Vorheizzeit.

Die Entscheidung, welche Regressionsform angewendet werden soll, hängt von der Anzahl der Einflussmerkmale und von der Streuungsform der Merkmalsausprägungen ab. Man unterscheidet zwischen linearer Regression mit nur einem x-Merkmal, multipler linearer Regression mit mehr als einem x-Merkmal und jeweils ihren nichtlinearen Formen.

Dieser Abschnitt zeigt eine Skizze, eines multiplen nichtlinearen Regressionsmodells zur Prognose der Vorheizzeit. Zuerst wird anhand des linearen Modells mit nur einem Eingabemerkmal gezeigt, welche Phasen mit dem Regressionsmodell durchlaufen werden, um die Vorheizzeit zu prognostizieren und welche Formeln benötigt werden.

Bei den in der Lernphase verwendeten x-Merkmalen und y-Merkmalen handelt es sich um Stichproben, die zuvor aus Experimenten ermittelt wurden.

### 4.1.2 Mathematik der Stichprobenregressionsanalyse

Die Stichprobenregressionsanalyse versucht einen möglichst genauen Zusammenhang zwischen Stichprobendaten zu finden. An dieser Stelle wird die bivariate lineare Regression mit einer Variable beschrieben und anschließend die multiple Regression mit mehreren Variablen<sup>44</sup>.

#### 4.1.2.1 Bivariate lineare Regression

Die lineare Regression geht davon aus, dass zwischen zwei Variablen  $x_i$  und  $y_i$  ein Zusammenhang wie Formel 3 besteht.

$$y_i = b_0 + b_1 * x_i + e_i$$

Formel 3: Lineare Stichprobenregressionsfunktion mit Stör-Term

Dabei ist  $y_i$  die zu schätzende Variable basierend auf  $x_i$ , der beeinflussenden Variable. Alle nichtmessbaren Einflüsse auf die Gleichung fließen in den nichtdeterministischen Term<sup>45</sup>  $e_i$  ein. Die Regressionsanalyse legt eine Gerade durch die Punktwolke, die durch  $y_i$  und  $x_i$  repräsentiert wird. Dabei ist  $b_0$  der y-Achsenabschnitt und  $b_1$  die Steigung der

---

<sup>44</sup> Beide Verfahren gehen davon aus, dass durch die 2-dimensionale Punktwolke eine Gerade oder einer höherdimensionalen Punktwolke eine zwei- oder mehrdimensionale Fläche gelegt werden kann.

<sup>45</sup> Dieser Term wird im Folgenden nicht weiter berücksichtigt.

Regressionsgerade. Die durch die von  $x_i$  erklärten  $y_i$  Werte liegen in optimaler Weise exakt oder sehr nahe auf der Regressionsgerade. Die durch den Schätzer prognostizierten Werte liegen auf der Regressionsgeraden und werden mit  $\hat{y}_i$  bezeichnet. Es wird nun versucht die Abstände zwischen den Variablen  $y_i$  und den Variablen  $\hat{y}_i$  zu minimieren. Oder mit anderen Worten, die Abweichung  $(y_i - \hat{y}_i)$  sollte möglichst klein sein, um möglichst passende Werte für  $b_0$  und  $b_1$  zu finden. Zu diesem Zweck bedient man sich der Methode der kleinsten Quadrate (siehe Formel 4).

$$f(b_0, b_1) = \sum_{i=1}^N (y_i - \hat{y}_i)^2 = \sum_{i=1}^N (y_i - b_0 - b_1 x_i)^2$$

Formel 4: Minimieren der Summe der kleinsten quadratischen Abstände

Das Minimum der Formel 4 erhält man indem man die erste Ableitung bildet und 0 setzt sowie die zweite Ableitung der Funktion  $f(b_0, b_1)$  bildet und dann überprüft, ob die Ableitungen größer 0 sind, wie dies in nachfolgenden Formeln gezeigt wird.

$$\frac{\partial f}{\partial b_0} = -2 \sum_{i=1}^N (y_i - b_0 - b_1 x_i) = -2N(\bar{y} - b_0 - b_1 \bar{x}) = 0$$

Formel 5: Erste Ableitung der vorangegangenen Formel nach  $b_0$

$$\frac{\partial f}{\partial b_1} = -2 \sum_{i=1}^N x_i (y_i - b_0 - b_1 x_i) = -2b_1 \sum_{i=1}^N x_i^2 - 2 \sum_{i=1}^N x_i y_i + 2b_0 N\bar{x} = 0$$

Formel 6: Erste Ableitung der vorangegangenen Formel nach  $b_1$

Aus Formel 5 kann der y-Achsenschnitt berechnet werden. Dies folgt in Formel 7.

$$b_0 = \bar{y} - b_1 \bar{x}$$

Formel 7: y-Achsenschnitt der Regressionsgeraden

Setzt man Formel 7 in Formel 6 ein und löst diese nach  $b_1$  auf, erhält man die Formel für die Steigung in Formel 8 aus der Kovarianz aus  $y$  und  $x$  im Verhältnis zur Varianz von  $x$ .

$$b_1 = \frac{\sum_{i=1}^N x_i y_i - N\bar{x}\bar{y}}{\sum_{i=1}^N x_i^2 - N\bar{x}^2} = \frac{\frac{1}{N} \sum_{i=1}^N (x_i - \bar{x})(y_i - \bar{y})}{\frac{1}{N} \sum_{i=1}^N (x_i - \bar{x})^2} = \frac{Cov(y, x)}{Var(x)}$$

Formel 8: Berechnung der Steigung der Regressionsgeraden

In Formel 9 und Formel 10 wird überprüft, ob ein Minimum der Funktion  $f(b_0, b_1)$  vorliegt.

$$\frac{\partial^2 f}{\partial b_0^2} = 2N > 0$$

Formel 9: Zweite Ableitung der Formel  $f(b_0, b_1)$  nach  $b_0^2$

$$\frac{\partial^2 f}{\partial b_1^2} = 2 \sum_{i=1}^N x_i^2 > 0$$

Formel 10: Zweite Ableitung der Formel  $f(b_0, b_1)$  nach  $b_1^2$

Beide berechneten Ausdrücke für  $b_0$  und  $b_1$  sind Minima der Funktion  $f(b_0, b_1)$  und damit optimale Parameter für die Regressionsgerade.

Die Güte der Regressionsgerade, also wie gut die Gerade in die Punktwolke der Messwerte passt, kann aus der Varianzzerlegung hergeleitet werden.  $y_i$  setzt sich, wie schon zuvor angesprochen, aus  $\hat{y}$  und einem stochastischen Teil  $e$  - den Stichprobenresiduen - zusammen, wie in Formel 11 gezeigt.

$$y_i = \hat{y}_i + e_i$$

Formel 11: Aufspaltung von  $y_i$  in einen „erklärten Teil“ und in einen „unerklärten Teil“

Die Varianz von  $y$  ist Formel 12 dargestellt.

$$Var(y) = Var(\hat{y}) + Var(e) + 2Cov(\hat{y}, e)$$

Formel 12: Varianz von  $y$ , erklärt durch die Varianz des Schätzwertes von  $y$  und der Varianz des stochastischen Fehlers von  $y$  und der Kovarianz des Schätzwertes von  $y$  und des systematischen Fehlers

Da die Kovarianz zwischen den geschätzten Werten  $\hat{y}$  und den Stichprobenresiduen  $e$  gleich 0 ist, fällt der letzte Term Formel 12 weg. Das ergibt die Varianz von  $y$  nach Formel 13.

$$Var(y) = Var(\hat{y}) + Var(e)$$

Formel 13: Varianz von  $y$  für die lineare Regression

Die Gesamtstreuung von  $y$  der linearen Regression berechnet sich aus Formel 14.

$$\frac{1}{N} \sum (y_i - \bar{y})^2 = \frac{1}{N} \sum (\hat{y}_i - \bar{y})^2 + \frac{1}{N} \sum (e_i - \bar{e})^2$$

Formel 14: Gesamtstreuung von y in der linearen Regression

Formel 14 lässt sich in Formel 15 vereinfachen, da  $\bar{e} = 0$  ist.

$$\sum (y_i - \bar{y})^2 = \sum (\hat{y}_i - \bar{y})^2 + \sum e_i^2 \Rightarrow TSS = ESS + SSR$$

Formel 15: Vereinfachte Formel der Gesamtstreuung von y in der linearen Regression

TSS steht für „Total Sum Squared“, ESS für „Explained Sum Squared“ und SSR für „Sum of Squared Residuals“. Das Bestimmtheitsmaß R ist definiert durch den Anteil der erklärten Streuung ESS zur gesamten Streuung TSS, wie in Formel 16 zu sehen ist.

$$R^2 = \frac{ESS}{TSS} = 1 - \frac{SSR}{TSS} = 1 - \frac{\sum (y_i - \hat{y}_i)^2}{\sum (y_i - \bar{y})^2} = 1 - \frac{\sum e_i^2}{\sum (y_i - \bar{y})^2}$$

Formel 16: Bestimmtheitsmaß R der linearen Regression

Der Wert von  $R^2$  liegt immer zwischen 0 und 1. Je näher das Bestimmtheitsmaß bei 1 liegt, desto besser ist der Zusammenhang zwischen der zu erklärenden Variable und der beeinflussenden Variable. Jedoch sollte der Zusammenhang auch mit einem Blick auf die Punktwolke überprüft werden.

#### 4.1.2.1.1 Beispiel bivariate lineare Regression für das Aufheizen eines Raumes

##### 4.1.2.1.1.1 Lernphase

Angenommen zwischen den 2 Variablen  $x$  und  $y$  besteht ein linearer Zusammenhang. Die Variable  $x$  repräsentiert die aktuelle Raumtemperatur  $T_{\text{Ist}}$  in Grad Celsius und  $y$  die Aufheizzeit in Minuten auf eine gewünschte Raumtemperatur  $T_{\text{Soll}}$  von  $22^\circ\text{C}$ . In Tabelle 8 sind die gemessenen Zusammenhänge abgebildet, aus denen eine Regressionsgerade, die durch die Parameter  $b_0$  und  $b_1$ , durch die Variablen  $x$  und  $y$  beschrieben wird, hergeleitet werden soll.

Beobachtung i	$x:=T_{\text{Ist}} [^\circ\text{C}]$	$y:=t_{\text{Aufheiz}} [\text{Minuten}]$
1	15	64
2	21	7
3	19	25
4	17,5	38
5	20,5	5
6	18	30
7	16,5	44
8	21,5	6
9	19,5	12
10	18,5	35

Tabelle 8: Beispiel lineare Regression - Ermitteln der Regressionsgerade

In der Tabelle 9 sind die Zwischenrechnungsschritte für die Berechnung von  $b_0$  in Formel 18 und  $b_1$  in Formel 17 abgebildet.

Beobachtung i	$x:=T_{\text{Ist}} [^\circ\text{C}]$	$y:=t_{\text{Aufheiz}} [\text{Minuten}]$	$x*y$	$x^2$
1	15	64	960	225
2	21	7	147	441
3	19	25	475	361
4	17,5	38	665	306,25
5	20,5	5	102,5	420,25
6	18	30	540	324
7	16,5	44	726	272,25
8	21,5	6	129	462,25
9	19,5	12	234	380,25
10	18,5	35	647,5	342,25
$\Sigma$	187	266	4626	3534,5

Tabelle 9: Zwischenrechnung - Beispiel lineare Regression

$$\underline{\underline{b_1}} = \frac{N \sum y_i x_i - \sum x_i \sum y_i}{N \sum x_i^2 - (\sum x_i)^2} = \frac{10 * 4626 - 187 * 266}{5 * 3534,5 - (187)^2} = \underline{\underline{-9,26}}$$

Formel 17: Steigungsparameter - Beispiel lineare Regression

$$\underline{\underline{b_0}} = \bar{y} - b_1 \bar{x} = \frac{266}{10} - (-9,26) * \frac{187}{10} = \underline{\underline{199,77}}$$

Formel 18: y-Achsen Schnittpunkt - Beispiel lineare Regression

Damit ergibt sich eine Regressionsgerade nach Formel 19.

$$y = 199,77 - 9,26x$$

Formel 19: Regressionsgerade für das Beispiel lineare Regression

Die Güte  $R^2$  der Regression in diesem Beispiel errechnet sich aus Formel 16 auf 0,947. In Abbildung 22 sieht man die Stichprobenwerte mit der eingezeichneten Regressionsgerade.

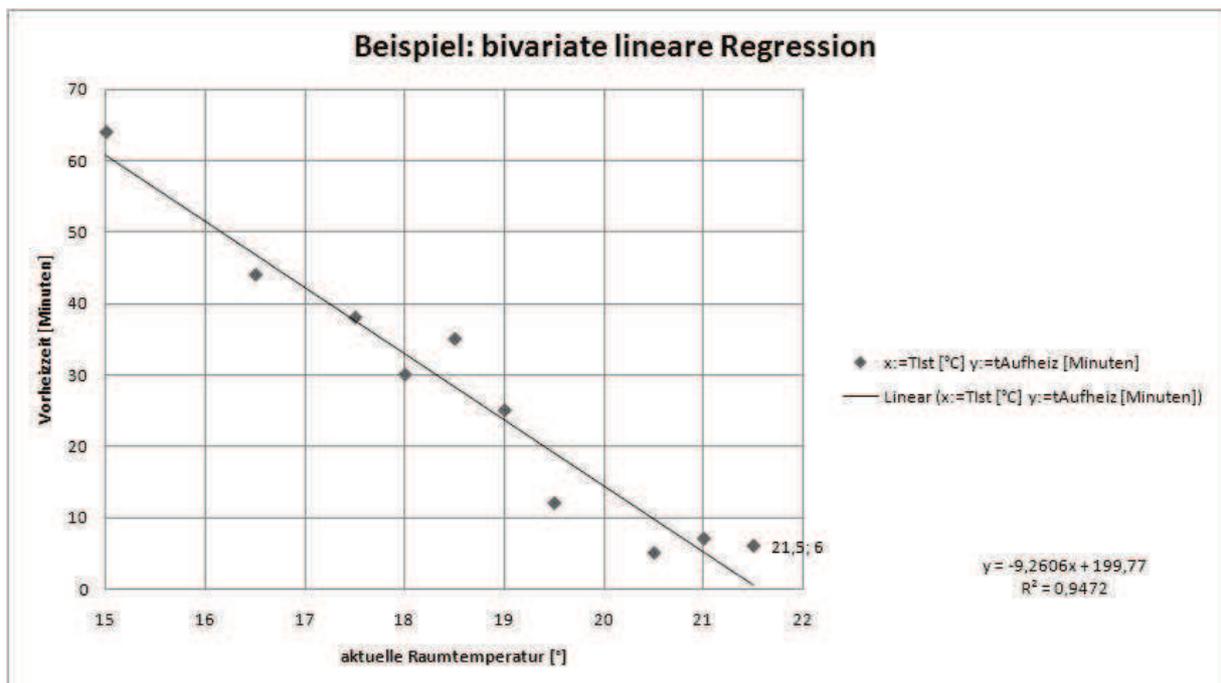


Abbildung 22: Punktvolke und Regressionsgerade für das Beispiel bivariate lineare Regression

#### 4.1.2.1.1.2 Operationale Phase

Um die operationale Phase zu illustrieren, nehmen wir an dieser Stelle eine gemessene Raumtemperatur von 19°C an. Der Raum soll auf 22°C aufgeheizt werden. Um einen Vorhersagewert für die Vorheizzeit zu bekommen, wird Formel 19 verwendet.

$$\underline{y} = 199,77 - 9,26 * 19 = \underline{\underline{23,83}}$$

Formel 20: Beispiel Vorhersage der Vorheizzeit

Somit erhält man eine auf ganze Minuten gerundete Vorheizzeit von 24 Minuten. Das heißt, das Heizungssystem muss 24 Minuten vor der Benützung des gewünschten Raumes mit der Erwärmung des Raumes beginnen.

#### 4.1.2.2 Multiple Regression

Bei physikalischen Vorgängen haben meist mehrere erklärende Variablen Einfluss auf die abhängige Variable. An dieser Stelle wird ein multiples Regressionsmodell vorgestellt, das mit zwei erklärenden Variablen arbeitet.

Das bisherige Modell wird, wie in Formel 21 gezeigt, erweitert.

$$y_i = b_0 + b_1 x_{1i} + b_2 x_{2i} + e_i$$

Formel 21: Multiples Regressionsmodell

Um die Parameter  $b_0$ ,  $b_1$  und  $b_2$  zu berechnen, wird die Summe der Abstandquadrate, wie in Formel 22 gezeigt, minimiert.

$$\min_{b_0, b_1, b_2} \sum e_i^2 = \sum (y_i - b_0 - b_1 x_{1i} - b_2 x_{2i})^2$$

Formel 22: Minimieren der Abstandskvadrat

Nun werden die erste Ableitungen der Formel 22 gebildet, wie in Formel 23, Formel 24 und Formel 25 ersichtlich.

$$\frac{\partial \sum e_i^2}{\partial b_0} = 2 \sum (y_i - b_0 - b_1 x_{1i} - b_2 x_{2i})(-1) = 0$$

Formel 23: Erste Ableitung nach  $b_0$  der obigen Minimierungsgleichung

$$\frac{\partial \sum e_i^2}{\partial b_1} = 2 \sum (y_i - b_0 - b_1 x_{1i} - b_2 x_{2i})(-x_{1i}) = 0$$

Formel 24: Erste Ableitung nach  $b_1$  der obigen Minimierungsgleichung

$$\frac{\partial \sum e_i^2}{\partial b_2} = 2 \sum (y_i - b_0 - b_1 x_{1i} - b_2 x_{2i})(-x_{2i}) = 0$$

Formel 25: Erste Ableitung nach  $b_2$  der obigen Minimierungsgleichung

Somit erhält man durch einige Umformarbeit die Lösung für die Parameter  $b_0$ ,  $b_1$  und  $b_2$  (Formel 26, Formel 27 und Formel 28).

$$b_0 = \bar{y} - b_1 \bar{x}_1 - b_2 \bar{x}_2$$

Formel 26: Berechnung des Parameters  $b_0$

$$b_1 = \frac{(\sum \ddot{y}_i \ddot{x}_{1i})(\sum \ddot{x}_{2i}^2) - (\sum \ddot{y}_i \ddot{x}_{2i})(\sum \ddot{x}_{1i} \ddot{x}_{2i})}{(\sum \ddot{x}_{1i}^2)(\sum \ddot{x}_{2i}^2) - (\sum \ddot{x}_{1i} \ddot{x}_{2i})^2}$$

Formel 27: Berechnung des Parameters  $b_1$ <sup>46</sup>

$$b_2 = \frac{(\sum \ddot{y}_i \ddot{x}_{2i})(\sum \ddot{x}_{1i}^2) - (\sum \ddot{y}_i \ddot{x}_{1i})(\sum \ddot{x}_{1i} \ddot{x}_{2i})}{(\sum \ddot{x}_{1i}^2)(\sum \ddot{x}_{2i}^2) - (\sum \ddot{x}_{1i} \ddot{x}_{2i})^2}$$

Formel 28: Berechnung des Parameters  $b_2$

#### 4.1.2.2.1 Beispiel multiple lineare Regression für das Aufheizen eines Raumes<sup>47</sup>

##### 4.1.2.2.1.1 Lernphase

Angenommen zwischen den 3 Variablen  $x$ ,  $y$  und  $z$  bestehe ein linearer Zusammenhang. Die Variable  $x$  repräsentiert die aktuelle Raumtemperatur  $T_{\text{Ist}}$  in Grad Celsius, die Variable  $y$  repräsentiert die aktuelle Außentemperatur  $T_{\text{außen,ist}}$  in Grad Celsius und die Variable  $z$  die Aufheizzeit in Minuten auf eine gewünschte Raumtemperatur  $T_{\text{Soll}}$  von 22°C. In Tabelle 10 sind die gemessenen Zusammenhänge abgebildet, aus denen eine Regressionsfläche, die durch die Parameter  $b_0$ ,  $b_1$  und  $b_2$  sowie den Variablen  $x$ ,  $y$  und  $z$  beschrieben wird, hergeleitet werden soll.

Beobachtung i	$x_1 := T_{\text{Ist}} [^\circ\text{C}]$	$x_2 := T_{\text{Außen}} [^\circ\text{C}]$	$y := t_{\text{Aufheiz}} [\text{Minuten}]$
1	15	0	64
2	16,5	5	44
3	17,5	10	38
4	18	16	30
5	18,5	14	35
6	19	12	25
7	19,5	15	12
8	20,5	16	5
9	21	12	7
10	21,5	12	6

Tabelle 10: Beispiel multiple Regression - Ermitteln der Regressionsfläche

<sup>46</sup>  $\ddot{x} = x_i - \bar{x}$

<sup>47</sup> Für die Darstellung der Punktwolke wurde die Statistiksoftware R in Verbindung mit MS Excel und RExcel [64] verwendet.

Die Punktwolke für die in Tabelle 10 gegebenen Datenpunkte ist in Abbildung 23 zu sehen.

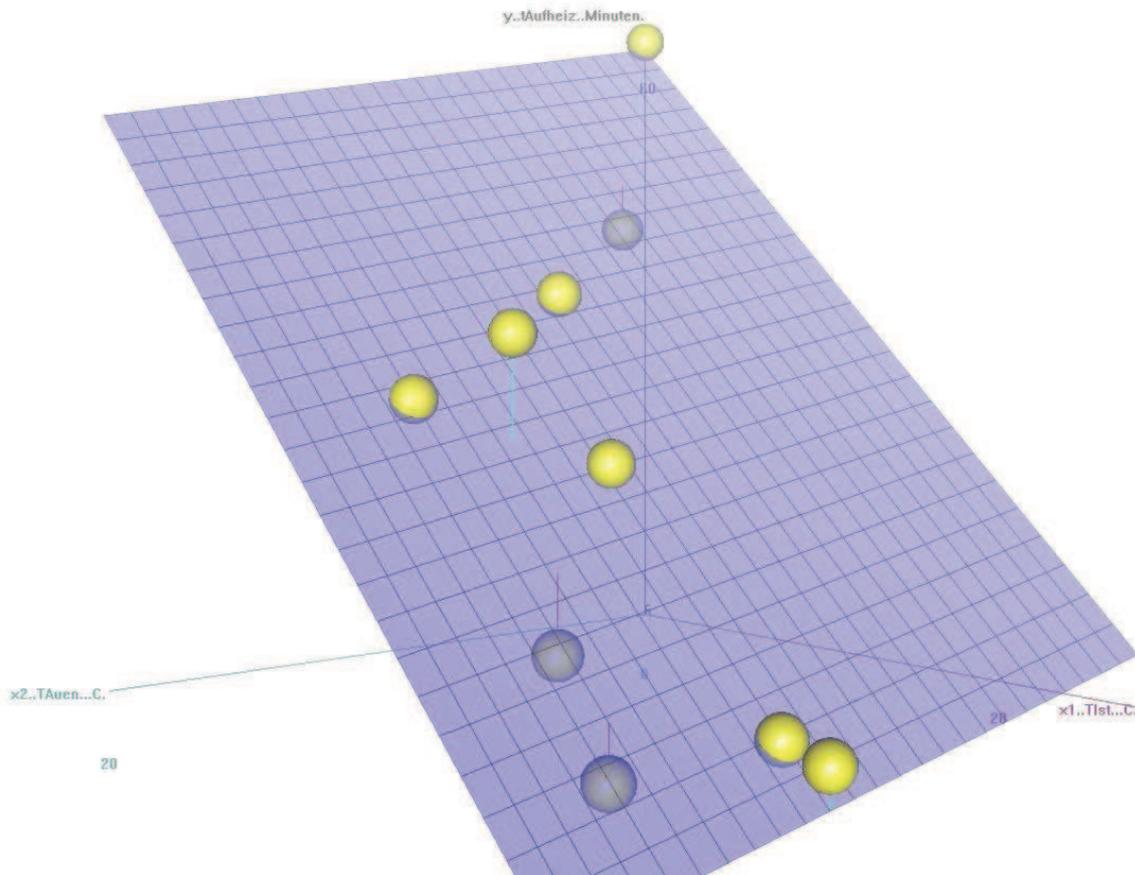


Abbildung 23: Punktwolke der multiplen Regression

Die Gleichung für die lineare multiple Regression ist in Formel 29 zu sehen.

$$y_i = 187,7550 - 8,3038 x_{1i} - 0,5244 x_{2i}$$

Formel 29: Vorhersage der Vorheizzeit mit Hilfe der multiplen Regression

Das Bestimmtheitsmaß berechnet sich zu  $R^2 = 0,9561$ .

#### 4.1.2.2.1.2 Operationale Phase

Um die operationale Phase zu illustrieren, nehmen wir an dieser Stelle eine gemessene Raumtemperatur von 17°C und eine Außentemperatur von 6°C an. Der Raum soll auf 22°C aufgeheizt werden, was auch als Annahme beim Aufstellen des Regressionsmodells eingeflossen ist. Um einen Vorhersagewert für die Vorheizzeit zu bekommen, wird in Formel 29 eingesetzt.

$$\underline{y} = 187,7550 - 8,3038 * 17 - 0,5244 * 6 = \underline{\underline{43,5}}$$

Formel 30: Beispiel für die Vorhersage der Vorheizzeit der multiplen Regression

Somit erhält man eine auf ganze Minuten aufgerundete Vorheizzeit von 44 Minuten. Das heißt, das Heizungssystem muss 44 Minuten vor der Benützung des gewünschten Raumes mit der Erwärmung des Raumes beginnen.

#### 4.1.3 Fazit für die Schätzung der Vorheizzeit mit Hilfe der multiplen Regression

Mit der Methode für Multiple Regression lässt sich ein guter Schätzer für die Vorheizzeit in Abhängigkeit von mehreren Variablen, wie aktuelle Raumtemperatur, aktuelle Außentemperatur für eine bestimmte Solltemperatur schätzen. Auch wenn die Temperaturverläufe nicht wie gezeigt linear verlaufen, kann man einen logarithmischen Verlauf in der Regressionsfläche leicht mit den gezeigten Formeln mitberücksichtigen, indem man jeweils die Variablen  $x_1$  und  $x_2$  in zum Beispiel Formel 29 mit dem Logarithmus Naturalis logarithmiert. Sollen viele verschiedene Variablen in das Schätzermodell einfließen, bieten sich für die komplexe Datenverarbeitung künstlich neuronale Netze in geeigneterer Weise für die gestellte Aufgabe an.

## 4.2 Künstliche neuronale Netze

Künstlich neuronale Netze haben ihr Vorbild in der Natur. Sie sind Gehirnen von Tieren und Menschen, die aus einem Netzwerk von Neuronen und Synapsen bestehen nachempfunden. Einzelne Neuronen „feuern“, wenn Sie über ihre Synapsen ein bestimmtes elektrisches Aktivitätslevel erreicht haben und leiten die elektrischen Impulse über Synapsen im Netzwerk weiter. Gehirne sind überaus leistungsfähige Signalverarbeitungs- und Speichersysteme, die über ihr Leben hinweg die Möglichkeit haben, neue Informationen zu lernen und zu einem späteren Zeitpunkt wieder anwenden zu können. Diese Eigenschaft werden mit künstlich neuronalen Netzwerken in einem mathematisch technischen System nachgebildet. Jedoch beschränkt man sich bei der Verwendung von künstlich neuronalen Netzen in einer bestimmten Anwendungsdomäne auf bis zu etwa 100 Knoten anstatt von ca. 20 Mrd. Neuronen im menschlichen Gehirn.

Künstlich neuronale Netzwerke bestehen, wie ihre Vorbilder aus der Natur auch, aus Neuronen und Synapsen. Die Neuronen werden Units oder Knoten und die Synapsen Kanten oder Links genannt. Der Aufbau und die Funktionsweise von Neuronen und Synapsen wird im ersten Unterkapitel vorgestellt.

Danach werden mögliche Netztypen für die Prognose der Vorheizzeit vorgestellt. Deren Kantengewichte, die das gespeicherte Wissen repräsentieren, müssen erst auf die spezielle Anwendung trainiert werden. Dieses Training wird im Kontext von künstlich neuronalen Netzen mit Lernen bezeichnet. Es werden Lernalgorithmen vorgestellt, die für die Anwendung der Prognose für die Vorheizzeit verwendet werden können.

Zum Schluss wird die für die Modellierung und Simulation von künstlichen neuronalen Netzen in dieser Arbeit verwendete Software MemBrain [30] vorgestellt.

### 4.2.1 Einführung

Die kleinsten Einheiten in einem künstlich neuronalen Netz sind Knoten, die Neuronen als ihr natürliches Vorbild haben. Knoten haben ein oder mehrere Inputs, mit denen sie ein

Aktivitätslevel bestimmen, über das der Output eines künstlichen Neurons berechnet wird. In Abbildung 24 kann man ein künstliches Neuron sehen. Auf der linken Seite befindet sich der Knoteneingang, der ein oder mehrere Inputs akzeptiert. Auf der rechten Seite des Knotens befinden sich ein oder mehrere Knotenausgänge, die alle denselben Wert für die Ausgabe bereitstellen.

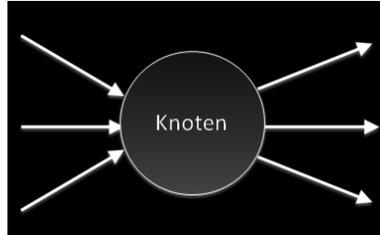


Abbildung 24: Künstliches Neuron

In Abbildung 25 sieht man, mit welchen Signalverarbeitungsblöcken ein Knoten intern aufgebaut ist.

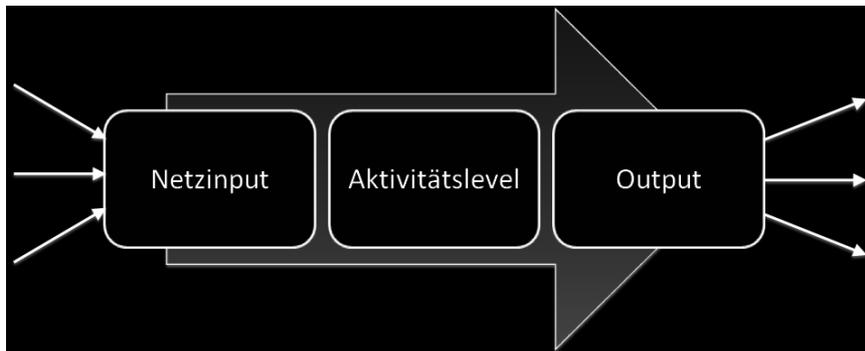


Abbildung 25: Interne Signalabfolge in einem Knoten

#### 4.2.1.1 Netzinput

Der erste Block Netzinput  $netinput_i$  bildet die gewichtete Summe der Inputwerte, die von den Vorgängerknoten  $j$  als Output  $a_j$  an die Eingängen eines Knotens  $i$  angelegt werden. Die Kantengewichte zwischen zwei miteinander verbundenen Knoten  $i$  und  $j$  werden mit  $w_{ij}$  in Formel 31 bezeichnet.

$$netinput_i = \sum_j a_j \cdot w_{ji}$$

Formel 31: Berechnung des Netzinputs

#### 4.2.1.2 Aktivitätsfunktionen

Aus dem Netzinput für den jeweiligen Knoten  $i$  folgt anschließend die Zuordnung eines Aktivitätslevels. Diese Zuordnung wird mit Hilfe einer Aktivitätsfunktion gebildet. Nachfolgend werden einige typische Aktivitätsfunktionen gezeigt. Die x-Achse stellt immer den Netzinput  $netinput_i$  dar und die y-Achse das korrespondierende Aktivitätslevel  $a_i$ .

#### 4.2.1.2.1 Lineare Aktivitätsfunktion

Die einfachste Aktivitätsfunktion ist die lineare Aktivitätsfunktion, wie sie beispielsweise in Abbildung 26 zu sehen ist.

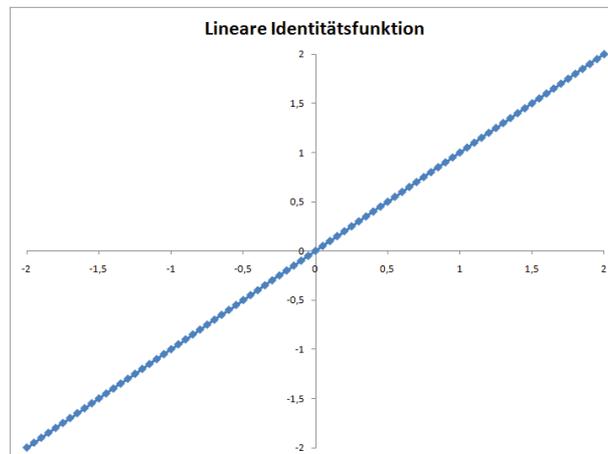


Abbildung 26: Lineare Aktivitätsfunktion

Die lineare Aktivitätsfunktion berechnet sich nach Formel 32.

$$a_i = k \cdot netinput_i$$

Formel 32: Lineare Aktivitätsfunktion

#### 4.2.1.2.2 Binäre Aktivitätsfunktion

Die binäre Aktivitätsfunktion beschreibt eine Funktion mit 2 Zuständen, den Aktivitätslevels 0 und 1, wie in Abbildung 27 und in Formel 33 zu sehen ist.

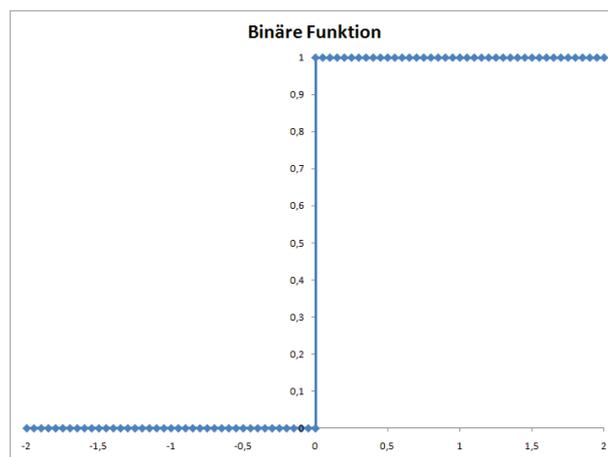


Abbildung 27: Heavyside-Funktion als Aktivitätsfunktion

$$a_i = \begin{cases} 1 & \text{wenn } netinput_i \geq 1 \\ 0 & \text{sonst} \end{cases}$$

Formel 33: Heavyside-Funktion

#### 4.2.1.2.3 Lineare Aktivitätsfunktion mit Schwelle

Eine weitere typische Form ist die lineare Aktivitätsfunktion mit Schwelle, wie man das zum Beispiel in Abbildung 28 in Formel 34 sehen kann. Hier ist der Sonderfall „lineare Identitätsfunktion mit Schwelle“ zu sehen.

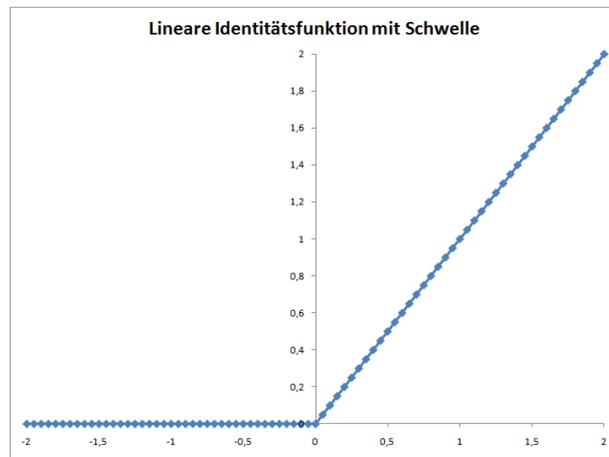


Abbildung 28: Lineare Aktivitätsfunktion mit Schwelle

$$y = \begin{cases} k \cdot netinput_i & \text{falls } netinput_i \geq 0 \\ 0 & \text{sonst} \end{cases}$$

Formel 34: Lineare Funktion mit Schwelle

#### 4.2.1.2.4 Sigmoidale Aktivitätsfunktionen

In den meisten Anwendungsfällen werden sigmoidale Aktivitätsfunktionen verwendet. Zu der Gruppe der sigmoiden Aktivitätsfunktionen gehören die logistische Aktivitätsfunktion und die Tangens Hyperbolicus Aktivitätsfunktion, die im Folgenden kurz beschrieben werden.

#### 4.2.1.2.4.1 Logistische Aktivitätsfunktion

Abbildung 29 und Formel 35 illustrieren die logistische Aktivitätsfunktion. Der Wertebereich von  $a_i$  liegt zwischen 0 und 1. Geht  $netinput_i$  gegen  $-\infty$ , dann geht  $y$  asymptotisch gegen 0. Bei  $netinput_i$  von 0 ist  $a_i$  gleich 0,5. Geht  $netinput_i$  gegen  $+\infty$ , dann geht  $a_i$  asymptotisch gegen +1.

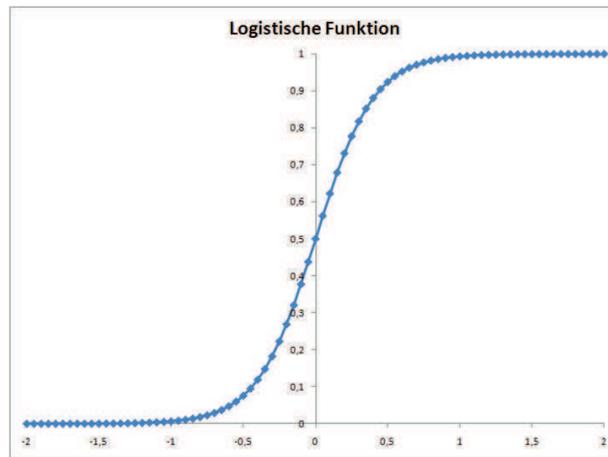


Abbildung 29: Logistische Aktivitätsfunktion

$$a_i = \frac{1}{1 + e^{-netinput_i}}$$

Formel 35: Logistische Aktivitätsfunktion

#### 4.2.1.2.4.2 Tangens Hyperbolicus Aktivitätsfunktion

Ähnlich wie die logistische Aktivitätsfunktion verhält sich die Tangens Hyperbolicus Aktivitätsfunktion. Der wesentliche Unterschied besteht darin, dass der Wertebereich von  $a_i$  zwischen -1 und +1 liegt. Die Funktion ist in Abbildung 30 und Formel 36 zu sehen.

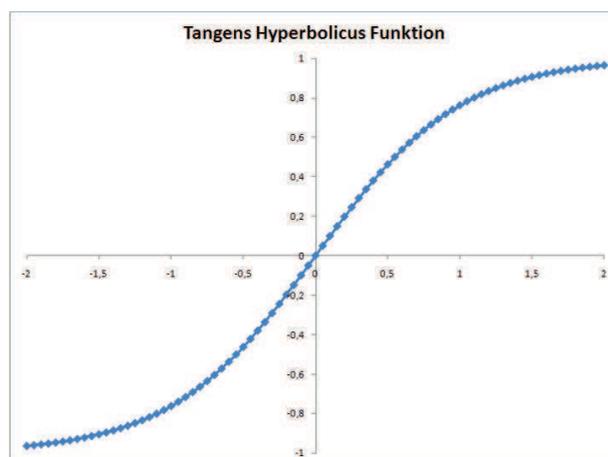


Abbildung 30: Tangens Hyperbolicus Aktivitätsfunktion

$$a_i = \tanh(netinput_i)$$

Formel 36: Tangens Hyperbolicus Aktivierungsfunktion

#### 4.2.1.2.5 Normalverteilte Aktivitätsfunktion

Nicht so oft verwendet, aber doch erwähnenswert, ist die normalverteilte Aktivitätsfunktion, wie in Abbildung 31 und Formel 37 zu sehen ist.

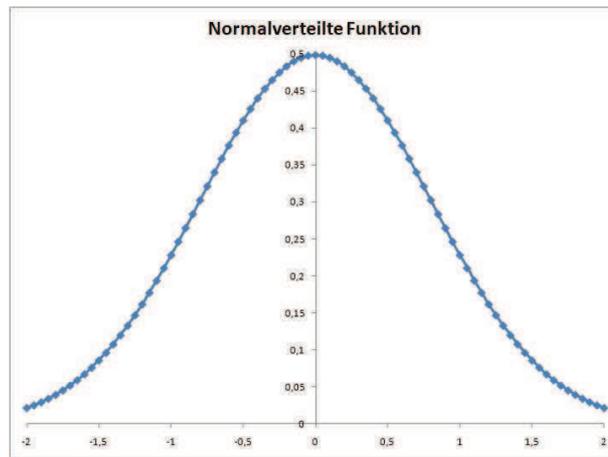


Abbildung 31: Normalverteilte Aktivitätsfunktion

$$a_i = \frac{1}{\sqrt{2 \cdot \pi}} \cdot e^{\frac{-netinput_i^2}{2}}$$

Formel 37: Normalverteilte Aktivitätsfunktion

#### 4.2.1.3 Output eines Neurons

Der Output  $o_i$  eines Neurons entspricht im einfachsten und gebräuchlichsten Fall der Identitätsfunktion der Aktivitätsfunktion  $a_i$ . Genauso wird das auch in der Software MemBrain behandelt. Ist  $a_i = 1$ , dann ist somit  $o_i = a_i = 1$ .

### 4.2.2 Trainings- und Testphase von künstlich neuronalen Netzen

Bei künstlich neuronalen Netze unterscheidet man zwischen Trainings- und Testphase.

#### 4.2.2.1 Die Trainingsphase

In der Trainingsphase werden Trainingsdaten<sup>48</sup> verwendet, um die Kantengewichte zwischen den Knoten zu verändern. Es gibt verschiedene Verfahren, nach denen die Kantengewichte trainiert werden können. Grundsätzlich wird zwischen den Lernmethoden Supervised Learning, Unsupervised Learning, Reinforcement Learning und Direct Design Methods unterschieden.

Beim überwachten Lernen werden dem künstlich neuronalen Netz die Trainingsdaten an die Eingänge angelegt und am Ausgang mit dem vorgegebenen Zielmuster verglichen. Je nachdem wie groß die Abweichung zwischen dem berechneten Ergebnis durch das künstlich neuronale Netz und der Sollwertvorgabe ist, werden die Kantengewichte angepasst, um den Fehler zu minimieren.

<sup>48</sup> Diese stellen einen ausgewählten Ausschnitt aus allen möglichen Eingabereizen dar.

Unüberwachtes Lernen erfolgt durch Anlegen von Eingangsmustern an das künstlich neuronale Netz. Es wird kein Sollwert für die Ausgabe vorgegeben. Die Kantengewichte werden in Abhängigkeit der Ähnlichkeit der Eingangsmuster verändert.

Bestärkendes Lernen ist ähnlich zu überwachtem Lernen. Dabei werden ebenfalls Eingabereize an das künstlich neuronale Netz angelegt und mit einem Sollwert verglichen. Jedoch wird im Vergleich zu überwachtem Lernen nur bewertet, ob das Ausgabemuster dem Sollausgabemuster entspricht oder nicht.

Bei den direkten Designmethoden werden die Kantengewichte vom Designer des künstlich neuronalen Netzes in der Designphase festgelegt.

#### **4.2.2.2 Die Testphase**

In der Testphase wird das zuvor trainierte künstlich neuronale Netz untersucht, wie es auf neue Inputreize reagiert. Damit wird überprüft, wie gut die Trainingsdaten generalisiert wurden, um neue Probleme zu lösen und damit eine Vorhersage mit neuen Reizen zu berechnen. Die durch das Netz berechneten Ausgangswerte werden mit Vergleichssollwerten verglichen. In dieser Phase werden die Kantengewichte nicht mehr modifiziert. Stellt das Netz das gewünschte Verhalten zur Verfügung, kann in die Anwendungsphase übergegangen werden.

#### **4.2.3 Auswahl des geeigneten Netztyps für die Vorhersage der Vorheizzeit**

In diesem Abschnitt werden ausgewählte Netztypen vorgestellt, die für die Vorhersage der Vorheizzeit in Betracht kommen. Generell kann man sagen, dass sich die Auswahl des geeigneten Typs nach der Art der Problemstellung richtet. Die Anpassungen, zum Beispiel wie viele Knoten sich in einem Layer befinden, die Anzahl der Layer und die Knotenverbindungen werden in der Praxis empirisch ermittelt.

In der Literatur findet man die Netztypen Pattern Associator, Rekurrente Netze, Kompetitive Netze, Kohonennetze und Constraint Satisfaction Netze. Welcher Netztyp für welche Anwendung zum Tragen kommt, wird nach den Eigenschaften der jeweiligen Netze entschieden. Die Eigenschaften gliedern sich in Kernkonzept, dem Vorhandensein von Rückkopplungsschleifen, der Existenz von Hidden Units und der Art der Lernregel. Bei den Lernregeln unterscheidet man zwischen überwachtem und unüberwachtem Lernen.

Das für diese Anwendung verwendete Netz soll eine Vorhersage für die Vorheizzeit liefern. Die Trainingsphase erfolgt überwacht, da die Zeit für das Heizen eines Raumes mit Hilfe des Heizungssimulationsmodells, das in einem späteren Kapitel vorgestellt wird, zur Laufzeit dynamisch ermittelt wird und dem Netz als Vergleichssollwert zugeführt wird. Unter diesen Gesichtspunkten kommen die Netztypen Pattern Associator und Rekurrente Netze in die nähere Auswahl, da diese Netztypen überwachtes Lernen unterstützen.

##### **4.2.3.1 Pattern Associator**

Die Anwendungsgebiete eines Pattern Associator Netzes findet man in der Mustererkennung, zum Beispiel bei der Gesichtserkennung. Dabei werden Eingangsreize

klassifiziert. Ein Pattern Associator besteht lediglich aus einer Eingabe- und einer Ausgabeschicht, wie in Abbildung 32 ersichtlich<sup>49</sup>.

Für die Anwendung zur Vorhersage der Vorheizzeit eignen sich Pattern Associator Netze nicht, da die Stärke der Pattern Associator in der Kategorisierung von Reizen liegt.

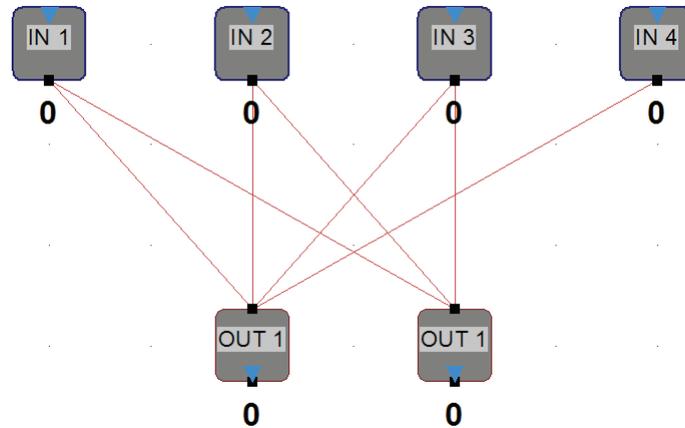


Abbildung 32: Beispiel für einen Pattern Associator

#### 4.2.3.2 Rekurrente Netze

Rekurrente Netze eignen sich, um Prognosen über die Zukunft anhand vorliegender Input-Daten zu treffen, beispielsweise über das Wetter oder das Klima und, wie in dem Kontext dieser Arbeit, eine Prognose über die Vorheizzeit eines Raumes.

Die Netze dieser Klasse können Rückkopplungen aufweisen. Dabei wird zwischen direkter, indirekter und seitlicher Rückkopplung sowie vollständigen Verbindungen unterschieden.

1. Eine direkte Rückkopplungen ist eine Kante vom Ausgang eines Knotens  $i$  zum Eingang des Knotens  $i$  (siehe Abbildung 33).
2. Die indirekte Rückkopplung ist eine Verbindung des Ausgangs eines Knotens  $j$  mit dem Eingang seines Vorgängerknotens  $i$  (siehe Abbildung 34).
3. Die seitliche Rückkopplungen beschreibt eine Kante zwischen dem Ausgang und dem Eingang eines Knotens derselben Schicht (siehe Abbildung 35).
4. Bei einem vollständig verbundenen Netz sind alle Knoten mit den anderen Knoten des Netzes verbunden (siehe Abbildung 36).

<sup>49</sup> Alle Netzgrafiken wurden mit der Software MemBrain erstellt.

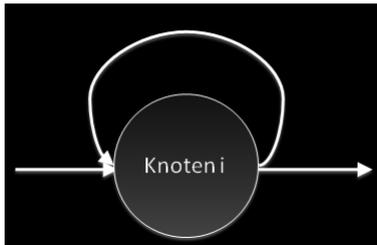


Abbildung 33: Direkte Rückkopplung

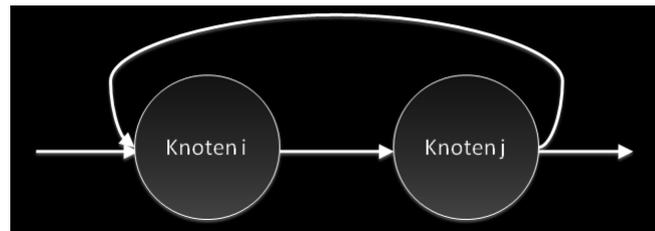


Abbildung 34: Indirekte Rückkopplung

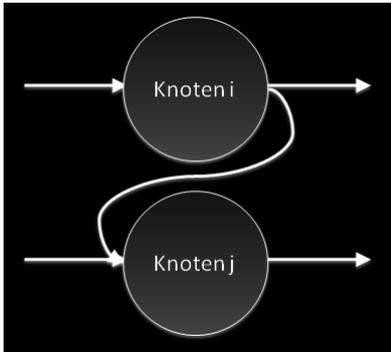


Abbildung 35: Seitliche Rückkopplung

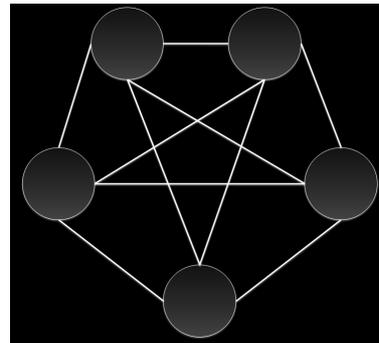


Abbildung 36: Vollständig verbundenes Netz

Für die Prognose der Vorheizzeit eignen sich potentiell Feedforward Netze, Simple Recurrent Networks und Jordan Netze, die an dieser Stelle kurz vorgestellt werden.

#### 4.2.3.2.1 Feedforward Netze

Feedforward Netze bestehen aus einer Eingabeschicht, einer oder mehreren verborgenen Schichten und einer Ausgabeschicht. Zwischen den Knoten existieren keine Rückkopplungen und Vermaschungen, wie in Abbildung 37 zu sehen,

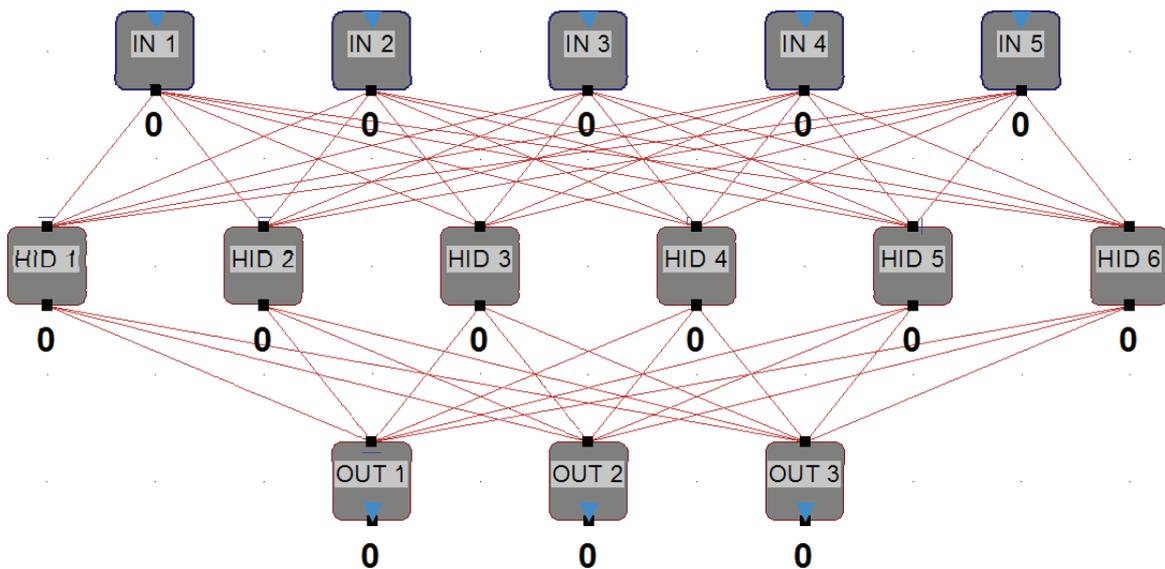


Abbildung 37: Beispiel für ein einfaches Feedforward Netz

sondern nur Verbindungen zwischen Neuronen zur darauffolgenden Schicht oder einem Knoten einer späteren nachfolgenden Schicht mittels einer sogenannten Shortcut-Verbindung (siehe beispielsweise Abbildung 38<sup>50</sup>).

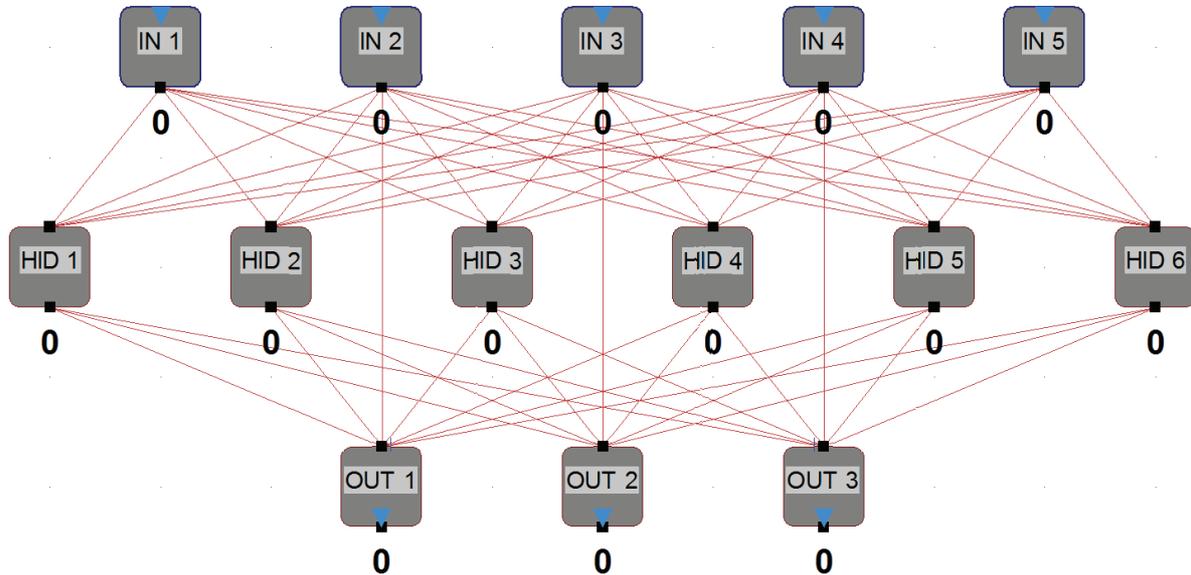
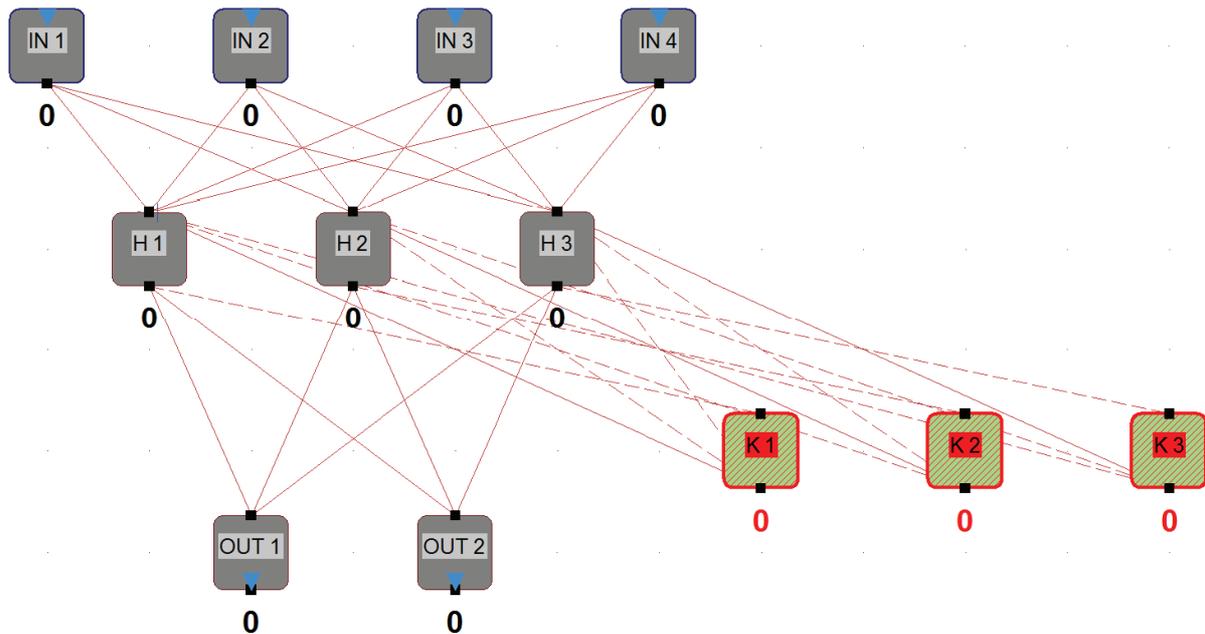


Abbildung 38: Beispiel für ein Feedforward Netz mit Shortcuts

#### 4.2.3.2.2 Simple Recurrent Networks

Dieser Netztyp weist indirekte Rückkoppelungen auf, die über eine Kontextschicht führen, wie in Abbildung 39 zu sehen ist. Knoten IN1, IN2, IN3 und IN4 sind in der Eingabeschicht, die Knoten H1, H2 und H3 sind in der Hidden Schicht und die Knoten O1 und O2 in der Ausgabeschicht. Die sogenannte Kontextschicht bilden die Knoten K1, K2 und K3, die in derselben Zahl wie die Knoten der Hidden-Schicht vorkommen müssen. Der Input eines Knotens der Kontextschicht ist mit den jeweiligen Ausgängen der Knoten der Hidden Schicht verbunden. Der Ausgang eines Knotens der Kontextschicht wird allen Knoten der Hidden-Schicht zugeführt.

<sup>50</sup> Besteht ein Feedforward Netz beispielsweise aus einer Eingabeschicht, einer verborgenen Schicht und einer Ausgabeschicht, dann kann auch eine Unit der Eingabeschicht, eine Verbindung direkt an die Ausgabeschicht aufweisen.

Abbildung 39: Beispiel für ein Simple Recurrent Network<sup>51</sup>

Die Kantengewichten zwischen den Knoten der Hidden- und der Kontextschicht sind mit 1 fixiert. Somit erhalten die Knoten der Kontextschicht immer eine exakte Kopie der Hidden-Schicht und sind dabei eine Art Pufferspeicher, der im nächsten Schritt den Vorgängerwert eines Knotens der Hidden-Schicht wieder an alle Knoten der Hidden-Schicht weitergeben wird. Damit wird ein dynamisches Gedächtnis modelliert, das willkürliche Sequenzen erlernen kann.

#### 4.2.3.2.3 Jordan-Netze

Genauso wie Simple Recurrent Networks besitzen Jordan Netze eine Hidden- und eine Kontextschicht, wie in Abbildung 40 zu sehen ist. Der Unterschied liegt jedoch darin, dass die Eingänge der Knoten der Kontextschicht an die Ausgänge der Knoten der Ausgangsschicht mit einem Kantengewicht von 1 angeschlossen sind. Die Ausgänge der Knoten der Kontextschicht sind mit den Eingängen der Knoten der Hidden-Schicht verbunden. Zusätzlich besteht eine direkte Rückkopplung der einzelnen Knoten der Kontextschicht mit fest vorgegebenen Gewichten.

<sup>51</sup> Schleifen werden von MemBrain automatisch erkannt und mit gestrichelten Linien gezeichnet.

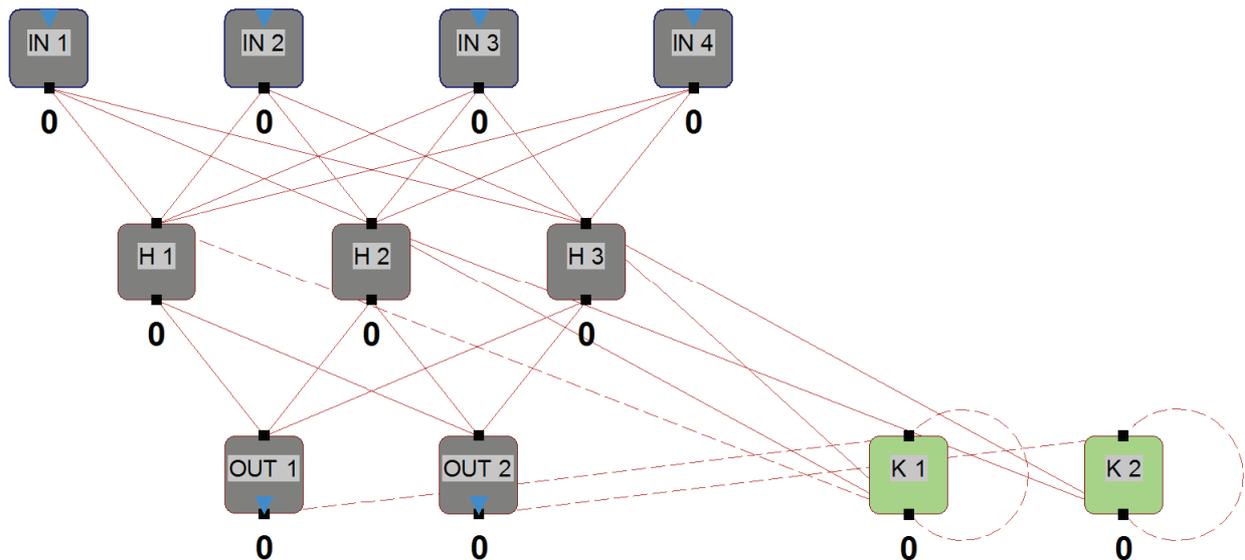


Abbildung 40: Beispiel für ein Jordan-Netz

#### 4.2.4 Lernregeln

An dieser Stelle wird das Prinzip einer Lernregel vorgestellt, die für die zuvor ausgewählten Netztypen angewendet wird. Dieser Algorithmus modifiziert die Kantengewichte in der Trainingsphase so, dass die Fehler zwischen Ist- und Sollwerten an den Ausgängen minimiert wird. Für die Veränderung der Kantengewichte von Feedforward Netzen, Simple Recurrent Networks und Jordan-Netze wird der Backpropagation-Algorithmus angewendet.

##### 4.2.4.1 Backpropagation-Algorithmus

Dieser Lernalgorithmus wird von Netzen mit einer oder mehreren Hidden-Schichten angewendet. Der Grundgedanke von diesem überwachten Algorithmus ist, dass der Fehler zwischen Ist- und Sollwert an den Ausgangsknoten berechnet wird und über eine Rechenvorschrift durch das Netz rückpropagiert wird, wobei schrittweise die Kantengewichte jeder einzelnen Schicht so modifiziert werden, dass der berechnete Fehler beim nächsten Durchlauf minimal ist.

Der Backpropagation-Algorithmus läuft in der Trainingsphase in 3 Subphasen ab:

##### 4.2.4.1.1 Forward-Pass

An die Inputneuronen werden Reize angelegt und das Netz berechnet mit Hilfe der Propagierungsfunktion den Output (siehe Abbildung 41). Die Berechnung des Ausgabewertes erfolgt stufenweise. Zuerst wird ein Output der Eingabeschicht, hervorgerufen durch externe Reize, berechnet. Die Ausgabe der Knoten der Eingabeschicht dient als Input der Knoten der Hiddenschicht, die wiederum einen Output berechnen. Schließlich berechnen die Knoten der Ausgabeschicht den Output des Netzes.

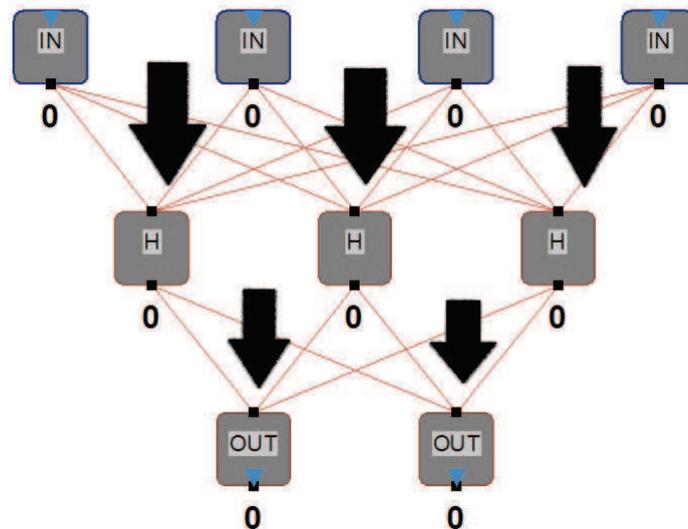


Abbildung 41: Backpropagation Algorithmus: Forward-Pass Phase

#### 4.2.4.1.2 Fehlerbestimmung

In dieser Phase werden die durch das Netz berechneten Ausgabewerten mit den jeweiligen Sollwerten verglichen. Die Fehlerterme werden quadriert und aufsummiert (siehe Formel 38<sup>52</sup>). Liegt der Fehlerterm über einer zuvor festgelegten Schwelle, dann wird die dritte Phase des Backpropagation-Algorithmus durchgeführt. Andernfalls bricht der Algorithmus an dieser Stelle ab.

$$E = \frac{1}{2} \sum_{i=1}^{\Omega} (y_i - y_{i,soll})^2$$

Formel 38: Quadrierte Summe des Fehlers an den Ausgabeknoten

#### 4.2.4.1.3 Backward-Pass

In dieser Phase werden die Fehler, wie in Abbildung 42 zu sehen ist, von den Knoten der Ausgabeschicht durch das Netz hindurch, bis hin zur Eingabeschicht durchlaufen. Um die Fehlerterme zu minimieren, werden die Kantengewichte des Netzes modifiziert (siehe Formel 39). Dies erfolgt mit Hilfe des Gradientenabstiegsverfahrens [31].

$$\delta_{w_{ij}} = -\varepsilon \frac{\partial E}{\partial w_{ij}}$$

Formel 39: Modifikation der Kantengewichte

<sup>52</sup> E steht für den Fehler,  $\Omega$  ist die Anzahl der Ausgabeknoten,  $y_i$  ist das durch das Netz berechnete Ausgangssignal und  $y_{i,soll}$  beschreibt den Sollwert des Ausgangssignals.

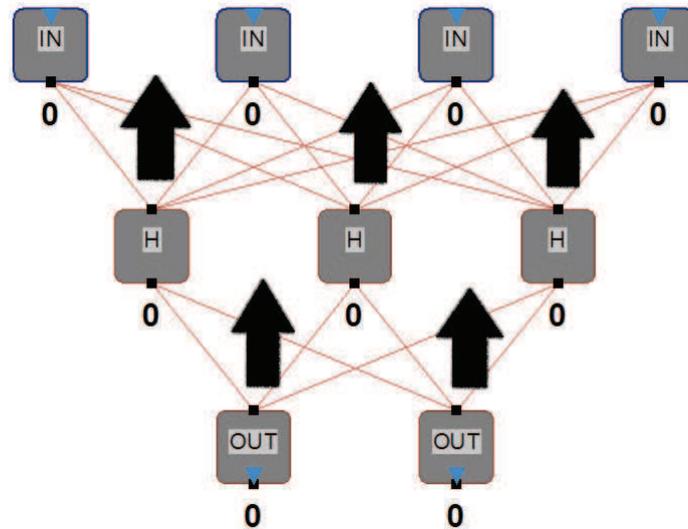


Abbildung 42: Backpropagation Algorithmus: Backward-Pass Phase

#### 4.2.5 Neuronale Netze Simulator - MemBrain

In diesem Abschnitt wird die in dieser Arbeit verwendete Simulationssoftware MemBrain<sup>53</sup> [30] beschrieben, die für die Modellierung von künstlich neuronalen Netzen sowie der Trainings- und Testphase verwendet wurde. Anschließend wird erwähnt, wie ein künstlich neuronales Netz, das mit MemBrain erstellt wurde, in eigene Applikationen eingebunden werden kann.

##### 4.2.5.1 Demo-Problem: XOR-Approximation mit einem künstlich neuronalen Netz

Zur Illustration der wichtigsten Funktionen von MemBrain soll ein XOR-Gatter mit einem künstlich neuronalen Netz approximiert werden. Die Wahrheitstabelle ist in Tabelle 11 abgebildet.

$x_1$	$x_2$	$y$
0	0	0
0	1	1
1	0	1
1	1	0

Tabelle 11: XOR Wahrheitstafel

Das Netz besteht aus 2 Eingängen  $x_1$  und  $x_2$  sowie einem Ausgang  $y$ . Für die Approximation des XOR-Gatters wird ein Feedforward-Netz mit einer Hidden Unit verwendet. Alle Knoten sind in vorwärtsstrebender Richtung untereinander verbunden, wie in Abbildung 50 zu sehen.

<sup>53</sup> Die Software funktioniert in aktuellen Windows Versionen (Windows XP SP3, Windows Vista und Windows 7).

#### 4.2.5.2 Erstellen eines Netzes mit MemBrain

Nach erfolgreicher Installation öffnet man das Programm MemBrain und sieht den Arbeitsbereich, wie in Abbildung 43 abgebildet.

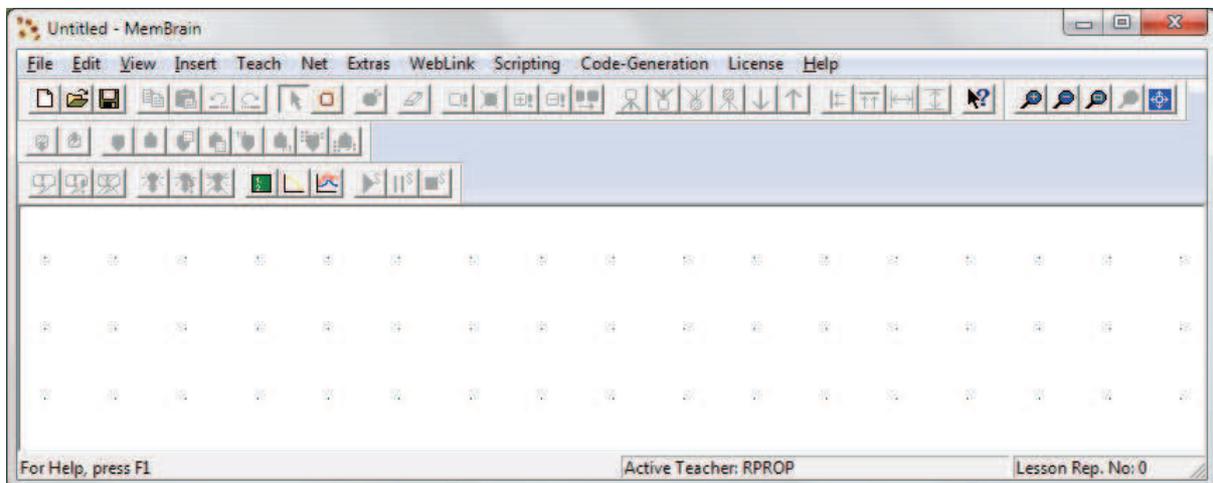


Abbildung 43: MemBrain - Hauptfenster

Anschließend fügt man Knoten über das Menü **Insert - New Neurons** hinzu, so wie in Abbildung 44 gezeigt.

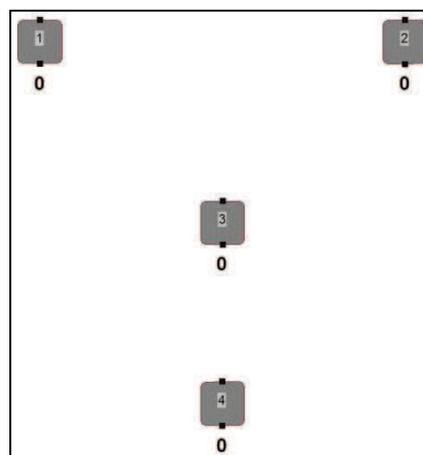


Abbildung 44: Neuronen einfügen

Jedes neu eingefügte Neuron ist in MemBrain standardmäßig eine Hidden Unit. Die Units 1 und 2 aus Abbildung 44 werden demzufolge Input Units und Knoten 4 eine Output Unit. Zusätzlich muss die Beschriftung der Knoten geändert werden. Knoten 1 und 2 werden in IN 1 und IN 2, die Hidden Unit - Knoten 3 - wird in H und das Neuron mit der Nummer 4 in OUT umbenannt.

Um dies durchzuführen, markiert man den betreffenden Knoten und wählt im Kontextmenü - Properties. Es öffnet sich ein Fenster, mit den Eigenschaften des Neurons auf 3 Tabs aufgeteilt. Die Änderungen werden im ersten Tab **Edit Neuron Properties** durchgeführt. Hier setzt man in der Gruppierung für generelle Einstellungen für das erste Inputneuron als Name *IN 1* ein und wählt den Typ *INPUT*. In der Eigenschaftengruppe **Activation Settings**

bleiben alle Einstellungen unverändert, genauso in der Gruppe der Eigenschaften für **Output Settings** (siehe Abbildung 45).

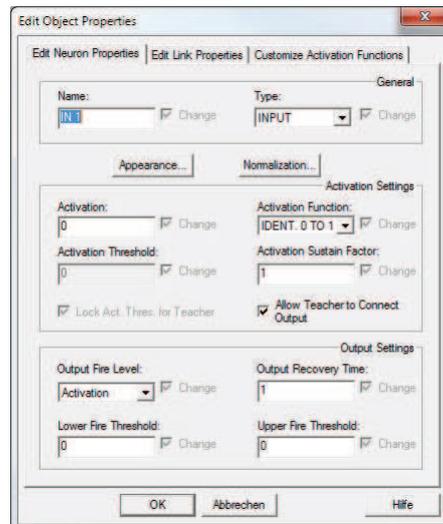


Abbildung 45: Unit - Object Properties Dialog

Das Tab **Edit Link Properties** ist dann interessant, wenn ein Verbindungsobjekt gewählt wurde. Hier kann ein Kantengewicht und die Signalverzögerung gesetzt werden sowie die Sichtbarkeit der Kantengewichte im Hauptfenster. Im letzten Tab **Customize Activation** können die 6 in MemBrain implementierten Aktivierungsfunktionen<sup>54</sup> eingestellt werden.

Nun werden die Kantenverbindungen durch Klicken auf die schwarzen Rechtecke der Neuronen (diese Rechtecke repräsentieren jeweils den Ein- und Ausgang eines Neurons) miteinander verbunden, wie dies in Abbildung 50 zu sehen ist.

#### 4.2.5.2.1 Trainingsphase

##### 4.2.5.2.1.1 Trainingsdaten einfügen

In der Trainingsphase muss das Verhalten des künstlich neuronalen Netzwerks durch überwachtes Lernen trainiert werden. Zu diesem Zweck müssen MemBrain die Trainingsdaten bekannt gemacht oder, falls es sich um eine externe CSV-Datei mit Trainingsdaten handelt, der Pfad zur Datei angegeben werden.

Da die Trainingsdaten für das hier illustrierte Beispiel sehr gering sind, werden die Trainingsdaten manuell eingeben. Zu diesem Zweck öffnet man den Lesson Editor im Menü unter **Teach - Lesson Editor** (siehe Abbildung 46). Im oberen Drittel sind die beiden Eingangsneuronen eingetragen, jeweils mit einem Wert 0. Im mittleren Drittel hat MemBrain das Ausgangsneuron eingetragen, ebenfalls mit 0 vorbelegt. Dem Lesson Editor muss die Korrespondenz, wie sie in Tabelle 11 abgebildet ist, bekannt gemacht werden. Die erste Korrespondenz kann übernommen werden. Die weiteren 3 Korrespondenzen können dem Lesson Editor durch Drücken der Schaltfläche **New Pattern** im mittleren Drittel rechts

<sup>54</sup> Verfügbare Aktivierungsfunktionen: Logistic, Identical, Identical 0 to 1, Tan Hyp, Binary und Min Euclid Dist.

bekannt gemacht werden. Wurden alle Korrespondenzen eingetragen, kann der Lesson Editor wieder durch Aktivieren der Schaltfläche **Close** geschlossen werden.

Liegen CSV-Dateien mit Trainingsdaten vor, dann können diese über die Menüleiste im Lesson Editor importiert werden.

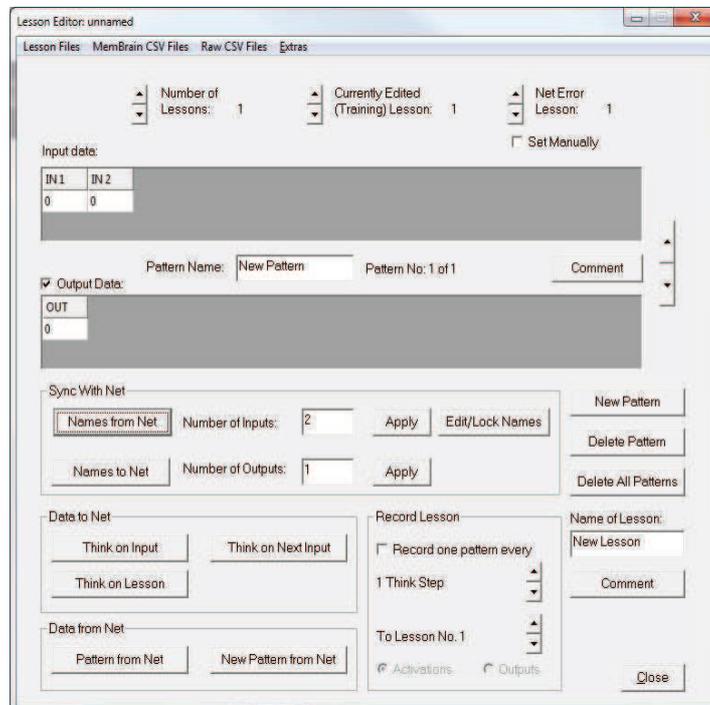


Abbildung 46: MemBrain - Lesson Editor

#### 4.2.5.2.1.2 Lernregel auswählen

Nun muss die Lernregel ausgewählt werden, mit der das Netz trainiert werden soll. Dazu wählt man im Menü **Teach - Teacher Manager** und klickt dann auf die Schaltfläche **New**. Es werden dann alle in MemBrain verfügbaren Lernalgorithmen aufgelistet, wie in Abbildung 47 zu sehen ist. Es kann zwischen den Algorithmen für überwachtes Lernen und für unüberwachtes Lernen gewählt werden. Für überwachtes Lernen stehen die Algorithmen *Backpropagation*, *Standard Backpropagation with Momentum*<sup>55</sup>, *Backpropagation with Momentum*, *Competitive with Momentum*, *Cascade Correlation Backpropagation*, *Cascade Correlation Resilient Backpropagation* und *Resilient Backpropagation* zur Verfügung. Jede der angeführten Lernregeln eignet sich für rekurrente Netze, bis auf die Lernregeln *Backpropagation* und *Standard BP with Momentum*. Für unüberwachtes Lernen steht der Algorithmus *The Winner Takes it All*, speziell zugeschnitten für *Self Organized Maps*, zur Verfügung.

Für das XOR-Beispiel wird die Lernregel RPROP verwendet. Wichtig ist, dass die Option *Reset Net Before Every Lesson* gewählt wurde. Das stellt sicher, dass die Trainings immer vom selben Ausgangszustand des Netzes beginnen. Alle weiteren Einstellungen, die

<sup>55</sup> Nur für Netze ohne Rückkopplungen.

durchgeführt werden können, wurden von MemBrain schon mit passenden Voreinstellungen versehen.

#### 4.2.5.2.1.3 Kantengewichte initialisieren

Die Kantengewichte zwischen den einzelnen Knoten werden von MemBrain für den Startzustand zufällig gewählt. Dies erreicht man im Menü unter **Net - Randomize Net**. Die Kantengewichte werden zwischen -1 und +1 gewählt.

Prinzipiell könnte man die Kantengewichte manuell wählen oder gar auf 0<sup>56</sup> belassen. Jedoch verspricht man sich von zufällig gewählten Kantengewichten, dass der Lernalgorithmus ein globales Minimum findet und nicht in einem lokalen Minimum stecken bleibt und die beste Lösung für die Kantengewichte eventuell nie ermitteln kann.

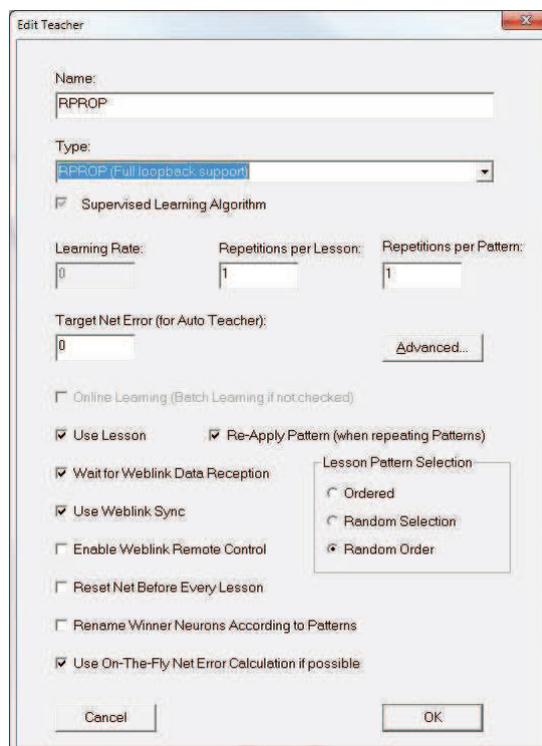


Abbildung 47: Lernregel wählen

#### 4.2.5.2.1.4 Kantengewichte trainieren

Für die eigentliche Trainingsphase öffnet man den **Net Error Viewer** im Menü unter **Teach - Net Error Viewer**. In diesem Fenster sieht man den mittleren quadratischen Fehler des Ausgangs in Abhängigkeit zu den durchgeführten Übungsaufgaben (siehe Abbildung 48). Ziel ist es, den Fehler zu minimieren, jedoch nicht gleich 0 werden lassen, um eine Überanpassung des Netzes an die Trainingsdaten zu verhindern. Typischerweise lässt man eine Trainingssitzung solange laufen, bis der mittlere quadratischen Fehler am Ausgang circa 0,01% aufweist.

<sup>56</sup> Das ist die Voreinstellung, wenn man eine neue Kante hinzufügt.

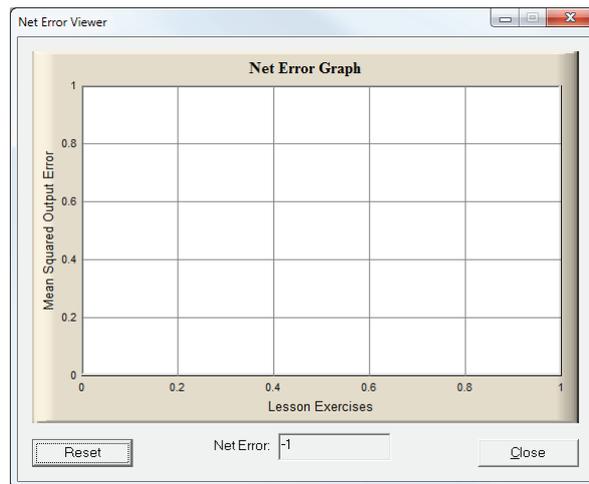


Abbildung 48: Net Error Viewer

Nun kann damit begonnen werden das Netz zu trainieren. Dazu startet man den „Lehrer“ im Menü unter **Teach - Start Teacher (Auto)**, der dem künstlich neuronalen Netz die Trainingsdaten in randomisierter Reihenfolge, die Eingaben und die Sollwerte für die Ausgaben zum Vergleich mit den durch das Netz berechneten Ausgabewerten zuführt.

Wird im Net Error Graph ein mittlerer quadratischer Fehler von circa 0,01% erreicht, kann die Trainingssitzung im Menü **Teach - Stop Teacher (Auto)** beendet werden. In Abbildung 49 sieht man einen möglichen Verlauf des mittleren quadratischen Fehlers der Ausgabeabweichung während des Trainingsverlaufs.

Ab circa 70 geübten Trainingsdatensätzen ist das Netz für die XOR-Aufgabe trainiert und das Lernen ist somit abgeschlossen.

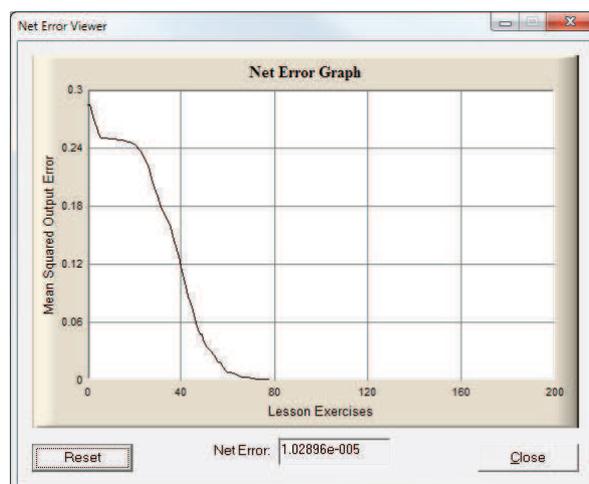


Abbildung 49: Möglicher Verlauf des mittleren quadratischen Fehlers

#### 4.2.5.2.2 Testphase

In der Testphase werden dem künstlich neuronalen Netz nur die Eingabedaten der Trainingsdaten vorgelegt. In MemBrain kann dies bequem mit Hilfe des Lesson Editors durchgeführt werden. Dazu öffnet man den Lesson Editor<sup>57</sup> und klickt im Bereich **Data to Net** auf die Schaltfläche **Think on Next Input** und überprüft die Ausgabe des Netzes auf der Basis der Eingabedaten.

Für die XOR-Aufgabe bedeutet das, dass das künstlich neuronale Netz die Wahrheitstabelle aus Tabelle 11 approximiert. Im vorliegenden Fall heißt das, wenn die Ausgabe den Wert 0 haben soll, dass das Netz eine Ausgabe nahe 0 berechnet. Gleiches gilt für den Sollwert 1, den das künstlich neuronale Netz, wie zum Beispiel in Abbildung 50 zu sehen, mit 0,9 berechnet.

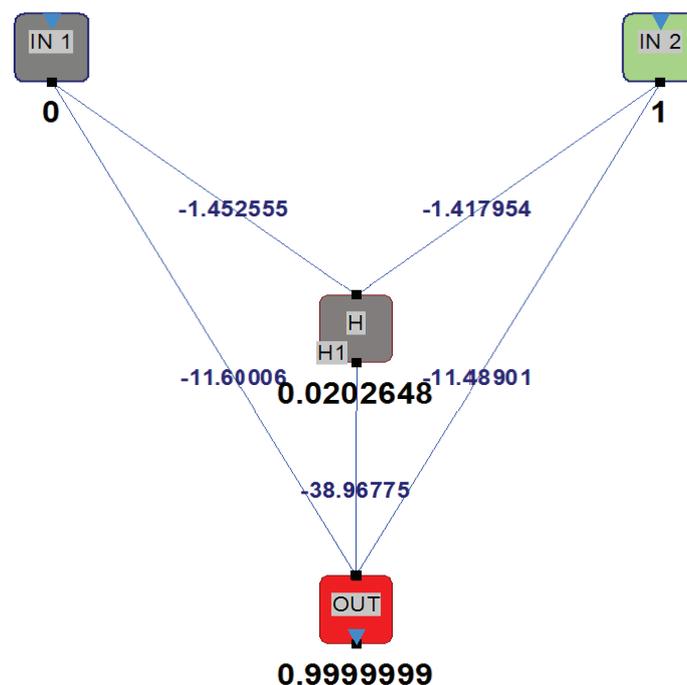


Abbildung 50: Künstlich neuronales Netzwerk für das XOR-Problem

#### 4.2.5.3 Einbinden von MemBrain in eigene Applikationen

Die mit MemBrain erstellten und trainierten Netze können über eine MemBrain DLL<sup>58</sup> in eigene Applikationen, die in einer beliebigen Programmiersprache für Windows erstellt wurden, eingebunden werden, ohne dabei auf das TCP/IP Interface oder auf die interne Skriptsprache von MemBrain zurückzugreifen. Diese Eigenschaft wird für die Start-Stopp-Vorhersage für die Vorheizzeit verwendet, die im späteren Verlauf dieser Arbeit vorgestellt wird.

Für die Programmiersprachen Java, C# und Visual Basic existieren vorgefertigte Wrapper-Klassen, die das Interfacing zur Programmiersprache erleichtern.

<sup>57</sup> Im Menü unter **Teach-Lesson Editor**.

<sup>58</sup> Dynamic Link Library

Die DLL muss entweder im Ordner der Applikation<sup>59</sup> verfügbar sein oder im Systemdateienordner von Windows<sup>60</sup>.

Die aktuelle DLL<sup>61</sup> unterstützt dabei

- das gleichzeitige Laden von mehreren neuronalen Netzen,
- Zuweisen von Inputwerten,
- Ausführen einer beliebigen Anzahl von Simulationsschritten,
- Auslesen der Ergebniswerte, die mit dem künstlich neuronalen Netz berechnet wurden,
- Trainieren von Netzen,
- Erstellen, Laden, Importieren und Exportieren von mehreren Datensätzen.

Zusätzlich zu diesen Eigenschaften, bietet MemBrain die Funktion ein erstelltes und trainiertes künstlich neuronales Netz als C-Programmcode zu exportieren. Damit könnte das neuronale Netz auch auf einem Mikrocontroller eingesetzt werden.

---

<sup>59</sup> Zum Beispiel *C:\Program Files\MyApp\bin.* unter der Voraussetzung, dass sich in diesem Ordner die ausführbaren Dateien für MyApp befinden.

<sup>60</sup> Beispielsweise *{Laufwerksbuchstabe}:\Windows\System32*

<sup>61</sup> Version 03.01.00.00 vom 30. Jänner 2010

## 5 Einsparungspotential von - Ein Vergleich zwischen künstlich neuronalen Netzen zu konventioneller Zeitsteuerung

In diesem Kapitel wird die Eignung von künstlich neuronalen Netzen für die Raumlufttemperaturregelung einer herkömmlichen Zeitsteuerung gegenübergestellt. Dafür wird ein Simulationsmodell verwendet, um die Physik zu modellieren, die mit einer Heizungsregelung einhergeht. Im Gebäudesimulationsmodell werden die Parameter Außentemperatur, Innentemperatur oder das Verhalten der Gebäudestruktur bei Temperaturänderungen berücksichtigt.

Die Simulation der Heizungsregelung findet in 3 Phasen statt, die im Belegungszeitplan wieder zu finden sind. Phase 1 ergibt sich aufgrund der Physiksimulation, die Phasen 2 und 3<sup>62</sup> aufgrund der Vorgehensweise beim Trainieren und Testen von künstlich neuronalen Netzen.

In der ersten Phase, die eine Woche dauert, werden die internen Massen des Simulationmodells eingeschwungen. Dazu werden die simulierten Wände, Fußböden und Raumdecken von einer Kerntemperatur von 0°C in den jeweiligen Bauteilen aufgeheizt.

In der zweiten Phase findet das Training der künstlich neuronalen Netze statt. Hier wird nach einem festen Zeitplan über 2 Wochen hinweg zu Beginn der jeweiligen Ereigniszeitpunkte<sup>63</sup> der Zustand der Heizung geändert und dann die Zeitspanne gemessen, bis die Raumtemperatur in vorgegebenen Temperaturtoleranzbereichen gebracht wurde.

Die dritte und letzte Phase dient dazu, die trainierten künstlich neuronalen Netze untereinander zu vergleichen und festzustellen, welche Kostenersparnis im Vergleich zu einer zeitgesteuerten Temperaturregelung quantifiziert werden kann.

Im Anschluss an die Simulationsergebnisse wird besprochen, inwiefern sich künstlich neuronale Netze für die Raumtemperaturregelung verwenden lassen.

### 5.1 Gebäudemodell

Die Physik des Gebäudes wird mit der Modelica [32] Bibliothek ATplus [33] modelliert, da die Sollwertvorgaben für die jeweilige Raumtemperatur über ASCII-Zustandsdateien modifiziert werden können und damit eine Anbindung über eine beliebige Programmiersprache ermöglicht wird.

Das Gebäudemodell beinhaltet im Wesentlichen ein Gebäude mit zwei Räumen mit je 50m<sup>3</sup> Volumen deren Wärmeübertragungsverhalten gekoppelt<sup>64</sup> ist. In den beiden Räumen befindet sich jeweils ein 6m<sup>2</sup> großes Fenster und ein Heizkörper, der von einem zentralen Heizkessel mit einer Leistung von 10kW über Heizungswasserverrohrung gespeist wird. Zusätzlich werden interne Lasten<sup>65</sup> modelliert, die den Räumen überdies, wenn auch in geringem Maß, Wärme zuführen. Die Außenhülle des Modells ist simulierten

<sup>62</sup> Für die 2 Phasen Trainings- und Testphase von künstlich neuronale Netzen, siehe Kapitel 4.2.2

<sup>63</sup> Der Start einer Komforttemperaturzeitspanne oder der Beginn einer Temperaturabsenkezeitspanne

<sup>64</sup> Dieser Sachverhalt modelliert den Wärmeübertragungseinfluss zweier benachbarter Räume.

<sup>65</sup> Menschen oder elektrische Verbraucher

Außenwetterbedingungen ausgesetzt. Die Räume sind nach den Himmelsrichtungen ausgerichtet. Raum 1 befindet sich in südlicher Lage, Raum 2 in nördlicher Lage. ATPlus unterscheidet Tag und Nacht beim Heizverhalten. Das heißt, am Tag heizt ein Raum schneller auf als in der Nacht, da die zusätzliche solare Infrarotstrahlung nachts fehlt.

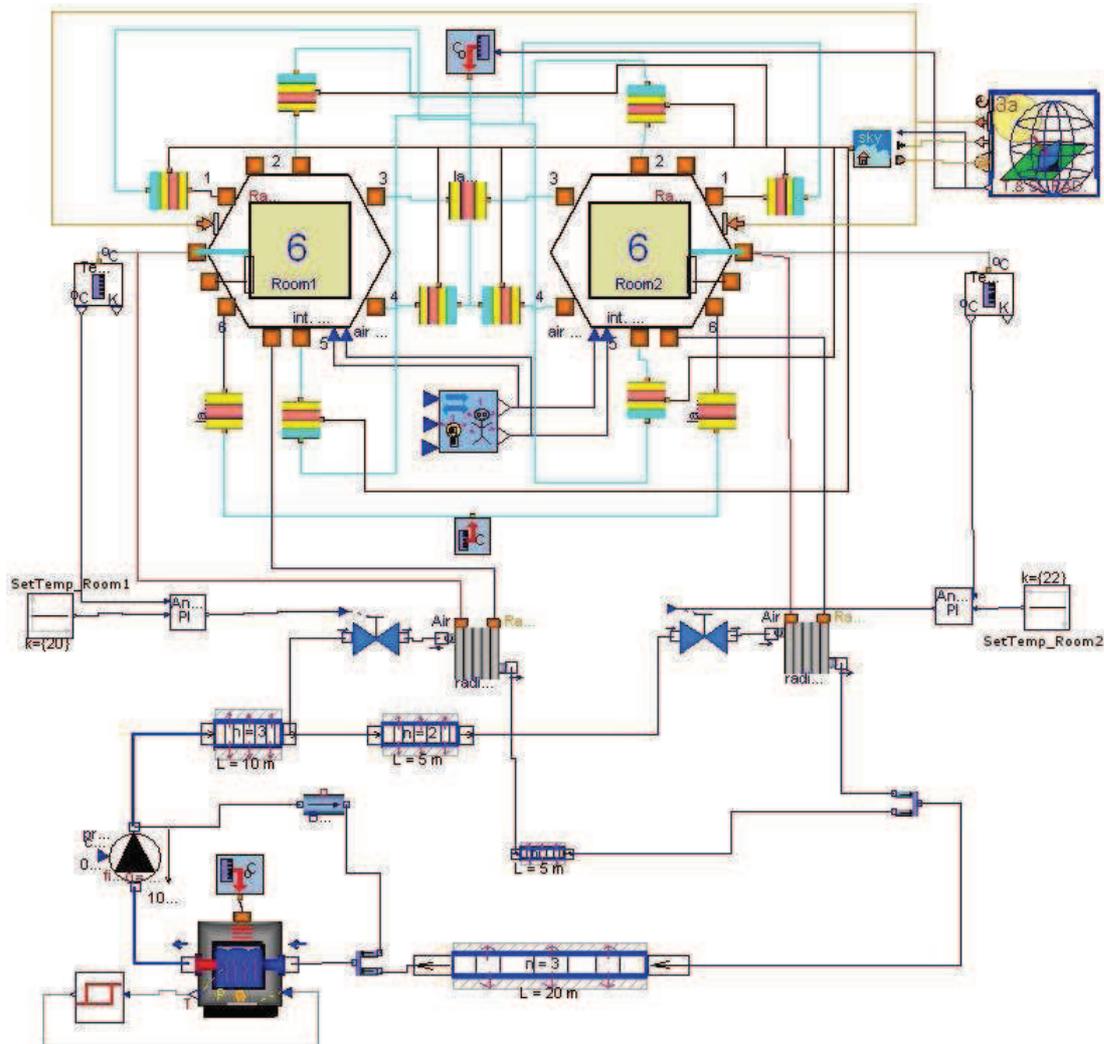


Abbildung 51: ATPlus - Gebäudemodell

Die wichtigsten Dateien von ATPlus und deren Bedeutung findet man in Tabelle 12.

Dateiname	Bedeutung
<b>dymosim.exe</b>	Von Dymola kompiliertes Simulationsmodell <sup>66</sup>
<b>humid.txt</b>	Luftfeuchtigkeitsdaten; wird zur Laufzeit benötigt
<b>tab_mrz2.txt</b>	Wetterdaten; werden zur Laufzeit benötigt
<b>start.txt</b>	Erste Eingabedatei für den Simulator
<b>after_xx_min.txt</b>	Zustand nach einer definierten Zeitspanne xx
<b>first_xx_mins.mat</b>	Verlauf der ersten xx simulierten Minuten

Tabelle 12: ATPlus Dateien

Die wichtigsten Modellparameter für das Simulationsmodell sind in Tabelle 13 zusammengefasst.

Variablename	Bedeutung	Einheit
<b>StartTime</b>	Startzeitpunkt der Iteration <sup>67</sup>	[sec]
<b>StopTime</b>	Ende des Iterationsschrittes	[sec]
<b>SetTemp_Room1.k[1]</b>	Sollwert für den Raum 1	[°C]
<b>SetTemp_Room2.k[2]</b>	Sollwert für den Raum 2	[°C]
<b>TempSensorRoom1.T0</b>	Gemessene Lufttemperatur im Raum 1	[°C]
<b>TempSensorRoom2.T0</b>	Gemessene Lufttemperatur im Raum 2	[°C]
<b>GroundTempFixed.Ts</b>	Fußbodentemperatur	[°C]
<b>OutsideAirTemp.T</b>	Außenlufttemperatur	[°C]
<b>Burner_OpTimeMeter.y[1]</b>	Brennerlaufzeit	[sec]

Tabelle 13: Für die Simulation notwendige ATPlus Variablen

Eine Simulation wird auf der Windows-Kommandozeile mit dem Befehl angestoßen:

```
> dymosim.exe -f after_xx_min.txt start.txt first_xx_mins.mat
```

## 5.2 Belegungsplan

Abbildung 52 stellt einen vorgegebenen Belegungsplan dar. Der Plan wird für die Initialisierungs-, Trainings- und Testphase der künstlich neuronalen Netze und der Referenzmessung verwendet. Es wird zwischen Komforttemperatur CT und Energiespartemperatur ET unterschieden. Die CT wird mit 22°C definiert (im Belegungsplan mit hellroter Farbe eingezeichnet), die ET mit 16°C (im Belegungsplan mit hellgrüner Farbe eingezeichnet).

Der Zeitplan beginnt mit dem 1. März 2007 um 0:00 Uhr und endet mit dem 28. März 2007 um 23:50 Uhr, da für diesen Zeitraum Außentemperaturwerte für das Simulationsmodell vorliegen.

<sup>66</sup> Unter Windows XP und ohne die Simulationmodellierungssoftware Dymola lauffähig

<sup>67</sup> Gemessen ab 1.3.2007; Dies ist vom ATPlus Paket, aufgrund der Luftfeuchtigkeitsdaten vorgegeben.

In der ersten Woche wird das Gebäudemodell initialisiert. In dieser Zeit werden beide Räume durchgehend mit einer Sollwerttemperatur von 25°C geheizt<sup>68</sup> (im Belegungsplan mit dunkelroter Farbe eingezeichnet). Damit erreichen alle Wände, Böden und Decken eingeschwungene Kerntemperaturen. Damit werden Verzerrungen aus Einschwingvorgängen des Modells für die Trainings- als auch für die Testphase eliminiert.

Die zweite und dritte Woche des Belegungsplans sind als Trainingsdaten für die künstlich neuronalen Netze gedacht. Eine Temperaturänderung erfolgt zu den eingetragenen Änderungszeitpunkten aus dem Zeitplan.

Die letzte Woche dient zum Testen und zum Überprüfen der trainierten Netze, ob diese tatsächlich in der Lage sind den optimalen Vorheizzeitpunkt zu schätzen und außerdem deren quantitativer Vergleich in Hinblick auf die Brennerlaufzeit zur Referenzbrennerlaufzeitmessung, wodurch man auf die Energieersparnis schließen kann.

---

<sup>68</sup> Das Gebäudemodell wird von Modelica mit 0°C Temperatur für Luft und Materialien kompiliert.

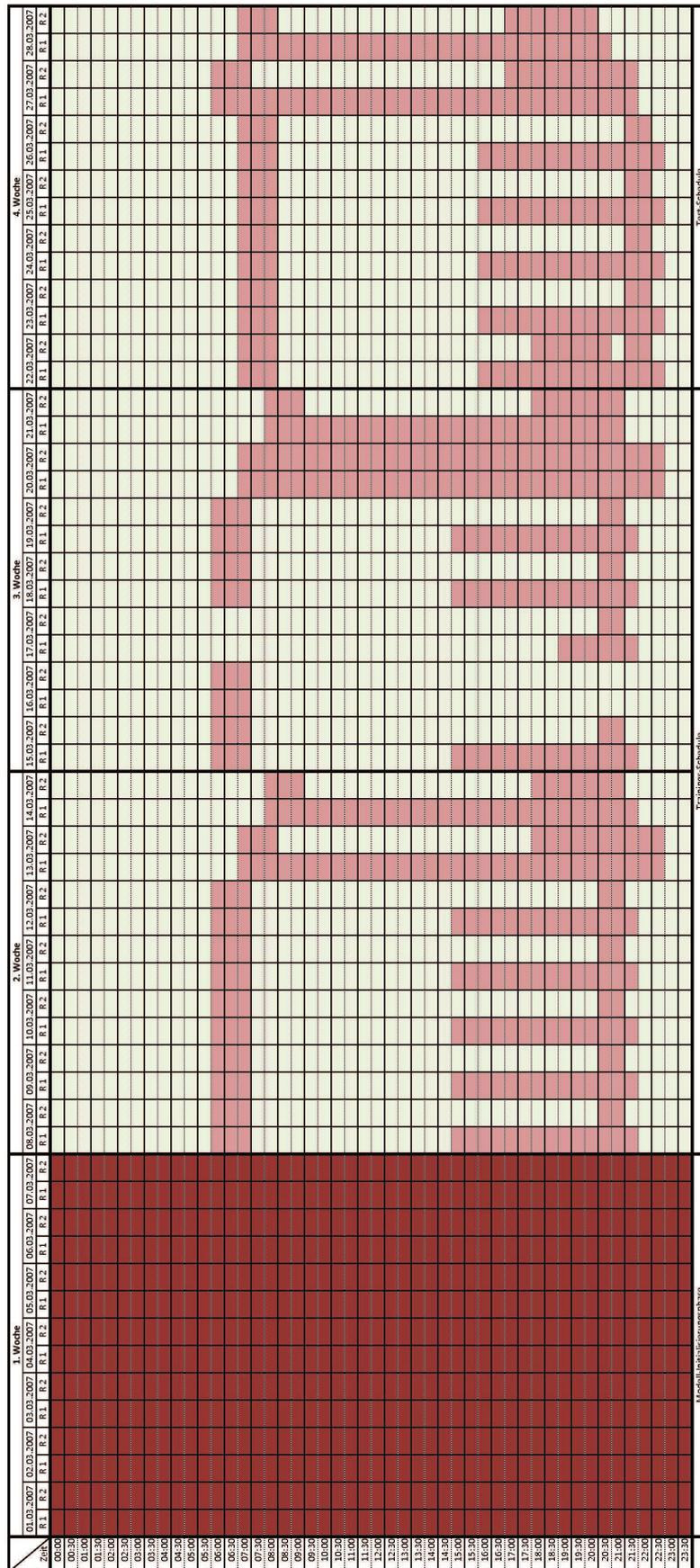


Abbildung 52: Zeitplan für die Initialisierungs-, Trainings- und Testphase

### 5.3 Verwendete Neuronale Netze

Für die Bestimmung des optimalen Start-Stopp-Zeitpunkts werden nachfolgende künstlich neuronale Netze hinsichtlich Einsparung der Brennerlaufzeit im Simulationsmodell miteinander verglichen. Für jeden Raum wird jeweils ein Netz verwendet, da sich jeder Raum in einem Haus unterschiedlich verhält, je nach Ausrichtung nach den Himmelsrichtungen und den Nachbarräumen.

Alle künstlich neuronale Netze haben die gleichen Ein- und Ausgangsneuronen, diese sind in Tabelle 14 zusammengefasst. Die erste Spalte korrespondiert direkt mit den Bezeichnungen der mit MemBrain modellierten künstlich neuronalen Netze. Zusätzlich findet man in der Übersicht der Ein- und Ausgabeneuronen die normierten Wertebereiche und deren physikalischen Einheiten. Die künstliche Neuronen, die mit MemBrain modelliert werden, akzeptieren nur Inputs im Wertebereich  $0 \leq x \leq 1$ . Daher muss eine Normierung erfolgen. Die Normierung erfolgt linear und ist in der Spalte Wertebereich angegeben. Die Ein- und Ausgabebereiche der versteckten Schichten müssen nicht normiert werden, da die Normierung nur an den Ein- und Ausgabeneuronen erfolgt.

Name des Neurons	Typ	Bedeutung	Wertebereich	Einheit
SET t	Input	Uhrzeit	$0 \leq t \leq 23,99$	[h]
SET T	Input	Sollwert für die Raumtemperatur	$10 \leq T \leq 30$	[°C]
CUR T	Input	Vorliegende Raumtemperatur zum Zeitpunkt der Sollwertänderung	$0 \leq T \leq 40$	[°C]
OUT T	Input	Vorliegende Lufttemperatur außerhalb des Hauses zum Zeitpunkt der Sollwertänderung	$-30 \leq T \leq 50$	[°C]
HEAT t	Output	Berechnete Zeitspanne bis zum Erreichen der Solltemperatur	$0 \leq t \leq 20000$	[sec]

Tabelle 14: Ein- und Ausgabeneuronen

Alle Inputneuronen haben eine lineare Identitätsfunktion. Die Eingangswerte werden auf einen definierten Wertebereich (siehe Tabelle 14) normiert. Das Ausgangsneuron verwendet eine logistische Aktivierungsfunktion, genauso wie die Neuronen der versteckten Schichten. Große negative Eingaben an den Neuronen der versteckten Schichten als auch des Ausgangsneurons bewirken eine Ausgabe nahe 0. Steigt die Eingabe an den Neuronen, dann steigt die Ausgabe zunächst langsam an. Der Anstieg wird immer steiler und ist zwischenzeitlich einer linearen Funktion sehr ähnlich. Ist die Eingabe an den Neuronen sehr hoch, nähert sich die Ausgabe asymptotisch an 1 an.

Alle Netze wurden mit Resilient Backpropagation Algorithmus RPROP [34] trainiert. Dieser Algorithmus ist Teil von MemBrain und bietet volle Unterstützung von Loop-Back-Schleifen. Gelernt werden die Trainingsdaten mit einer Wiederholung pro Lerneinheit und einer Wiederholung pro Ein- und Ausgabemuster. Da die Daten in einem zeitlichen Zusammenhang stehen, müssen auch die Lernmuster in dieser Reihenfolge gelernt werden. Vor jeder Lernphase werden die Kantengewichte auf einen zufälligen Wert zurückgesetzt. Das hat den Hintergrund, dass der Lernalgorithmus während der Lernphase nicht in einem

lokales Optimum hängen bleibt, sondern das globale Optimum der Kantengewichte findet. Die Lernphase wird als beendet betrachtet, wenn der Netzfehler unter 0,1‰ gelangt ist.

### 5.3.1 Feed Forward Network

Das FeedForward Netz (siehe Abbildung 53) besteht aus vier Eingängen und einem Ausgang<sup>69</sup>. Dazwischen befinden sich doppelt so viele verdeckte Neuronen HID 1 bis HID 8 in einer verdeckten Schicht wie Inputneuronen. Das Netz besitzt keine Rückkopplungen.

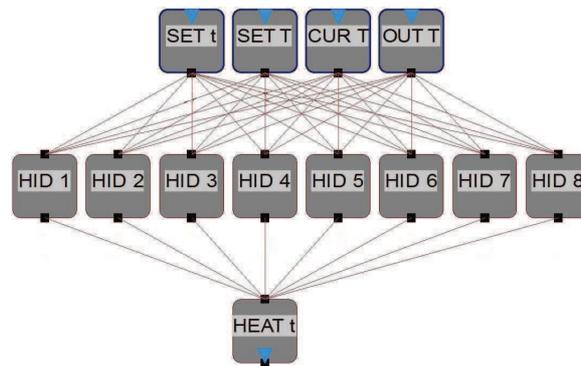


Abbildung 53: Optimum Start-Stop FeedForward Netz

### 5.3.2 Simple Recurrent Network

Die vier Knoten SET t, SET T, CUR T und OUT T bilden die Eingabeschicht des Simple Recurrent Network (siehe Abbildung 54). Die Ausgabeschicht bildet der Knoten HEAT t. Dazwischen findet man eine verdeckte Schicht mit den Knoten HID 1, HID 2 und HID 3. Die Anzahl der Knoten in der verdeckten Schicht ergibt sich aus der Anzahl der Eingabeknoten minus einem Knoten. Jeder Knoten der verdeckten Schicht hat jeweils einen Rückkopplungsknoten CON 1 bis CON 3. Jeder Rückkopplungsknoten erhält seine Eingabe aus dem zugehörigen verdeckten Knoten. Die Ausgabe der Rückkopplungsknoten wird jedem Knoten der verdeckten Schicht zugeführt. Wie bei Jordan-Netzen im Kapitel 5.4.2.2.2, bilden die Rückkopplungsknoten das zeitliche Gedächtnis des künstlich neuronalen Netzes.

<sup>69</sup> Für die Bedeutung der Input- und Outputneuronen siehe Tabelle 14.

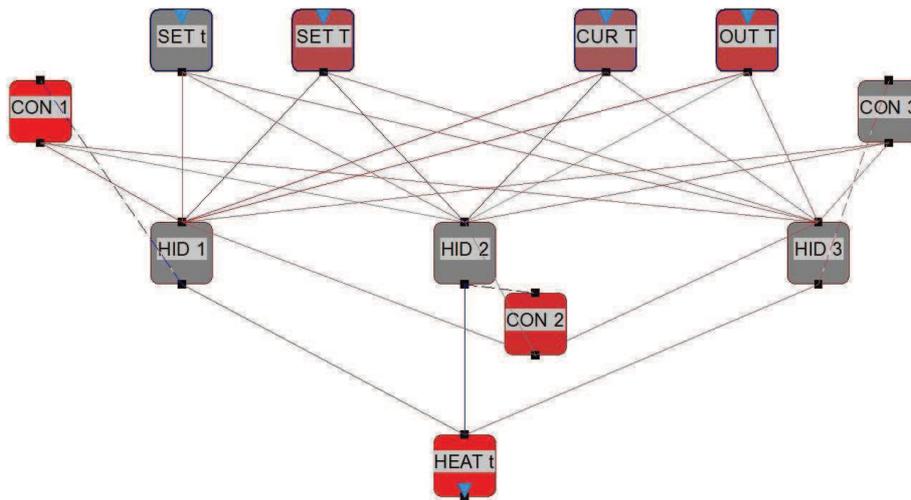


Abbildung 54: Optimum Start-Stop Simple Recurrent Network

### 5.3.3 Jordan Network

In der Eingabeschicht des Jordan Netzes (siehe Abbildung 55) befinden sich vier Eingänge und ein Ausgang, wie in Tabelle 14 angeführt. In der verdeckten Schicht findet man 3 verdeckte Neuronen, die sich aus der Anzahl der Eingabeneuronen minus 1 Neuron ergibt. Das Jordan Netz hat einen Rückkopplungsknoten mit der Beschriftung CON 1. Der Rückkopplungsknoten erhält seine Eingabe über eine indirekte Rückkopplung aus der Ausgabe des Output-Neurons HEAT t und zusätzlich über eine direkte Rückkopplung. Die Ausgabe des Rückkopplungsknotens wird über eine indirekte Rückkopplung an die Eingänge der verdeckten Neuronen der verdeckten Schicht rückgeführt. Dadurch wird ein zeitliches Gedächtnis über vorangegangene Ereignisse während des Normalbetriebs modelliert.

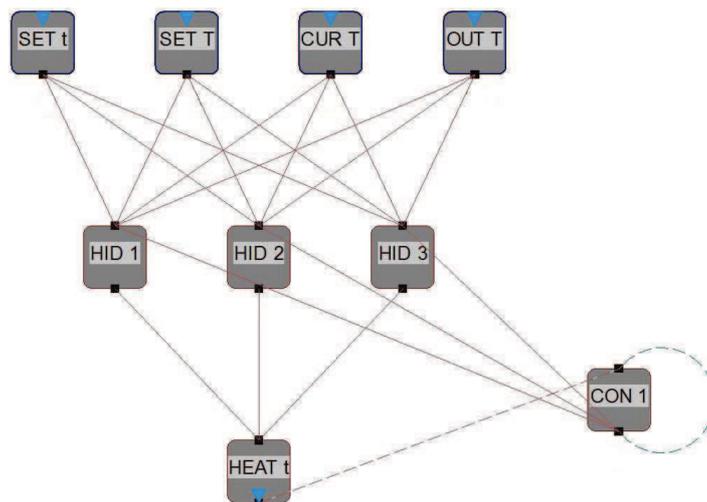


Abbildung 55: Optimum Start-Stop Jordan Netz

## 5.4 Simulation

Für den Vergleich der künstlich neuronalen Netze mit einer statischen Vorheizzeit wurde ein Simulator für den optimalen Start-Stopp-Zeitpunkt der Raumheizung in Microsoft C# .NET 4.0 geschrieben. Evaluiert werden soll, ob künstlich neuronale Netze im Vergleich zu einer statischen im Vorfeld ermittelten<sup>70</sup> Vorheizzeit von 90 Minuten den Raum auch aufheizen können, sodass die Raumtemperatur im Raumtemperaturtoleranzbereich liegt. Gleichzeitig wird getestet, ob bei der Verwendung von künstlich neuronalen Netzen Heizkosten gespart werden können.

### 5.4.1 Der Simulationsablauf

Die Simulation ist in die 3 Phasen Initialisierungs-, Trainings- und Testphase aufgeteilt, wie dies in Abbildung 56<sup>71</sup> dargestellt ist. Zuerst wird die Simulationskonfigurationsdatei geladen, die nicht zwingend vorhanden sein muss.

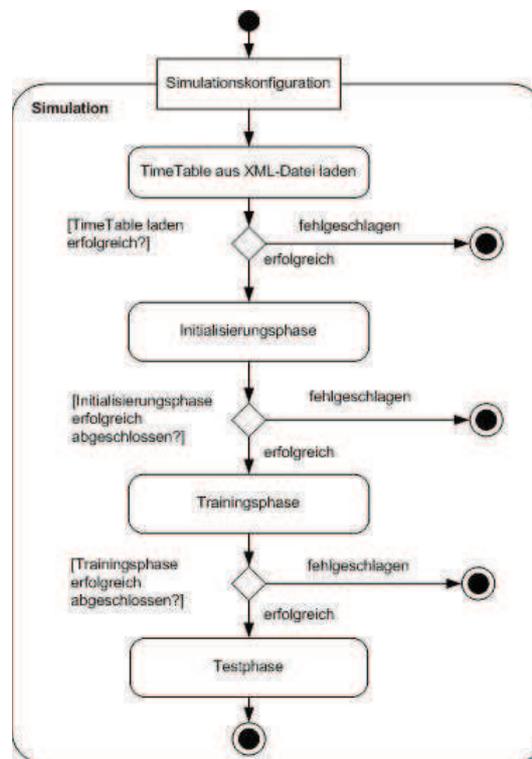


Abbildung 56: Übersicht über den Simulationsablauf

#### 5.4.1.1 Die Simulationskonfiguration: SimConfig.xml

In der Simulationskonfigurationsdatei, die in Abbildung 57 zu sehen ist, befinden sich die Pfade und die Namen, der für die in der Evaluierung verwendeten künstlich neuronalen Netze. Im zweiten Block findet man den Pfad und den Dateinamen des Zeitplans sowie die Anzahl der Tage, die nach der Initialisierungsphase als Trainingstage gelten sollen. Den

<sup>70</sup> Ermitteln des Maximums der Vorheizzeiten.

<sup>71</sup> Die UML-Diagramme, die den Simulationsablauf beschreiben, sind als Aktivitätsdiagramme verfasst.

letzten Block bildet der Konfigurationsblock für die Bibliothek ATPlus der Dymosim-Physikmodellsoftware.

```
<?xml version="1.0" encoding="utf-8" ?>
<Configuration Path="ANN">
  <KNNs Path="ANN" FileNameEnding="mbn">
    <KNN ID="1" FileName="FeedForward" NickName="FFN">Feed Forward Network</KNN>
    <KNN ID="2" FileName="SRN" NickName="SRN">Simple Recurrent Network</KNN>
    <KNN ID="3" FileName="Jordan" NickName="JN">Jordan Network</KNN>
  </KNNs>
  <TimeTables Path="TimeTables" FileNameEnding="xml" TrainingsData_FirstDays="14">
    <TimeTable ID="1" FileName="TimeTable">Standard TimeTable</TimeTable>
  </TimeTables>
  <DymosimConfig>
    <ProgramConfig Path="bin_Dymosim" Name="dymosim.exe" Arguments="-f" />
    <DataConfig FileNameEnding="txt" FileNamePattern="{0:yyyy}-{0:MM}-{0:dd} {0:HHmm}" DataPath="Results">
      <InitDataConfig>2007-03-01_0000_INIT</InitDataConfig>
    </DataConfig>
  </DymosimConfig>
</Configuration>
```

Abbildung 57: SimConfig.xml

#### 5.4.1.2 Der Zeitplan in TimeTable.xml

Die Kernkomponente der Simulation ist der Zeitplan TimeTable.xml. Schlägt das Laden des Zeitplans fehl, dann wird die komplette Simulation abgebrochen. Werden die einzelnen Simulationsphasen nicht fehlerfrei durchlaufen, wird das Simulationsprogramm ebenfalls abgebrochen, da die erzeugten Daten der einzelnen Phasen aufeinander aufbauen.

Die Zeitplan-Datei besteht aus dem Block *SimulationInput*, der die Blöcke *Temperatures*, *RoomNames* und *OccupationTimeTable* beinhaltet.

Der Block *Temperatures* legt die Energiespartemperatur und die Komforttemperatur fest. Beinhaltet sind die Maßeinheit, die jeweiligen eindeutigen Bezeichner sowie die Kurzbezeichnungen für die Temperaturen.

In *RoomNames* werden die in ATPlus vorhanden Räume mit ihrer Bezeichnung im Physikmodell eingetragen.

Das Herzstück dieser Simulation ist die *OccupationTimeTable*, die angibt, wann welche Räume wie warm sein sollen. Die Zeittafel setzt sich aus 28 Tagesblöcken zusammen, die wiederum in 1 bis beliebig viele Zeitblöcke geteilt werden können. In den Zeitblöcken befindet sich die Information, welcher Raum welche Temperatur haben soll.

Das Datumsformat für das Date-Attribut ist in C# nach dem Muster *yyyy-MM-dd* formatiert. Das Zeitformat für das Time Attribut hat als Formatmuster *HH:mm*.

```
<?xml version="1.0" encoding="utf-8" ?>
<SimulationInput>
  <Temperatures ValueUnit="Celsius">
    <Temperature ID="1" Name="ComfortTemperature" ShortName="CT" Value="22" />
    <Temperature ID="2" Name="EconomicTemperature" ShortName="ET" Value="16" />
  </Temperatures>
  <RoomNames>
    <RoomName ID="1" Name="Room_1" DymolaVarName="SetTemp_Room1.k[1]" />
    <RoomName ID="2" Name="Room_2" DymolaVarName="SetTemp_Room2.k[1]" />
  </RoomNames>
  <OccupationTimeTable>
    <Day Date="2007-03-01">
      <Time Time="00:00">
        <RoomTemp ID="1">ET</RoomTemp>
        <RoomTemp ID="2">ET</RoomTemp>
      </Time>
      ...
    </Day>
    ...
  </OccupationTimeTable>
</SimulationInput>
```

Abbildung 58: TimeTable.xml

### 5.4.1.3 Initialisierungsphase

Die erste Phase der Simulation ist die Initialisierungsphase. Das Ziel dieser Phase ist, die thermische Gebäudephysik, die mit ATPlus simuliert wird, in einen eingeschwungenen Zustand überzuführen. Das bedeutet, dass die Solltemperatur in beiden Räumen, die bei Simulationsbeginn 0°C kalt sind, auf 25°C gestellt wird und dann sieben Tage nicht verändert wird. Somit heizt sich die Gebäudestruktur auf mindestens Komforttemperatur auf. Dadurch wird vermieden, dass die künstlich neuronalen Netze in ihrer Trainingsphase Initialisierungsaufheizzeit mitberücksichtigen. Der Einsatz der künstlich neuronalen Netze liegt im Normalbetrieb, wenn die Gebäudestruktur eingeschwungen ist.

Die Initialisierungsphase besteht aus einem unterbrechbaren Aktivitätenpool. Darin wird die Trainingszeitspanne ausgelesen und die Eingabedatei von ATPlus dementsprechend konfiguriert. Anschließend wird der Simulationsprozess angestoßen, der jederzeit mit einem Fehler abgebrochen werden kann. Die Unterbrechung kann vom Abbrechen-Knopf des GUI-Interfaces ausgelöst werden oder ist von einem Timeout *SimulationTimeoutTime* bestimmt, der verhindert, dass ATPlus im ungünstigen Fall die interne Abbruchbedingung nie erreicht und dann keine Möglichkeit mehr besteht den ATPlus-Prozess aus dem Simulationsprogramm heraus abzubrechen. Als Ausgabe einer erfolgreichen Phase erhält man eine ASCII-Datei, die als Input für den ersten Durchlauf der Trainingsphase verwendet wird.

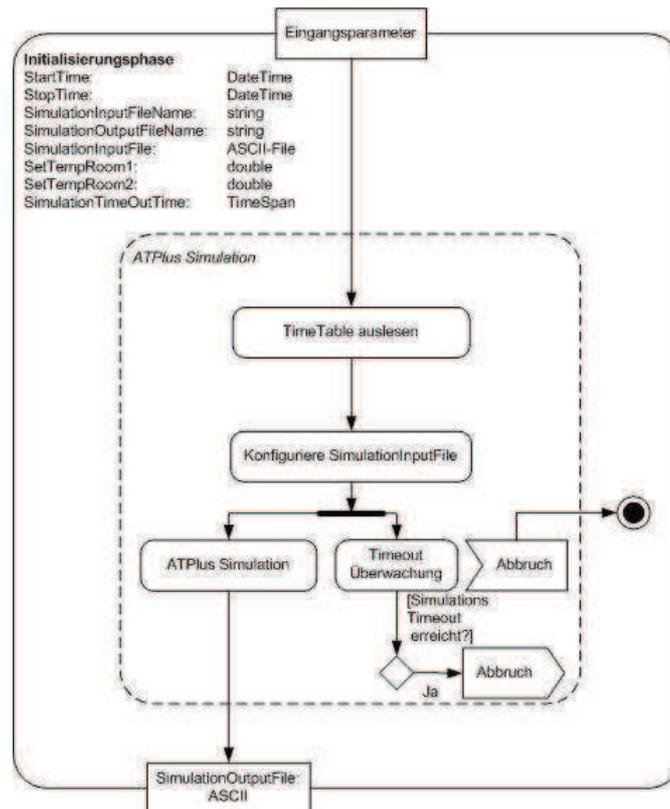


Abbildung 59: Übersicht über die Initialisierungsphase

#### 5.4.1.4 Trainingsphase

Als nächstes folgt die Trainingsphase. In dieser Phase wird der Zeitplan ausgelesen und die Solltemperaturen der einzelnen Räume für den Gebäudesimulator zu dem Zeitpunkt verändert, an dem diese im Zeitplan auftreten. Jeder Simulationsschritt dauert *SimulationStep* lange. Für den in dieser Arbeit durchgeführten Vergleich wurde eine Simulationsschrittweite *SimulationStep* von 300 Sekunden gewählt. Die Trainingsphase ist beendet, wenn der Stoppzeitpunkt *StopTime* durch Erhöhung der Startzeit *StartTime* um die Simulationsschrittweite *SimulationStep* erreicht wurde.

Die Ergebnisse für das Training der künstlich neuronalen Netze werden für jeden Raum individuell festgehalten. Ein Eintrag wird erstellt, nachdem gemessen wurde, wie lange es ab dem Einstellen der Solltemperatur dauert, dass die Raumtemperatur des jeweiligen Raumes im Toleranzbereich liegt. Als Toleranzbereich wurde beim Aufheizen von der Energiespartemperatur auf Komforttemperatur  $20^{\circ}\text{C} \leq T \leq 22^{\circ}\text{C}$  und beim Absenken von der Komforttemperatur auf die Energiespartemperatur  $16^{\circ}\text{C} \leq T \leq 16,5^{\circ}\text{C}$  gewählt. Das hat den Grund, dass Räume schneller auskühlen als aufheizen und es vorkommen kann, dass der Sollwert noch vor dem Erreichen der vorangegangenen Sollwertvorgabe zu oft geändert wird und dadurch zu wenige Trainingsdaten gesammelt werden. Der Fall, dass der Istwert zwischen zwei Sollwertvorgaben nicht erreicht werden kann, wird ebenfalls festgehalten. In diesem Fall wird die Zeitspanne zwischen den beiden Sollwerttemperaturänderungen gemessen.

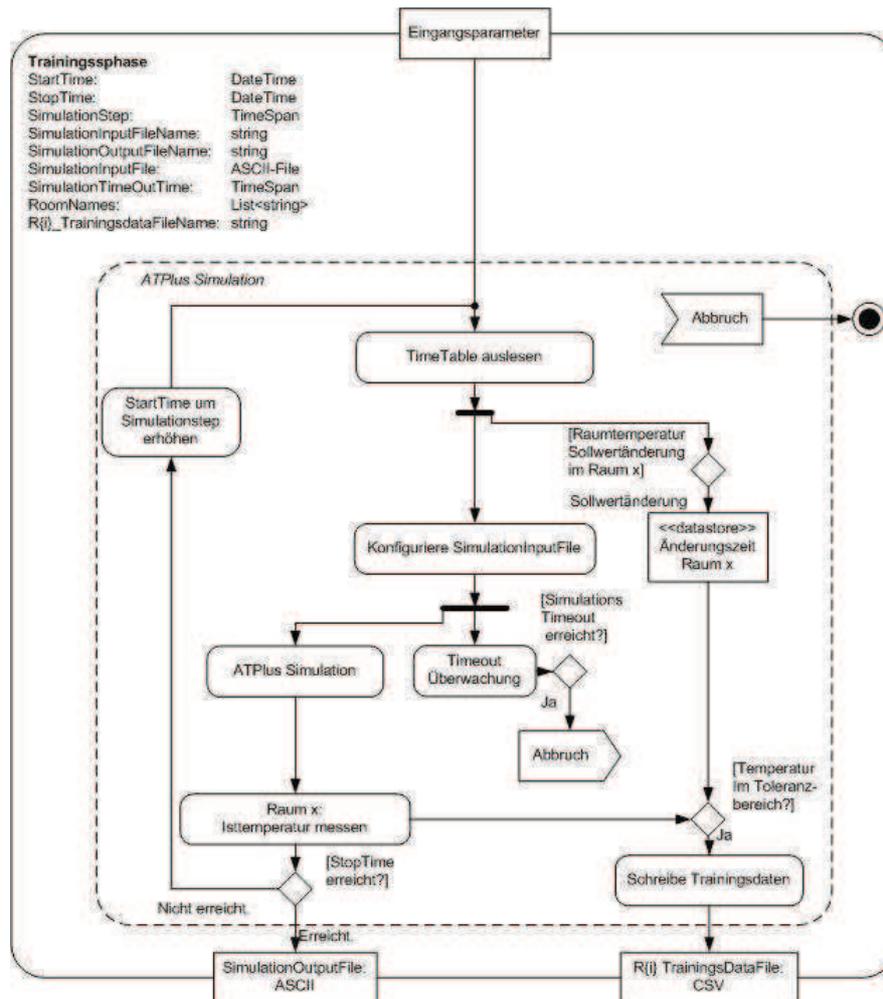


Abbildung 60: Trainingsphase

Trainingsdateien für jeden Raum beinhalten die Felder<sup>72</sup>, die mit den Input- und Outputneuronen, wie sie in Tabelle 14 zu sehen sind, korrespondieren.

Wurde die Trainingsphase im Simulator abgeschlossen, dann müssen die künstlich neuronalen Netze trainiert werden. Dafür wird für jeden Raum jeweils ein künstlich neuronales Netz verwendet. Das Training erfolgt in MemBrain, analog wie im Kapitel 4.2.5.2.1 beschrieben.

### 5.4.1.5 Testphase

In der Testphase werden die einzelnen Entscheidungen der künstlich neuronalen Netze mit einer Referenz verglichen. Wie auch schon bei der Trainingsphase beginnt die Phase zu einem bestimmten Startzeitpunkt und endet durch Hinzugeben einer Zeitspanne *SimulationStep* für einen Simulationsschritt von 5 Minuten. Der Startzeitpunkt *StartTime* beginnt nach Ende der Trainingszeit.

<sup>72</sup> SET t ist als 4-stellige Dezimalzahl im Bereich von  $0,000 \leq t \leq 23,9999$  angegeben. SET T, CUR T und OUT T sind 2 stellige Dezimalzahlen und HEAT t als eine Zeitspanne  $0 \leq t$  in Sekunden ausgedrückt

Zu jedem Simulationsschritt wird in der Raumbelegungstabelle *TimeTable* nachgesehen, ob ein Raum in den nächsten 90 Minuten benützt wird. Falls ja, werden die jeweiligen künstlich neuronalen Netze ab diesem Zeitpunkt bis zum optimalen Temperaturänderungszeitpunkt mit den aktuellen Umgebungsbedingungen der jeweiligen Räume konsultiert. Als Ergebnis liefern die künstlich neuronalen Netze die Aufheizzeit der Räume, womit man direkt auf den optimalen Startzeitpunkt für den Temperaturübergang von der Energiespartemperatur auf Komforttemperatur schließen kann. Ist der optimale Startzeitpunkt erreicht, wird die Solltemperatur für den jeweiligen Raum im Physikmodell geändert. Soll die Temperatur von Komforttemperatur auf Energiespartemperatur abgesenkt werden, wird der Übergangszeitpunkt aus dem Belegungsplan als optimaler Startzeitpunkt für die Temperaturabsenkung verwendet.

Für die Referenzmessung wird die Solltemperatur bei einer Änderung von Energiespartemperatur auf Komforttemperatur 90 Minuten im Vorhinein geändert.

Die Physiksimulation mit ATPlus läuft analog wie in der Physiksimulation der Trainingsphase, jedoch für jedes verwendete künstlich neuronale Netz und für die Referenzmessung getrennt ab. Informationen über den Zeitpunkt der Sollwertänderung, den neuen Sollwert, die aktuell vorliegende Raumlufttemperatur, die Außentemperatur und die Brennerlaufzeit, werden für jeden Raum und für jedes verwendete künstlich neuronale Netz und die Referenzmessung getrennt gesammelt. Einen Überblick über die Abläufe in der Testphase findet man in Abbildung 61.

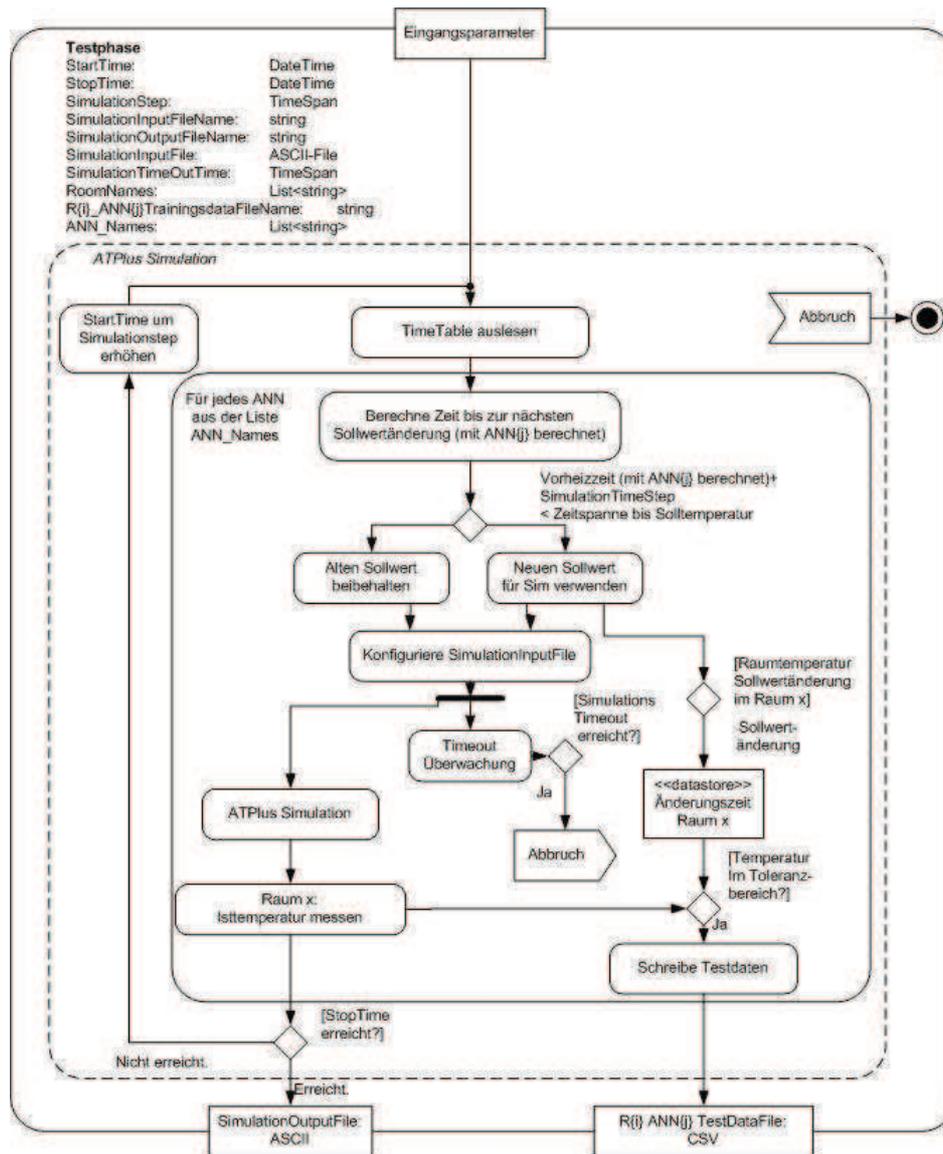


Abbildung 61: Testphase

## 5.4.2 Auswertung der Simulationsergebnisse

In diesem Abschnitt werden die Simulationsergebnisse in grafischer Form aufbereitet dargestellt.

### 5.4.2.1 Trainingsphase

Die Trainingsphase startet nach der Initialisierungsphase mit dem Datum 8.3.2007, um 0:00 Uhr und endet nach 14 Tagen am 21.3.2007 um 23:59 Uhr.

Auf der Primärachse sind in der Farbe Blau der Temperatursollwert SET T und in roter Farbe die aktuelle Temperatur CURT im betrachteten Raum aufgetragen. Die Sekundärachse skaliert in grüner Farbe die Brennerlaufzeit, welche für beide Räume gleich ist. Auf der x-Achse ist die Zeit aufgetragen.

In Abbildung 62 und Abbildung 63 sieht man die Temperatur- und Brennerlaufzeitverläufe. Man kann sehr gut das Temperaturreauschen in der Aufheizphase erkennen, welches direkt von ATPlus erzeugt wird. Außerdem kann man erkennen, dass wenn die Räume nicht auf

Komforttemperatur geheizt werden, die Aufheizzeit nachfolgender Komforttemperaturbereichsphasen länger dauert. Konkret kann man das in Abbildung 62 für Raum 1 sehen, wo das Physikmodell vom 16.3.2007 7:30 Uhr bis 17.3.2007 19:00 Uhr auf Energiespartemperatur betrieben wird.

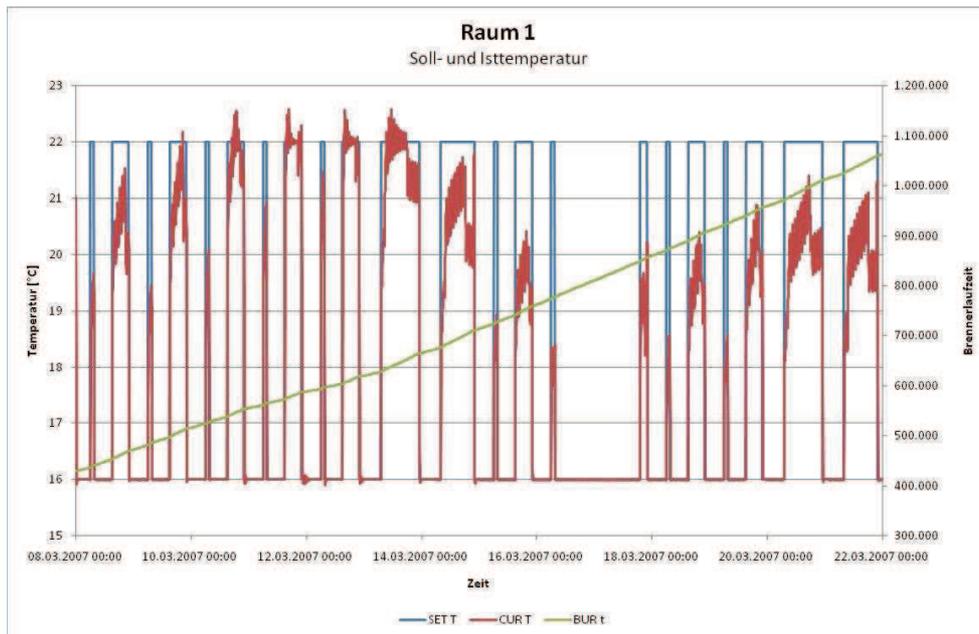


Abbildung 62: Ergebnisse der Trainingsphase für Raum 1<sup>73</sup>

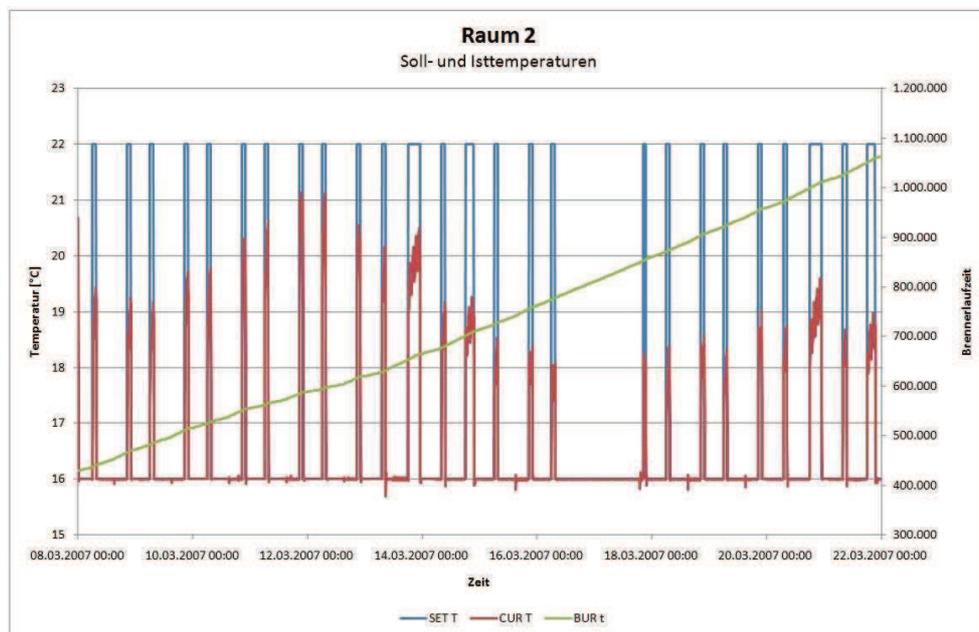


Abbildung 63: Ergebnisse der Trainingsphase für Raum 2

<sup>73</sup> In jeder Grafik findet man ein Rauschen auf dem Signal CUR T. Dieses lässt sich auf den verwendeten Simulator ATplus zurückführen.

### 5.4.2.2 Testphase

Die Testphase beginnt am 22.3.2007 um 0:00 Uhr und endet am 28.3.2007 um 23:59 Uhr. In der Farbe Blau ist die Solltemperatur ab dem Zeitpunkt eingezeichnet, wo entweder die prognostizierte Zeitspanne laut konsultierten künstlich neuronalen Netzen für die Aufheizzeit beginnt oder 90 Minuten vor dem eigentlichen Zeitpunkt der laut Zeitplan für die Referenzmessung eingetragen ist. In Rot ist die aktuelle Raumtemperatur eingezeichnet. Auf der Sekundärachse findet man die Brennerlaufzeit in grüner Farbe eingezeichnet.

#### 5.4.2.2.1 FeedForward Network

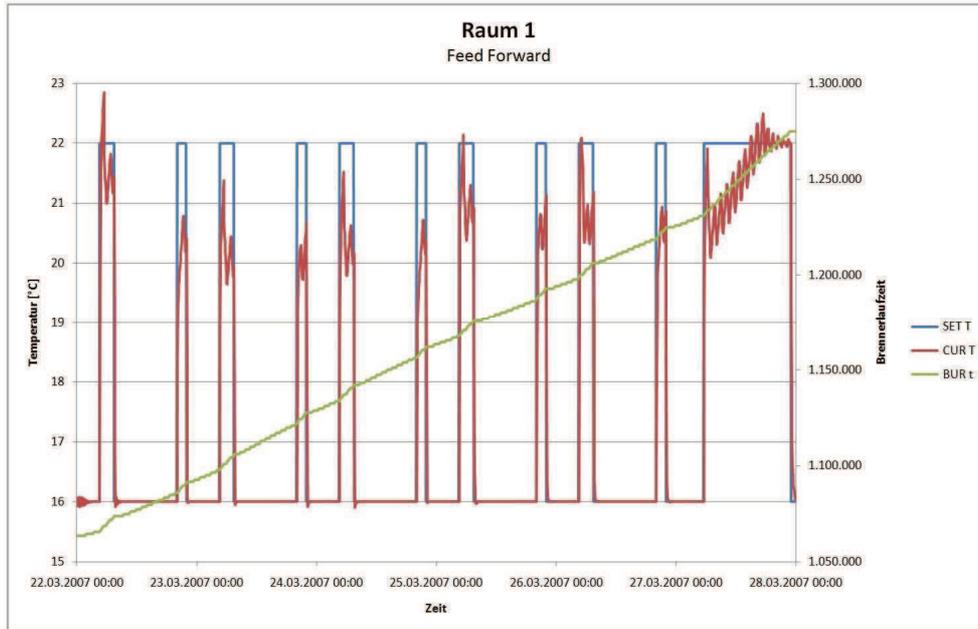


Abbildung 64: Testphase - FeedForward Network für Raum 1

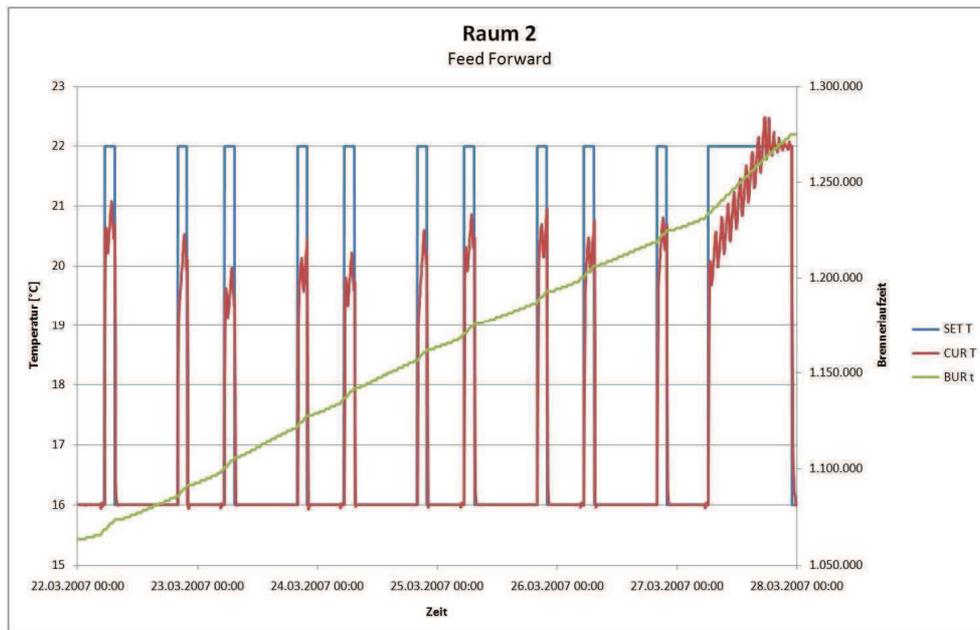


Abbildung 65: Testphase - FeedForward Network für Raum 2

5.4.2.2.2 Jordan Network

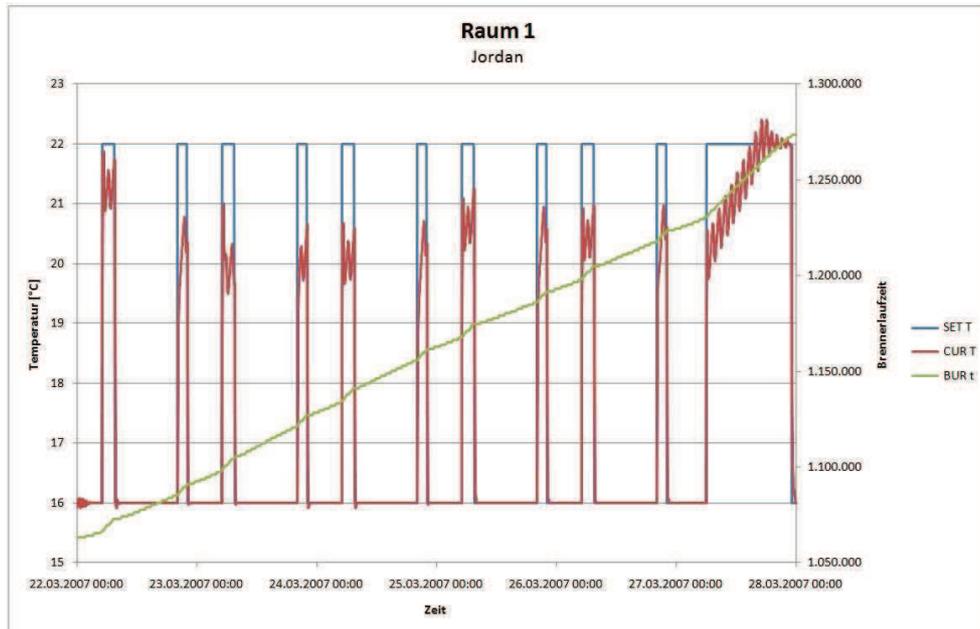


Abbildung 66: Testphase - Jordan Network für Raum 1

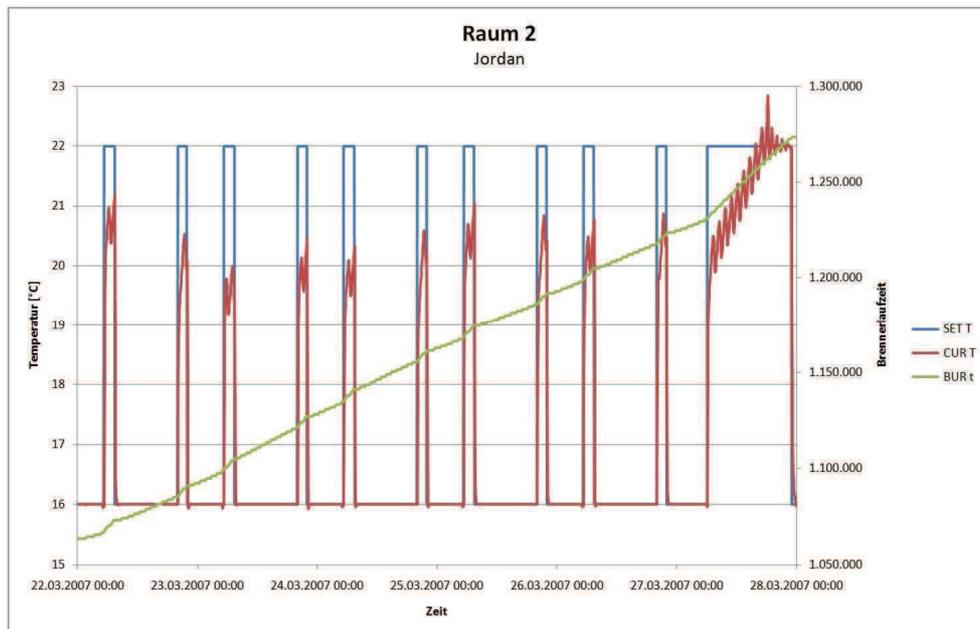


Abbildung 67: Testphase - Jordan Network für Raum 2

### 5.4.2.2.3 Simple Recurrent Network

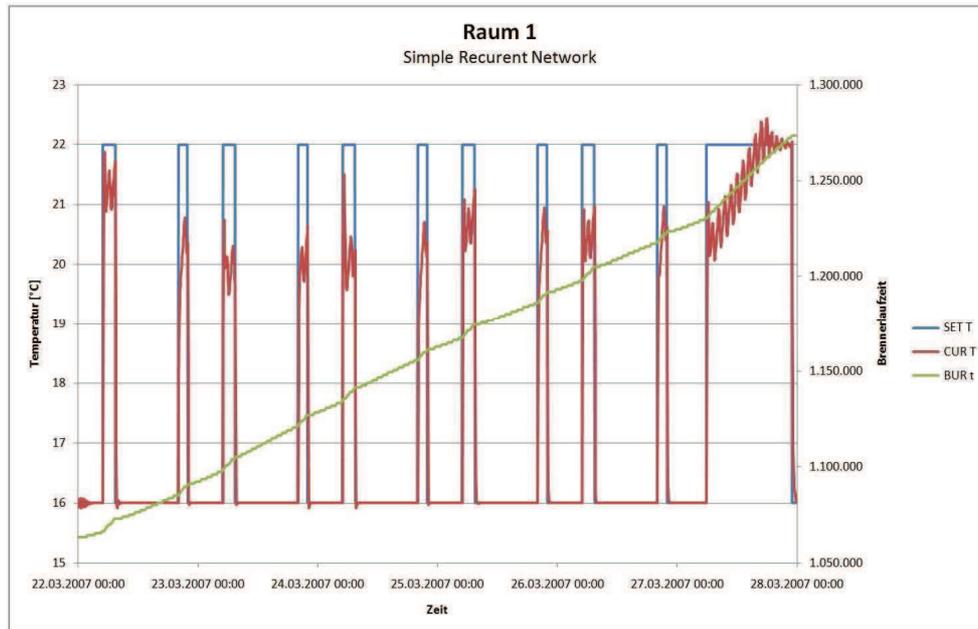


Abbildung 68: Testphase - Simple Recurrent Network für Raum 1

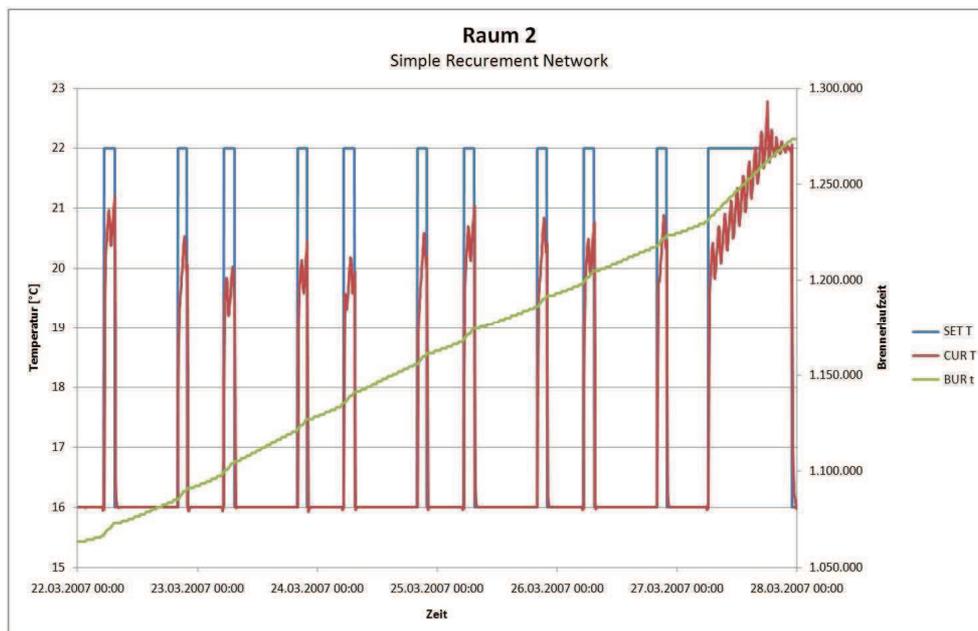


Abbildung 69: Testphase - Simple Recurrent Network für Raum 2

5.4.2.2.4 Referenzmessung

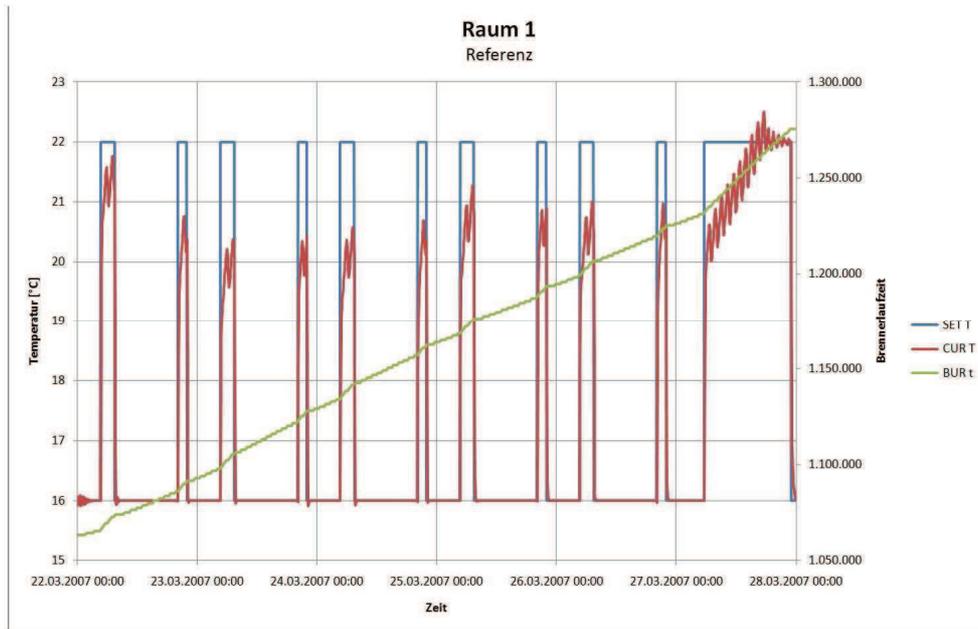


Abbildung 70: Testphase - Referenz für Raum 1

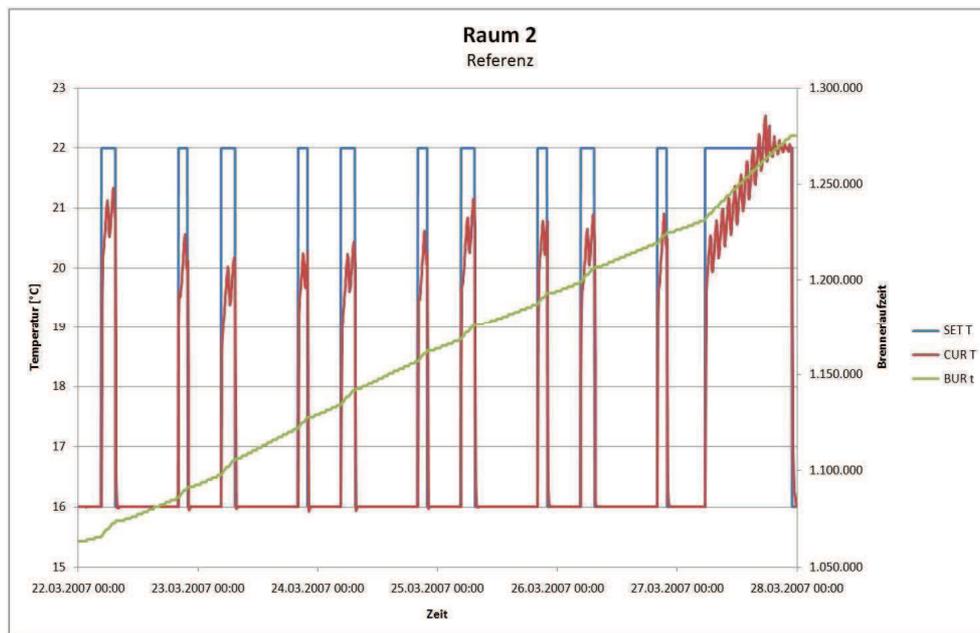


Abbildung 71: Testphase - Referenz für Raum 2

### 5.5 Schlussfolgerungen über die Simulationsergebnisse

Die Brennerlaufzeiten sind in der Abbildung 72 und Tabelle 15<sup>74</sup> zusammengefasst. Auf der y-Achse ist die Brennerlaufzeit in Sekunden aufgetragen und auf der x-Achse die jeweiligen Strategien für den optimalen Start-Stop-Zeitpunkt.

Die Betriebsdauer ist die Brennerlaufzeit, ab dem 1.3.2007 0:00 Uhr, dem Simulationsbeginn und dem Beginn der Initialisierungsphase des Physikmodells bis zum Ende der Testphase. In der Spalte Einsparung werden die Sekunden im Vergleich zur Referenz gerechnet. In der letzten Spalte findet man die relative Einsparung gemessen an der Referenz.

Man kann erkennen, dass jene Netzwerke mit Gedächtnisfunktion, wie beispielsweise das Jordan Network und Simple Recurrent Network, bei der Gegenüberstellung am besten abschneiden. Aber auch mit dem FeedForward Netz lassen sich Energiekosteneinsparungen erkennen.

Um das Rauschen der aktuellen Temperaturwerte CUR T zu eliminieren oder zu reduzieren, sollte für zukünftige Simulationen eine andere Simulationssoftware oder -bibliothek verwendet werden.

	Brennerbetriebsdauer [sec]	Einsparung [sec]	Einsparung [%]
<b>Jordan Network</b>	1273490	1883	0,15%
<b>Simple Recurent Network</b>	1273574	1799	0,14%
<b>Feed Forward Network</b>	1274767	606	0,05%
<b>Referenz</b>	1275373	0	0,00%

Tabelle 15: Gegenüberstellung der Brennerlaufzeiten

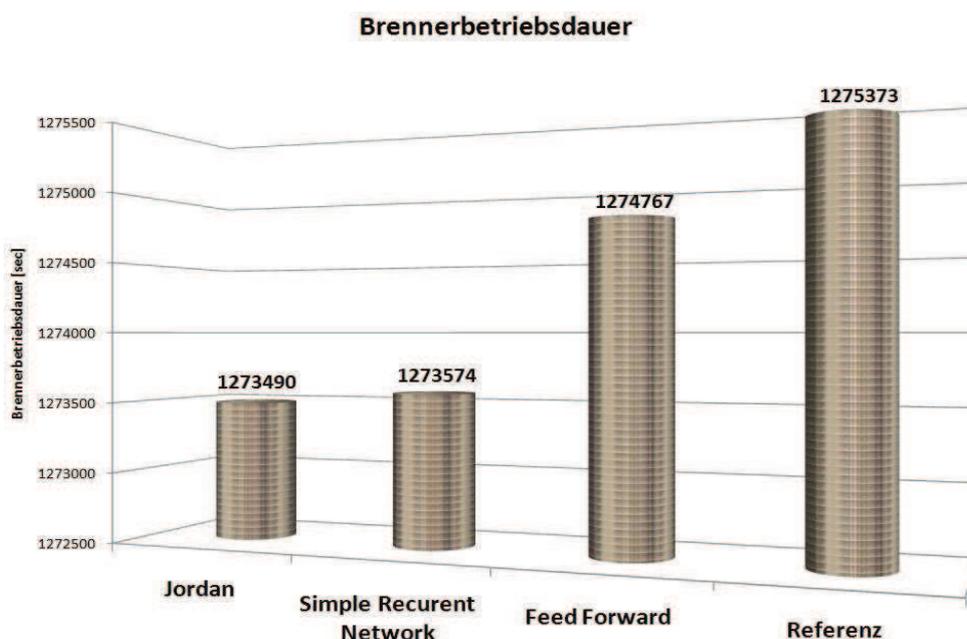


Abbildung 72: Gegenüberstellung der Brennerlaufzeiten

<sup>74</sup> Die Einträge sind nach Einsparung der Brennerlaufzeit geordnet.

## 6 Multiagentensystem für die Heim- und Gebäudeautomation

Eine ubiquitäre Gebäudetechniksteuerung, unter Einbeziehung der Präferenzen der Bewohner eines Hauses und des Energieverbrauchs sowie sonstigen Nebenbedingungen, ist eine große Herausforderung für das Softwareengineering. Um die unterschiedlichen Gewerke mit der Gebäudetechniksteuerung abzudecken, eignen sich Multiagentensysteme [35], um den komplexen Ansprüchen Genüge zu tragen.

Ein Multiagentensystem besteht aus einzelnen Softwareagenten. Daher werden an dieser Stelle Softwareagenten näher betrachtet. Begonnen wird mit den Eigenschaften von Softwareagenten und anschließend wird ein Heimautomations-Multiagentensystem vorgestellt.

### 6.1 Einführung

In diesem Abschnitt wird ein kurzer Einblick in die Themen des Softwareengineerings zu Softwareagenten und ihrer komplexeren Form für Multiagentensysteme (MAS) gegeben. Bei MAS handelt es sich um ein Abstraktionskonzept, ähnlich der objektorientierten Programmierung. Mit diesem lassen sich Systeme durch Konzepte spezifizieren, die dem menschlichen Denken und Handeln nachempfunden sind.

#### 6.1.1 Softwareagenten

Bei Softwareagenten handelt es sich um Programme, die auf einem oder mehreren Rechnereinheiten laufen und stellvertretend für Personen und Systeme operieren. Am besten stellt man sich Softwareagenten als menschliche Akteure vor. Wie ihre natürlichen Vorbilder können Softwareagenten ihre Umgebung über geeignete Schnittstellen wahrnehmen und verändern und mit anderen Agenten kommunizieren.

##### 6.1.1.1 Eigenschaften von Softwareagenten

Die Definition von Softwareagenten ist nicht standardisiert, jedoch findet man in der Literatur meist die folgenden kennzeichnenden Eigenschaften von Softwareagenten: Autonomie, Reaktivität und Proaktivität, Kommunikativität, Lernfähigkeit und Intelligenz, Interaktivität, Mobilität und stationäre Agenten.

Softwareagenten können ohne externe Einwirkungen autonom entscheiden, welche Aktion oder Abfolge von Aktionen anhand der vorliegenden Kriterien ausgeführt werden sollen, um ihre Ziele zu erreichen.

Unter Reaktivität bei Softwareagenten versteht man, dass diese auf Änderungen der Umgebung reagieren. Die Veränderungen werden mit geeigneten Sensoren wahrgenommen und die Reaktion auf Basis von Regeln eingeleitet. Reagieren Softwareagenten nicht nur auf ihre Umgebung, sondern verändern diese auch aufgrund von internen Parametern oder des aktuellen Zustandes, dann spricht man von Proaktivität - Eigenschaften.

Einzelne Agenten müssen mit ihrer Umgebung und eventuell mit anderen Agenten oder ganzen Agentensystemen kommunizieren können. Damit jeder Agent andere Agenten versteht, wird eine gemeinsame Sprache die Agent Communication Language ACL [36] als gemeinsame Interaktions- und Kommunikationsprotokoll verwendet. Eine gemeinsame Weltansicht wird mit einer Ontologie beschrieben.

Softwareagenten können auf verschiedene Art und Weise lernen, wenn eine geeignete Einheit zur Wissensrepräsentation und Auswertung in einem Agenten vorhanden ist. Die einfachste Form ist die Verwendung von vorhandenem Wissen für die eigene Wissensbasis. Während der Laufzeit führen Agenten Anweisungen aus, die nach ihrer Ausführung intern durch geeignete Regeln oder extern von Benutzern und Systemen auf ihre Güte bewertet werden können.

Die Interaktion mit Benutzern oder anderen Systemen erfolgt über geeignete Schnittstellen. Beispielsweise kann mit Benutzern über grafische oder akustische Schnittstellen kommuniziert werden.

Menschliche Benutzer, aber auch technische Systeme, müssen sich nicht immer am selben Ort aufhalten, daher können Agenten von einer Plattform auf eine andere Plattform migrieren. Dies setzt voraus, dass der Programmcode von einer Plattform auf eine andere übertragen wird und die Zielplattform den Programmcode des Agenten interpretieren kann.

#### **6.1.1.2 Philosophisches BDI-Konzept und das praktische BGP- Konzept für planende Agenten**

Softwareagenten für die Anwendung in der Gebäudeautomation müssen ihre Aufgaben planen. So kann zum Beispiel aus dem Wissen über die Ankunft eines Bewohners im Haus der Vorheizzeitpunkt für die Raumheizung geplant werden. Daher werden planende Softwareagenten für diese Aufgabe benötigt. Softwareagenten handeln ähnlich wie Menschen. Der philosophische Belief, Desire, Intention (BDI) Ansatz [37] beschreibt ein menschlich handelndes Agentenarchitekturmodell. Agenten sind mit Sensoren ausgestattet, die es ihnen erlauben die Umwelt wahrzunehmen und Effektoren, mit denen Agenten Einfluss auf die Umwelt nehmen können. Das heißt, ein Agent akzeptiert Eingaben und liefert mit Hilfe eines BDI-Interpreters Ausgaben.

Die Beliefs, das Wissen über die Umgebung in der ein Agent arbeitet, stellen die Information über den aktuellen Zustand der Welt dar, aus denen ein Agent schlussfolgern kann. Dieses Wissen wird laufend aktualisiert und in einer Wissensdatenbank abgelegt. Die Wissensdatenbank stellt somit den internen Zustand eines Agenten dar.

Die Desires, die Ziele<sup>75</sup> eines Agenten, legen fest, wie sich ein Agent verhält. Im Prinzip handelt es sich bei Desires um eine Menge von Zielen mit bestimmten Vor- und Nachbedingungen, die bei der Auswahl einer Absicht von entscheidender Bedeutung sind.

Die Intentions, die Absichten eines Agenten, werden aus einer hierarchisch angeordneten Plandatenbank ausgewählt, um die Ziele zu erreichen. Ein ausgewählter Plan wird Intention genannt.

---

<sup>75</sup> Nicht unbedingt frei von Konflikten.

Das vorgestellte Modell lässt sich mit den Methoden der Informatik nicht eindeutig umsetzen. Aus diesem Grund wurde ein Belief, Goal und Plans Modell [38] entwickelt. Der Unterschied liegt darin, dass Goals konfliktfreien Desires entsprechen. Aus Beliefs und Goals werden zyklisch Pläne, analog dem BDI-Ansatz, zur Ausführung ausgewählt. Unterschiedlich ist die Intention, die aktuell laufende Pläne darstellt.

### 6.1.1.3 Ontologien

Agenten sind kommunikative<sup>76</sup> Einheiten. Damit jeder Agent dieselbe Ansicht auf die Welt hat, dasselbe Vokabular zur Interaktion untereinander verwendet, Terme gleich interpretiert und die Strukturen eines Ausdrucks gleich sind, werden Ontologien definiert. Mit diesen Punkten haben Softwareagenten das gleiche Verständnis von einer Darstellung oder eines Konzepts. Das hilft die Interoperabilität zu ermöglichen und das Anwendungswissen vom Lösungswissen zu trennen.

### 6.1.1.4 Die FIPA-ACL

Zu klassischen Ontologiebeschreibungssprachen zählen beispielsweise RDF und OWL. Aktuelle Softwareagentensysteme, wie JADE [39], verwenden die Standards der FIPA.

Die FIPA-ACL[40] beschreibt Kommunikationsabläufe zwischen verschiedenen Agenten. Es werden der Aufbau einer Nachricht und die Sprechakte zwischen den Agenten beschrieben.

Eine Nachricht beschreibt die Inhaltsblöcke *Teilnehmer*, *Inhaltsbeschreibung* und *Interaktionskontrolle*. Der Block *Teilnehmer* beinhaltet den Sender und den Empfänger einer Nachricht. Der *Inhaltsblock* beinhaltet die eigentliche Nachricht, in welcher Sprache die Nachricht verfasst und codiert wurde sowie die verwendete Ontologie für die Wissensrepräsentation. Der *Interaktionsblock* listet das verwendete Interaktionsprotokoll, die Kennung der Konversation, die Nachrichtenennung für eine Antwort, den Bezug auf eine vorangegangene Nachricht und eine Zeitangabe, bis wann eine Nachricht erwartet wird, auf.

Die Sprechakte sind beispielsweise REQUEST, die Anforderung einer Aktion, oder QUERY, die Anfrage nach einer Information.

Ein beispielhafter Sprechakt zwischen 2 Agenten ist in Abbildung 73 zu sehen. In diesem Beispiel fordert Agent A die aktuelle Außentemperatur von Agent B an. Die Kommunikationsbereitschaft wird jeweils mit „Hallo.“ eingeleitet. Anschließend wird die eigentliche Anfrage von Agent A gestartet und von Agent B passend beantwortet. Die Kommunikation wird von Agent A mit dem Schlüsselwort „Danke.“ für beendet erklärt.

Die genauen Sprechakte für Agenten, wie zum Beispiel der Ablauf eines REQUESTs, werden in den FIPA-Standards spezifiziert.

---

<sup>76</sup> Ein definierter Kommunikationsablauf zwischen Agenten wird Sprechakt genannt.

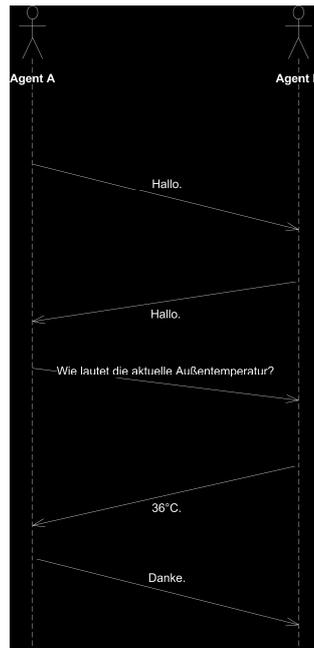


Abbildung 73: Beispiel eines Interaktionsprotokolls

### 6.1.2 Multiagentensysteme

Multiagentensysteme bestehen aus mindestens zwei oder mehreren Softwareagenten, die verschiedene, ähnliche oder gleiche Ziele haben können. Dabei obliegt es den einzelnen Agenten selbst, wie sie eine Aufgabe auf der Basis des Wissens über ihre Umwelt in der sie agieren, erfüllen und ihre Intentionen reagieren, geeignete Pläne zu wählen, um ein konkretes Ziel zu erreichen. Dabei kann das Handeln eines Agents alleine maßgeblich für die Erreichung eines Ziels sein oder aber auch ein Zusammenschluss und eine Abstimmung mehrerer Agenten mit denselben oder ähnlichen Zielen.

## 6.2 Design - und Modellierungswerkzeuge für Multiagentensysteme

### 6.2.1 Design von Multiagentensystemen mit Prometheus

Zum Designen eines Agentensystems reichen die klassischen Methoden des Softwareengineering, wie zum Beispiel Unified Modelling Language [41], nicht aus, da Standard-UML die Agentenmethodologien nicht unterstützt.

Diese Aufgabe haben sich mehrere Arbeitsgruppen zum Thema gemacht. Es gibt viele brauchbare Agentenmodellierungswerkzeuge. In dieser Arbeit wird das auf Java basierende und damit plattformunabhängige frei verfügbare Prometheus Design Tool PDT [42] verwendet. PDT ist ein grafischer Editor für die Modellierung von Multiagentensystemen. Es wird in 3 Phasen vorgegangen - vom Grobdesign bis zum Feindesign. Jede Änderung, die in einer unterliegenden Phase durchgeführt wird, wird in die höheren Abstraktionsschichten propagiert.

Die erste Phase ist die Systemspezifikationsphase, in der die mit dem Agentensystem kommunizierenden Akteure identifiziert werden. Es werden die Schnittstellen zu den

Systemen und den Aktionen und Empfindungen festgelegt. Jedem Agenten werden Ziele vorgegeben und diese in Szenarien beschrieben.

Die zweite Stufe ist das High Level Architectural Design. Es werden die in dem Agentensystem verfügbaren Agentenklassen spezifiziert und eine Übersicht über das gesamte Multiagentensystem inklusive der verwendeten internen Protokolle der gültigen Nachrichtensequenzen modelliert.

Das detaillierte Design, die dritte Modellierungsstufe, beschäftigt sich mit dem Innenleben eines Agenten. Es werden Ereignisse, Pläne und Daten modelliert.

PDT liefert eine XML-Datei mit dem Design und auf Wunsch einen Code-Rumpf für das kommerziell erhältliche Agentenentwicklungswerkzeug JACK [43].

### 6.2.2 Modellierung eines Multiagentensystems mit JADEX

Für die Modellierung eines Multiagentensystems gibt es eine Vielzahl von kommerziellen und frei verfügbaren Werkzeugen. Von besonderem Interesse sind Projekte die frei verfügbar sind und die gewünschten Standards von FIPA einhalten. Eine Agentenkommunikationsplattform, die alle gewünschten Standards des BDI Konzepts implementiert, ist JADEX [44].

Die Plattform JADEX basiert auf dem Open Source Projekt JADE [39]. Beide Plattformen wurden in Java entwickelt JADEX erweitert JADE um das BDI Verhalten und liefert außerdem eine integrierte Programmierumgebung IDE. Die Erweiterung mit JADEX ermöglicht den Agenten zielorientiertes Handeln und die Erzeugung von Zielen und Subzielen, eine Meta-Level Schlussfolgerung und eine Speicherung der Beliefs in einer Wissensdatenbank.

Ein Agent wird in einem sogenannten „Agent Description File“ beschrieben. Dies enthält die Pläne, initiale Ziele, initiales Wissen über die Umgebung und Filter. Diese Informationen werden in einem Property-File gespeichert, mit dem zur Laufzeit Konfigurationsänderungen ermöglicht werden.

Die IDE von JADEX erlaubt eine komfortable Möglichkeit, die internen Nachrichten eines Multiagentensystems zu betrachten und auf Fehler zu testen.

## 6.3 Vorgeschlagene Multiagentenplattform

Eine dynamisch anpassbare lernfähige Multiagentenplattform für die Heimautomation ist in Abbildung 74 zu sehen. In dieser Abbildung ist das System Overview Diagram<sup>77</sup> festgehalten, das mit Prometheus erstellt wurde.

Die Plattform besteht aus 7 Agententypen, wobei jeder Agententyp mit einem oder mehreren konkreten Agenten in der Plattform vertreten sein kann. Im Multiagentensystem befinden sich der Interface Agent, der Presence Agent, der Trend Agent, der Building Analysis Agent, der Control Agent, der Weather Forecast Agent, der oder die User Agents und der Conflict Resolution Agent. Die Ereignisse, Aktionen, Wahrnehmungen, Nachrichten

---

<sup>77</sup> Die Diagrammsymbole sind in Tabelle 16 abgebildet.

und von den Agents verwendete Daten werden in den Agentenbeschreibungen wiedergegeben.

Die vom Multiagentensystem aufgenommenen Wahrnehmungen werden mit *Sensor Change*, *User Interactions* und *Passing of Time* modelliert. Die *Sensor Changes* und *User Interactions* werden vom Interface Agent entgegen genommen. Die Sensoränderungsinformationen sendet das unterliegende Gebäudeautomationsnetzwerk an das Multiagentensystem. Die Benutzerinteraktionen werden von den Bewohnern des Hauses in expliziter Weise mit dem Multiagentensystem ausgelöst. Dies tritt auf, wenn beispielsweise ein Bewohner seine Vorlieben für die Raumtemperatur in einem bestimmten Raum für eine bestimmte Tätigkeit bekannt gibt. *Passing of Time* wirkt auf den Building Analysis Agent, den Control Agent und den User Agent. Damit wird den jeweiligen Agenten die aktuelle Uhrzeit mitgeteilt.

Im Multiagentensystem kommen die Nachrichten *Actuator Change Request Message*, *Sensor Change*, *Owner Desires* und *Current and Expected Future Set Points* vor. Die Nachricht *Actuator Change Request Message*, mit den jeweiligen Stellwerten, wird vom Control Agent an den Interface Agent geschickt, wenn eine Aktuatoränderung gemacht werden soll. Bei den *Sensor Changes* handelt es sich um die geänderten Sensorwerte mit ihrer semantischen Bedeutung. Diese werden vom Interface Agent dem Trend Agent und dem Presence Agent zur Verfügung gestellt. Die *Owner Desires* werden von den jeweiligen Users Agents an den Conflict Resolution Agent gesendet. Dabei handelt es sich um einen Satz von Benutzervorzügen für eine bestimmte Tätigkeit und Energiesparregeln. Die *Current and Expected Future Set Points* werden vom Conflict Resolution Agent an den Control Agent gesendet. Diese beinhalten die Sollwerte für beispielsweise die Helligkeit und Temperatur in einem bestimmten Raum.

Die gesammelten Daten im System sind *Process Image*, *Process Trend*, *Room Behaviour* und Webabfragen *WWW*. Bei den *Process Image* Daten handelt es sich um aktuell vorliegende Sensordaten aus der eingesetzten physikalischen Umgebung, die ständig bei jeder Änderung vom Interface Agent aktualisiert werden. Im *Process Trend* werden die aktuellen Sensordaten gespeichert, so dass sich eine History der Sensorwerte ergibt. Die *Room Behaviour* Daten stellen gesammelte Daten über das dynamische Temperaturverhalten eines Raumes dar. Die Daten aus dem Internet werden vom Weather Forecast Agent benötigt. Es handelt sich also um gesammelte Daten von Internetwetterdiensten für die Region, in der sich das Haus befindet.

In der vorgestellten Agentenplattform werden die Protokolle *Owner Tracking*, *User Feedback* und *Weather Forecast Protocol* verwendet. Protokolle sind verzahnte Nachrichtenabläufe zwischen 2 oder mehrerer Kommunikationspartnern. Das *Owner Tracking Protocol* wird zwischen User Agent und Presence Agent verwendet, um vom betreffenden User Agent den zugehörigen Bewohner in den Räumen des Hauses lokalisieren zu können. Das *User Feedback Protocol (UFP)* wird zum Einen zwischen User Agent und Interface Agent und zum Anderen zwischen Conflict Resolution Agent und Interface Agent ausgeführt. Im ersten Fall dient das UFP zum Lernen der jeweiligen Wohnervorzüge, die der jeweilige User Agent vom Interface Agent entgegen nimmt. Im zweiten Fall dient das UFP zur Auflösung von Konflikten der Präferenzen zwischen User Agents stellvertretend für ihre jeweiligen Bewohner. Wünschen sich zum Beispiel zwei User Agents im selben Raum zur selben Zeit für ihre zugeordneten Bewohner unterschiedliche

Raumtemperaturen, kann das Multiagentensystem die betreffenden Bewohner über den Interface Agent in Kenntnis setzen, dass ein Konflikt aufgetreten ist und eine Konfliktlösung durch die Bewohner anfordern, um den aktuellen Konflikt zu lösen. Die Lösung wird für zukünftige Situationen gespeichert.

Ein Multiagentensystem für ein Heimautomationssystem benötigt die Informationen über die Lage des Hauses, in dem das Multiagentensystem eingesetzt wird, die Aufteilung des Hauses, die verfügbare Gebäudetechnik und deren Installationsorte sowie alle verfügbaren technischen Kommunikationstechnikkomponenten.

Symbol	Bedeutung
	Notiz
	Agent
	Aktion
	Wahrnehmung
	Nachricht
	Daten
	Protocol
	Kante

Tabelle 16: Prometheus System Overview Diagram - Symbole

### 6.3.1 Beschreibung von Softwareagenten

In diesem Abschnitt werden die einzelnen Agentenklassen vorgestellt. Zuerst wird eine verbale Beschreibung über die Funktionen der jeweiligen Agentenklasse geliefert. Anschließend wird eine Übersicht gegeben, wie viele Agenten es in dem Agentensystem geben kann. In den Übersichtstabellen findet man die Abschnitte Lebenszyklus, Kommunikation und Daten. Der Lebenszyklus beschreibt die Initialisierung, die Terminierung und die Lebenszeit der beschriebenen Agentenklasse. Im Abschnitt Kommunikation findet man die Ereignisse, Aktionen und Interaktionen der Agenten. Im Bereich Daten wird aufgelistet, welche Daten ein Agent in eine Datenbank liefert oder verwendet.

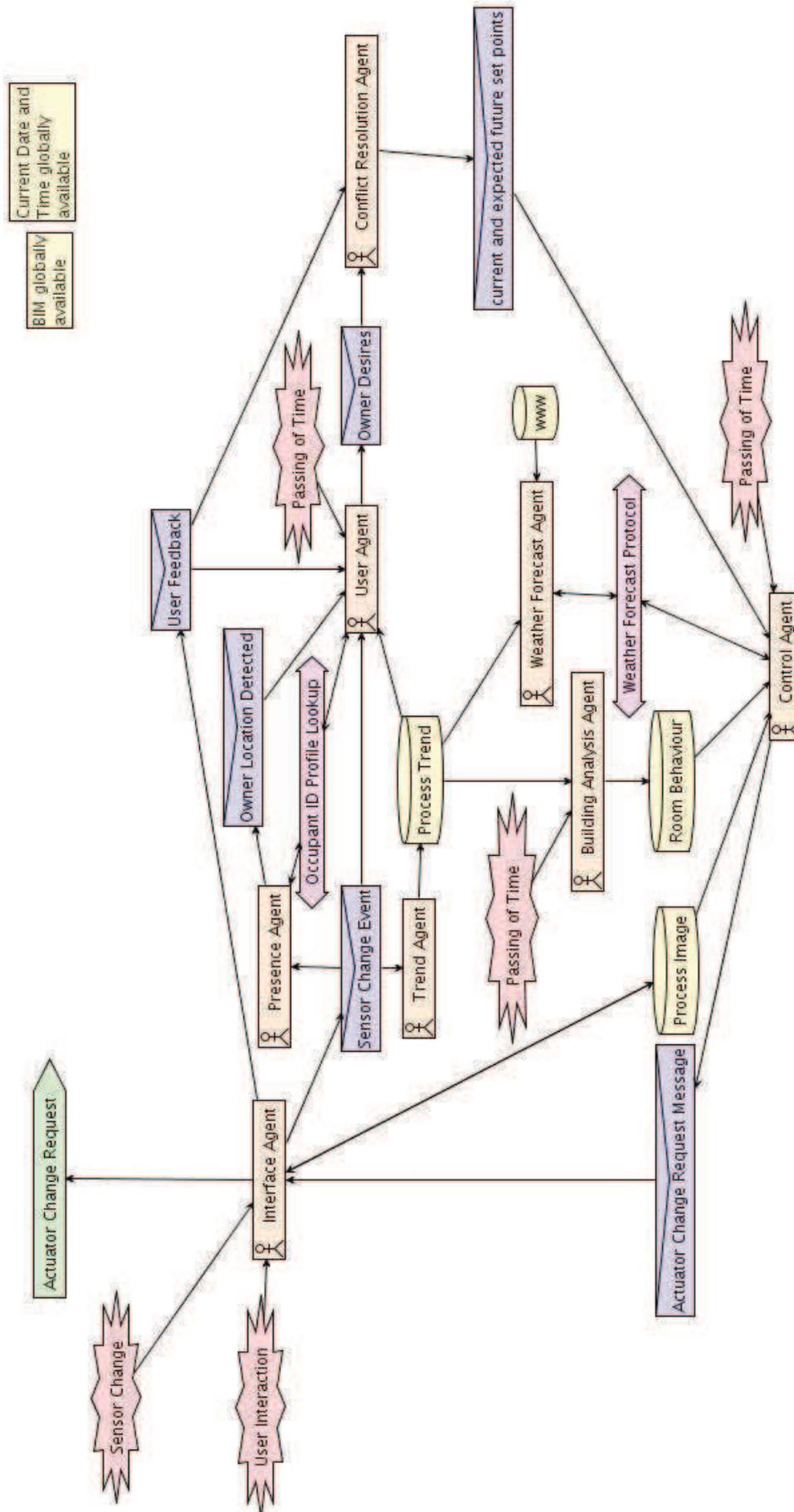


Abbildung 74: Übersicht MAS für die Gebäudesteuerung

### 6.3.2 Interface Agent

Der Interface Agent stellt die Verbindung zwischen der Multiagentenplattform, der Benutzerschnittstelle und der technischer Gebäudeausrüstung<sup>78</sup> her. Der Interface Agent nimmt aktuelle Sensorwerte entgegen und registriert jede Änderung eines Sensorwertes. Gibt das Multiagentensystem eine Sollwertgröße für ein zu steuerndes Gerät vor, dann wird diese über den Interface Agent angenommen und so angepasst, dass die Gebäudeausrüstung die Befehle ausführen kann. Der Interface Agent hält auch das Process Image auf den aktuellen Stand und versendet eine Broadcast-Nachricht über Zustandsänderungen oder Benutzerinteraktionen an die übrigen Agenten im Multiagentensystem.

	Kardinalität	Dieser Agententyp kommt im Agentensystem einmal vor. (=1)
Lebenszyklus	Initialisierung	Wird mit dem Start des Multiagentensystems gestartet.
	Terminierung	Terminiert mit der Beendigung des Multiagentensystems.
	Lebenszeit	Unbegrenzt
Kommunikation	Ereignisse	<ul style="list-style-type: none"> <li>• <i>User Interaction</i>, wenn ein Bewohner mit dem System in bewusster und expliziter Weise interagiert.</li> <li>• <i>Sensor Change</i>, wenn sich ein Sensorwert geändert hat.</li> </ul>
	Aktionen	<ul style="list-style-type: none"> <li>• <i>Action Change Request Message</i>, fordert eine Aktuatoränderung an.</li> </ul>
	Interaktionen	<ul style="list-style-type: none"> <li>• <i>Presence Agent</i>, wird über ein Sensor Change Event informiert</li> <li>• <i>User Agent</i>, informiert über ein Sensor Change Event und einem User Feedback Event</li> <li>• <i>Conflict Resolution Agent</i>, informiert über einen Sensor Change Event und ein User Feedback Event</li> </ul>
Daten	Verwendete Daten	<ul style="list-style-type: none"> <li>• <i>aktuelle Sensordaten</i></li> <li>• <i>Process Image</i></li> </ul>
	Erzeugte Daten	<ul style="list-style-type: none"> <li>• <i>Process Image</i>, diese Daten spiegeln die aktuellen Sensor- und Aktuatorwerte wider.</li> </ul>

Tabelle 17: Datenblatt des Interface Agent

<sup>78</sup> Beispielsweise eine Anbindung an KNX.

### 6.3.3 Trending Agent

Die Aufgabe des Trending Agent besteht darin, bestimmte Sensorwerte und Aktuatorzustände aufzuzeichnen.

	Kardinalität	Dieser Agententyp kommt im Agentensystem einmal vor. (=1)
Lebenszyklus	Initialisierung	Wird mit dem Start des Multiagentensystems gestartet.
	Terminierung	Terminiert mit der Beendigung des Multiagentensystems.
	Lebenszeit	Unbegrenzt
Kommunikation	Ereignisse	Der Trending Agent löst keine Ereignisse aus.
	Aktionen	Der Trending Agent löst keine Aktionen aus.
	Interaktionen	<ul style="list-style-type: none"> <li>• <i>Interface Agent</i></li> </ul>
Daten	Verwendete Daten	<ul style="list-style-type: none"> <li>• <i>Process Image</i>, aktuelle Sensor- und Aktuatorwerte sowie die aktuelle Uhrzeit.</li> </ul>
	Erzeugte Daten	<ul style="list-style-type: none"> <li>• <i>Process Trend</i>- Daten</li> </ul>

Tabelle 18: Datenblatt des Trending Agent

### 6.3.4 Presence Agent

Die Lokalisierung der Personen im Haus wird vom Presence Agent ausgeführt. Dieser fordert die User Agents auf, ihre gespeicherten persönlichen Erkennungsmerkmalprofile ihrer jeweiligen Bewohner zu übermitteln. Der Presence Agent identifiziert anhand der Erkennungsmerkmalprofile die Bewohner. Die Lokalisierungsdaten eines erkannten Bewohners werden an den jeweiligen User Agent gemeldet.

	Kardinalität	Dieser Agententyp kommt im Agentensystem einmal vor. (=1)
Lebenszyklus	Initialisierung	Wird mit dem Start des Multiagentensystems gestartet.
	Terminierung	Terminiert mit der Beendigung des Multiagentensystems.
	Lebenszeit	Unbegrenzt
Kommunikation	Ereignisse	Der Presence Agent löst keine Ereignisse aus.
	Aktionen	Der Presence Agent löst keine Aktionen aus.
	Interaktionen	<ul style="list-style-type: none"> <li>• <i>Occupant ID Profile Lookup Protocol</i>, zum Austausch der Bewohnerprofile mit den <i>Users Agents</i></li> <li>• <i>Owner Location Detected</i>, Lokalisierungsinformationen der identifizierten Bewohner an den zuständigen <i>User Agent</i></li> <li>• <i>Interface Agent</i>, über einen <i>Sensor Change Event</i></li> </ul>
Daten	Verwendete Daten	<ul style="list-style-type: none"> <li>• <i>Process Image</i></li> <li>• <i>Erkennungsmerkmalprofile</i>, der im Multiagentensystem vorhanden User Agents</li> </ul>
	Erzeugte Daten	<ul style="list-style-type: none"> <li>• <i>keine</i></li> </ul>

Tabelle 19: Datenblatt des Presence Agent

### 6.3.5 User Agent

In einem multiagentenbasierten Regelungssystem stehen ein oder mehrere *User Agents* zur Verfügung. Dabei existieren zwei Formen von *User Agents*:

1. *Guest User Agents* und
2. *Resident User Agents*.

*User Agents* stellen dem Presence Agent gespeicherte Profilinformationen über ihre Besitzer zur Verfügung. Die *User Agents*, die einem Benutzer zugeordnet sind, versuchen für ihre jeweiligen zugeordneten Bewohner die aktuelle Tätigkeit, das Sequenzmuster, in dem eine Tätigkeit vorkommen kann und die Wunschbedingungen vorherzusagen. Durch das Feedback der zugeordneten Besitzer kann ein User Agent Tätigkeiten erkennen, Sequenzmuster voraussagen und Wunschbedingungen ableiten lernen.

Personalisierte *User Agents* sind auch in der Abwesenheit ihrer jeweiligen zugeordneten Bewohner aktiv und melden zukünftige Wunschbedingungen an. Dies stellt sicher, dass die Wunschbedingungen des jeweiligen Bewohners beim Betreten eines Raumes im Haus eingestellt sind. Die Abläufe, die an dieser Stelle für personalisierte *User Agents* vorgestellt wurden, gelten auch für generische *User Agents* mit voreingestellten generischen Tätigkeitsmustern, generischen Sequenzmustern und generischen Wunschbedingungen. Die erlernten Eigenschaften werden, nachdem ein Gast das Haus verlassen hat, gelöscht, außer ein Gast wechselt zu der Gruppe der ständigen Bewohner des Hauses<sup>79</sup>.

---

<sup>79</sup> Bei der Inbetriebnahme des Multiagentensystems sind dem System keine Bewohner bekannt und daher werden alle Personen als Gast eingestuft.

	Kardinalität	Dieser Agententyp kommt im Agentensystem mindestens einmal oder beliebig oft vor, jedoch für jeden Bewohner garantiert einmal. ( $\geq 1$ )
Lebenszyklus	Initialisierung	Wird mit beim Betreten des Hauses durch einen unbekanntem Bewohner gestartet.
	Terminierung	Es wird zwischen 2 Fällen unterschieden: <ol style="list-style-type: none"> <li>1. Die Person war ein Gast, dann terminiert der <i>Guest User Agent</i> sofort, nachdem die Gastperson das Haus verlassen hat.</li> <li>2. Die Person ist eine bekannte Person. Dann terminiert der <i>User Agent</i>, wenn eine Person aus dem Haus auszieht<sup>80</sup>.</li> </ol>
	Lebenszeit	Abhängig von der Initialisierung und der Terminierung des jeweiligen <i>User Agents</i> .
Kommunikation	Ereignisse	<ul style="list-style-type: none"> <li>• <i>Passing of Time</i></li> </ul>
	Aktionen	Keine.
	Interaktionen	<ul style="list-style-type: none"> <li>• <i>Sensor Change Event</i> vom <i>Interface Agent</i></li> <li>• <i>Occupant ID Profile Lookup</i> mit dem <i>Presence Agent</i></li> <li>• <i>Owner Desires</i> mit dem <i>Conflict Resolution Agent</i></li> </ul>
Daten	Verwendete Daten	<ul style="list-style-type: none"> <li>• <i>Process Trend</i></li> </ul>
	Erzeugte Daten	Keine.

Tabelle 20: Datenblatt für die User Agents

<sup>80</sup> Das muss dem betreffenden *Resident User Agent* manuell mitgeteilt werden.

### 6.3.6 Conflict Resolution Agent

Befinden sich mehrere Bewohner in einem Raum, dann können Konflikte zwischen den einzelnen *Owner Desires*, repräsentiert durch die einzelnen *User Agents* der jeweiligen Bewohner, auftreten. Es können auch Konflikte zwischen *User Agents* und den Energiesparvorgaben hervorgerufen werden. Die eventuell entstandenen Konflikte werden vom *Conflict Resolution Agent* aufgelöst und *Current and Future Set Points* berechnet.

	Kardinalität	Dieser Agententyp kommt im Agentensystem einmal vor. (=1)
Lebenszyklus	Initialisierung	Wird mit dem Start des Multiagentensystems gestartet.
	Terminierung	Terminiert mit der Beendigung des Multiagentensystems.
	Lebenszeit	Unbegrenzt
Kommunikation	Ereignisse	<ul style="list-style-type: none"> <li>• <i>Keine.</i></li> </ul>
	Aktionen	<ul style="list-style-type: none"> <li>• <i>Keine.</i></li> </ul>
	Interaktionen	<ul style="list-style-type: none"> <li>• <i>User Feedback</i> über den <i>Interface Agent</i></li> <li>• <i>Owner Desires</i> von den <i>User Agents</i></li> <li>• <i>Current and Expected Future Set Points</i> an den <i>Control Agent</i></li> </ul>
Daten	Verwendete Daten	<ul style="list-style-type: none"> <li>• <i>Gebäudestrukturmodell</i></li> </ul>
	Erzeugte Daten	<ul style="list-style-type: none"> <li>• <i>Keine.</i></li> </ul>

Tabelle 21: Datenblatt des Conflict Resolution Agent

### 6.3.7 Control Agent

Der Control Agent kennt die Gebäudestruktur und das jeweilige Verhalten der einzelnen Räume eines Hauses und die aktuellen Wetterdaten sowie die Wetterprognosen für die kommenden Stunden und die nächsten Tage. Aufgrund der *Current and Future Set Points* werden Stellwerte für einen bestimmten Raum berechnet.

	Kardinalität	Dieser Agententyp kommt im Agentensystem einmal vor. (=1)
Lebenszyklus	Initialisierung	Wird mit dem Start des Multiagentensystems gestartet.
	Terminierung	Terminiert mit der Beendigung des Multiagentensystems.
	Lebenszeit	Unbegrenzt
Kommunikation	Ereignisse	<ul style="list-style-type: none"> <li>• <i>Passing of Time.</i></li> </ul>
	Aktionen	<ul style="list-style-type: none"> <li>• Berechnete Stellwerte</li> </ul>
	Interaktionen	<ul style="list-style-type: none"> <li>• <i>Current and Expected Future Set Points</i> vom <i>Conflict Resolution Agent</i></li> <li>• <i>Weather Forecast Protocol</i> mit dem <i>Weather Forecast Agent</i></li> <li>• <i>Actuator Change Request</i> an den <i>Interface Agent</i></li> </ul>
Daten	Verwendete Daten	<ul style="list-style-type: none"> <li>• <i>Room Behaviour</i></li> <li>• <i>Process Image</i></li> </ul>
	Erzeugte Daten	Keine.

Tabelle 22: Datenblatt des Control Agent

### 6.3.8 Weather Forecast Agent

Die aktuellen Wetterdaten und die Wettervorhersage aus den lokalen Wetterdaten einer optional vorhandenen Wetterstation und Wetterdaten von Wetterdiensten aus dem Internet werden vom Weather Forecast Agent berechnet.

	Kardinalität	Dieser Agententyp kommt im Agentensystem einmal vor. (=1)
Lebenszyklus	Initialisierung	Wird mit dem Start des Multiagentensystems gestartet.
	Terminierung	Terminiert mit der Beendigung des Multiagentensystems.
	Lebenszeit	Unbegrenzt
Kommunikation	Ereignisse	<ul style="list-style-type: none"> <li>• <i>Keine.</i></li> </ul>
	Aktionen	<ul style="list-style-type: none"> <li>• <i>Keine.</i></li> </ul>
	Interaktionen	<ul style="list-style-type: none"> <li>• <i>Weather Forecast Protocol</i> mit dem <i>Control Agent</i></li> </ul>
Daten	Verwendete Daten	<ul style="list-style-type: none"> <li>• <i>World Wide Web - Datenabfrage</i></li> <li>• <i>Process Trend</i></li> </ul>
	Erzeugte Daten	Keine.

Tabelle 23: Datenblatt für den Weather Forecast Agent

### 6.3.9 Building Analysis Agent

Der Building Analysis Agent wertet periodisch für jeden Raum individuell und, unter der Einbeziehung von Sensor- und Aktuatorendaten, das Raumverhalten aus und speichert dieses in *Room Behaviour* ab. Damit ist es dem *Control Agent* möglich, die Regelung energieoptimal zu berechnen.

	Kardinalität	Dieser Agententyp kommt im Agentensystem einmal vor. (=1)
Lebenszyklus	Initialisierung	Wird mit dem Start des Multiagentensystems gestartet.
	Terminierung	Terminiert mit der Beendigung des Multiagentensystems.
	Lebenszeit	Unbegrenzt
Kommunikation	Ereignisse	<ul style="list-style-type: none"> <li>• <i>Passing of Time.</i></li> </ul>
	Aktionen	Keine.
	Interaktionen	<i>Keine.</i>
Daten	Verwendete Daten	<ul style="list-style-type: none"> <li>• <i>Process Trend</i></li> </ul>
	Erzeugte Daten	<ul style="list-style-type: none"> <li>• <i>Room Behaviour.</i></li> </ul>

Tabelle 24: Datenblatt für den Building Analysis Agent

## 7 Schlussfolgerungen und Ausblick

Im Rahmen der Diplomarbeit wurde zuerst der Stand der Wissenschaft und Technik zum großen Thema Smart Homes und deren Disziplinen erarbeitet. Bei den technischen Implementierungen und Feldstudien findet man interessante Umsetzungen. Beispielsweise das Produkt myGekko, das eine drahtlose Steuerung und Regelung von Beleuchtungskörpern und Heizung erlaubt. Das Produkt arbeitet auf der Basis von Szenarien, die statisch im System vorliegen. Wenn man über die Einschränkung von vordefinierten Szenarien hinwegsieht, dann hat myGekko ein großes Potential. Schon heute können eine große Anzahl Geräte unterschiedlicher Gewerke mit dem System bedient werden. Ein sehr wichtiges Thema für Smart Homes ist die Erkennung von unterschiedlichen Aktivitäten und Szenarien.

Im weiteren Verlauf dieser Arbeit wurden typische Anwendungsszenarien mit Hilfe von UML modelliert, die Aufschluss über die Komplexität des Zusammenspiels zwischen Menschen, Technik und Umwelt gaben.

Heizungsenergie sparen wird in einem zunehmenden Maß auch ein Thema in Privathaushalten. Das Thema selbstlernende optimierte Kontrollstrategien zur Heizungssteuerung nimmt dies als Anlass. Es wird immer wichtiger, primäre und sekundäre Wärmequellen effizient zu nutzen. Das beginnt bei beim optimalen Heizungsstart. Wie das technisch umgesetzt werden kann, wurde mit Hilfe von multipler Regression und künstlich neuronalen Netzen gezeigt. Die beiden Methoden arbeiten auf der Basis verschiedener physikalisch verfügbaren Größen und ermöglichen eine Schätzung des optimalen Heizungsstartzeitpunktes.

Künstlich neuronale Netze haben sich als beste Methode im Umgang mit unterschiedlichen nichtlinearen Einflussgrößen für die Schätzung der Vorheizzeit erwiesen. Aus diesem Grund wurde das Thema vertieft. Es wurde ein Heizungsregelkreis mit neuronalen Netzen implementiert und Feed Forward, Jordan und Simple Recurrent Networks mit einer klassischen Vorheizzeitschätzmethode in einem Gebäudesimulationsmodell verglichen. Als Fazit der Simulation wurde festgestellt, dass mit künstlich neuronalen Netzen Heizungskosten eingespart werden können.

Das verwendete Simulationsmodell wurde mit der frei erhältlichen Bibliothek ATplus für Modelica umgesetzt. Das Modell besitzt ein sehr starkes Rauschen bei den Istwerten für die Raumtemperatur. Um dieses Rauschen zu minimieren oder zu eliminieren, müsste bei ATplus in zukünftigen Arbeiten auf diesem Gebiet nachgebessert werden. Der Code liegt im Modelica-Format vor und kann modifiziert werden. Sollten die Verbesserungen das Rauschen nicht beseitigen, dann bietet sich als Ausweichmöglichkeit der kommerziell erhältliche Gebäudesimulator TRNSYS [45] an. Dieser erlaubt, ähnlich wie es mit ATplus möglich ist, eine Manipulation der Variablen während eines Simulationsdurchlaufes. Für die Verwendung von TRNSYS müssten die Schnittstellen des erstellten künstlich neuronalen Netze - Controllers angepasst werden

Als nächstes wurde ein Multiagentensystem mit dem Modellierungstool Prometheus entworfen. Es zeigt, welche Agenten in einem Multiagentensystem für die Heim- und Gebäudeautomation enthalten sein müssen und wie diese Agenten interagieren könnten.

Das Multiagentensystem für die Heim- und Gebäudeautomation könnte in zukünftigen Arbeiten beispielsweise mit JADEX implementiert werden. Der Prototyp könnte zunächst eingeschränkte Funktionalität aufweisen. So könnte er zunächst nur für eine Aufgabe zuständig sein, zum Beispiel zur Ansteuerung der Heizung. Im nächsten Schritt könnte das Multiagentensystem mit 2 stationären User Agents ausgestattet werden, deren Vorzüge fest vorgegeben sind. Funktioniert dies, könnten die User Agents mobil gemacht werden, um sie über ein geeignetes Medium von einem Multiagentensystem zu einem anderen zu transferieren. Anschließend sollte man sich dem Lernen der Bewohnerwünsche und in welchem Kontext die Bewohnerwünsche auftreten widmen. Verlaufen die Prototypentests positiv, könnte die Steuerung von weiteren Gewerken in Angriff genommen werden. Dafür müssten auch die User Agents erweitert werden.

Von einer Heimautomationsanlage, die die Steuerung und Regelung aller möglichen in einem Haus vorhandenen Geräte für die Bewohner nach deren Wünschen automatisch erledigt, sind die momentanen Entwicklungen noch einige Schritte weit entfernt. Es fehlen in erster Linie Haushaltsgeräte, die sich über ein Hausautomationssystem ansprechen lassen, und nötige Standards, wie mit den Geräten über eine Computerschnittstelle kommuniziert werden kann. Es gibt zwar bereits Heimautomationsbusse, jedoch beschränken sich diese meist auf die Kommunikation mit Sensoren und Aktuatoren eines klassischen Heimautomationssystems. Die Grenze zu vorhandenen Computernetzen, Unterhaltungssystemen und Haushaltsgeräten muss schwinden, um Smart Homes umsetzen zu können. Sei es mit übergelagerten Systemen, wie das in dieser Diplomarbeit vorgestellte Multiagentensystem, das die verwendeten logischen Kommunikationsnetze über einen Interface Agent abstrahiert, um Regelungs- und Steuerungsaufgaben in mitdenkender Weise für die Bewohner erfüllen zu können.

Natürlich erscheinen Smart Home-Systeme aus heutiger Sicht noch nicht greifbar. Das wird sich in der Zukunft zum einen (hoffentlich) durch neue Standards, durch die es möglich ist, alle Gewerke überspannend Geräte zu regeln und steuern, als auch durch die höhere Akzeptanz der Bauherrn durch die Massenfertigung der Smart Home Produkten ändern. Nicht nur der Anschaffungspreis eines Smart Homes fließt in die Return on Invest Rechnung ein, sondern auch der Betrieb. Außerdem besitzt ein Smart Home so viele Mehrwerte wie beispielsweise die Unterstützung bei der Altenbetreuung, dass der ideelle Wert erst gar nicht mit Zahlen beziffert werden kann.

## 8 Anhang

### I. Abbildungsverzeichnis

Abbildung 1: Generisches Feedforward neuronales Netz.....	9
Abbildung 2: 3 Phasen des strukturierten Szenarienerkennungssystems .....	12
Abbildung 3: Beispiel eines Image Templates.....	13
Abbildung 4: Beispiel Scenario Template .....	14
Abbildung 5: Implementierungsphasen des Hidden-Markov-Modells.....	14
Abbildung 6: Beispiel eines Hidden-Markov-Model.....	15
Abbildung 7: Amigo Middleware Architektur .....	18
Abbildung 8: Lernen mit Benutzerrückmeldung .....	20
Abbildung 9: Systemarchitektur von MASBO.....	21
Abbildung 10: Verarbeitung von Sensordaten in MASBO.....	22
Abbildung 11: Funktionelle Architektur von MASBO .....	23
Abbildung 12: CACHE Regelschleife .....	34
Abbildung 13: CACHE Architekturübersicht .....	34
Abbildung 14: Anwendungsfall: Automatische Raumtemperaturregelung nach einem eingelernten Belegungsplan .....	39
Abbildung 15: Aktivitätsdiagramm zur Raumtemperaturregelung .....	41
Abbildung 16: Aktivitätsdiagramm für den Raumtemperaturregler.....	42
Abbildung 17: Sequenzdiagramm für das thermische Wohlbefinden eines Bewohners in einem Raum.....	44
Abbildung 18: Anwendungsfall zur automatischen Raumtemperaturregelung für einen Mehrpersonen Haushalt mit Konfliktlösung.....	46
Abbildung 19: Sequenzdiagramm zur Auflösung eventuell auftretender Raumtemperaturkonflikte zwischen den Bewohnern.....	48
Abbildung 20: Anwendungsfall zu Aufstehen am Morgen während der Arbeitswoche .....	52
Abbildung 21: Sequenzdiagramm zur Lokalisierung der Bewohner im Haus .....	55
Abbildung 22: Punktwolke und Regressionsgerade für das Beispiel bivariate lineare Regression.....	64
Abbildung 23: Punktwolke der multiplen Regression .....	67
Abbildung 24: Künstliches Neuron.....	69
Abbildung 25: Interne Signalabfolge in einem Knoten .....	69
Abbildung 26: Lineare Aktivitätsfunktion.....	70
Abbildung 27: Heavyside-Funktion als Aktivitätsfunktion .....	70
Abbildung 28: Lineare Aktivitätsfunktion mit Schwelle.....	71
Abbildung 29: Logistische Aktivitätsfunktion .....	72
Abbildung 30: Tangens Hyperbolicus Aktivitätsfunktion .....	72

Abbildung 31: Normalverteilte Aktivitätsfunktion .....	73
Abbildung 32: Beispiel für einen Pattern Associator .....	75
Abbildung 33: Direkte Rückkopplung .....	76
Abbildung 34: Indirekte Rückkopplung .....	76
Abbildung 35: Seitliche Rückkopplung .....	76
Abbildung 36: Vollständig verbundenes Netz.....	76
Abbildung 37: Beispiel für ein einfaches Feedforward Netz .....	76
Abbildung 38: Beispiel für ein Feedforward Netz mit Shortcuts .....	77
Abbildung 39: Beispiel für ein Simple Recurrent Network .....	78
Abbildung 40: Beispiel für ein Jordan-Netz.....	79
Abbildung 41: Backpropagation Algorithmus: Forward-Pass Phase .....	80
Abbildung 42: Backpropagation Algorithmus: Backward-Pass Phase.....	81
Abbildung 43: MemBrain - Hauptfenster.....	82
Abbildung 44: Neuronen einfügen .....	82
Abbildung 45: Unit - Object Properties Dialog .....	83
Abbildung 46: MemBrain - Lesson Editor .....	84
Abbildung 47: Lernregel wählen .....	85
Abbildung 48: Net Error Viewer.....	86
Abbildung 49: Möglicher Verlauf des mittleren quadratischen Fehlers .....	86
Abbildung 50: Künstlich neuronales Netzwerk für das XOR-Problem .....	87
Abbildung 51: ATPlus - Gebäudemodell.....	90
Abbildung 52: Zeitplan für die Initialisierungs-, Trainings- und Testphase .....	93
Abbildung 53: Optimum Start-Stop FeedForward Netz .....	95
Abbildung 54: Optimum Start-Stop Simple Recurrent Network .....	96
Abbildung 55: Optimum Start-Stop Jordan Netz.....	96
Abbildung 56: Übersicht über den Simulationsablauf .....	97
Abbildung 57: SimConfig.xml .....	98
Abbildung 58: TimeTable.xml .....	99
Abbildung 59: Übersicht über die Initialisierungsphase .....	100
Abbildung 60: Trainingsphase.....	101
Abbildung 61: Testphase .....	103
Abbildung 62: Ergebnisse der Trainingsphase für Raum 1 .....	104
Abbildung 63: Ergebnisse der Trainingsphase für Raum 2 .....	104
Abbildung 64: Testphase - FeedForward Network für Raum 1 .....	105
Abbildung 65: Testphase - FeedForward Network für Raum 2 .....	105
Abbildung 66: Testphase - Jordan Network für Raum 1 .....	106
Abbildung 67: Testphase - Jordan Network für Raum 2 .....	106

Abbildung 68: Testphase - Simple Recurrent Network für Raum 1 .....	107
Abbildung 69: Testphase - Simple Recurrent Network für Raum 2.....	107
Abbildung 70: Testphase - Referenz für Raum 1 .....	108
Abbildung 71: Testphase - Referenz für Raum 2.....	108
Abbildung 72: Gegenüberstellung der Brennerlaufzeiten .....	109
Abbildung 73: Beispiel eines Interaktionsprotokolls .....	113
Abbildung 74: Übersicht MAS für die Gebäudesteuerung .....	117

## II. Tabellenverzeichnis

Tabelle 1: Zusammenfassung der ermittelten optimalen Werte für ein Feedforward KNN... 10	10
Tabelle 2: Image Template Basiselemente.....	12
Tabelle 3: Wahrscheinlichkeit mit der ein Image Templates zutrifft .....	13
Tabelle 4: Beschreibung des Anwendungsfalls - automatische energiesparende Raumtemperaturregelung für eine Person nach einem eingelernten Belegungsplan <sup>40</sup>	
Tabelle 5: Beschreibung des Anwendungsfalls - automatische Raumtemperaturregelung für eine Mehrpersonenbelegung mit Konfliktlösungseinheit .....	47
Tabelle 6: Beschreibung des Anwendungsfalls - Singlehaushalt: Aufstehen am Morgen während der Arbeitswoche .....	54
Tabelle 7: Betrachtete physikalische Messgrößen.....	58
Tabelle 8: Beispiel lineare Regression - Ermitteln der Regressionsgerade .....	63
Tabelle 9: Zwischenrechnung - Beispiel lineare Regression.....	63
Tabelle 10: Beispiel multiple Regression - Ermitteln der Regressionsfläche .....	66
Tabelle 11: XOR Wahrheitstafel .....	81
Tabelle 12: ATPlus Dateien .....	91
Tabelle 13: Für die Simulation notwendige ATPlus Variablen .....	91
Tabelle 14: Ein- und Ausgabeneuronen .....	94
Tabelle 15: Gegenüberstellung der Brennerlaufzeiten.....	109
Tabelle 16: Prometheus System Overview Diagram - Symbole.....	116
Tabelle 17: Datenblatt des Interface Agent .....	118
Tabelle 18: Datenblatt des Trending Agent.....	119
Tabelle 19: Datenblatt des Presence Agent .....	120
Tabelle 20: Datenblatt für die User Agents .....	122
Tabelle 21: Datenblatt des Conflict Resolution Agent .....	123
Tabelle 22: Datenblatt des Control Agent .....	124
Tabelle 23: Datenblatt für den Weather Forecast Agent .....	125
Tabelle 24: Datenblatt für den Building Analysis Agent .....	126

### III. Literaturverzeichnis

1. ZEW - Center for European Economic Research. ZEW Energiebarometer. [Online].; 2010 [cited 2010 05 04. Available from: <ftp://ftp.zew.de/pub/zew-docs/zn/schwerpunkte/energiemarkt/Energiemarkt0110.pdf>.
2. Löser DG, Natsch B. ECOTrinova e.V. [Online].; 2005 [cited 2010 05 04. Available from: <http://www.ecotrinova.de/downloads/stromsparbroschwww051109coll.pdf>.
3. Statistik Austria - Die Informationsmanager. Statistik Austria - Sterbetafeln. [Online].; 2007 [cited 2010 Mai 3. Available from: [http://www.statistik.at/web\\_de/static/sterbetafeln\\_seit\\_186871\\_nach\\_geschlecht\\_022541.xls](http://www.statistik.at/web_de/static/sterbetafeln_seit_186871_nach_geschlecht_022541.xls).
4. Yang IH, Yeo MS, Kim KW. Application of artificial neural network to predict the optimal start time for heating system in building. Energy Conversion & Management. 2003: p. 2791-2809.
5. Mozer MC. The Neuronal Network House: An Environment that Adapts to its Inhabitants. In Proceedings of the American Association for Artificial Intelligence Spring Symposium on Intelligent Environments; 1998; Menlo Park. p. 110-114.
6. Lang R, Bruckner D, Pratl R, Velik R, Deutsch T. Scenario Recognition in Modern Building Automation. In Proceedings of the 7th IFAC International Conference on Fieldbuses & Networks in Industrial & Embedded Systems (FeT 2007); 2007. p. 305-312.
7. Amigo Project. Amigo Home. [Online].; 2008 [cited 2010 05 27. Available from: <http://www.amigo-project.org/>.
8. Augusto JC, Nugent CD. Smart Homes Can Be Smarter. Lecture Notes in Computer Science. 2006.
9. Davidsson P, Boman M. A Multi-Agent System for Controlling Intelligent Buildings. In Proceedings Fourth International Conference on MultiAgent Systems ; 2000; Boston. p. 377-378.
10. Siemens AG. Siemens Synco Living. [Online].; 2009 [cited 2010 05 20. Available from: [http://www.buildingtechnologies.siemens.com/bt/de/Produkte\\_und\\_Systeme/Gebaeudeautomationssysteme/home\\_automation\\_system/Seiten/Hausautomation.aspx](http://www.buildingtechnologies.siemens.com/bt/de/Produkte_und_Systeme/Gebaeudeautomationssysteme/home_automation_system/Seiten/Hausautomation.aspx).
11. myGEKKO -.einfach leben. [Online].; 2010 [cited 2010 05 21. Available from: <http://www.my-gekko.com>.
12. Adhoco AG. Intelligenz für Ihr Zuhause. [Online].; 2010 [cited 2010 05 22. Available from: <http://www.adhoco.com>.
13. Incropera FP, De Witt DP. Fundamentals of heating and mass transfer. 5th ed. Sons HW&, editor. New Jersey: Hohn Wiley & Sons; 1990.
14. Dimitrov T. Ambient Intelligence for the networked home environment. In KNX Scientific - Session 4; 2007; Fraunhofer Institut IMS, Duisburg, Germany. p. 57-79.

15. Qiao B, Kecheng L, Guy C. A Multi-Agent System for Building Control. In Proceedings of the IEEE/WIC/ACM International Conference on Intelligent Agent Technology (IAT'06); 2006; Hong Kong. p. 653-659.
16. Callaghan V, Clarke G, Colley M, Hagrais H. A soft-computing distributed artificial intelligence architecture for intelligent buildings. In Loia V, Sesa S. Soft Computing Agents - New Trends for Designing Autonomous Systems. Berlin: Springer-Verlag; 2001. p. 117 ff.
17. Davidsson P, Boman M. Distributed monitoring and control of office buildings by embedded agents. Information Sciences. 2005 May; 141(4): p. 15.
18. Stamper R, Liu K, Ades Y. Understanding the roles of signs and norms in organizations - a semiotic approach to information systems design. Behaviour & Information Technology. 2000 Jänner; 19(1): p. 15-27.
19. KNX Association [Official website]. [Online].; 2010 [cited 2010 05 20. Available from: <http://www.knx.org/>.
20. ZigBee Alliance. ZigBee Alliance. [Online].; 2010 [cited 2010 05 22. Available from: <http://www.zigbee.org/>.
21. Umbach-Daniel A, Schumann T. Pilotprojekt für das Heimautomations- und Meldesystem der Adhoco AG zur Unterstützung des selbstständigen Wohnens im Alter. sozialwissenschaftliche Begleitstudie. Schweiz: Age Stiftung Schweiz, Sozialwissenschaft; 2008.
22. Living Tomorrow. Living Tomorrow. [Online].; 2010 [cited 2010 05 25. Available from: <http://www.livingtomorrow.com/>.
23. Mozer M. The adaptive house. In Intelligent Building Environments, 2005. The IEE Seminar on (Ref. No. 2005/11059); 2005; University of Essex, Colchester, United Kingdom. p. 39-79.
24. Object Management Group. Unified Modeling Language v 2.2. [Online].; 2009 [cited 2010 05 13. Available from: <http://www.uml.org/#UML2.0>.
25. Forbrig P. Objektorientierte Softwareentwicklung mit UML. 3rd ed. München: Carl Hanser Verlag; 2007.
26. Stoiber G. Energieausweis in Österreich. [Online].; 2009 [cited 2010 05 05. Available from: <http://www.energieausweis.at/>.
27. Weiß C. Basiswissen Medizinische Statistik. 3rd ed. Heidelberg: Springer; 2005.
28. Galton F. Natural inheritance London: Maxmillan & Co.; 1989.
29. Pearson K. Regression, Heredity, and Panmixia. Mathematical Contributions to the Theory of Evolution. 1896: p. 253-318.
30. Jetter T. MemBrain Neuronale Netze Editor und Simulator. [Online].; 2010 [cited 2010

- 06 30. Available from: <http://www.membrain-nn.de/>.
31. Stöcker H. Überwachtes Lernen - Prinzip des überwachten Lernens. In Stöcker H. Taschenbuch mathematischer Formeln und moderne Verfahren. 4th ed. Frankfurt am Main: Wissenschaftlicher Verlag Harri Deutsch GmbH; 2003. p. 743-744.
32. Modelica Association. Modelica and the Modelica Association. [Online].; 2010 [cited 2010 11 18. Available from: <https://www.modelica.org/>.
33. Dipl.-Ing.Felgner F. HVAC Control of Buildings Based on Comprehensive Modeling. [Online].; 2006 [cited 2010 November 18. Available from: <http://www.eit.uni-kl.de/litz/forschung/BuildingSimulation/BuildingModelling.html>.
34. Riedmiller M, Braun H. A direct adaptive method for faster backpropagation learning: The Rprop algorithm. In Proceedings of the IEEE International Conference on Neural Networks; 1993. p. 586-591.
35. Wooldridge M. An Introduction to MultiAgent Systems. 2nd ed. Hoboken NJ, editor. Chichester: Wiley; 2009.
36. Foundation for Intelligent Physical Agents. FIPA Agent Communication Language Specifications. [Online].; 2002 [cited 2010 07 18. Available from: <http://www.fipa.org/repository/aclspecs.html>.
37. Bratman ME. Intentions, Plans, and Practical Reason Inf CftSoLal, editor. Chicago; 1999.
38. Rao AS, Georgeff MP. BDI Agents: From Theory to Practice. In Proceedings of the First International Conference on Multiagent Systems; 1995. p. 312-319.
39. Telecom Italia. Jade - Java Agent DEvelopment Framework. [Online].; 2010 [cited 2010 07 18. Available from: <http://jade.tilab.com/>.
40. FIPA - Foundation for Intelligent Physical Agents. Foundation for Intelligent Physical Agents. [Online].; 2010 [cited 2010 07 18. Available from: <http://www.fipa.org/>.
41. Object Management Group. Object Management Group - UML. [Online].; 2010 [cited 2010 07 18. Available from: <http://www.uml.org>.
42. Padgham L, Thangarajah J, Winikoff M. RIMT Agents Group. [Online].; 2009 [cited 2010 07 18. Available from: <http://www.cs.rmit.edu.au/agents/pdt/index.shtml>.
43. AOS Group. Automation Decision-Making Software. [Online].; 2010 [cited 2010 07 18. Available from: <http://www.agent-software.com.au/products/jack/index.html>.
44. Braubach L, Pokahr A. Jadex - BDI Agent System. [Online].; 2010 [cited 2010 07 18. Available from: <http://jadex-agents.informatik.uni-hamburg.de/xwiki/bin/view/About/Overview>.
45. Qiao B, Liu K, Guy C. A Multi-Agent System for Building Control. In International Conference on Intelligent Agent Technology (IAT '06); 2006; Hong Kong: IEE/WIC/ACM. p. 653-659.

46. Cook DJ, Youngblood M, Heiermann IEO, Gopalratnam K, Rao S, Litvin A, et al. MavHome: An Agent-based Smart Home. In IEEE International Conference on Pervasive Computing and Communications; 2003; Dallas-Fort Worth, Texas. p. 521-524.
47. Kalogirou SA. Application of artificial neural-network for energy systems. *Applied Energy*. 2000 September;(67): p. 17-35.
48. Hagrais H, Callaghan V, Colley M, Clarke G, Pounds-Cornish A, Duman H. Creating an Ambient-Intelligence Environment Using Embedded Agents. *IEEE Intelligent Systems*. 2004 Jul-Aug; 1(19): p. 12-20.
49. Goncalves D. Ubiquitous Computing and AI Towards an Inclusive Society. In EC/NSF Workshop on Universal Accessibility of Ubiquitous Computing: Providing for the Elderly; 2001; Alcácer do Sal, Portugal. p. 37-40.
50. Liang NS, Fu LC, Wu CL. An integrated, flexible, and Internet-based control architecture for home automation system in the Internet Era. In Proceedings of the 2002 IEEE International Conference on Robotics & Automation; 2002; Washington, DC. p. 1101-1106.
51. Zhang H, Wang FY, Ai Y. An OSGi and Agent Based Control System Architecture for Smart Home. In IEEE , editor. *Networking, Sensing and Control*, 2005. Proceedings.; 2005; Tucson, Arizona. p. 13-18.
52. Choi J, Shin D, Shin D. Research and Implementation of the Context-Aware Middleware for Controlling Home Appliances. *Consumer Electronics*. 2005 November: p. 1223-1226.
53. Heuer S. Rollende Kraftwerke. *Technology Review*. 2004 Oktober: p. 104-105.
54. Rutishauser U, Joller J, Douglas R. Control and Learning of Ambience by an Intelligent Building. *IEEE Transactions on Systems, Man, And Cybernetics Society*. 2005 Jänner: p. 121-132.
55. Malinowsky B, Neugschwandtner G, Kastner W. *Calimero: Next Generation*. Vienna: Vienna University of Technology, Institute of Automation - Automation System Group; 2005.
56. Lindelöf D, Morel N. A field investigation of the intermediate light switching by users. *Energy and Buildings*. 2006: p. 790-801.
57. Bodmer J, inventor; AG HTS, assignee. Home and Building Automation System. Switzerland patent PCT/CH97/00194. 1997 Mai 20.
58. Bodmer J, inventor; AG A, assignee. Automation System for the Home and for Buildings. Switzerland patent PCT/CH03/00118. 2003 February 18.
59. Degen DIM, Liebich DT. IFC-Austauschformat für die TGA. *Fach.Journal*. 2006|2007: p. 172-175.
60. Vallée M, Ramparany F, Vercouter L. A Multi-Agent System for Dynamic Service

- Composition in Ambient Intelligence Environments. PhD Thesis - Paper. France Telecom R&D, IST Amigo; 2004.
61. Liebich T, Hoffeller T, von Both P, Degen M, Geiger A, Gerschning E, et al. Anwendungshandbuch Datenaustausch BIm/IFC. 11th ed. München: IAI Industriellianz für Interoperabilität; 2006.
  62. Tesauo G. Temporal Difference Learning and TD-Gammon. Communications of the ACM. 1995 März.
  63. United Nations. Kyoto Protocol. In Kyoto Protocol to the United Nations Framework Convention on Climate Change; 1998; Kyoto. p. 21.
  64. Baier T, Neuwirth E. statconn: RAndFriends, RExcel. [Online].; 2010 [cited 2010 Juni 29. Available from: <http://rcom.univie.ac.at/>.
  65. University of Wisconsin. A TRaNsient SYstems Simulation Program. [Online].; 2010 [cited 2010 10 1. Available from: <http://sel.me.wisc.edu/trnsys/index.html>.