



Hochschule für Angewandte Wissenschaften Hamburg
Hamburg University of Applied Sciences

Master Thesis

Jens Ellenberg

Ontologiebasierte Aktivitätserkennung im Smart
Home Kontext

Jens Ellenberg
Ontologiebasierte Aktivitätserkennung im Smart
Home Kontext

Master Thesis eingereicht
im Studiengang Informatik Master
am Department Informatik
der Fakultät Technik und Informatik
der Hochschule für Angewandte Wissenschaften Hamburg

Betreuender Prüfer : Prof. Dr. rer. nat. Kai von Luck
Zweitgutachter : Prof. Dr. rer. nat. Gunter Klemke

Abgegeben am 30. Dezember 2011

Jens Ellenberg

Thema der Ausarbeitung

Ontologiebasierte Aktivitätserkennung im Smart Home Kontext

Stichworte

Aktivitätserkennung, Ontologie, Smart Home, Kontext, Klassifikation, allgegenwärtige Computer

Kurzzusammenfassung

In dieser Arbeit wurde eine Aktivitätserkennung in einem Smart Home entwickelt. Um die Aktivitäten des Bewohners zu beschreiben wurde eine Ontologie eingesetzt. Die Aktivitäten werden indirekt aus den Sensordaten mit einem regelbasiertem Klassifikationsverfahren erkannt. Die Regeln für die Klassifikation wurden automatisch aus der Ontologie generiert. Als Regelmaschine und Übersetzer der Regeln wurde die Anwendung „Scenior“ von der Universität Hamburg eingesetzt.

Jens Ellenberg

Title of the paper

Ontology-based Activity Recognition in Smart Home Context

Keywords

Activity Recognition, Ontology, Smart Home, Context, Classification, Ubiquitous Computing

Abstract

In this work were an activity recognition for a smart home developed. An ontology was used to describe the activities of the resident. The activities were indirect recognized with a rule based classification out of the sensor data. The rules of the classification were automated generated from the ontology. As a rule translator and rule engine were used the application "Scenior" from the University Hamburg.

Inhaltsverzeichnis

Abbildungsverzeichnis	vii
1. Einleitung	1
1.1. Motivation	1
1.2. Ziel der Masterarbeit	2
1.3. Aufbau der Arbeit	3
2. Grundlagen	4
2.1. Livingplace Hamburg	4
2.2. Kontext	5
2.3. Ontologien	8
2.3.1. RDF	9
2.3.2. RDFS	10
2.3.3. OWL	10
2.3.4. SWRL	11
2.4. Klassifikation	12
2.4.1. Grundlagen	12
2.4.2. Beurteilung der Klassifikation	13
2.5. Übersicht vergleichbarer Arbeiten	14
3. Analyse der Aktivitätserkennung im Smart Home	16
3.1. Architektur der Aktivitätserkennung	16
3.1.1. Modell zur Aktivitätserkennung	16
3.1.2. Aufbau der Aktivitätserkennung	21
3.1.3. Architektur der Aktivitätserkennung	23
3.1.4. Bedingungen im Living Place	24
3.2. Beschreibung der Aktivitäten	28
3.2.1. Darstellung der Dimensionen	29
3.2.2. Definition der Aktivitäten	34
3.2.3. Klassifikationsverfahren	38
3.3. Daten im Living Place	41
3.3.1. Situationsbedingte Anwendungen im Living Place	42
3.3.2. Übersicht	43

3.3.3. Sensor-Daten	44
3.3.4. Interpretierte Daten	44
3.3.5. Externe Daten	45
4. Entwicklung einer Ontologie für Smart Homes	47
4.1. Rolle der Ontologie im System	47
4.1.1. Modell der Eingangsdaten	49
4.1.2. Modell der Klassen	49
4.1.3. Zuordnungsfunktion	50
4.2. Auswahl der Werkzeuge für die Arbeit mit Ontologien	51
4.2.1. Bearbeitungszyklus für Ontologien	51
4.2.2. Auswahl von Werkzeugen	52
4.3. Erweiterung der Upper-Ontologie	53
4.4. Aufbau einer Domain-Ontologie für das Living Place	56
4.4.1. Agenten-Begriffe	58
4.4.2. Objekt-Begriffe	59
4.4.3. Zonen-Begriffe	59
4.4.4. Kontext-Begriffe	60
4.4.5. Event-Begriffe	61
4.4.6. State-Begriffe	63
4.5. Erstellen von SWRL Regeln	64
4.6. Extraktion der ontologischen Primitiven	65
4.6.1. Physische Primitive	66
4.6.2. Event Primitive	66
4.6.3. Temporale Primitive	67
4.7. Evaluation der Ontologie	67
4.7.1. Ontologie als Modell der Realität	67
4.7.2. Evaluierung des Ontologie Modells	69
5. Design und Evaluation	71
5.1. Übersicht der Kontext-Nutzung im Living Place	71
5.1.1. Kontexterkenkung im Living Place	71
5.1.2. Einbettung der Aktivitätserkenkung in das Living Place	74
5.2. Einbindung der Aktivitätserkenkung in das Living Place	75
5.2.1. Anpassung von Scenior	76
5.2.2. Anwendung zum Erzeugen von Events	77
5.2.3. Anwendung zum Versenden von Aktivitäten	78
5.2.4. Beschreibung von Scenior	78
5.3. Entwicklung einer Testumgebung	79
5.3.1. Nachrichten Sender und Empfänger	79
5.3.2. Nachrichten-Aufnahme und -Abspielgerät	81

5.3.3. Weitere Simulationswerkzeuge	83
5.4. Evaluation der Aktivitätserkennung	83
5.4.1. Evaluation der Arbeitsschritte	83
5.4.2. Evaluation des Design	85
5.4.3. Evaluation durch eine Testumgebung	86
5.4.4. Einsatz der Software im Living Place	88
6. Zusammenfassung und Ausblick	90
6.1. Zusammenfassung	90
6.2. Ausblick	91
Literaturverzeichnis	93
Anhang	100
A. Definition der States und Events	100
A.1. States (Aktivitäten)	100
A.2. Komplexe States (Aktivitäten)	104
A.3. Events	104
A.3.1. Zone betreten	104
A.3.2. Zone verlassen	105
A.3.3. Objekt aktivieren	105
A.3.4. Objekt deaktivieren	105
A.4. Komplexe Events	106
A.4.1. Start Events	106
A.4.2. End Events	106
B. Ontologie Code	107
B.1. Header	107
B.2. Klasse Bewohner	108
B.3. Klasse Bett	108
B.4. Klasse Nutzungsbereich des Bettes	108
B.5. State Schlafen	109
B.6. SWRL Regeln in Protege	110
B.7. SWRL Rule Aktivität Schlafen	113
C. Testlauf mit Scenior	117
C.1. Schlafen	117
C.2. Duschen	121
C.3. Frühstück	125
C.4. Morgenszenario	128

Abbildungsverzeichnis

2.1. 3D Modell des Living Place Hamburg	4
2.2. Übersicht der Semantik-Web-Sprachen (vgl. Grimm u. a. (2007))	9
3.1. Aktivitätserkennung	22
3.2. Blackboard als Zentrale Kommunikationsschnittstelle	24
3.3. Darstellung von Kontext (vgl. Bolchini u. a. (2007))	29
3.4. Ontologie des 5W1H-Modells	34
3.5. Aktivität als zentrale Dimension	35
3.6. Upper Ontologie (vgl. Bohlken u. a. (2011))	36
4.1. Wissensbasis	48
4.2. Protege 3.4.7	52
4.3. Upper Ontologie physischer Teil	55
4.4. Upper Ontologie konzeptueller Teil	55
4.5. Vererbungsstruktur von Scenior (vgl. Bohlken u. a. (2011))	56
4.6. Vererbungsstruktur im Living Place	57
4.7. Event Ontologie	62
4.8. State Ontologie	63
4.9. Zusammenhang zwischen Events und States	65
4.10. Zeitstrahl der Aktivitätserkennung	69
5.1. Gesamt-Architektur	73
5.2. Arbeitsschritte der Wissenbasis	75
5.3. Komponenten der Aktivitätserkennung	76
5.4. Ablauf der Event-Erzeugung	77
5.5. Eventverarbeitung von Scenior	78
5.6. Test Nachrichten Sender	80
5.7. Test-Nachricht-Empfänger	81
5.8. Nachrichten Wiedergabe	82
5.9. Morgen Szenario	86
C.1. Schlafen Start-Event erst die Zone dann das Objekt	118
C.2. BEENDET: Schlafen Start-Event erst das Objekt dann die Zone	118

C.3. Schlafen End-Event ausgelöst durch die Zone	119
C.4. Schlafen End-Event ausgelöst durch das Objekt	119
C.5. Schlafen State beendet durch die Zone	120
C.6. Schlafen State beendet durch das Objekt	120
C.7. Duschen Start-Event erst die Zone dann das Objekt	121
C.8. BEENDET: Duschen Start-Event erst das Objekt dann die Zone	122
C.9. Duschen End-Event ausgelöst durch die Zone	122
C.10. Duschen End-Event ausgelöst durch das Objekt	123
C.11. Duschen State beendet durch die Zone	123
C.12. Duschen State beendet durch das Objekt	124
C.13. Frühstück Start-Event erst die Zone dann das Objekt	125
C.14. BEENDET: Frühstück Start-Event erst das Objekt dann die Zone	126
C.15. Frühstück End-Event ausgelöst durch die Zone	126
C.16. Frühstück End-Event ausgelöst durch das Objekt	127
C.17. Frühstück State beendet durch die Zone	127
C.18. Frühstück State beendet durch das Objekt	128
C.19. Morgen-Szenario ohne eine erkannte Aktion	129
C.20. Morgen-Szenario Schlafen erkannt	129
C.21. Morgen-Szenario Schlafen und Duschen erkannt	130
C.22. Morgen-Szenario Schlafen, Duschen und Frühstück erkannt	130

1. Einleitung

1.1. Motivation

In der Vision aus [Weiser \(1999\)](#) sind Computer im 21. Jahrhundert allgegenwärtig. Die Computer unterstützen den Menschen in seinen Tätigkeiten, ohne dass die Computer dabei im Vordergrund stehen. In seiner Arbeit beschreibt [Weiser](#) einen Morgen im Leben von Sal. In diesem Szenario wacht Sal auf und wird von einer Stimme gefragt, ob sie einen Kaffee wünscht. Der Computer erkennt das Aufwachen von Sal und weiß, wie er Sal bei der Morgenroutine unterstützen kann. Die Wohnung aus der Vision von [Weiser](#) ist aus heutiger Sicht ein Smart Home.

Computer, die Personen in Ihrer Wohnung unterstützen, werden in verschiedenen Bereichen der Gesellschaft gebraucht. In einer alternden Gesellschaft (vgl. [Scheidt-Nave u. a. \(2010\)](#)) wird die Frage immer wichtiger, wie die benötigte Unterstützung für ältere Leute geleistet werden kann. Hier können die Computer einen Teil der Aufgaben übernehmen, beispielsweise in einem Smart Home. Ebenso ist es wahrscheinlich nur eine Frage der Zeit, bis in jedem Haushalt die Computer eine größere Rolle spielen. Vor fünfzehn Jahren konnte man sich nicht vorstellen, dass jeder da telefonieren möchte wo er gerade ist. Heute ist es durch das Mobiltelefon zur Normalität geworden. Ebenso wird es die Zeit zeigen, welche Aufgaben der Computer in der Wohnung übernehmen wird. Es ist gut möglich, dass ein intelligentes Haus in naher Zukunft, wie beim Mobiltelefon, ein Hilfsmittel sein wird, dass jeder gerne nutzt und jeder besitzt, auch wenn ein Leben ohne diese Hilfe immer noch möglich wäre.

Wenn die Computer den Menschen in seiner Wohnung unterstützen sollen, ist das in der Regel eine Unterstützung bei seinen allgemeinen Tätigkeiten. Hierfür müssen die Computer bzw. die Anwendungen zwei Dinge wissen:

- Wie soll die Unterstützung erfolgen?
- Wann soll die Unterstützung erfolgen?

Wie die Unterstützung erfolgen soll, ist von der Art und von den Möglichkeiten der Anwendung abhängig. Das Wissen darüber liegt in der Regel bei den Anwendungen selbst. Wann die Unterstützung erfolgen soll, ist von der Situation in der Wohnung abhängig. Die Situation in der Wohnung bildet also den Kontext, in dem die verschiedenen Anwendungen arbeiten.

Daher wird im weiteren Verlauf der Arbeit die Situation und der Kontext synonym verwendet. Das Wissen über die Situation ist somit anwendungsunabhängig aber die Anwendungen müssen für die Unterstützung des Bewohners die Situation bzw. Kontext in der Wohnung kennen.

Das Wissen aus den Interpretationen über die Situation und den Aktivitäten in der Wohnung bildet somit eine Voraussetzung des Smart Home, damit die Wohnung ihr Verhalten entsprechend der Situation anpassen kann. Wie die Computer oder Anwendungen in der Wohnung an das Wissen gelangen, um sich „intelligent“ zu verhalten, ist Inhalt dieser Arbeit.

1.2. Ziel der Masterarbeit

Das Ziel dieser Masterarbeit ist es ein Framework zu entwickeln, mit dessen Hilfe die Aktivitäten des Bewohners der Wohnung erkannt werden können. Diese Informationen sollen für weitere Programme nutzbar sein. Es gibt nahezu unendlich viele Möglichkeiten verschiedene Aktivitäten in einer Wohnung zu definieren. Jede dieser Aktivitäten könnte für ein bestimmtes Programm wichtig sein und sollte daher vom Framework erkannt werden. Es ist nicht möglich alle Aktivitäten in dieser Arbeit zu beachten. Das Framework muss daher auf unterschiedliche Aktivitätserkennungen anpassbar sein, damit immer genau die Aktivität hypothetisiert wird, die in der aktuellen Wohnung für die Anwendungen relevant ist. In dieser Arbeit soll ein morgendliches Szenario als Ausgangsbasis für die Aktivitätserkennung dienen.

Das Erkennen dieser Kontexte soll durch einen Klassifikationsalgorithmus erfolgen. Es muss ein Algorithmus gefunden werden, der den Anforderungen dieser Arbeit genügt. Dieser Algorithmus soll für die Wohnung umgesetzt werden. Hierbei sind die verschiedenen Herausforderungen der Klassifikation zu beachten.

Das Framework soll ein Teil des „Living Place Hamburg“ (vgl. [von Luck u. a. \(2010\)](#)) werden. Im Living Place sind viele verschiedene Sensoren (vgl. [Johannsen \(2011\)](#)) installiert. Mit diesen Sensoren sollen die Aktivitäten in dieser Arbeit erkannt werden. Die Aktivitäten erschließen sich nicht direkt aus einem bestimmten Sensor, sondern sie bilden sich in der Regel aus den Werten verschiedener Sensoren. Dabei ist zum Einen der Zeitpunkt wichtig, an dem der Sensor seinen Wert ermittelt. Außerdem könnte die Änderung eines Sensorwertes über einen bestimmten Zeitraum relevant sein um die Aktivitäten zu identifizieren. Aktivitäten können auch mit anderen Informationsquellen eingegrenzt werden. Andere Quellen sind beispielsweise Informationsdienste aus dem Internet, die das Wetter zur Verfügung stellen (wetterfeste Kleidung anziehen) oder Auskunft über die öffentlichen Verkehrsmittel geben (die Wohnung verlassen, um den Bus zu erreichen). Die Ausgaben der Sensoren müssen in der Regel umgewandelt werden, da sie sonst für eine Klassifikation von Aktivitäten nicht geeignet sind. Die Sensor- und Informationsquellen werden dann zu Merkmalen

oder Features¹ aus denen die Aktivitäten erkannt werden können. In dieser Arbeit sollen die Möglichkeiten des „Living Place“ identifiziert werden, welche Sensoren zur Verfügung stehen und welche noch fehlen um die Aktivitätserkennung aufzubauen.

1.3. Aufbau der Arbeit

In dieser Arbeit werden im nachfolgenden als erstes in den *Grundlagen* die Themen beschrieben, die zum Verständnis dieser Arbeit notwendig sind. Zu den Grundlagen gehört eine Beschreibung des Forschungsprojektes „Living Place Hamburg“ der HAW Hamburg von dem diese Arbeit ein Teil ist. Diese Arbeit gehört in den Bereich des „Context Aware Computing“² und es werden Ontologien zur Lösung der Herausforderungen eingesetzt. Dementsprechend wird in den Grundlagen der Kontext-Begriff definiert und es werden Ontologien und deren Sprachen aus dem Semantik Web beschrieben. Nach einem Abschnitt über Klassifikation enden die Grundlagen mit einer Übersicht über vergleichbare Arbeiten.

In der *Analyse der Aktivitätserkennung im Smart Home* ist der erste Schritt die Aktivitäten zu definieren. Anschließend wird nach einer Struktur gesucht, mit der die Aktivitäten gespeichert werden können. In der Analyse wird auch die Klassifikation beschrieben, mit der die Aktivitäten erkannt werden sollen und es werden die Voraussetzungen (Eingangsdaten) für die Aktivitätserkennung im Living Place beschrieben.

In dem Kapitel *Entwicklung einer Ontologie für Smart Homes* werden die Ergebnisse aus der Analyse umgesetzt. Es werden die Aktivitäten definiert und es werden die Zusammenhänge zwischen dem Living Place und den Aktivitäten des Bewohners beschrieben. Teil der Ontologie sind auch die Beschreibung der Regeln mit denen die Klassifikation im nächsten Kapitel umgesetzt wird.

Im Kapitel *Design und Evaluation* wird eine Übersicht über die Kontext-Erzeugung im Living Place gegeben, dessen Teil die Aktivitätserkennung ist. Anschließend wird die Klassifikation der Aktivitäten in das Living Place integriert. Um die Aktivitätserkennung zu entwickeln und zu testen sind verschiedene Simulations-Werkzeuge notwendig. Abschließend wird die Evaluation der Aktivitätserkennung beschrieben.

Im letzten Kapitel wird die Arbeit zusammengefasst und es wird ein Ausblick gegeben über die Aufgabe, die in der Zukunft noch erwartet werden.

¹Merkmale und Features werden hier synonym verwendet

²Situationsabhängige Berechnungen

2. Grundlagen

Im Folgenden werden die Grundlagen dieser Arbeit beschrieben. Die Arbeit ist ein Teil des Living Place Hamburg (2.1). Es soll der Kontext (2.2) in der Wohnung erkannt werden. Dabei kommen die verschiedenen Sensoren des Livingplace zum Einsatz.

2.1. Livingplace Hamburg

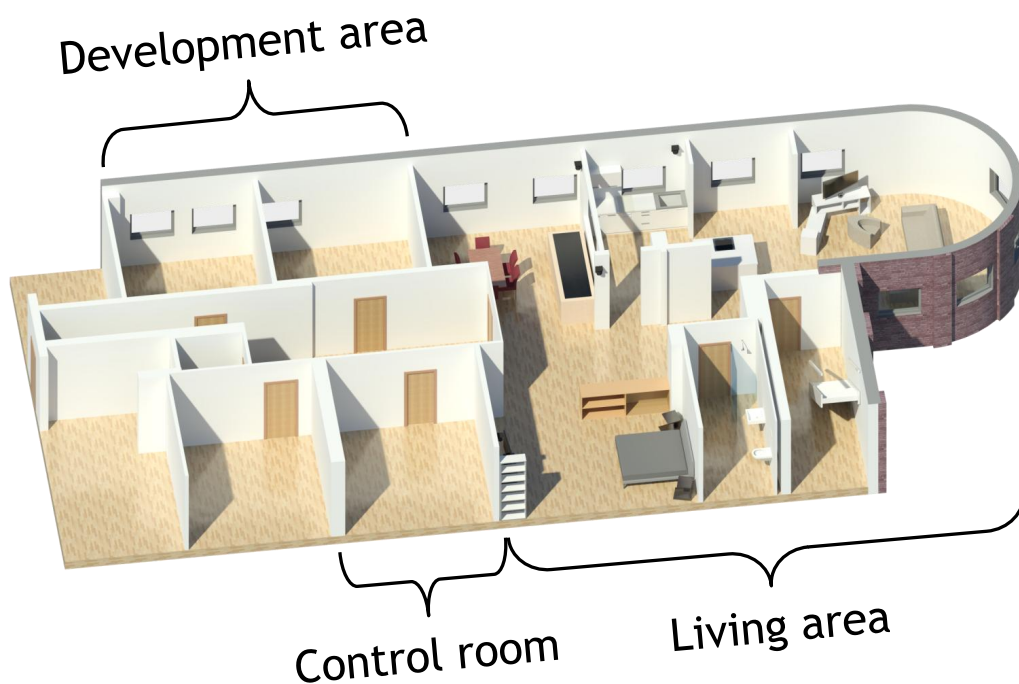


Abbildung 2.1.: 3D Modell des Living Place Hamburg

Das „Livingplace Hamburg“ ist ein Forschungsprojekt der HAW Hamburg (vgl. [von Luck u. a. \(2010\)](#)). In diesem Projekt sollen die Möglichkeiten erforscht werden, wie das Leben in einer Wohnung durch Computer unterstützt werden kann. Ausgehend von der Vision von Mark Weiser (vgl. [Weiser \(1999\)](#)) soll hier ein „Smart Home“ entstehen.

Die Vision von Weiser stammt aus den 80er Jahren. Die Technik und die Möglichkeiten sind seit dieser Zeit sehr fortgeschritten. Daher ist die Frage berechtigt, warum diese Vision noch nicht Wirklichkeit geworden ist (vgl. [Bell und Dourish \(2007\)](#)). Vielleicht ist die Vision noch nicht Realität, weil sie noch niemand umgesetzt hat. Die Möglichkeiten sollten in der heutigen Zeit alle existieren aber das Hauptproblem aus dem Paper wurde bisher noch nicht gelöst: Die Computer stellen für den Menschen eine Barriere da.

Daher ist ein Schwerpunkt im Livingplace die barrierefreie, nahtlose Interaktion des Bewohners mit der Wohnung (vgl. [Rahimi und Vogt \(2011\)](#)).

Ein weiterer Schwerpunkt der in der Wohnung umgesetzt werden soll, sind Computer oder Programme die den Bewohner intelligent unterstützen. Beispiele für solche Projekte sind das intelligente Bett (vgl. [Hardenack \(2011\)](#)), die intelligente Couch (vgl. [Dreschke \(2011\)](#)) oder der Wecker 2.0 (vgl. [Ellenberg \(2011\)](#)). Die Computer in der Wohnung sollten keine Barriere darstellen. Sie sollen den Menschen im Sinne von [Weiser](#) unterstützen ohne selbst im Vordergrund zu stehen. In der Wohnung sind die Computer allgegenwärtig, ohne dass man sie auf den ersten Blick wahr nimmt.

Die Wohnung ist mit vielen verschiedenen Sensoren und Kameras ausgestattet. Mit diesen Sensoren sollen die Funktionen und das Verhalten der Anwendungen im Living Place angepasst werden. Die Sensoren der Wohnung werden detaillierter in [Abschnitt 3.3.3](#) beschrieben. Die Anwendungen sollen den Bewohner bei seinen täglichen Aktionen unterstützen oder ihm arbeiten abnehmen. Hierbei steht nicht ein bestimmtes Ziel im Vordergrund, sondern das Living Place bietet eine Plattform für viele Projekte. Diese Projekte haben durch das Labor auch die Möglichkeit sich gegenseitig zu ergänzen und aufeinander aufzubauen.

Um diese Ziele zu verifizieren ist die Wohnung auch als Usability-Labor ausgelegt. Die Wohnung ist voll bewohnbar, so dass es in Zukunft möglich ist einen Probanden mehrere Tage in der Wohnung zu beobachten. Über einen Kontrollraum (vgl. [Abbildung 2.1](#)) ist es möglich alle Anwendungen in der Wohnung zu steuern und über die Kameras und Mikrofone den Bewohner zu beobachten. Außerdem können die Bild- und Tonsignale für die spätere Auswertung und Analyse aufgezeichnet werden.

2.2. Kontext

In dieser Arbeit geht es hauptsächlich um die Interpretation von Kontext-Informationen in einer intelligenten Wohnung. Aus den Kontext-Informationen soll weiteres Wissen extrahiert werden. In diesem Kapitel wird beschrieben wie sich Kontext im Bezug zu dieser Arbeit definiert, welche Erkennungsmöglichkeiten vorhanden sind und wo die Grenzen der Kontexterkennung liegen.

Bei einem bestimmten Kontext kann es sich um einen einfachen oder komplexen Kontext handeln. Ein einfacher Kontext ist etwas, das direkt von Sensoren abgeleitet werden kann. Beispielsweise könnte ein RFID-Chip, den der Bewohner bei sich trägt, direkt den Schluss zulassen, dass „der Bewohner in der Wohnung anwesend ist“. Für einen komplexeren Kontext („der Bewohner schläft in seinem Bett“) reicht es nicht aus festzustellen, dass der Bewohner sich gerade im Bett befindet. Hierfür sind mehrere Daten notwendig, die sich vielleicht auch erst aus anderen Kontext-Informationen ergeben.

Kontext ist ein allgemeiner Begriff, der in verschiedenen Arbeiten unterschiedlich definiert wird. Eine gängige Definition von Kontext ist:

„Context is any information that can be used to characterize the situation of an entity. An entity is a person, place, or object that is considered relevant to the interaction between a user and an application, including the user and applications themselves.“ [Abowd u. a. \(1999\)](#)

Diese Definition besagt, dass alle Informationen, die in einer intelligenten Wohnung zur Verfügung stehen können, auch möglicher Weise für eine Anwendung von Relevanz sind. Hierbei sind aber nicht nur Informationen gemeint, die von Sensoren erfasst werden können, sondern auch Informationen, die sich aus den Sensoren ableiten lassen und Informationen, die durch externe Wissensquellen oder Expertenwissen den Anwendungen zugänglich gemacht werden können.

Diese Auffassung von Kontext soll auch in dieser Arbeit angewandt werden.

Neben der allgemeinen Definition von [Abowd u. a. \(1999\)](#) gibt es noch andere Arbeiten über Kontext. In [Greenberg \(2001\)](#) werden drei unterschiedliche Theorien von Kontext beschrieben. Diese Theorien zeigen, dass „Kontext“ ein sich ständig änderndes und situationsabhängiges Konstrukt ist. Die Theorien unterscheiden sich dahingehend, wo der Fokus bei der Interpretation von Kontext liegt.

Situationsbedingte Aktion: In dieser Theorie wird davon ausgegangen, dass die ausgeführten Aktionen durch die vorherrschende Situation bestimmt wird. Pläne existieren, aber sie werden laufend der Situation angepasst und legen daher keinen bindenden Ablaufplan fest. Die Ziele einer Aktivität bestimmen hier den Kontext.

Aktivitätstheorie: Hier definieren die Aktionen selbst den Kontext. Es müssen bei der Erstellung des Kontextes nicht nur die beteiligten Personen und Artefakte betrachtet werden, sondern auch die Wünsche und Ziele der beteiligten Personen, da diese sich ändern können. Der Fokus liegt aber auf der Aktion.

Soziale Netzwerke: Je nachdem in welcher sozialen Umgebung sich eine Person befindet, definiert das den aktuellen Kontext. Dabei muss beachtet werden, dass sich eine

Person in vielen verschiedenen Umgebungen gleichzeitig befinden kann und, je nach Kombination, ein anderer Kontext entstehen kann.

In einem Beispiel für das Living Place stellen sich die 3 Theorien in einem morgendlichen Szenario wie folgt da.

Das Szenario:

Der Bewohner befindet sich in seiner Wohnung und ist gerade aufgestanden. Er ist hungrig und bereitet sich sein Frühstück zu, das er anschließend am Küchentisch einnimmt.

Situationsbedingte Aktion: Den Kontext bildet das Vorhaben, dass der Bewohner seinen Hunger nach dem Aufstehen stillen möchte. Diese Situation hat zur Folge, dass der Bewohner sich sein Frühstück zubereitet und isst.

Aktivitätstheorie: Der Kontext sind die Aktivitäten: Frühstück zubereiten und Frühstück essen.

Soziale Netzwerke: Der Kontext ist der Bewohner in seiner privaten Umgebung, morgens nach dem Aufstehen und er ist allein.

Eine weitere Möglichkeit Kontext-Informationen zu unterscheiden sind die Art der Informationen bzw. was die Informationen beschreiben.

Externe Informationen: Diese sind physikalische Werte und Eigenschaften der Einrichtung und Personen des Hauses. Externe Informationen können mit verschiedenen Sensoren relativ einfach erfasst werden. Um diese Informationen anschließend mit bestimmten höherwertigen Kontexten Beschreibungen in Verbindung zu bringen, ist hingegen eine komplexere Aufgabe.

Interne Informationen: Diese sind abstrakte Informationen über die Interessen und Ziele einer Person. Hierzu gehört auch was die Person bisher getan hat und welche Fortschritte ihre aktuellen Aktivitäten aufweisen. Diese Informationen sind sehr viel schwieriger zu gewinnen als die externen Informationen. In [Greenberg \(2001\)](#) geht man davon aus, dass man diese Informationen abschätzen muss oder „informiertes Raten“ angewandt werden könnte, um sich den internen Informationen anzunähern.

Des Weiteren bringt die Arbeit mit Kontext verschiedene Risiken. Es muss die Frage geklärt werden wie verschiedene Kontexte sich voneinander trennen lassen. Beispielsweise stellt sich die Frage, ob der Kontext „Schlafen“ mit dem Liegen im Bett beginnt oder ob dieser Kontext erst mit einem schlafenden Zustand startet. Möglicherweise ist die Position „Bett“ für den Kontext „Schlafen“ auch nicht notwendig aber hinreichend, da der Bewohner auch im Wohnzimmer schlafen kann. Dann muss ermittelt werden wie fein der Kontext definiert werden muss. Beispielsweise könnte „Schlafen“ mit dem Ausziehen anfangen und mit einer angezogenen Person am nächsten Morgen enden. Oder die Aktionen „Ausziehen“, „Ins Bett

gehen“, „Lesen“, „Licht ausschalten“, „Einschlafen“ u.s.w. sind eigene Kontexte. Hier könnte jeder Kontext noch beliebig verfeinert werden.

2.3. Ontologien

Ontologien stammen ursprünglich aus der Philosophie. Mit Ontologien wurden in der Philosophie die fundamentalen Fragen von „Was ist sein?“ und „Welcher Art von Dingen gibt es?“ bearbeitet (Craig, 1998, Seite 117 - 118). In der Informatik nutzt man Ontologien um in einer maschinenlesbaren Form Wissen in einer Domäne zu speichern und abrufbar zu machen. Mit Ontologien hat man unter anderem die Möglichkeit Begriffe zu klassifizieren und die Zusammenhänge zwischen diesen Begriffen zu definieren.

In der Informatik werden Ontologien unter anderem im Semantik-Web-Bereich eingesetzt. Laut Grimm u. a. (2007) ist die gängige Definition¹ von Ontologien:

„An ontology is formal explicit specification of a shared conceptualisation of a domain of interest.“ Gruber (1993)

Aus dem Semantik Web Bereich stammen die Beschreibungssprachen:

RDF: Resource Description Framework

RDFS: Resource Description Framework Schema

OWL: Web Ontology Language²

SWRL Semantic Web Rule Language

In Abbildung 2.2 werden die Zusammenhänge zwischen den einzelnen Sprachen dargestellt. Wichtig ist in dieser Abbildung die Schwelle der Entscheidbarkeit. SWRL und OWL-Full sind zwar sehr ausdrucksstark aber nicht mehr entscheidbar.

¹Im Semantik-Web-Umfeld

²Die Abkürzung OWL und nicht WOL hat ihren Hintergrund in <http://lists.w3.org/Archives/Public/www-webont-wg/2001Dec/0169.html> 1.) Es ist klar, wie OWL auszusprechen ist (nämlich wie das englische Wort für Eule). 2.) Das Akronym eignet sich hervorragend zur Erstellung von Logos. 3.) Eulen werden mit Weisheit assoziiert. 4.) Es gibt eine interessante Hintergrundgeschichte. (Zuletzt abgerufen am 07.12.2011)

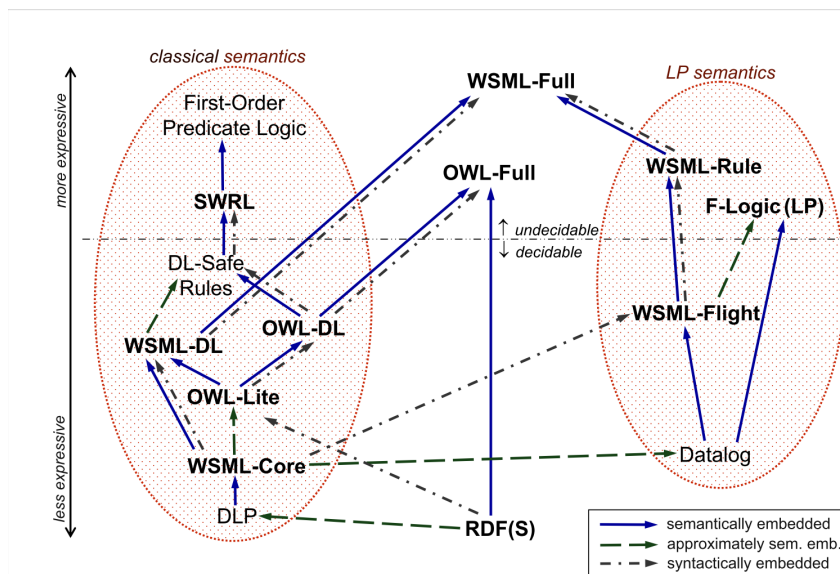


Abbildung 2.2.: Übersicht der Semantik-Web-Sprachen (vgl. Grimm u. a. (2007))

2.3.1. RDF

Das W3C (World Wide Web Consortium) beschreibt die Sprachen RDF wie folgt:

„*RDF* is a standard model for data interchange on the Web. *RDF* has features that facilitate data merging even if the underlying schemas differ, and it specifically supports the evolution of schemas over time without requiring all the data consumers to be changed.

RDF extends the linking structure of the Web to use URIs to name the relationship between things as well as the two ends of the link (this is usually referred to as a „triple“). Using this simple model, it allows structured and semi-structured data to be mixed, exposed, and shared across different applications.

This linking structure forms a directed, labeled graph, where the edges represent the named link between two resources, represented by the graph nodes. This graph view is the easiest possible mental model for *RDF* and is often used in easy-to-understand visual explanations.“ W3C³

RDF bildet das Grundgerüst von *OWL*. Durch *RDF* kann schon Wissen in einer computerlesbaren Art gespeichert und abrufbar gemacht werden. Die einzelnen Objekte in *RDF* werden durch eine URI⁴ dargestellt. Die einzelnen Objekte werden durch Prädikate verbunden. Die-

³<http://www.w3.org/RDF/> (Zuletzt abgerufen am 07.12.2011)

⁴Uniform Resource Identifier <http://tools.ietf.org/html/rfc1630> (zuletzt abgerufen am 07.12.2011)

se Verbindung sind in RDF die Tripel.

$$\text{Subjekt} \quad \Rightarrow \text{Prädikat} \quad \Rightarrow \text{Objekt} \quad (2.1)$$

$$LP : \text{Bewohner} \quad \Rightarrow LP : \text{benutzt} \quad \Rightarrow LP : \text{Dusche} \quad (2.2)$$

Durch die Darstellung von RDF mit der Hilfe von XML⁵ ist es möglich Ontologien in Namensräume aufzuteilen und „qualifizierte Namen“ zu verwenden. Beispielsweise ist aus dem Tripel 2.2 das „LP:“ ein qualifizierter Name, der der URI „<http://livingplace.informatik.haw-hamburg.de/livingplaceOntologie#>“ entspricht.

2.3.2. RDFS

Das W3C beschreibt die RDFS wie folgt:

„Descriptions used by these applications can be modeled as relationships among Web resources. The RDF data model, as specified in [RDFMS], defines a simple model for describing interrelationships among resources in terms of named properties and values. RDF properties may be thought of as attributes of resources and in this sense correspond to traditional attribute-value pairs. RDF properties also represent relationships between resources. As such, the RDF data model can therefore resemble an entity-relationship diagram. The RDF data model, however, provides no mechanisms for declaring these properties, nor does it provide any mechanisms for defining the relationships between these properties and other resources. That is the role of *RDF Schema*. „ W3C⁶

Da RDF und RDFS eng zusammenhängen wird in der Regel auch von RDF(S) gesprochen, was eine Kombination der beiden Konzepte bedeutet. Mit RDFS können zu den RDF Objekten Metadaten hinzugefügt werden. Des weiteren ist eine Klassenstruktur und das Vererbungskonzept ein Teil von RDFS. Das Vererbungskonzept gilt in RDFS auch für die Prädikate aus RDF.

2.3.3. OWL

Das W3C beschreibt die RDFS wie folgt:

⁵Extensible Markup Language <http://www.w3.org/XML/> (Zuletzt abgerufen am 07.12.2011)

⁶<http://www.w3.org/TR/2000/CR-rdf-schema-20000327/> (Zuletzt abgerufen am 07.12.2011)

„OWL is intended to be used when the information contained in documents needs to be processed by applications, as opposed to situations where the content only needs to be presented to humans. OWL can be used to explicitly represent the meaning of terms in vocabularies and the relationships between those terms. This representation of terms and their interrelationships is called an ontology. OWL has more facilities for expressing meaning and semantics than XML, RDF, and RDF-S, and thus OWL goes beyond these languages in its ability to represent machine interpretable content on the Web. OWL is a revision of the DAML+OIL web ontology language incorporating lessons learned from the design and application of DAML+OIL.“
W3C⁷

Mit OWL ist es möglich in den Strukturen von RDF(S) weitere Regeln und Axiome hinzuzufügen. Da OWL-full mit seiner hohen Ausdrucksstärke nicht mehr entscheidbar ist, gibt es die Varianten OWL-Lite und OWL-DL. Diese Varianten sind eine Teilmenge von OWL-full und beschränken den Umfang der Prädikaten-Logik auf die Logik der ersten Ordnung. Durch diese Reduktion bleibt die Sprache entscheidbar (vgl. Abbildung 2.2).

Durch die Logik von OWL ist es möglich mit Folgerungen aus dem bestehenden Wissen einer Ontologie neues Wissen zu generieren. Beispielsweise könnte in der Ontologie festgelegt werden, dass sich der Bewohner in einer Wohnung nur in einem Raum aufhalten kann. Wenn nun der Bewohner im Bett liegt, dann kann aus der Ontologie auch die Frage eindeutig beantwortet werden, ob der Bewohner sich im Bad befindet.

2.3.4. SWRL

Das W3C beschreibt die SWRL wie folgt:

„Semantic Web Rule Language (SWRL) based on a combination of the OWL DL and OWL Lite sublanguages of the OWL Web Ontology Language with the Unary/Binary Datalog RuleML sublanguages of the Rule Markup Language. SWRL includes a high-level abstract syntax for Horn-like rules in both the OWL DL and OWL Lite sublanguages of OWL. A model-theoretic semantics is given to provide the formal meaning for OWL ontologies including rules written in this abstract syntax. An XML syntax based on RuleML and the OWL XML Presentation Syntax as well as an RDF concrete syntax based on the OWL RDF/XML exchange syntax are also given.“
W3C⁸

Durch SWRL werden die Elemente aus RuleML in OWL-DL integriert. Dadurch stehen in der Ontologie Horn-artige Regeln zur Verfügung. SWRL hat allerdings den Nachteil, dass

⁷<http://www.w3.org/TR/owl-features/> (Zuletzt abgerufen am 07.12.2011)

⁸<http://www.w3.org/Submission/SWRL/> (Zuletzt abgerufen am 07.12.2011)

die Ontologie die Entscheidbarkeit verliert. Um SWRL trotzdem einsetzen zu können wurde eine Variante von SWRL entwickelt, die nur Elemente verwendet mit denen die Sprache entscheidbar bleibt. Mehr zu SWRL und den „DL-Safe-Rules“ kann in [Motik u. a. \(2004\)](#) nachgelesen werden.

2.4. Klassifikation

2.4.1. Grundlagen

Klassifikation bedeutet mit einer Funktion Daten auf verschiedene vordefinierte Klassen abzubilden (vgl. [Fayyad u. a. \(1996\)](#)). Diese Funktion wird entweder vorgegeben oder sie kann erlernt werden. Ein Klassifikationsalgorithmus durchläuft in der Regel die folgenden Schritte:

1. Definieren der Funktion oder diese mit Trainingsdaten erlernen
2. Überprüfen/Testen der erlernten Funktion
3. Anwenden der Funktion auf neue Daten
4. Optional: Anpassen der Funktion während der Anwendung

Formal bedeutet Klassifikation die Umsetzung der Funktion

$$\Phi : D \times C \rightarrow \{T, F\} \quad (2.3)$$

D die Menge der zu klassifizierenden Objekte

C die Menge der Kategorien

T True

F False

Die Zuordnung der Klassen in der Funktion [2.3](#) ist wie folgt definiert:

$$\Phi(x, c) = \begin{cases} True & x \text{ Element der Klasse } c \\ False & \text{sonst} \end{cases} \quad (2.4)$$

Ein Problem bei der Klassifikation ist die exakte Zuordnung eines Objektes zu einer Klasse. In vielen Fällen ist die Entscheidung subjektiv. Das Problem besteht in den verschiedenen Kontexten, in denen eine Information beurteilt werden soll.

Beispielsweise könnte ein Bewohner, der allein in seinem Bett schläft, zu der Klasse „Schlafen“ gehören. Es könnten aber auch die Klassen „Inaktiv“ sein, wenn der Fokus anders gelegt wird.

Bei der Klassifikation gibt es die Unterscheidung zwischen Single Label und Multi Label. Bei Single Label gehört ein Objekt immer zu einer Klasse und nur zu dieser Klasse. In dem Beispiel oben ist es entweder ein „Schlafen“ oder „Inaktiv“, aber nicht beides. Die Aktion muss aber in eine der beiden Kategorien eingeordnet werden. Eine spezielle Form der Single Label Klassifikation ist die Binary Klassifikation. Bei dieser Form gibt es nur zwei Möglichkeiten der Klassifizierung. Bei diesen Möglichkeiten ist die eine Klasse das Gegenteil der anderen Klasse. Beispielsweise „aktiv“ und „nicht aktiv“. Bei der Multi Label Klassifikation ist es erlaubt, dem Bewohner beide und weitere Klassen zuzuweisen. Es ist bei Multi Label auch erlaubt dem Bewohner keine Kategorie zuzuteilen. Dieses könnte nötig sein, wenn die Informationen für die Klassifizierung nicht vorliegen, oder es gefordert ist, dass nur eindeutige Klassifizierungen durchgeführt werden sollen. Siehe hierzu auch [Sebastiani \(2005\)](#).

2.4.2. Beurteilung der Klassifikation

Die Kombination aller Möglichkeiten der Funktion 2.3 nennt man auch den Hypothesenraum. Die einzelnen Elemente im Hypothesenraum können in der Regel nicht sicher zugeordnet werden. Um die Leistungsstärke der Funktion zu beurteilen, kann ihre Fehlerwahrscheinlichkeit mit der folgenden Funktion berechnet werden:

$$L_{(g)} = \mathbf{P}\{g(X) \neq Y\} \quad (2.5)$$

Wenn die Wahrscheinlichkeit der Verteilung zwischen den beiden Klassen im Voraus bekannt ist, dann ist die optimale Entscheidung bezogen auf die kleinste Fehlerwahrscheinlichkeit die Bayes Entscheidungsregel: „ g^* “. Die Entscheidungsregel nutzt einfach die bekannten und die beobachteten Verteilungswahrscheinlichkeiten.

$$\eta_0(x) = \mathbf{P}\{Y = 0|X = x\} \quad \eta_1(x) = \mathbf{P}\{Y = 1|X = x\} \quad (2.6)$$

Hier wird, nach Bayes, die Klasse mit der höheren Wahrscheinlichkeit gewählt.

Die Leistungsstärke der Bayes Entscheidungsregel ist:

$$L^* = L_{g^*} = \mathbf{E}[\min\{\eta_0(X), \eta_1(X)\}] \quad (2.7)$$

Leider sind in vielen Anwendungen diese Wahrscheinlichkeiten unbekannt oder nur zum Teil

bekannt. Daher werden vorher zugeteilte Feature-Vektoren benötigt (Trainingsdaten):

$$D_n = \{(X_1, Y_1), \dots, (X_n, Y_n)\} \quad (2.8)$$

2.5. Übersicht vergleichbarer Arbeiten

In [Augusto und Nugent \(2006\)](#) wird untersucht welche Potenziale in den Techniken der künstlichen Intelligenz liegen um ein Smart Home weiter zu verbessern. Folgende Techniken werden in der Arbeit untersucht:

- Räumlich-zeitliche Folgerungen; kausale Folgerungen; fallbasierte Folgerungen
- Aktionen planen
- Aktionen lernen
- Einsatz von Entscheidungsbäumen, neuronalen Netzwerken und multiagenten Systemen

[Augusto und Nugent](#) kommen dabei zu dem Schluss, dass diese Maßnahmen dazu beitragen den Kontext in der Wohnung besser zu identifizieren. Dadurch lassen sich die Reaktionen der Anwendungen besser und auch intelligenter gestalten.

In [Simpson u. a. \(2006\)](#) wird untersucht, wie Pläne in einem Smart Home erstellt werden können. Das Ziel dieser Arbeit ist es, durch das Erstellen dieser Pläne den Bewohner bei seinen Tätigkeiten zu unterstützen oder dem Bewohner Arbeiten abzunehmen. Das Erstellen der Pläne ist eine Grundvoraussetzung die Aktivitäten des Bewohners zu beobachten und zu identifizieren. Das Wissen, welche Aktivitäten der Bewohner vor hat und wie er gedenkt diese Aktivitäten umzusetzen, werden als die wichtigsten Herausforderungen für die Erstellung von Plänen in der Arbeit identifiziert.

Die Arbeit [Leake u. a. \(2006\)](#) untersucht die Möglichkeiten von fallbasierten Folgerungen in Smart Homes. In der Arbeit werden die folgenden Punkte besonders herausgestellt:

- Das Erlernen der Fälle ist eine einfache Methode das Verhalten des Smart Home anzupassen. Das Verhalten des Smart Home kann für die verschiedenen Fälle direkt aufgezeichnet werden.
- Wenn alte Fälle des fallbasierten Folgerns angewandt werden, ist es einfach möglich Erklärungen für Anomalien zu finden.
- Es können vom Bewohner einfach Trainingsdaten erzeugt werden, da die Fälle einen direkten Zusammenhang mit den tatsächlichen Tätigkeiten des Bewohners haben.

- Der Zusammenhang der Fälle zur Realität und den daraus resultierenden Aktionen des Smart Home ist ein intuitiver Weg dem Bewohner das Verhalten seines Hauses zu erklären.

Durch den Zusammenhang zwischen den Fällen und den Aktivitäten des Bewohners wird das Vertrauen des Bewohners in die Technik verstärkt, da es so möglich ist das Verhalten des Smart Home nachzuvollziehen. Dieses ist das Wichtigste, so die Autoren der Arbeit, um ein Smart Home in der Realität umzusetzen, da nur Techniken eingesetzt werden denen vertraut wird.

3. Analyse der Aktivitätserkennung im Smart Home

Bei der Aktivitätserkennung wird das Verhalten einer Person und deren Umgebung beobachtet und analysiert. Aus diesem Wissen soll dann auf Aktivitäten geschlossen werden, welche den Beobachtungen zu Grunde liegen. Hierfür sind die folgenden Schritte notwendig (vgl. [Chen und Nugent \(2009\)](#)):

- Ein Modell der Aktivitäten erstellen, in dem durch Software Veränderungen und logisches Schließen möglich sind.
- Das Verhalten der Person und Veränderungen in ihrer Umgebung beobachten.
- Aus den Informationen durch Zusammenfassen und Kombinieren eine High-Level-Abstraktion der Situation erzeugen.
- Einen Algorithmus zur Aktivitätserkennung auswählen und umsetzen.

Im Folgenden wird die Architektur für die Aktivitätserkennung beschrieben und wie die oben genannten Punkte im Living Place umgesetzt werden.

3.1. Architektur der Aktivitätserkennung

3.1.1. Modell zur Aktivitätserkennung

In einem ersten Schritt muss festgelegt werden, was eine Aktivität beschreibt und wie die beschriebenen Aktivitäten sich voneinander abgrenzen. Für das Erkennen der Aktivitäten ist es außerdem noch interessant, ob sich Muster erkennen lassen, nach denen die einzelnen Aktivitäten auftreten.

Nachdem diese Begriffe definiert sind, werden im Folgenden die Möglichkeiten beschrieben, die zur Verfügung stehen, um die Merkmale für die Aktivitätserkennung zu erzeugen. Anschließend wird die Softwarearchitektur beschrieben und erläutert, welche Möglichkeiten das Living Place Hamburg zur Aktivitätserkennung bietet oder bieten kann.

3.1.1.1. Definition von Aktivitäten

Um Aktivitäten zu beschreiben müssen die entsprechenden Begriffe definiert werden. Hierfür sind Erkenntnisse der Linguistik hilfreich. In der Linguistik werden Sätze unter anderem auf ihre Bedeutung hin untersucht. Dabei stehen auch Sätze zur Beschreibung von Handlungen im Fokus der Linguisten. Neben einem Prädikat aus einem Satz, welches sich auf die Handlung direkt bezieht, können die Subjekte und Objekte verschiedene Rollen einnehmen. Zum Beispiel:

- Sal schneidet ein Brötchen auf. (Subjekt: Sal -> ist die Handelnde oder ein Agent, welcher die Aktion ausführt)
- Das Messer schneidet ein Brötchen auf. (Subjekt: Das Messer -> ist ein Instrument, das der Agent nutzt)

Daher kann ein Satz nicht nur mit seiner ihm zugrundeliegenden Grammatik auf die Bedeutung der Wörter hin interpretiert werden. Von [Fillmore](#) werden semantische Rollen vorgestellt (siehe ([Fillmore, 1968](#), Seite 46/47)). Diese Rollen beschreiben die tieferliegende Bedeutung der Worte (Tiefenkasus), mit der die Worte in einem Satz eingesetzt werden. Die folgenden Rollen wurden von [Fillmore](#) spezifiziert:

„**Agentive (A)** the case of the typically animate perceived instigator of the action identified by the verb.

Instrumental (I) the case of the inanimate force or object causally involved in the action or state identified by the verb.

Dative (D) the case of the animate being affected by the state or action identified by the verb.

Factitive (F) the case of the object or being resulting from the action or state identified by the verb, or understood as a part of the meaning of the verb.

Locative (L) the case which identifies the location or spatial orientation of the state or action identified by the verb.

Objective (O) the semantically most neutral case, the case of anything representable by a noun whose role in the action or state identified by the verb is identified by the semantic interpretation of the verb itself; conceivably the concept should be limited to things which are affected by the action or state identified by the verb. The term is not to be confused with the notion of direct object, nor with the name of the surface case synonymous with accusative.“

[Fillmore \(1968\)](#)

Diese Liste ist laut [Fillmore](#) eine sehr grobe Einteilung und bedarf einer Erweiterung um die Möglichkeiten im Sprachgebrauch besser abzubilden. Zu erkennen ist hier, dass bei einer Handlung der Handelnde, der Ort der Handlung, betroffene sowie beteiligte Objekte eine bestimmte Rolle einnehmen.

In [Polenz \(1988\)](#) werden 19 verschiedene Rollen vorgestellt. Bei diesen Rollen werden der Agent der Handlung, das genutzte Objekt, der Ort und der Zeitpunkt weiter spezialisiert.

Eine andere Methode um eine Situation zu beschreiben ist die „5W1H-Methode“¹. In dieser Methode stehen die 6 Fragen: Wer?, Was?, Wo?, Wann?, Warum? und Wie?² im Mittelpunkt. Nach Kipling kann mit diesen Fragen eine Handlung oder Situation erfasst werden, indem man jede dieser Fragen beantwortet. Mit diesen Fragen wird nach der Rolle der Beteiligten einer Handlung gefragt. Daher ist diese Methode ein allgemeiner Weg, wenn kein bestimmter Satz vorliegt, sondern eine bestimmte Situation, in der die Beteiligten und ihre Rollen noch erst festgestellt werden müssen. Vergleicht man die 5W1H-Methode mit Fillmore:

Who Agent ⇒ nach Fillmore der Lebendige Initiator

What Beteiligtes Objekt ⇒ wie auch nach Fillmore

Where Ort der Handlung ⇒ wie nach Fillmore

When Zeitpunkt der Handlung ⇒ nicht bei Fillmore enthalten

Why Grund der Handlung ⇒ das erwartete Resultat, welches Fillmore direkt als Rolle beschreibt

How Beschreibung der Aktivitäten selbst ⇒ nach Fillmore das Verb des Satzes auf den sich die Fälle beziehen

Bei dieser 5W1H-Methode wird neben den Rollen von Fillmore zusätzlich die Zeit als beschreibendes Element hinzugezogen. Bei Polenz wurde die Zeit als Rolle auch schon aufgeführt.

Zusammenfassend lässt sich feststellen, dass in den Methoden zur Beschreibung von Handlungen oder Aktivitäten

- der Agent
- das benutzte Objekt
- der Ort
- der Zeitpunkt³

eine Rolle spielen. Diese Punkte werden durch die 5W1H-Methode beachtet. Zudem stellt die 5W1H-Methode ein Verfahren dar, mit der allgemein Aktivitäten analysiert und beschrieben werden können.

¹Die 5W1H-Methode ist auch als Kipling-Methode bekannt. Diese geht zurück auf eine Geschichte aus [Kipling \(1902\)](#)

²How, Where, What, When, Why, Who

³Zeit wurde nicht bei Fillmore aufgeführt

3.1.1.2. Abgrenzung von einzelnen Aktivitäten

Im vorherigen Abschnitt wurde eine Aktivität durch einen handelnden Agenten, ein genutztes Objekt, einen Ort und eine Zeit beschrieben. Der Begriff der Zeit ist hier noch sehr allgemein gehalten. Dieser Begriff soll im Folgenden genauer untersucht werden.

Allgemein kann festgestellt werden:

- Eine Aktivität beginnt zu einem bestimmten Zeitpunkt.
- Eine Aktivität endet an einem späteren Zeitpunkt.
- Aktivitäten können zusammengefasst werden.
- Aktivitäten können aus verschiedenen Teilaktionen bestehen.

Beispielsweise könnte das morgendliche Waschen aus dem „Betreten des Bades“, dem „Waschen am Waschbecken“ und dem „Verlassen des Bades“ bestehen. Dieses Waschen wäre wiederum Teil einer „Morgen-Routine“. Es ist auch möglich, das Waschen am Waschbecken wieder in einzelne Teilaktionen zu unterteilen. Es ist theoretisch möglich, diese Unterteilung immer weiter fortzuführen oder alle Aktivitäten zum Leben eines Menschen zusammenzufassen.

Für den praktischen Einsatz der Aktivitätserkennung ist sehr feine Unterteilung von Aktionen nicht notwendig und eine zu allgemeine Zusammenfassung von Aktivitäten ist auch nicht sinnvoll. Wenn beispielsweise die Lichtsteuerung der Wohnung das passende Licht für die Nutzung des Badezimmers bereitstellen will, dann ist es nicht notwendig, jeden einzelnen Arbeitsschritt im Bad zu erkennen und dann der Lichtsteuerung mitzuteilen. Andererseits benötigt die Lichtsteuerung eine genauere Aufteilung der Aktivitäten als die Information, dass der Bewohner sich in seiner „Morgen-Routine“ befinde. Daher sollte für die Aufteilung der Aktivitäten nicht ausschlaggebend sein, was alles möglich ist, sondern was in der Wohnung notwendig ist. Sollten sich die Anforderungen in der Wohnung ändern, muss es dann außerdem möglich sein, auch die erkennbaren Aktivitäten anzupassen.

3.1.1.3. Struktur der Aktivitäten

In jedem Bereich gibt es Aktionen, die erwartet werden und andere die aufgrund der äußeren Gegebenheiten ausgeschlossen werden können. Beispielsweise wird auf einem Flughafenvorfeld das Ankommen eines Flugzeuges erwartet oder ein Tanker, der das Flugzeug betankt [Bohlken u. a. \(2011\)](#). Es kann aber ausgeschlossen werden, dass ein Tanker zum Tanken auf das Rollfeld fährt, wenn das Flugzeug noch nicht da ist. In diesem und anderen technischen Feldern sind die Aktivitäten sehr strukturiert. Es werden nur wenige Aktionen

zu einem gewissen Zeitpunkt erwartet und viele andere können ausgeschlossen werden. Überraschungen deuten in der Regel auf einen Fehler im allgemeinen Ablauf hin.

In anderen Bereichen sind die Strukturen der Aktivitäten nicht mehr so streng strukturiert. Ein Beispiel hierfür ist eine Büroumgebung (vgl: Dai und Xu (2008)). In einem „Meeting-Raum“ gibt es beispielsweise Vorträge, Besprechungen, Pausen oder keine Veranstaltungen. Diese Veranstaltungen sind auch strukturiert aber der gesamte Ablauf hat verschiedene Freiheiten. Bei einem Vortrag könnte auch ein Zuschauer dem Redner Fragen stellen oder in einer Besprechung gibt es einen Vortrag als Eröffnung. Unerwartete Ereignisse bedeuten nicht immer einen Fehler. Beispielsweise kann es eine Pause zu jedem Zeitpunkt geben, auch wenn diese nicht im vorher festgelegten Ablauf vorgesehen ist. Sollte die Pause aber innerhalb der ersten fünf Minuten erfolgen, könnte es sich doch um einen Fehler handeln.

Im Bereich des Smart Home gibt es kaum Strukturen oder Erwartungen. Die Handlungen verschiedener Bewohner gleichen sich in der Regel nicht und unerwartete Aktionen sind jederzeit möglich. Strukturen können nur in wenigen Teilbereichen gefunden werden oder sind auf einen bestimmten Bewohner zugeschnitten. Beispielsweise kann nach einer „Zu Bett gehen“-Aktion eine „Aufstehen“-Aktion erwartet werden. Wann die „Aufstehen“-Aktion stattfindet kann aber nicht mit Sicherheit vorausgesagt werden. Es ist möglich, dass der Bewohner mitten in der Nacht aufwacht und aufsteht oder dass er trotz Wecker bis zur letzten Minute im Bett bleibt und erst dann aufsteht. Teile einer „Morgenroutine“ sind möglicherweise „Aufstehen“, „Anziehen“, „Frühstücken“, und „kurz Waschen“. Diese Aktionen müssen aber nicht in dieser Reihenfolge ablaufen oder überhaupt eintreten. Beispielsweise könnte ein Frühstück in der Firma angesetzt sein und der Bewohner lässt es daher in der Wohnung ausfallen. Dieses ist dann kein Fehler oder Problem im Ablauf.

3.1.1.4. Möglichkeiten Aktivitäten zu erkennen

Um die Aktivitäten zu erkennen, müssen entweder die Aktivitäten direkt oder indirekt beobachtet werden.

Ein Beispiel für eine direkte Beobachtung von Aktivitäten ist der Einsatz von Kameras. Hier werden die Bilder der Kamera analysiert um aus den einzelnen Bildern Aktionen abzuleiten. Beispielsweise wurde in Sung u. a. (2011) mit einer Kinect⁴ die Bildfolge einzelner Körperpositionen auf Aktivitäten übertragen. Grundsätzlich sind Kameras in der Lage alle Aktivitäten zu erkennen. Die Wohnung muss nicht mit zusätzlichen Sensoren ausgestattet werden und auch der Bewohner braucht keine Teile bei sich tragen. Problematisch bei der Bildauswertung wird es, je mehr sich die Aktivitäten in ihrem äußeren Bild gleichen. Dann wird es umso

⁴Gerät um einen Körper inklusive Tiefeninformationen zu scannen. http://www.microsoft.com/Presspass/press/2010/mar10/03-31PrimeSensePR.aspx?rss_fdn=Press%20Releases zuletzt abgerufen am 20.12.2011

schwieriger diese Unterschiede aus den Bildern herauszufiltern. Weitere Nachteile beim Einsatz von Kameras ist die Ausleuchtung des Raumes und tote Winkel in dem Blickfeld der Kameras. Wenn beispielsweise sich zwei Personen im Raum aufhalten, könnte es sein, dass die eine Person die zweite verdeckt. Die Bildverarbeitung hat zurzeit auch noch Probleme Ergebnisse in Echtzeit oder nahe Echtzeit zu liefern. Beispielsweise benötigen die aktuellen Algorithmen ca. 10 Sekunden um ein Gesicht zu identifizieren (vgl. [Verma u. a. \(2010\)](#)).

Indirekte Beobachtung von Aktivitäten bedeutet, die Ursache und die Wirkung einer Aktivität zu interpretieren. Wenn beispielsweise der Bewohner in seinem Bett schläft können die folgenden indirekten Merkmale beobachtet werden:

- Der Bewohner befindet sich im Bett.
- Das Licht in der Wohnung ist ausgeschaltet.
- Das Bett registriert eine Person.
- Die Bewegung auf dem Bett ist kaum vorhanden.
- Wenige Geräusche sind in der Wohnung zu vernehmen.
- Es ist nachts.

Neben den bauartbedingten Schwankungen der Messwerte liefern die Sensoren relativ eindeutige Werte. Eine Auswertung der Sensoren ist mit recht geringem Rechenaufwand möglich. Aber wenn ein Sensorwert eine bestimmte Schwelle überschreitet, bedeutet es nicht in jedem Fall, dass auch eine Aktion mit diesem Sensor gemessen werden kann. Der Sensor ist nur ein indirektes Merkmal. Liefern beispielsweise die Sensoren die oben genannten Informationen, dann kann es auf einen schlafenden Bewohner hindeuten. Es könnte aber auch sein, dass er gerade am telefonieren ist und dem Gesprächspartner zuhört ohne zu sprechen. Für die indirekte Beobachtung von Aktivitäten muss daher immer auch davon ausgegangen werden, dass die Erkenntnisse nicht zutreffen können.

Neue Gegenstände im Haushalt müssen in das System integriert werden. Wenn es notwendig ist zu erkennen, dass Teller auf dem Küchentisch stehen, dann müssen neue Teller auch mit Sensoren versehen werden.

3.1.2. Aufbau der Aktivitätserkennung

Die Aktivitätserkennung soll nicht auf eine spezielle Wohnung beschränkt sein. Der Fokus dieser Arbeit liegt auf dem Modellieren und Erkennen von Aktivitäten. Als Grundlage für die Erkennung der Aktivitäten sollen die Sensoren der Wohnung dienen. Dabei können die Kameras auch als Sensoren betrachtet werden. Beispielsweise könnte eine Kamera einen Teller auf einem Tisch erkennen, ohne dass dieser mit einem Sensor versehen werden muss

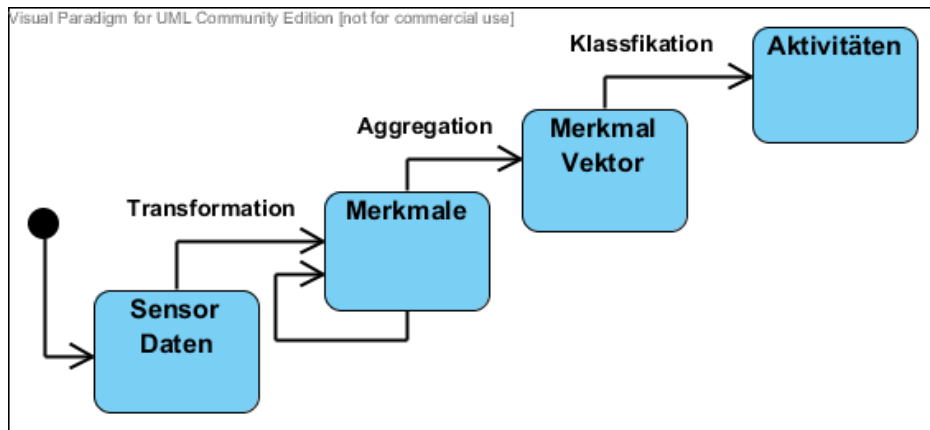


Abbildung 3.1.: Aktivitätserkennung

(vgl. Najem (2011)). Aus den Bilddaten könnte auch erkannt werden, wie viel Bewegung in der Wohnung existiert⁵ (vgl. Teske (2011)).

Allgemein ist bei der Erkennung der Aktivitäten nicht der einzelne Sensor relevant, sondern die Information für die der Sensor in der Wohnung eingesetzt wird. Beispielsweise ist es unwichtig, mit welchem System ermittelt wird, dass sich der Bewohner im Bereich des Bettes befindet. Nur die Information, „Bewohner ist im Bereich des Bettes“ wird benötigt. Genauso ist es uninteressant, ob ein Schalter in der Wohnung gerade schaltet. Die relevante Information ist, dass eine Tür offen oder geschlossen ist. Daher ist eine Abstraktionsebene sinnvoll, in der aus den Rohdaten der Sensoren die relevanten Informationen oder Features für die Aktivitätserkennung generiert werden. Hierbei muss man sich bewusst machen, dass durch die Abstraktion die Informationen zu Hypothesen werden. Beispielsweise könnte ein Drucksensor im Bett auf einen Stapel Bücher ansprechen. Die Interpretation würde dann „Person liegt im Bett“ melden, obwohl diese Information mit der Realität nicht übereinstimmt.

Wahrscheinlich ist in vielen Fällen für die Erkennung der Aktivitäten nicht jedes Features relevant. Beispielsweise muss für eine Aktivität im Bett nicht der Wasserfluss im Bad überprüft werden. Wenn für die Erkennung die einzelnen Features zu speziellen Features-Vektoren zusammengefasst werden, kann so der Aufwand für die Weiterverarbeitung der Daten reduziert werden.

In Abbildung: 3.1 sind die Komponenten für die Aktivitätserkennung zusammengefasst. Sie teilen sich in die folgenden Bereiche auf:

⁵Mit geringem Aufwand ist dieses durch Differenzbilder möglich. Dabei wird nicht unterscheiden, ob diese Bewegung der Bewohner oder beispielsweise ein Vorhang erzeugt.

- Daten-Transformation
- Features-Aggregation
- Klassifikation der Aktivitäten als High-Level-Kontext

Die Daten-Transformation überführt die Werte der verschiedenen Sensoren oder der externen Informationsquellen in ihre Merkmale oder Features (vgl. [Pautz \(2010\)](#)). Dabei kann ein Features auch aus mehreren Sensoren oder anderen Features erzeugt werden. Es ist auch möglich, einen Sensor über einen Zeitraum zu interpretieren, um die erste oder zweite Ableitung der Zeit des Sensorwertes zu erhalten. Verschiedene Features zusammengefasst ergeben einen Features-Vektor. Aufgrund dieses Features-Vektors wird mit Klassifikationsverfahren die aktuelle Situation der Wohnung erkannt.

3.1.3. Architektur der Aktivitätserkennung

Wie im letzten Abschnitt festgestellt wurde, ist die Anzahl der Sensoren oder die Anzahl der verschiedenen Aktivitäten nicht festgelegt. Diese Parameter können sich je nach Wohnung, in der das System eingesetzt wird, ändern. Die Architektur muss daher so gewählt werden, dass die verschiedenen Information und Komponenten dem Prozess hinzugefügt oder wieder entfernt werden können, ohne dass das gesamte System neu entwickelt werden muss. Außerdem soll eine neue Definition einer Aktivität integrierbar sein, wenn sie von einem Modul benötigt wird.

Das Ziel der Aktivitätserkennung ist Kontext-Informationen über die Wohnung zur Verfügung zu stellen. Die Aktivitäten sind die Kontext-Informationen um die es in dieser Arbeit geht. Verschiedene Module der intelligenten Wohnung könnten aber außerdem Interesse an anderen Kontext-Informationen haben. Hier stellt sich die Frage, welche Informationen noch zum Kontext gehören. [Abowd u. a.](#) definieren Kontext wie folgt:

„Context is any information that can be used to characterize the situation of an entity. An entity is a person, place, or object that is considered relevant to the interaction between a user and an application, including the user and applications themselves“ [Abowd u. a. \(1999\)](#)

Nach dieser Definition können alle Sensor-Daten, alle Informationen aus den Zwischenschritten und die Ergebnisse der Aktivitätserkennung selbst für eine Anwendung von Interesse sein. Daher muss die gesamte Architektur so gewählt werden, dass alle Informationen jeder Instanz zugänglich sind, welche diese benötigt. Des Weiteren ist eine lose Kopplung der einzelnen Komponenten wichtig, damit jederzeit zusätzliche Komponenten in das System integriert werden können.

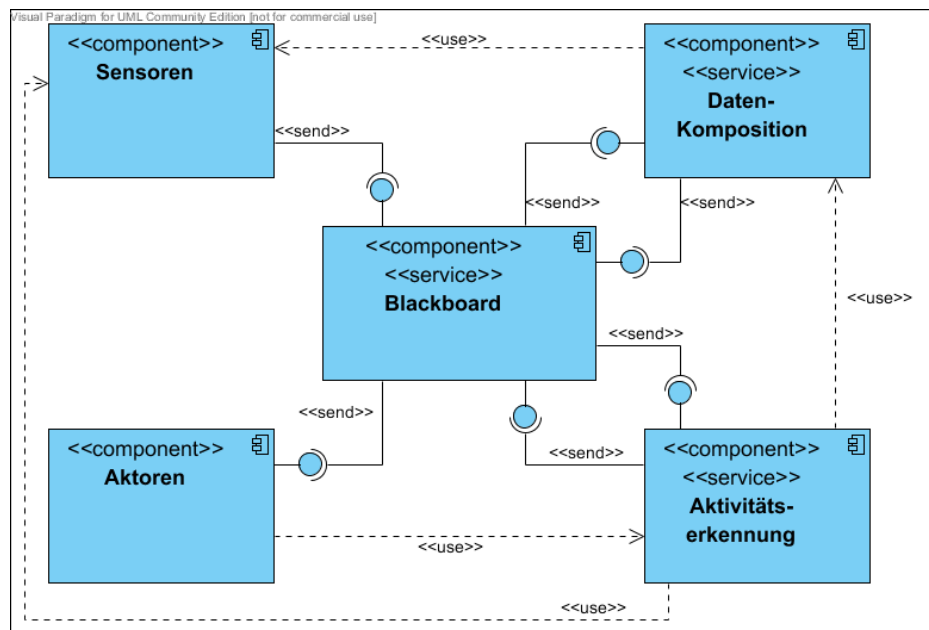


Abbildung 3.2.: Blackboard als Zentrale Kommunikationsschnittstelle

Um die Dynamik der Komponenten zu gewährleisten und die Kommunikation offen zu halten, soll in dieser Arbeit die Blackboard-Architektur (vgl. [Erman u. a. \(1980\)](#)) eingesetzt werden. Das Blackboard ist eine Kommunikationsschnittstelle zwischen Komponenten mit der die Komponenten gleichberechtigt und entkoppelt miteinander kommunizieren können. Auf dem Blackboard werden alle Informationen (Sensordaten, Features, Kontext) für alle anderen Komponenten zur Verfügung gestellt. Die einzelnen Komponenten melden sich für genau die Informationen am Blackboard an, die sie benötigen. So können verschieden Erzeuger oder Abonnenten von Informationen dem System hinzugefügt oder entfernt werden. Der Nachteil dieser Entkopplung besteht in der Gefahr, dass es für eine Information keinen Erzeuger geben könnte oder dass Informationen auf dem Blackboard hinterlegt werden, die keinen Abnehmer haben. Dieses muss bei der Entwicklung der Programme berücksichtigt werden. In [Abbildung: 3.2](#) ist zu sehen, wie das Blackboard als zentrale Schnittstelle eingesetzt wird. Die Sensor-Daten werden über verschiedene Ebenen interpretiert. Am Ende steht in diesem Beispiel die Aktivitätserkennung, die von anderen Komponenten (z. B. Aktoren) genutzt wird. Weitere Informationen zum Zusammenspiel der verschiedenen Ebenen sind in [Ellenberg u. a. \(2011\)](#) genauer erläutert.

3.1.4. Bedingungen im Living Place

Die oben vorgestellten Methoden sollen im Living Place umgesetzt werden. Nach der 5W1H-Methode sind Agent, Ort, Gegenstand und Zeit der Handlung die beschreibenden Dimensio-

nen der Aktivitätserkennung. Im Folgenden werden die Bedingungen des Living Place in diesen Dimensionen beschreiben.

Die „Why-Dimension“ ist nicht Gegenstand dieser Arbeit. Wie in [Greenberg \(2001\)](#) beschrieben wurde, ist es nicht möglich die Intention einer Person zu messen. Daher kann man nur durch statistische und andere Verfahren sich diesem Wissen nähern. Diese Verfahren umzusetzen würde den Rahmen dieser Arbeit übersteigen.

3.1.4.1. Who-Dimension

Mit dieser Dimension soll der Handelnde einer Aktion identifiziert werden. Dieses kann beispielsweise über eine Gesichtserkennung oder mit einem Sensor geschehen, den der Bewohner mit sich trägt. Die Gesichtserkennung hätte den Vorteil, dass der Bewohner keinen Gegenstand tragen muss. Der Sensor hingegen ist einfacher und schneller auszuwerten, gegenüber dem Rechenaufwand, den eine Gesichtserkennung benötigen würde. Da in der Wohnung schon eine Identifizierung des Bewohners durch einen Sensor existiert, soll dieses System hier eingesetzt werden (vgl. [Otto und Voskuhl \(2011\)](#)). Durch die Blackboard-Architektur ist es später ohne größeren Aufwand möglich, das Sensor-System durch eine Gesichtserkennung zu ersetzen oder beide Systeme parallel laufen zu lassen.

3.1.4.2. Where-Dimension

Der Ort der Aktion hilft auf die Aktion zu schließen, die von dem Bewohner durchgeführt wird. Grundsätzlich kann man davon ausgehen, dass die Aktion des Bewohners auch an der Stelle ausgeführt wird, an der er sich befindet. Daher ist ein Feature für die Klassifikation die Position des Bewohners. In der Regel muss sich der Bewohner innerhalb eines bestimmten Bereiches um den Gegenstand befinden, den er für seine Aktion nutzt. Hierfür wird ein Features aus dem „Nutzungsbereich“ (oder Functional-Space siehe: [Bhatt u. a. \(2009\)](#)) und der aktuellen Position des Bewohners gebildet werden. In diesem Feature wird angegeben, in welchem Functional-Space sich der Bewohner gerade befindet und somit welche Gegenstände er gerade benutzen könnte. In [Karstaedt und Wendholt \(2011\)](#) wird ein dreidimensionales Modell der Wohnung erstellt. In diesem Modell sind auch alle Gegenstände mit ihrer Position und Functional-Spaces hinterlegt. Das 3D Modell wird außerdem zur Realität aktuell gehalten. Wenn beispielsweise ein Stuhl in der Realität verschoben wird, dann verschiebt er sich auch direkt in dem Modell. In dem Modell wurden die Functional-Spaces modelliert. Somit kann das Modell die Ortsinformation des Bewohner und die Informationen über die Functional-Spaces der Gegenstände an Interessierte Anwendungen liefern. Interessierte Anwendungen sind beispielsweise die Aktivitätserkennung.

3.1.4.3. When-Dimension

Wie in den „Möglichkeiten zur Aktivitätserkennung“ (Abschnitt: 3.1.1.4) beschrieben wurde, sind die Aktivitäten in einer Wohnung kaum strukturiert. Es ist kein Problem oder ein Fehler, wenn der Bewohner nachts um 4 Uhr ohne Termin aufsteht. Trotzdem gibt es im alltäglichen Leben gewisse Strukturen. Bestimmte Aktivitäten treten die in der Regel nur zu bestimmten Zeiten auf. Aufgrund der zeitlichen Komponente sind manche Aktionen dadurch wahrscheinlicher als andere. Beispielsweise ist es an einem „Arbeitstag“ wahrscheinlich, dass der Bewohner morgens aufsteht und auch innerhalb eines gewissen Zeitintervalls das Haus verlässt. Hingegen wird das Frühstück an einem Wochenende länger sein und könnte durch andere Aktivitäten unterbrochen werden. Die Strukturen teilen sich in drei verschiedene Aspekte auf:

- Erwartete temporal geordnete Folgen von Aktionen, z.B.:
 - Auf ein „Aufstehen“-Event folgt meistens auch „Kleidung anziehen“-Event.
 - Wenn es kein „Kleidung anziehen“ gegeben hat, wird es auch kein „Wohnung verlassen“ geben.
- Wechselseitige Handlungen, z.B.:
 - Einen Bereich betreten \Leftrightarrow Den Bereich verlassen.
 - Kleidung anziehen \Leftrightarrow Kleidung ausziehen.
- Temporale Intervalle an denen Aktionen erwartet werden, z.B.:
 - Nachts wird im Bett „geschlafen“ und tagsüber ist „TV schauen“ wahrscheinlicher, auch wenn in beiden Fällen der TV aktiv ist.
 - An einem Arbeitstag ist das „Arbeitskleidung anziehen“ wahrscheinlich, an einem arbeitsfreien Tag unwahrscheinlich.

Diese Umstände können ausgenutzt werden, um Aktionen zu beschreiben. Beispielsweise könnten bestimmte Szenarien für einen Arbeitstag beschrieben werden. Diese Szenarien müssen zwar dynamisch bleiben aber sie unterscheiden sich von den Szenarien für einen arbeitsfreien Tag.

Eine Herausforderung besteht in der Definition der ungenauen temporären Begriffe, wie beispielsweise „morgens“ oder „mittags“. Diese Begriffe haben nicht für alle Bewohner die gleiche Bedeutung. Beispielsweise könne mit „mittags“ der Zeitpunkt 12 Uhr oder 13 Uhr gemeint sein oder „morgens“ beginnt um 4 Uhr. Der „Morgen“ könnte aber auch erst um 8 Uhr beginnen. Auch könnte dieser Begriffe an verschiedenen Tagen für denselben Bewohner eine andere Bedeutung zukommen. Beispielsweise beginnt der Morgen an einem Arbeitstag um 5 Uhr während dieser Zeitpunkt an einem arbeitsfreien Tag noch „nachts“ bedeutet.

Bei der Verwendung dieser Begriffe, muss darauf geachtet werden, ob die unterschiedliche Verwendung die Definitionen der verschiedenen Aktionen verändert. Es ist andererseits auch möglich, dass durch die Verwendung solcher unscharfen Zeiten die Definition der Aktivitäten sich dem Bewohner dynamisch anpassen. Beispielsweise könnte das Frühstück als Mahlzeit am Morgen definiert werden. An einem Arbeitstag wäre dann eine Mahlzeit um 11:30 kein Frühstück, an einem arbeitsfreien Tag schon.

3.1.4.4. What-Dimension

In dieser Dimension werden die Werkzeuge oder Objekte beschrieben, mit denen eine Aktivität ausgeführt wird. Allgemein kann man feststellen, dass nicht bei jeder Aktivität ein Gegenstand erforderlich ist. Beispielsweise kann man im Bett schlafen aber auch sonst überall in der Wohnung. Sollte aber das Bett genutzt werden und die anderen Dimensionen passen zu einer schlafenden Person, dann ist dieser „benutzte“ oder „aktive“ Gegenstand ein weiteres Features für die Klassifikation der Aktivität „schlafen“. Somit geht es in dieser Dimension darum festzustellen, ob ein Gegenstand zur Nutzung zur Verfügung steht oder „aktiv“ ist.

Es gibt grundsätzlich zwei Wege, um zu erkennen ob ein Gegenstand gerade „aktiv“ ist.

1. Der Gegenstand kann es selbst erkennen und diesen Umstand über das Blackboard mitteilen.
2. Der Gegenstand wird beobachtet und der Beobachtende gibt über das Blackboard darüber Auskunft, ob ein Gegenstand benutzt werden kann.

Beispielsweise ist das intelligente Bett in der Lage zu erkennen, ob der Bewohner das Bett gerade nutzt (vgl. [Hardenack \(2011\)](#)). Ein Beispiel für die zweite Möglichkeit wäre die Dusche der Wohnung. Mit einem Durchflusssensor kann die Funktion der Dusche beobachtet und mitgeteilt werden. In der Arbeit von [Najem \(2011\)](#) geht es darum, Objekte in der Wohnung durch die Kameras zu identifizieren. Somit ist es möglich, durch die Kameras indirekt die „Aktivierung“ eines Gegenstandes mitzuteilen. Beispielsweise könnte durch die Kamera erkannt werden, wenn der Bewohner Teller aus dem Schrank holt und diese auf den Tisch stellt. Diese Teller auf dem Tisch würden dann zur Nutzung zur Verfügung stehen.

Wie oben beschreiben wurde, kann eine Beobachtung falsche Informationen erzeugen. Daher muss bei der beobachteten Funktion von Gegenständen, die ihre Aktivität nicht selbst erkennen, von einer Hypothese ausgegangen werden und nicht von einer Tatsache. Eine weitere Herausforderung stellt die Blackboard-Architektur dar. Durch die lose Kopplung kann es passieren, dass mehrere Beobachter das gleiche Objekt beobachten und auch seine Aktivität mitteilen oder es gibt niemanden, der in der Lage ist diesen Service anzubieten. Eine Lösung dieses Problems könnte ein zentraler Provider sein, der Auskunft über alle Sensoren und Dienste im Living Place anbietet (vgl. [Johannsen \(2011\)](#)).

3.1.4.5. How-Dimension

In dieser Dimension soll beschrieben werden, wie eine Aktivität ausgeführt wird⁶. In dieser Arbeit werden die Voraussetzungen und Ergebnisse der Aktivitäten in den oben beschriebenen Dimensionen beobachtet. Aus den Beobachtungen werden anschließend in dieser Dimension Hypothesen aufgestellt, um welche Aktivitäten es sich handelt. Jede Aktivität hat bestimmte Voraussetzungen (Features), die erfüllt sein müssen um eine Hypothese aufzustellen und andere Features, welche die Hypothese zusätzlich unterstützen. Beispielsweise kann der Bewohner nur dann Duschen, wenn er sich im Bad befindet (Where-Dimension) und die Dusche auch aktiv ist (What-Dimension). Ein unterstützendes Feature wäre hier die Helligkeit im Bad, da es wahrscheinlich ist, dass der Bewohner Licht zum Duschen benötigt.

Wie in [Bohlken u. a. \(2011\)](#) beschrieben, muss man zwischen den Aktivitäten und den Ereignissen (Events) unterscheiden. Aktivitäten sind hier Handlungen über einen bestimmten Zeitraum und gehören zu dieser Dimension. Events sind die Übergänge zwischen den Aktivitäten. Sie haben in der Regel auch keinen Zeitraum sondern nur einen Zeitpunkt. Dieser Zeitpunkt (Anfang und Ende) bzw. der Zeitraum einer Aktivität, wird durch das Auftreten der Features in einer der anderen Dimensionen festgelegt.

Am besten wird der Unterschied zwischen Events und Aktivitäten durch das Event „Kleidung anlegen“ verdeutlicht. Dieses Event könnte in einem ersten Ansatz als eine Aktivität über einen bestimmten Zeitraum interpretiert werden. Hier stellt man aber fest, dass man entweder nur einen messbaren „Anfang“ oder ein messbares „Ende“ der Handlung finden kann und nicht beides. Beispielsweise könnte die Handlung damit beginnen, wenn die Kleidung aktiviert wird. Dann stellt sich aber die Frage, wann die Handlung endet. Umgekehrt könnte die „aktive“ Kleidung das Ende der Handlung markieren. Dann ist aber die Frage, was den Beginn der Handlung bestimmen soll. Vielmehr ist es so, dass dieses Event den Übergang zwischen den Aktivitäten „keine Kleidung tragen“ und „Kleidung tragen“ markiert, mit den Zeitpunkten „Kleidung aktiv“ und „Kleidung inaktiv“. Dieses Event ist somit Teil der „What-Dimension“, die erkennt, ob die Kleidung benutzt wird.

3.2. Beschreibung der Aktivitäten

In diesem Abschnitt wird analysiert, welche Möglichkeiten es gibt, die Aktivitäten darzustellen und zu erkennen. Im letzten Abschnitt wurde das 5W1H-Modell zur Beschreibung der Situation in der Wohnung beschrieben. Im ersten Teil dieses Abschnittes werden verschiedene Datenstrukturen und Sprachen analysiert, mit denen die Dimensionen des 5W1H-Modells

⁶bzw. die Aktivität selbst wird beschrieben

System	Space	Time	Space/Time coordinates (Relative or Absolute)	Context history	Subject (User or Application)	User profile (Role or Features based)	Variable context granularity	Valid context constraints	Type of formalism: Key-value-based	Type of formalism: Mark-up based	Type of formalism: Logic-based	Type of formalism: Graph-based	Type of formalism: Ontology-based	Formality level (High or Low)	Flexibility	Context construction (Distributed or Centralized)	Context reasoning	Context quality monitoring	Ambiguity/Incompleteness mgmt.	Automatic learning features	Multi-context model
ACTIVITY	+		A	+	U	F	+							L	+	C	+				+
CASS	+	+	A	+	U	A								L	D	D	+				
CoBrA	+	+	A		A	F				+				H	+	D					
CoDaMoS	+	+	R/A		A	F								H	+	D	+				
COMANTO	+	+	R/A		A	F								H	+	D					
Context-ADDICT	+	+	R/A		A	R	+		+					H	+	C	+				
Conceptual-CM	+	+	R	+	A	R								L	+	C	+				
CSCP					A	F								L	+	C					
EXPDOC		+	R	+	U	F								H	+	C					
FAWIS				+	U	F		+						H	+	C					
Graphical-CM	+	+	R		A	F	+							H	+	C	+				
HIPS/HyperAudio	+	+	A	+	U	F			+					L	+	C	+				
MAIS	+	+	A		A	F	+							H	+	C					
SCOPEs					A	A				+				H	+	D					
SOCAM	+	+	R/A		A	F								H	+	D	+				
U-Learn	+	+	A		U	F	+							H	+	D	+				

Abbildung 3.3.: Darstellung von Kontext (vgl. Bolchini u. a. (2007))

beschrieben werden können. Der zweite Teil befasst sich mit der Definition der Aktivitäten und den Zusammenhängen zwischen der „How-Dimension“ und den anderen Dimensionen. Anschließend werden die Klassifikationsverfahren analysiert, welche die Aktivitäten erkennen sollen.

3.2.1. Darstellung der Dimensionen

3.2.1.1. Übersicht

In diesem Kapitel werden die Kriterien beschrieben, mit denen die verschiedenen Datenstrukturen bewertet werden sollen. Das 5W1H-Modell beschreibt allgemein die Situation oder den Kontext in der Wohnung. Es soll mit der Datenstruktur möglich sein dieses Modell abzubilden.

In Bolchini u. a. (2007) wurden verschiedene Arten der Notation von Kontext-Informationen gegenübergestellt. Dabei wurden die folgenden Kategorien unterschieden und bewertet (vgl. Tabelle 3.3). Diese Kriterien sollen auch in dieser Arbeit die Grundlage für die Analyse der Datenstrukturen liefern.

Dimension:

- Ist das Modell in der Lage **räumliche** Komponenten darzustellen?
- Ist das Modell in der Lage **zeitliche** Komponenten darzustellen?
- Sind die Repräsentationen von Zeit und Raum absolut oder relativ?
- Kann die Vergangenheit Teil des aktuellen Kontextes werden?
- Was ist das Subjekt der Beschreibung?
- Gibt es ein Profil für das Subjekt der Beschreibung?

Repräsentationsmerkmale:

- Art der Formalisierung (z. B. key-value-, mark-up-schema-, logik-, graph- oder ontologiebasiert).
- Existiert eine formale Definition der Formalisierung?
- Ist das Modell flexibel im Bezug auf verschiedene Domänen oder Anwendungen?
- Ist die Tiefe der Auflösung veränderbar und gibt es verschiedene Detail-Tiefen?
- Kann man Regeln für die Gültigkeit des Kontextes einfügen?

Verwaltung und Nutzung:

- Kann der Kontext zentral oder verteilt erstellt werden?
- Sind Schlussfolgerungen aus dem Kontext möglich?
- Gibt es eine Qualitätsüberwachung um beispielsweise fehlerhafte Sensoren aufzudecken?
- Wie ist die Mehrdeutigkeit und Fehler-Verwaltung (Fehler-Transparenz) gelöst?
- Ist automatisches Lernen von Kontext möglich?
- Kann mehrfacher Kontext verwaltet werden?

3.2.1.2. Dimensionen des Modells

Aus dem 5W1H ergeben sich die sechs Dimensionen Agent, Objekt, Zeit, Raum, Intension und Aktivität. Die Datenstruktur muss in der Lage sein alle diese Dimensionen mit ihren Besonderheiten darstellen zu können.

Die Raum-Dimension erfordert einen hierarchischen Aufbau. Räume können andere Räume enthalten. Räume stehen manchmal in einem bestimmten Verhältnis zueinander oder gehören zu einem bestimmten Objekt. Es muss in der Datenstruktur möglich sein diese Zusammenhänge abzubilden.

Das Modell muss mit verschiedenen Identitäten umgehen können. In diesem Projekt handelt es sich um einen Ein-Personen-Haushalt, in dem es nur einen Bewohner gibt. Das Modell muss aber auch hier zwischen einem Besucher und dem Bewohner unterscheiden können. Grundsätzlich soll es in der Datenstruktur möglich sein verschiedene Bewohner darzustellen.

Die zeitliche Komponente muss durch das Modell auch unterstützt werden. Wie im letzten Kapitel beschrieben, lassen sich verschiedene temporäre Strukturen erkennen, mit denen die möglichen Aktivitäten eingeschränkt werden können. Für diese Dimension ist die „Allen’s Temporal Logic“ (vgl. [Allen \(1983\)](#)) wichtig, um die temporären Zusammenhänge der Zeitintervalle auszudrücken.

Die Aktivitäten stehen meistens in Beziehung zu einem Gegenstand. Daher muss neben dem Bewohner in der Datenstruktur die gesamte Wohnung mit allen relevanten Gegenständen modelliert werden, die in einer Aktivität gebraucht werden. Des Weiteren muss es auch möglich sein, die Beziehungen zwischen den Gegenständen und der Aktivität darzustellen.

Die Dimension der Aktivitäten selbst muss mit den anderen Dimensionen in Zusammenhang gebracht werden können. Es ist notwendig einen zeitlichen Ablauf der Aktivitäten zu modellieren, da die zeitliche Struktur auch zwischen den verschiedenen Aktivitäten besteht, im Gegensatz zu der Zeit-Dimension, die die Zeit der Aktivität beschreibt aber nicht die temporalen Verhältnisse.

3.2.1.3. Repräsentationsmerkmale des Modells

Das Modell muss unterschiedliche Detailstufen unterstützen. Die Anwendungen, die auf die verschiedenen Aktivitäten reagieren, geben vor, wie detailliert die Aktionen definiert werden müssen. Hier kann es Anwendungen geben, denen die Aktivität „Morgen-Szenario“ genügt. Andere Anwendungen könnten auf den Vorgang des Kaffeekochens reagieren und müssen diesen Schritt explizit im Morgen-Szenario mitgeteilt bekommen.

Eine hohe Formalisierung bedeutet, dass die Begriffe eindeutiger definiert sind und die Freiheiten eingeschränkt werden. Die Begriffe können weiterhin eine ungenaue Bedeutung haben, beispielsweise „Draußen“. Durch die Formalisierung werden aber die Verhältnisse zu den anderen Begriffen eindeutiger und es wird klarer welchem Rahmen sich diese Ungenauigkeit hat. In dem Living Place gibt es verschiedene Projekte die auch in Zukunft die Aktivitätserkennung nutzen. Um bei dieser Zusammenarbeit die Fehlerquellen gering zu halten, ist es sinnvoll möglichst eindeutige Schnittstellen zu erstellen. Somit ist eine hohe Formalisierung der Datenstruktur erforderlich.

Zwischen den Aktivitäten sind komplexe Zusammenhänge zu erwarten, da die Freiheitsgrade im Ablauf in einer Wohnung sehr hoch sind. Um diese Zusammenhänge abbilden zu können müssen zwischen den Daten Regeln erstellt werden. Es wäre von Vorteil, wenn diese Regeln schon ein Teil der Datenstruktur sind.

Welche Aktivitäten in der Zukunft modelliert werden sollen kann nicht abgeschätzt werden. Daher muss die Datenstruktur flexibel anpassbar sein, um neue Aktivitäten aufzunehmen. Die neuen Aktivitäten müssen auch später noch in das Modell integrierbar sein, auch wenn sie völlig neu definiert werden.

3.2.1.4. Verwaltung/Nutzung des Modells

Wenn die Wohnung einmal modelliert wurde, kann man in der Regel davon ausgehen, dass sich nicht viel verändert. Es werden aber immer wieder Gegenstände in der Wohnung hinzugefügt oder entfernt. Hier muss dann das Modell entsprechend angepasst werden. Ebenso kann es neue Anwendungen geben, die eine andere Aktivität erkannt haben wollen. Diese Aktivität muss dann auch in das Modell eingefügt werden. Um die Übersicht zu wahren, sollte es auch möglich sein, nicht mehr benötigte Aktivitäten zu entfernen.

Schlussfolgerungen über neue Informationen werden in dem Modell eher nicht benötigt. Die Datenstruktur hat die Aufgabe die Wohnung und die Aktivitäten abzubilden. Daher soll das benötigte Wissen vollständig modelliert werden.

Eine automatische Validierung der Datenstruktur wäre von Vorteil. Das Modell kann durchaus sehr komplex werden und daher könnte eine Möglichkeit zur automatischen Fehlererkennung hilfreich sein. Dies ist aber nicht zwingend erforderlich. Fehler können auch vermieden werden, wenn die Datenstruktur entsprechend strukturiert und formalisiert ist.

3.2.1.5. Darstellungsform für das Living Place

Um den oben genannten Anforderungen zu genügen, wird eine Datenstruktur mit den folgenden Eigenschaften benötigt:

- Stark formalisiert
- Strukturiert
- Abbildung von (räumlichen-)Hierarchien
- Abbildung von „Allen´s Temporal Logic“
- Erweiterbar
- Wiederverwendbar

Eine Ontologie (vgl. [Wand und Weber \(1990\)](#), ([Russell und Norvig, 2004](#), Seite: 397 ff.)) ist eine formale und bekannte Datenstruktur. Sie unterstützt Hierarchien und mit der Regelsprache SWRL⁷ können auch temporale Zusammenhänge abgebildet werden. Wie auch schon in [Ellenberg \(2010a\)](#) untersucht wurde, bieten Ontologien die geforderten Eigenschaften um die Dimensionen des 5W1H-Modells darzustellen. Auch in dem Projekt von [Bohlken u. a. \(2011\)](#) wurde eine Ontologie erfolgreich bei der Aktivitätserkennung eingesetzt. In der Arbeit von [Winkler \(2007\)](#) wurden Ontologien auf ihre Eigenschaften untersucht, ob sie sich für die Darstellung von Kontext eignen. Dabei ist [Winkler](#) auch zu dem Ergebnis gekommen, dass hier Ontologien geeignet sind.

Ontologien sind ein komplexes System, in das sich ein Anwender erst einarbeiten muss. Diese Einarbeitung ist aber nur notwendig, um die Aktivitätserkennung an neuen Bedingungen anzupassen. Nutzer der Schnittstellen zur Aktivitätserkennung haben kaum etwas mit der Ontologie zu tun. Außerdem schränken Ontologien den Anwender in seiner Freiheit ein. Begriffe außerhalb der Ontologie können nicht beachtet werden und bestehende Begriffe sind meistens eindeutig definiert. Dadurch müssen alle Ausnahmen oder Sonderfälle definiert werden. Das ist aufwändig und kann das gesamte System noch zusätzlich komplizierter machen. Dieses Problem kann man aber dahingehend reduzieren, in dem man genau analysiert, welche Ausnahmen auch wirklich notwendig sind und welche Sonderfälle weggelassen werden können.

Einen weiteren Vorteil über diese Arbeit hinaus bietet die Ontologie zusätzlich. Da sie von Computern lesbar ist und das Living Place in ihr vollständig definiert werden kann, kann man diese Ontologie auch als Grundlage zur Kommunikation im Living Place nutzen. Das Living Place besteht aus vielen verschiedenen Anwendungen die miteinander kommunizieren müssen. Wenn jede Anwendung für die Kommunikation ein eigenes Protokoll nutzt, kann es zu Missverständnissen und Fehlern kommen. Eine zentrale Grundlage für die Kommunikation kann helfen diese Probleme zu vermeiden.

⁷<http://www.w3.org/Submission/SWRL/> abgerufen 30. Dezember 2011

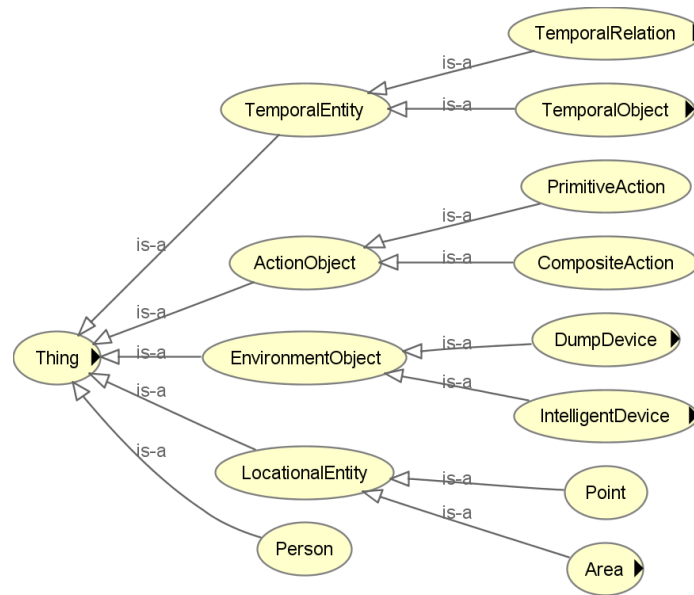


Abbildung 3.4.: Ontologie des 5W1H-Modells

3.2.2. Definition der Aktivitäten

In diesem Abschnitt werden die Aktivitäten selbst diskutiert. Im letzten Abschnitt wurde eine Ontologie als geeignete Datenstruktur zur Beschreibung des Kontext und der Aktivitäten erarbeitet. Hier sollen nun diskutiert werden, wie die Dimensionen am besten mit der Ontologie abgebildet werden können.

Ein Ansatz, die Dimensionen in der Ontologie zu modellieren, ist die Begriffe und Konzepte jeder Dimension in einem Zweig der Ontologie zu beschreiben. Es gibt dann in der Ontologie sechs⁸ Elternknoten unter denen die Begriffe gelegt werden, die in diese Dimension gehören. Die Ontologie für diesen Ansatz ist in Abbildung 3.4 zu sehen.

Dieser Ansatz spiegelt aber nicht die zentrale Bedeutung der Aktivitäten wider. Alle Dimensionen stehen gleichberechtigt auf einer Ebene. Um die Bedeutung der Aktivitäten zu modellieren benötigt die How-Dimension eine besondere Stellung in der Ontologie. Ein weiteres Problem besteht darin, dass die Konzepte der Ontologie verfälscht werden. Elternknoten bedeuten in einer Ontologie eine Generalisierung von Begriffen. Die Einteilung in die verschiedenen Dimensionen ist aber keine Generalisierung, sondern es sind besondere Eigenschaften der Begriffe in diesem Kontext. In einer Ontologie ist die Notation hierfür zum Beispiel: Bewohner → hat_Dimension → Who-Dimension.

⁸In dieser Arbeit sind es fünf Dimensionen, da die Why-Dimension fehlt.

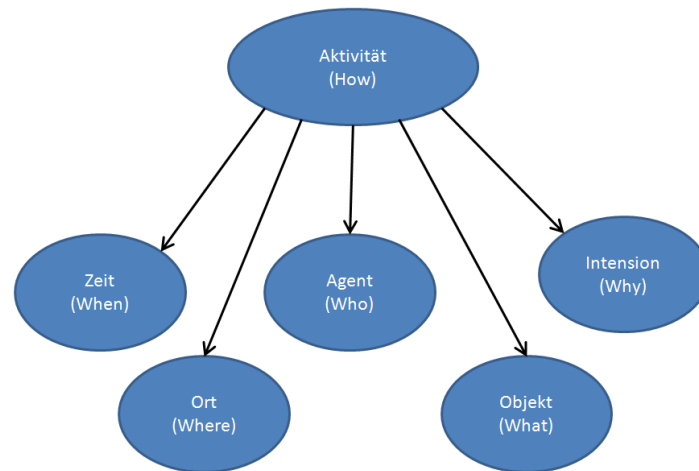


Abbildung 3.5.: Aktivität als zentrale Dimension

Die besondere Rolle der Aktivitäten besteht in dem Zusammenhang, dass die anderen Begriffe mit den Aktivitäten zusammenhängen. Die Dimensionen müssen also aus der Sicht der Aktivität betrachtet werden.

Das bedeutet, dass eine Aktivität:

- von einer Person ausgeführt wird
- an einem Ort ausgeführt wird
- mit einem Gegenstand ausgeführt wird
- zu einem bestimmten Zeitpunkt ausgeführt wird

Diese Zusammenhänge (siehe Abbildung: 3.5) sollen in die Ontologie übertragen werden. In Abschnitt 3.1.1.4 wurde für diese Arbeit die indirekte Erkennung von Aktivitäten für diese Arbeit festgelegt. Dieses Vorgehen spiegelt sich ebenso in dieser Darstellung der Dimensionen wider und kann auch in die Ontologie übertragen werden. Die Informationen in den verschiedenen Dimensionen beschreiben die Aktivität, wie die Sensoren die Aktivitäten indirekt erfassen. Somit sind die Quellen der oben genannten Dimensionen die Sensoren bzw. die Interpretierten Daten, die von den Sensoren geliefert werden. Anders ausgedrückt: Die Instanzen der Ontologie werden aus den Sensoren durch die Interpretationsebene erzeugt.

Eine Generalisierung der Begriffe aus den verschiedenen Dimensionen sind die Kategorien physisch und konzeptuell. Die Dimensionen Where und What repräsentieren die physische Welt des Living Place. Hierzu zählt auch die Who-Dimension, in der der Agent der Aktivität abgebildet wird. Der Agent oder Bewohner wird auch von den Sensoren erfasst. In dem konzeptuellen Zweig werden nicht die Elemente der intelligenten Wohnung definiert, sondern es werden die Aktivitäten modelliert, die aus der physischen Welt abgeleitet werden sollen.

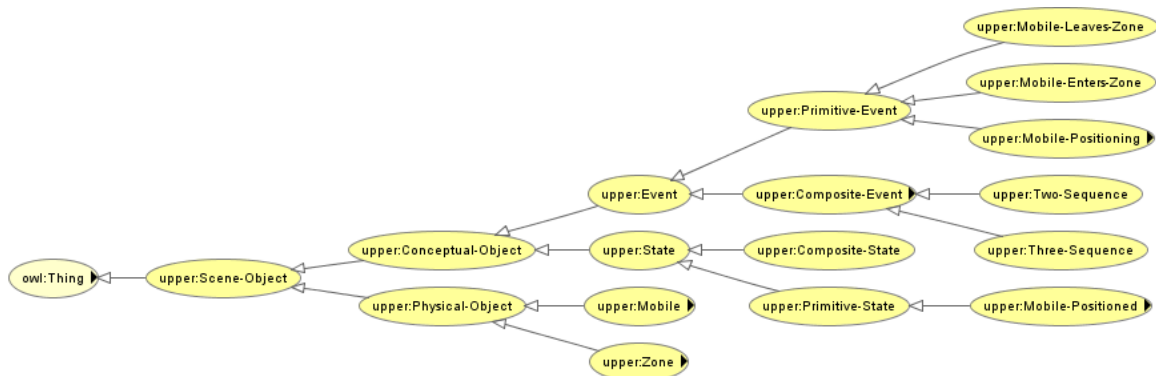


Abbildung 3.6.: Upper Ontologie (vgl. [Bohlken u. a. \(2011\)](#))

Zwischen den physikalischen und konzeptuellen Begriffen gibt es Zusammenhänge, die auch in der Ontologie abgebildet werden können.

Dieser Ansatz nutzt die Konzepte der Ontologie um die Begriffe in ihrer Bedeutung zusammenzufassen. Außerdem werden die besondere Stellung und die Eigenschaften der Aktivitäten besser als im ersten Ansatz dargestellt. Die Zusammenhänge zwischen den Aktivitäten und der physischen Welt lassen sich besser modellieren.

In [Bohlken u. a. \(2011\)](#) wurde eine Aktivitätserkennung auf einem Flugplatzvorfeld umgesetzt. Das Flugplatzvorfeld, die mobilen Objekte und die Aktivitäten wurden in einer Ontologie modelliert. Auf dem Flugplatzvorfeld gibt es im Gegensatz zum Living Place klare Abläufe und Strukturen. Diese Eigenschaft reduziert die erwarteten Aktivitäten deutlich. Das hauptsächliche Problem bei Scenior waren die fehlerbehafteten Daten, mit denen die Erkennung der Aktivitäten umgesetzt werden musste. [Bohlken u. a.](#) haben einer „Upper-Ontologie“ (siehe Abbildung: 3.6) entwickelt, die auch für Aktivitätserkennungen in anderen Umgebungen einsetzbar sein soll. Diese Upper-Ontologie soll auch in dieser Arbeit angewandt werden. Sie bildet die allgemeinen Strukturen gut ab und diese Strukturen entsprechen auch dem 5W1H-Modell, das in dieser Arbeit genutzt werden soll.

3.2.2.1. Physische Begriffe

Zu den physischen Begriffen zählen praktisch alle relevanten Objekte, die in der Wohnung zu finden sind. Hierzu zählen unter anderem:

- **Möbel** z. B.: Couch, Schrank, Bett, Tisch, Stuhl
- **Fenster** z. B.: Fenster im Küchenbereich
- **Bereiche** z. B.: vor dem Ofen, auf dem Bett, bei dem Esstisch

- **Räume** z. B.: Badezimmer, Küchenbereich
- **Küchengeräte** z. B.: Ofen, Waschbecken, Kaffeemaschine
- **Unterhaltungselektronik** z. B.: Fernseher, Telefon, Laptop

Es besteht bei den physischen Begriffen die Möglichkeit, dass es sehr viele werden und dadurch die Übersicht und die Handhabbarkeit verloren gehen. Um dies zu vermeiden, ist es sinnvoll die Begriffe frühzeitig in Gruppen einzuteilen und so Hierarchien zu erstellen.

Die Begriffe, die in der Ontologie modelliert werden sollen, sind genau die, welche in den Szenarien benötigt werden. Des Weiteren können auch weitere Begriffe in der Ontologie modelliert werden ohne dass es zu Problemen in diesem Projekt kommt. Diese Begriffe könnten in der Kommunikation zwischen den Anwendungen des Living Place genutzt werden.

3.2.2.2. Konzeptionelle Begriffe

Die Begriffe, die nötig sind, die „How-Dimension“ zu beschreiben, gehören in diese Kategorie. Hierzu zählen alle Aktivitäten in ihren verschiedenen Detailstufen. Beispielsweise ist ein Teil der „Morgenroutine“ die Aktivität „frühstücken“. Ein Teil des „frühstückens“ ist die Aktivität „Kaffee kochen“ usw. .

Um Aktivitäten zu beobachten gibt es zwei grundsätzliche Vorgehensweisen. Entweder werden die Aktivitäten durchgehend identifiziert oder es gibt einen definierten Anfang und ein definiertes Ende der Aktivität. Im letzteren Fall kann die Hypothese aufgestellt werden, dass die Aktivität von dem erkannten Anfang bis zum erkannten Ende vorliegt. In diesem Zusammenhang bilden Events in der Ontologie eine Sonderstellung. Sie markieren wie oben beschrieben eine Änderung der aktuellen Aktivitäten. Somit werden durch die Events ein definierter Anfang und ein definiertes Ende der Aktivitäten beschrieben.

Diese Änderungen haben in der Regel einen direkten Zusammenhang mit einem messbaren physischen Ereignis in der intelligenten Wohnung. Somit entstehen aus den Daten der Sensoren direkt Events mit denen Aktivitäten hypothetisiert werden können. Das heißt auch, dass es direkte Zusammenhänge zwischen den physischen Begriffen und den Events gibt, die in der Ontologie modelliert werden können.

3.2.2.3. Andere Begriffe

Die Zeit zählt nicht zu den physischen oder konzeptionellen Begriffen. Dennoch ist es notwendig, die Aktivitäten in eine temporäre Relation zu bringen und in der „When-Dimension“ die Aktivitäten einem Zeitraum zuzuordnen. Um die Abfolge der Aktivitäten zu beschreiben, wurde in Abschnitt [3.2.1.2](#) die temporale Logik von Allen diskutiert, die mit SWRL in der

Ontologie umgesetzt werden kann. Um die Aktivitäten einer Zeit zuzuordnen, werden in der Ontologie diese Zeitlichen Begriffe benötigt. Zu diesen Begriffen gehören unter anderem:

- **allgemein definierbare Zeiträume:** Diese Zeiträume sind in der Regel nicht von dem Bewohner abhängig Definiert. z. B.: morgens, mittags, abends
- **subjektive Zeiträume:** Diese Zeiträume werden durch den Bewohner und sein Leben bestimmt. z. B.: Freizeit, Arbeitszeit, Wochenende
- **relative Zeiträume:** Diese Zeiträume definieren sich im Bezug zu einem Ereignis. z. B.: nachDemAufstehen, vorDemMittagEssen

Weitere Begriffe existieren außerhalb des Living Place. Beispielsweise die Orte an denen der Bewohner arbeitet oder Termine an solchen Plätzen. Ein Termin könnte Einfluss auf eine Aktivität des Bewohners haben. Somit könnte der Zusammenhang mit diesem Termin in der When-Dimension beschrieben werden und muss somit in der Ontologie definiert werden.

Weitere Begriffe beziehen sich auf den Zustand des Bewohners. Das Event „Kleidung anziehen“ markiert nicht den Übergang von einer Aktivität zur nächsten, sondern der Zustand des Bewohners ändert sich von „ohne Kleidung“ in „mit Kleidung“.

Diese Begriffe können zu einem „Kontext“ zusammengefasst werden. Die anderen Begriffen der Ontologie können dann Verbindungen zu diesen Begriffen herstellen und so das gesamt Modell der Aktivitätserkennung vervollständigen.

3.2.3. Klassifikationsverfahren

3.2.3.1. Problembeschreibung

In dem 5W1H-Modell sind die Fragen nach dem „Who“, „Where“, „What“ und „When“ relativ einfach zu beantworten.

„**Who**“ In diesem Projekt wird von einem Ein-Personen-Haushalt ausgegangen. Die Erweiterung auf mehrere Personen ist in der Ontologie möglich, würde aber den Rahmen dieser Arbeit sprengen. Daher ist bei dem „Who“ hier immer der Bewohner der Wohnung gemeint.

„**Where**“ Da die Aktivität vom Bewohner der Wohnung ausgeführt wird, ist der Ort der Aktivität auch immer der Ort des Bewohners.

„**When**“ Es handelt sich bei dieser Aktivitätserkennung um ein nahe Echtzeitsystem. Daher ist der Zeitpunkt der Aktivität immer die aktuelle Zeit. Diese Zeit kann um mehrere Sekunden von der tatsächlichen Zeit abweichen, da die Definition der Aktivitäten voraussichtlich im Minuten- oder Stundenbereich liegt.

„**What**“ Die Gegenstände der intelligenten Wohnung sind zum größten Teil selbst in der Lage ihre Aktivität zu kommunizieren. Andere Gegenstände können in der Wohnung beobachtet werden.

Die Frage des „Why“ wird in dieser Arbeit nicht behandelt. Somit bleibt zu klären, wie die Fragen des „How“ beantwortet werden können. Eine Aktion kann selten direkt erkannt werden. Das Tippen einer Tastatur wäre ein Beispiel für eine direkte Möglichkeit. Aber schon das Arbeiten an einem PC wird von Pausen des Schreibens unterbrochen in denen man über den Text nachdenkt. Außerdem können sich Aktivitäten überlagern. Während der Arbeit ist es möglich auch etwas zu essen. Grundsätzlich kann man annehmen, dass eine Aktivität von ihrem Anfang durchgehend bis zum ihrem Ende besteht. Daraus ergeben sich, wie oben beschrieben, zwei Ansätze:

Aktivität durchgehend erkennen Alle Sensorwerte, müssen durchgehend zu der Aktivität passen. Passen die Sensorwerte nicht, ist die Aktion unterbrochen oder beendet. Der Datenstrom der Sensorwerte wird durchgehend analysiert.

Anfang und Ende der Aktivität identifizieren Es werden nur Anfang und Ende einer Aktivität in den Sensorwerten erkannt. Ein Vorteil dieses Ansatzes ist, wenn der Anfang erkannt wurde, kann sich die Klassifizierung auf das Ende der Aktivität konzentrieren.

Durch die Ontologie und die Aufteilung der Begriffe in Events und States sind die Voraussetzungen für das zweite Verfahren direkt gegeben. Daher muss bei der Erkennung der Aktivitäten nur noch der Zusammenhang zwischen den Events und States erarbeitet werden und Struktur innerhalb der Events und States definiert werden.

Um eine Hypothese über den Anfang einer Aktivität aufzustellen, müssen zumindest die Voraussetzungen der anderen Dimensionen gegeben sein.

„**Who**“ Der Agent der Aktivität muss anwesend sein.

„**What**“ Der Gegenstand der Aktivität muss aktiv sein. Aktiv bedeutet hier, dass der Gegenstand zur Nutzung bereit ist (z. B. TV-Gerät) oder aktiv genutzt wird (z. B. Bett).

„**When**“ Der Zeitraum für die Aktivität muss passen. Beispielsweise muss es morgens sein, damit der essende Bewohner frühstückt⁹.

„**Where**“ Der Ort für die Aktivität muss stimmen. Hier bedeutet das, dass der Agent oder Bewohner sich an dem Ort befinden muss, an dem die Aktivität stattfinden kann. Beispielsweise muss der Bewohner im Bad sein, um sich zu duschen.

Zwei Probleme für die bei der Umsetzung der Aktivitätserkennung eine Lösung gefunden werden muss sind:

⁹Wenn das Frühstück als Mahlzeit in den ersten Stunden des Tages definiert ist

Unterbrechungen Eine Aktion könnte abgebrochen werden und anschließend wieder aufgenommen. In diesem Fall könnte der Gegenstand durchgehend aktiv sein (z. B. PC, Herd oder TV).

Gleichzeitigkeit Es sind verschiedene Aktivitäten denkbar, die gleichzeitig ablaufen können. Beispielsweise das Arbeiten am PC und das Fernsehen oder morgens während des Frühstücks sich weiter anzuziehen. In diesen Fällen besteht die Frage, wie man die eine Aktivität von der zweiten abgrenzt oder ob die Aktivitätserkennung mehrere Aktivitäten zulässt.

3.2.3.2. Analyse der Algorithmen

Aus der Problembeschreibung geht hervor, dass aus eingehenden (Sensor-)Daten weiterführende Informationen erkannt werden sollen. Es ist bekannt, welche Informationen bzw. Klassen erkannt werden sollen. Gesucht wird eine Funktion oder ein Algorithmus, der die aktuellen eingehenden Daten den Klassen zuordnet. Dieses sind die klassischen Voraussetzungen und Aufgabenbeschreibungen für Klassifikationsverfahren der Informatik. In [Phyu \(2009\)](#) werden verschiedene Klassifikationsverfahren gegenübergestellt.

Für diese Arbeit soll sich auf Klassifikationsverfahren ohne statistische Elemente beschränkt werden. Statistische Verfahren benötigen Heuristiken und einen hinreichend großen Datensatz, damit die Heuristiken entwickelt und beurteilt werden können. Für diese Arbeit liegt ein entsprechender Datensatz nicht vor. Daher werden statistische Klassifikationsverfahren für diese Arbeit nicht weiter untersucht. Aus [Phyu \(2009\)](#) sind die folgenden Verfahren interessant:

- Entscheidungsbaum
- K-Nächste Nachbarn
- Regelbasierte Entscheidungen

Die Klassifizierung in einem Entscheidungsbaum nimmt in der Regel einen nachvollziehbaren deterministischen Weg. Diese Eigenschaft erleichtert es die Klassifizierung nachzuvollziehen und Fehler bei der Klassifizierung zu finden. In einem Entscheidungsbaum werden verschiedene Knoten durchlaufen. In einem Entscheidungsbaum wird angenommen, dass in den Knoten für die eingehenden Daten eine Entscheidung möglich ist, die eine Zuordnung der Klassen ermöglicht. Das bedeutet, dass die verschiedenen Werte des Merkmal-Raums sich für die Aktivitäten unterscheiden müssen.

Die K-nächste-Nachbarn-Klassifikation benötigt eine Abstandsfunktion mit der die eingehenden Daten beurteilt werden können. Diese Funktion wird durch die Datenstruktur der eingehenden Daten beeinflusst. Das bedeutet, dass je nach Datenstruktur die Abstandsfunktionen unterschiedliche Ergebnisse liefern können. Vorteilhaft ist, dass der Algorithmus des Systems insgesamt einfach ist.

Ein regelbasiertes System beschreibt direkt die Klassen, die erkannt werden sollen. Dieses hat den Vorteil, dass Expertenwissen direkt für jede Klasse definiert werden kann. Bei einem neu entwickelten System ist es dann möglich die Klassifikation direkt zu modellieren, auch wenn noch keine umfangreichen Daten zum Erlernen der Klassifikation vorliegen. Ein Nachteil bei regelbasierten Systemen sind die starren Grenzen der Regeln. Die Regeln können nicht alle Fälle beachten und es muss für jede Ausnahme eine eigene Regel erstellt werden.

3.2.3.3. Zusammenfassung

Durch die oben vorgestellten Verfahren können die Aktivitäten erkannt werden. Sie bieten aber kein Verfahren an, das gleichzeitig die Sicherheit angibt, mit der diese Erkenntnisse gewonnen wurden.

Da es sich bei der Aktivitätserkennung im Living Place um ein Projekt im Entwicklungsstadium handelt, gibt es noch nicht ausreichend Testdaten oder Möglichkeiten diese Testdaten zu erzeugen. Daher wird ein Klassifikationssystem benötigt, das in einem ersten Ansatz nur aus Expertenwissen besteht. Hier bieten sich ein regelbasiertes System an, wie es in dem Projekt von [Bohlken u. a. \(2011\)](#) schon existiert.

Durch die Regeln ist es möglich Hypothesen über Aktivitäten in der Wohnung aufzustellen. In einem zweiten Schritt ist es anschließend notwendig, eine Wahrscheinlichkeit anzugeben, mit der die Hypothesen wahr sind. Bei der Berechnung der Wahrscheinlichkeiten ist es denkbar noch weitere Sensoren einzubeziehen, die keine notwendigen Voraussetzungen für eine Hypothese sind. Beispielsweise könnte für die Aktivität in der Wohnung der Helligkeitswert beachtet werden.

3.3. Daten im Living Place

In diesem Abschnitt werden die Daten analysiert, die im Living Place zur Verfügung stehen. Durch die Blackboard-Architektur stehen der Aktivitätserkennung neben den Sensordaten auch alle Daten zur Verfügung, die aus der Interpretation von Sensordaten resultieren. Beispielsweise werden in [Voskuhl \(2012\)](#) die Positionsdaten mit zusätzlichen Informationen

angereichert. Im Folgenden wird erst eine Übersicht über die Sensoren im Living Place gegeben. Außerdem wird beschrieben welche interpretierten Daten zur Verfügung stehen und auf welche externen Daten die Aktivitätserkennung sich stützen kann.

3.3.1. Situationsbedingte Anwendungen im Living Place

Das Ziel fast aller intelligenten Wohnungen ist es, die Situation in der Wohnung zu bestimmen und die Wohnung entsprechend der Situation anzupassen (vgl. [Yamazaki \(2007\)](#)). Wie die Situation definiert wird ist den meisten „Smart Homes“ unterschiedlich. Auch im Living Place Hamburg soll sich die Wohnung intelligent Verhalten und den Bewohner in seinen Aktivitäten zu unterstützen. Hierbei geht es nicht darum neue Aktivitäten zu ermöglichen, sondern das alltägliche Leben (vgl. [Katz und Akpom \(1976\)](#)) zu unterstützen.

Eine Zielgruppe einer intelligenten Wohnung sind Menschen, die mit Einschränkungen leben. Beispielsweise brauchen ältere Menschen schon bei den einfachsten Aktivitäten Unterstützung. Wenn ältere Menschen aus dem Bett aufstehen, könnte es sinnvoll sein, wenn:

- Schon einmal das Licht im Zimmer angeht und sie nicht erst nach einem Schalter suchen müssen
- Sie mit einer Person anderer Tür reden können, ohne zur Tür zu gehen.
- Hilfe gerufen wird, wenn sie es selbst nicht mehr können.

Eine andere Zielgruppe wären Leute, die ihr Leben angenehmer gestalten möchten. Wenn die Wohnung Tätigkeiten erleichtert oder die Informationen auf die Situation anpassen die gerade in der Wohnung besteht. Beispielsweise möchte eine berufstätige Person:

- Morgens entsprechend dem Terminkalender geweckt werden.
- Die wichtigen Informationen für den kommenden Tag zur Verfügung haben.
- Während des Essens nicht von einem Telefon gestört werden aber sonst immer erreichbar sein.

Dieses sind nur zwei Beispiele, wie Kontext genutzt werden kann um einen Bewohner zu unterstützen. Viele weitere sind denkbar und können sich auf jede Aktivität in der Wohnung erstrecken.

3.3.2. Übersicht

Daten können aus primitiven Sensoren erzeugt werden. Diese Sensoren messen beispielsweise die Intensität des Lichtes und leiten die Werte über eine Schnittstelle weiter an das Blackboard. Des Weiteren gibt es komplexe Systeme, wie beispielsweise das „intelligente Bett“ (vgl. [Hardenack \(2011\)](#)), welches in der Lage ist komplexe Daten zu erzeugen. Komplexe Daten sind hier unter anderem interpretierte Schlafphasen des Bewohners. Außerdem werden auch Daten über Schnittstellen aus fremden Systemen bezogen und für das Living Place aufbereitet. Hier kann als Beispiel ein Programm genannt werden, dass aus dem Internet das aktuelle Wetter holt und es auf dem Blackboard zur Verfügung stellt (vgl. [Voskuhl \(2012\)](#)).

Die Sensoren lassen sich auf verschiedenen Ebenen betrachten:

Operationsbereich: Es sind Sensoren gemeint, die über eine bestimmte Zone im Living Place Informationen liefern. Bei dieser Betrachtung soll beispielsweise ein bestimmter Bereich in der Wohnung interpretiert werden. Relevant sind alle Informationen, die geliefert werden können, egal welche Art von Sensor dafür eingesetzt wird und um welche Art von Information es sich handelt.

Art des Sensors: Die Art des Sensors, der die Daten liefert, steht im Fokus. Die Sensoren werden nach ihrem Typ unterschieden und nicht nach der Art der Funktionen oder dem Ort, in dem die Informationen erfasst werden.

Art der Information: Hier steht die Art der Informationen im Vordergrund. Beispielsweise könnte die Position des Bewohners benötigt werden. Dabei ist es egal welche Art von Sensor diese Information erzeugt.

Für die Kontext-Erkennung sind die Anzahl und die Art der Sensoren nicht wichtig. Es kommt auf die Art der Informationen an, die die Sensoren liefern beziehungsweise den Service, den die Sensoren zur Verfügung stellen. Beispielsweise ist für verschiedene Kontexte die Position des Bewohners interessant. Diese Position könnte über eine Kamera erfasst werden, durch induktive Sensoren oder durch das UbiScience-System. Die Position des Sensors ist nicht direkt wichtig aber deren Aktionsbereich, wenn dieser sich nicht über die gesamte Wohnung erstreckt. Es muss die geforderte Funktion für jeden relevanten Bereich der Wohnung zur Verfügung stehen. Eventuell ist es notwendig, verschiedene Sensoren für die gleiche Funktion zu nutzen. Die Art des Sensors spielt in dieser Arbeit eine untergeordnete Rolle. Bei der Entwicklung der Features muss darauf geachtet werden, dass die Sensoren auch existieren und die Daten liefern. Es könnte später eine bestimmte Art von Information über ein zentrales Informationssystem angefordert werden wie es in [Johannsen \(2011\)](#) entwickelt werden soll. In diesem Informationssystem soll eine Übersicht über die verschiedenen Funktionen der Sensoren im Living Place zur Verfügung gestellt werden.

3.3.3. Sensor-Daten

Ein Merkmal einer intelligenten Wohnung sind viele verschiedene Sensoren, die in der Wohnung verteilt sind. Mit diesen Sensoren sollen die Merkmale in der Wohnung registriert werden, um diese später zu nutzen. Im Living Place gibt es verschiedene Sensoren mit sehr unterschiedlichen Aufgaben und Möglichkeiten der Datenerfassung.

Sensoren haben den Vorteil, dass sie relativ billig und einfach zu beschaffen sind. Es gibt sie in einer großen und vielfältigen Anzahl. Jedes Gerät in der Wohnung kann mit einem Sensor überwacht werden. Dabei gibt es auch eine Auswahl an vielen verschiedenen Sensoren für eine Aufgabe. Beispielsweise könnte eine Kaffeemaschine durch einen zusätzlichen Schalter in der Maschine überwacht werden oder der Stromverbrauch der Kaffeemaschine wird überwacht. Sensoren, die Daten über den Bewohner direkt aufnehmen sollen, müssen häufig vom Bewohner getragen werden. Hier besteht ein Problem der Toleranz des Bewohners

Einfache Sensoren sind beispielsweise Schalter, die nur zwei Zustände haben und anzeigen, ob eine Tür offen oder geschlossen ist. Kameras sind hingegen sehr viel komplexere Sensoren. Sie liefern einen Datenstrom, der für eine weitere Nutzung erst noch analysiert werden muss. Die Sensoren im Living Place werden wahrscheinlich ständig erweitert und andere Sensoren werden wieder demontiert oder ersetzt.

Die Daten der verschiedenen Sensoren sind heterogen und für eine direkte Nutzung meistens nicht geeignet. Sie liefern meistens Zahlen in einem individuellen Wertebereich ohne Einheit. Beispielsweise werden aus dem Ortungssystem „UbiSense“ zwei Vektor-Werte geliefert, aus denen die Position im Raum berechnet werden kann (vgl. [Tennstedt \(2009\)](#)). In diesem Schritt wird auch mit der Vorverarbeitung von den Sensoren abstrahiert. Dadurch kann das System leichter an neue Situationen angepasst werden.

Ein weiteres Beispiel ist ein Schalter, der an der Eingangstür angebracht ist. Dieser Schalter zeigt an, ob die Tür offen oder geschlossen ist. Diese Informationen liefert aber kein Schalter. Die Daten eines Schalters sind in der Regel nur „True“ und „False“. Die Informationen werden erst brauchbar, wenn sie mit anderem Wissen kombiniert werden. Beispielsweise, dass der Schalter die Eingangstür überwacht und dass „True“ eine „Offene Eingangstür“ bedeutet und „False“ eine „Geschlossene Eingangstür“.

3.3.4. Interpretierte Daten

Es sind nicht nur die direkten Sensordaten für die Klassifikation von Interesse, sondern auch Daten verschiedener Sensoren in der Kombination oder die Daten eines Sensors in der Ableitung über die Zeit. Beispielsweise könnte das UbiSense (vgl. [Tennstedt \(2009\)](#)) dazu genutzt werden, die Geschwindigkeit der Person zu bestimmen oder die Position eines Stuhls

kombiniert mit der Position einer Person, könnte zu einem „Person ist im Nutzungsbereich des Stuhls“-Feature werden.

Eine Möglichkeit um Daten in Smart Homes zu erfassen sind Kameras (vgl. [Gómez-Romero u. a.](#)). Kameras können auf verschiedene Bereiche der Wohnung gerichtet werden und sind so in der Lage Informationen zu den Aktivitäten in diesem Bereich zu liefern. Unter anderem können die Umrisse von Personen erkannt werden. Dadurch ist es möglich Bewegungen zu registrieren.

Die Nachteile bei kamerabasierten Systemen sind die schlechte Skalierbarkeit. Der Aufwand der Bildanalyse ist noch relativ hoch, wenn man gute und verlässliche Ergebnisse erzielen will. Systeme die für eine bestimmte Umgebung entwickelt wurden, lassen sich kaum auf eine andere Umgebung übertragen. Jede Wohnung hat ihren eigenen Einrichtungsstil und die Bewohner unterscheiden sich auch in Form, Kleidung und Art ihrer Bewegungen.

Bei weniger komplexen Interpretationen der Kamerabilder sind in relativ kurzer Zeit gute Ergebnisse möglich. Beispielsweise könnte für die Erkennung von Objekten die Arbeit von [Najem \(2011\)](#) genutzt werden um Besteck oder Teller auf dem Esstisch zu entdecken. In der Arbeit von [Teske \(2011\)](#) werden Bewegungen aus den Kamerabildern extrahiert. Diese Informationen können genutzt werden, um in Kombination mit der Position eines Gegenstandes dessen Aktivität zu ermitteln. Durch diese Möglichkeiten der Bildanalyse kann die Hypothese über die Aktivität eines Bewohners unterstützt werden.

3.3.5. Externe Daten

Für die Interpretation der Situation in der Wohnung sind nicht nur Daten über die Wohnung selbst von Interesse. Aktivitäten haben in der Regel einen Grund (Why-Dimension), der auch außerhalb des Messbereichs der Sensoren der Wohnung liegen kann.

Ein einfaches Beispiel hierfür sind die Termine in einem Kalender. Der erste Termin des Tages könnte die Hypothese der morgendlichen Aktivitäten unterstützen. Die Termine des Kalenders sind somit ein Teil der Rahmenbedingungen, in denen die Aktivitäten erkannt werden sollen.

Des Weiteren benötigt die Aktivitätserkennung Daten über die Zeit (When-Dimension). Diese Informationen müssen auch von einer Anwendung auf dem Blackboard zur Verfügung gestellt werden. Hierbei ist nicht nur die aktuelle Uhrzeit gemeint, sondern auch interpretierte Zeiträume. Beispielsweise könnte für das Erkennen der Aktivitäten die Mittagszeit von Interesse sein. Dann muss aus der aktuellen Uhrzeit der Mittags-Zeitraum interpretiert werden, damit dieser als Merkmal zur Verfügung steht.

Folgende weitere externe Informationen sind auch denkbar um die Hypothesen zu unterstützen:

- Wetterbedingungen
- Sonnenaufgang und Sonnenuntergang
- Verkehrsinformationen
- Besondere Tage im Jahr (z. B. Geburtstage oder Neu-Jahr)

Die externen Daten eignen sich hauptsächlich zur Unterstützung von Hypothesen. Es sind meistens keine notwendigen Voraussetzungen um eine Hypothese über eine Aktivität aufzustellen. Die meisten externen Informationen ordnen sich in die „Why-Dimension“ ein. Sie geben einen weiteren Hinweis darauf, dass eine Aktivität wahrscheinlich ist. Da diese Dimension den Rahmen dieser Arbeit übersteigt, werden auch die externen Informationen in diesem Ansatz nicht weiter verfolgt. Weitere Informationen zu externen Informationen finden sich in [Voskuhl \(2012\)](#).

4. Entwicklung einer Ontologie für Smart Homes

Wie in Abschnitt 3.2.1.5 diskutiert wurde, soll zur Darstellung der Aktivitäten und anderer Begriffe eine Ontologie eingesetzt werden. In diesem Abschnitt wird beschrieben, mit welchen Werkzeugen die Ontologie erstellt, verändert und gewartet wird. Des Weiteren teilt sich die Ontologie in eine allgemeine „Upper-Ontologie“ und eine spezielle „Domain-Ontologie“ auf. Es wird der Aufbau und Inhalt dieser beiden Ontologie-Teile beschrieben. Ein weiterer Teil der Ontologie sind die „SWRL-Rules“. Mit diesen Regeln werden die Zusammenhänge zwischen den Aktivitäts-Begriffen beschrieben. Abschließen wird die Ontologie mit Elementen befüllt, die dem Szenarien aus dem Living Place entsprechen.

4.1. Rolle der Ontologie im System

In dieser Arbeit sollen die Voraussetzungen geschaffen werden, dass Anwendungen ihr Verhalten an die Situation im Living Place anpassen können. Die Anwendungen sollen sich also „intelligent“ verhalten. Nach Brooks (1991) muss intelligentes Verhalten nicht zwingend auf einem Modell beruhen. Es ist genauso möglich, in den Anwendungen die Reaktion auf die Sensorwerte direkt zu implementieren.

Wenn die Anwendungen direkt auf die Sensoren reagieren sollen, sind in Funktionen starr an die Sensoren gekoppelt. Durch diese Kopplung, kann auf Veränderungen während des Betriebes der Anwendungen kaum reagiert werden. Beispielsweise führen dann Veränderungen an den Sensoren zu komplexen Änderungen in der Software. Daher ist es sinnvoll die Interpretation der Sensoren von der Reaktion der Anwendungen zu entkoppeln. Die Idee dieser Entkopplung entspricht der Idee aus Newell (1982) vom allgemeinen Agenten. Es soll bei diesem Agenten der Prozess der Interpretation der Umgebung unabhängig von dessen Einsatzgebiet aufgebaut werden. Diese Unabhängigkeit kann erreicht werden, wenn aus dem Interpretationsprozess das Wissen heraus gelöst wird und als Wissensbasis zur Verfügung steht.

Die Abbildung 4.1 zeigt den Ansatz einer Wissensbasis, mit der die oben genannten Ideen umgesetzt werden können. Die Wissensbasis enthält beispielsweise die Daten über die

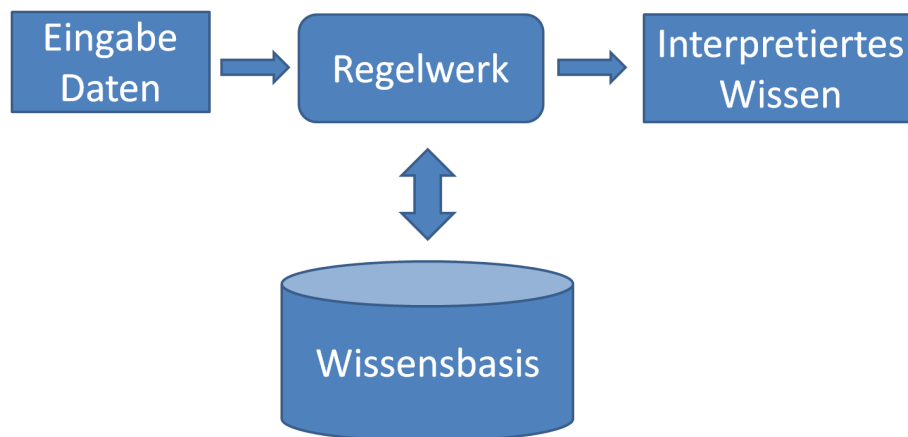


Abbildung 4.1.: Wissensbasis

Wohnung und die Zusammenhänge zwischen den einzelnen Elementen der Wohnung. Des Weiteren enthält die Wissensbasis die Zuordnungsfunktion für die Klassifikation der Aktivitäten. Mit dieser Funktion können die Eingangs-Daten den Klassen zugeordnet werden. Außerdem enthält die Wissensbasis Informationen darüber, welche Klassen für die Zuordnung gibt. Durch Änderungen in der Wissensbasis ist es dann auch möglich, das Verfahren der Aktivitätserkennung in einer anderen Domäne als dem Smart Home Kontext einzusetzen (vgl. [Bohlken u. a. \(2011\)](#)).

Im Detail müssen die folgenden Informationen in der Wissensbasis hinterlegt werden:

- Modell der Eingangsdaten
- Modell der zu erkennenden Klassen
- Funktion für die Zuordnung

Das in Abschnitt [3.1.1](#) diskutierte Modell für die Aktivitätserkennung bringt die oben genannten Eigenschaften mit, die für die Wissensbasis benötigt werden.

Modell der Eingangsdaten Das Modell der Eingangsdaten ist der Bewohner, besondere Zeiten, das Living Place, die Einrichtungsgegenstände und die Orte des Living Place. Die Eingangsdaten stellen somit die Dimensionen „Who“, „When“, „What“ und „Where“ da.

Modell der zu erkennenden Klassen Hier wird die „How“-Dimension modelliert. Dieses Modell besteht aus den Aktivitäten, die erkannt werden sollen, und den Ereignissen, mit denen eine Hypothese über die Aktivitäten aufgestellt werden kann.

Funktion für die Zuordnung Die Funktionen beinhalten die Zusammenhänge zwischen dem eingehenden Modell und den zu erkennenden Klassen. Hier wird beschrieben,

mit welchen Ereignissen eine Aktivität beginnt, und durch welche Ereignisse diese wieder beendet wird.

4.1.1. Modell der Eingangsdaten

Das Modell der eingehenden Daten besteht aus dem Living Place und den Zusammenhängen zwischen seinen Elementen. Daraus folgt, dass sich dieses Modell in zwei Teile aufteilt.

- Bestandteile des Living Place
- Zusammenhänge, die für eine Anwendung relevant sind¹

Die Primitiven des Living Place stehen von vorn herein fest. Sie bilden den statischen Teil des Modells. Zwischen den Bestandteilen des Living Place können verschiedene Beziehungen auftreten. Beispielsweise könnte der Bewohner im Bett sein oder im Badezimmer. Diese Möglichkeiten müssen in der Ontologie modelliert werden. Die Primitiven in der Ontologie, die diese Zusammenhänge repräsentieren, werden erst zur Laufzeit erzeugt oder wieder gelöscht.

4.1.2. Modell der Klassen

Das Modell der zu erkennenden Klassen besteht in dieser Arbeit aus den Aktivitäten des Bewohners und den Ereignissen, die eine Hypothese über die Aktivitäten zulassen. Dieses Modell teilt sich somit auf in:

- Ereignisse für den Anfang oder das Ende einer Aktivität
- Die Aktivitäten
- Zusammenhänge auf welche Teile des Eingangs-Modells sich die Ereignisse beziehen²

Die Instanzen der Ereignisse und der Aktivitäten werden zur Laufzeit erstellt. Sie ergeben sich aus der Zuordnungsfunktion, mit der aus den eingehenden Daten die Klassen des Ausgangs-Modells erkannt werden.

¹In dieser Arbeit beziehen sich die Zusammenhänge auf das Aufstellen einer Hypothese über Aktivitäten des Bewohners

²Diese Zusammenhänge entsprechen nicht der Zuordnungsfunktion, werden aber von der Funktion benötigt.

4.1.3. Zuordnungsfunktion

Die Zuordnungsfunktion beschreibt, wann die Veränderungen des Eingangsmodells zu den Ereignissen führen, die in dem Klassenmodell erstellt wurden. Jedes modellierte Ereignis ergibt sich aus Veränderungen im Modell des Living Place. Die Ereignisse sind hier nicht die Primitiven des Eingangs-Modells, sondern es sind die Veränderungen der Zusammenhänge. Wenn beispielsweise der Bewohner aus dem Bett aufsteht, dann hat das zur Folge, dass der Bewohner sich nicht mehr in einem Bereich befindet, in dem er das Bett nutzen kann. Diese Veränderung des Zusammenhangs zwischen dem Bewohner und dem Nutzungsbereich des Bettes ist eine Veränderung, die zu einem Ereignis wird.

Aus den Ereignissen folgen wiederum die Aktivitäten. Auch dieser Zusammenhang wird mit den Zuordnungsfunktionen modelliert. Bei dem oben genannten Beispiel folgt aus dem Ereignis, dass der Bewohner nicht mehr in der Nähe des Bettes ist, dass er nicht mehr schläft, also dass die Aktivität „Bewohner schläft“ beendet ist.

In der Zuordnungsfunktion wird für dieses Beispiel modelliert, dass das Ereignis „Bewohner verlässt Nutzungsbereich des Bettes“ (mit dem „Bewohner“ als Agenten und der Zone „Bett“) der Bewohner die Aktivität schlafen beendet hat.

Zusammengefasst hat dieses Beispiel die folgenden Elemente:

Eingangs-Modell :

- Bewohner
- Bett
- Nutzungsbereich des Bettes

Zusammenhänge im Eingangs-Modell :

- Bewohner - **hat_Zone** - Nutzungsbereich_des_Bettes
- Bett - **hat_Zone** - Nutzungsbereich_des_Bettes

Ereignisse : ⇒ Bewohner_verlässt_Nutzungsbereich_des_Bettes

- Bewohner_verlässt_Nutzungsbereich_des_Bettes
- **hat_Zone** - Nutzungsbereich_des_Bettes
- Bewohner_verlässt_Nutzungsbereich_des_Bettes
- **hat_Agent** - Bewohner

Klasse : ⇒ Bewohner_schläft

- Bewohner_schläft - **hat_Zone** - Nutzungsbereich_des_Bettes

- Bewohner_schläft - **hat_Agent** - Bewohner
- Bewohner_schläft - **hat_Objekt** - Bett

Die an dem Beispiel gezeigte Möglichkeit die Zusammenhänge zu modellieren macht die Vorteile der Wissensbasis gegenüber einer engen Kopplung deutlich. Das System, um die Hypothesen über die Aktivitäten aufzustellen, muss nicht verändert werden, wenn sich das Living Place verändert oder eine neue Aktivität erkannt werden soll. Es reicht aus die Zuordnungsfunktionen an die neue Situation im Living Place anzupassen. Oder eine neue Klasse mit ihren Events und Funktionen können neu definiert werden.

4.2. Auswahl der Werkzeuge für die Arbeit mit Ontologien

Wie in Abschnitt 2.3 beschrieben wurde, sind Ontologien komplexe Strukturen, die Begriff und die Zusammenhänge zwischen diesen Begriffen beschreiben. Daher ist es kaum möglich, Ontologien direkt in ihrem „OWL-Format“ zu definieren, zu warten oder an neue Bedingungen anzupassen. Aus diesem Grund ist es notwendig, die Ontologie mit der Hilfe von Werkzeugen zu bearbeiten.

4.2.1. Bearbeitungszyklus für Ontologien

Für die Arbeit mit der Ontologie werden Werkzeuge für die folgenden Schritte benötigt:

Klassen erstellen \Rightarrow *Zusammenhänge modellieren* \Rightarrow *Primitive erzeugen*

Im späteren Verlauf ist es notwendig, die Ontologie den laufenden Veränderungen des Living Place anzupassen.

Die Begriffe in einer Ontologie werden Grundsätzlich über ihre Klassenstruktur beschrieben. Zusätzlich zu der Klassenstruktur werden die Verhältnisse der Klassen untereinander beschrieben. Bei der Definition dieser Klassen und ihrer Zusammenhänge ist es hilfreich, wenn diese durch ein Werkzeug dargestellt werden können. Des Weiteren ist eine graphische Darstellung der Klassenhierarchie bei der Arbeit mit Ontologien hilfreich.

Bei der Erstellung von Klassen und Primitiven in der Ontologie, ist es hilfreich, die Begriffe neben dem Ontologie-Präfix mit zusätzlichen Präfixen oder Suffixen zu versehen. Beispielsweise könnten alle Bereiche im Living Place mit „Area“ enden oder Aktivitäten des Bewohners mit „Resident“ beginnen. Hier könnten automatische Werkzeuge helfen, die verschiedenen Begriffen automatisch Prä- und Suffixe zuzuordnen.

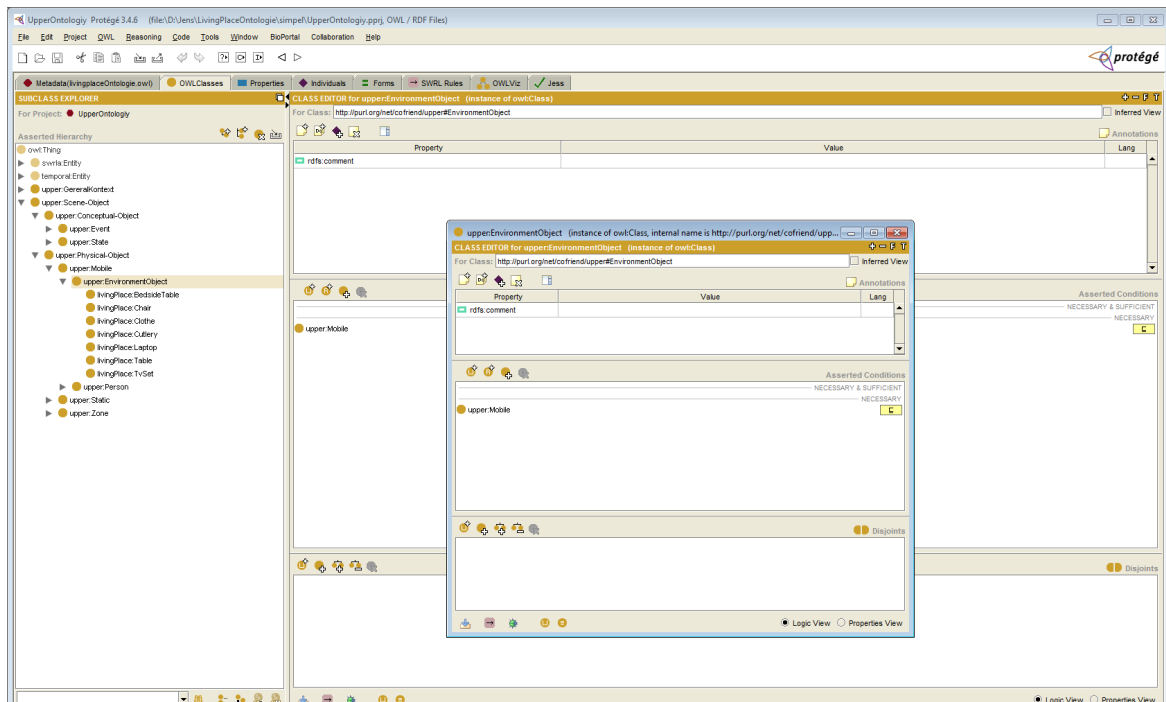


Abbildung 4.2.: Protege 3.4.7

Die „SWRL-Rules“ können unübersichtlich werden. Bei der Erstellung der Regeln könnte ein Überprüfung der Syntax der SWRL-Rules helfen Fehler zu vermeiden. Außerdem müssen die Klassen in den SWRL-Rules den Klassen in der Ontologie entsprechen. Auch hier sind Werkzeuge hilfreich, die diese Beziehung automatisch herstellen.

4.2.2. Auswahl von Werkzeugen

Es gibt verschiedene Tools um Ontologien zu bearbeiten. Beispiele für solche Werkzeuge sind unter anderem:

- [Ontoprise](http://www.ontoprise.de/)³
- [Knoowl](http://www.knoowl.com/)⁴
- [Protege](http://protege.stanford.edu/)⁵

³<http://www.ontoprise.de/> zuletzt Abgerufen: 21.11.2011

⁴<http://www.knoowl.com/> zuletzt Abgerufen: 21.11.2011

⁵<http://protege.stanford.edu/> zuletzt Abgerufen: 21.11.2011

Bei einer genaueren Untersuchung der verschiedenen Werkzeuge hat sich gezeigt, dass Protege genau die Möglichkeiten bietet, die für dieses Projekt gefordert sind. In Abbildung 4.2 ist ein Arbeitsfenster von Protege zu sehen.

Klassen erstellen In Protege lassen sich die Klassen der Ontologien hierarchisch darstellen und es gibt verschiedene Werkzeuge um Klassen automatisch zu erstellen.

Zusammenhänge modellieren Die komplexen Zusammenhänge in der Ontologie können in Protege dargestellt werden und über verschiedene Zusatzprogramme lassen sich die Funktionen von Protege erweitern. Ein weiterer Vorteil von Protege gegenüber anderen Werkzeugen ist die Unterstützung von SWRL Regeln. Diese Regeln können Werkzeug-Unterstützt modelliert werden.

Primitive erzeugen Die Primitiven lassen sich auch durch Werkzeuge automatisch darstellen. Es gibt für die Modellierung von Primitiven zusätzliche Tools und Eingabemasken.

SWRL-Regeln Für das Erstellen der Regeln steht ein einfacher Editor zur Verfügung. Der Editor hat eine einfache Syntax-Prüfung und stellt die Verbindungen zur Ontologie mit einem Werkzeug her.

Aufgrund der oben genannten Vorzüge wurde Protege in dieser Arbeit eingesetzt, um die Ontologie zu erstellen, und im weiteren Verlauf der Arbeit die Ontologie zu analysieren und anzupassen.

4.3. Erweiterung der Upper-Ontologie

Als „Upper-Ontologie“ soll die Ontologie aus dem Projekt von [Bohlken u. a. \(2011\)](#) eingesetzt werden. In der darunter liegenden „Domain-Ontologie“ werden anschließend die physischen Gegebenheiten des Living Place und die Aktivitäten des Bewohners modelliert.

Die Upper-Ontologie aus [Bohlken u. a. \(2011\)](#) ist aus der Domain eines Flugplatzvorfeldes hervorgegangen. Daher ist es nötig die generellen Konzepte der Upper-Ontologie zu erweitern. Diese Erweiterungen wurden in Zusammenarbeit mit der Universität Hamburg durchgeführt.

Eine Analyse der „Upper-Ontologie“ hat ergeben, dass die folgenden Ober-Begriffe fehlen:

- Die Kategorie der statischen Objekte
- Temporale Begriffe
- Allgemeine Zustände
- Unterschiedliche Zonen der Objekte

In einer Wohnung sind Aktivitäten mit Hilfe von statischen Objekten denkbar. Beispielsweise wird in einer „Dusche“ die Aktivität „Duschen“ ausgeführt oder mithilfe eines (Objektes) „Ofen“ wird „gekocht“.

Es gibt in der Wohnung einige Objekte, bei denen die Zuordnung in die statische oder mobile Kategorie unklar ist. Das Bett ist unter anderem ein solches Objekt. Es wäre theoretisch möglich das Bett zu verrücken. Es bleibt aber in der Regel immer an der gleichen Stelle in der Wohnung. In dieser Arbeit wurde sich dafür entschieden, dass alle Objekte statisch sind, von denen nicht erwartet wird, dass sie bewegt werden.

Mit den temporalen Begriffen werden im Living Place Aktivitäten genauer beschrieben und auch unterschieden. In *Scenior* wurden die Aktivitäten nur in eine temporale Relation gebracht. Der Zeitpunkt, an dem die Aktivitäten stattfanden, spielte keine Rolle. Dieses ist im Living Place anders. Beispielsweise könnten sich das Frühstück und das Abendessen bei den messbaren Auswirkungen kaum unterscheiden. Hier wäre nur durch den Zeitpunkt der Aktivitäten ein Unterschied festzustellen und somit eine Zuordnung möglich.

Die Auswirkungen einer Aktivität werden in dieser Arbeit auch eingesetzt, um die Aktivität selbst zu erkennen. Manche Auswirkungen bestehen darin, dass sich der Stand eines Objektes oder des Bewohners ändert. Beispielsweise ändert sich der Zustand des Bewohners durch das Anlegen von Kleidung. Diese verschiedenen Zustände müssen auch in der Ontologie beschrieben werden. Hierfür werden Begriffe für die „allgemeinen Zustände“ benötigt.

In der Arbeit von [Bhatt u. a. \(2009\)](#) wurden Zonen nach ihren Funktionen unterteilt. Die Bezeichnungen von [Bhatt u. a.](#) für die Verschiedenen Zonen sind:

Funktional Space Der Bereich, in dem es möglich ist, mit einem bestimmten Gegenstand zu interagieren. z. B. Der Bereich direkt vor einem Ofen

Range Space Der Bereich, den ein Gegenstand mit seiner Funktion erreicht. z. B. Der Bereich, in den ein Fernsehbild strahlt.

Operational Space Der Bereich, in dem ein Gegenstand sich bewegen kann. z. B. Der Schwingbereich einer Tür

Die „Upper-Ontologie“ wurde um diese Einteilung der Zonen erweitert, um die einzelnen Bereiche besser voneinander zu trennen, und die Begriffe besser zu strukturieren. Der Bereich um mit einer Tür zu interagieren, ist nicht nur der Bereich in dem eine Tür bewegt wird. Interaktion ist beispielsweise auf beiden Seiten der Tür möglich aber sie schwingt in der Regel nur in eine Richtung. Um einen Fernseher am Netzschalter ein oder aus zu schalten, muss man sich direkt vor dem Fernseher befinden. Das Bild kann hingegen in einem Kegel vor dem Fernseher gesehen werden. Um die neuen Zonen von den Bereichen des natürlichen Sprachgebrauch, wie beispielsweise Küche oder Bad, zu unterscheiden, wurde

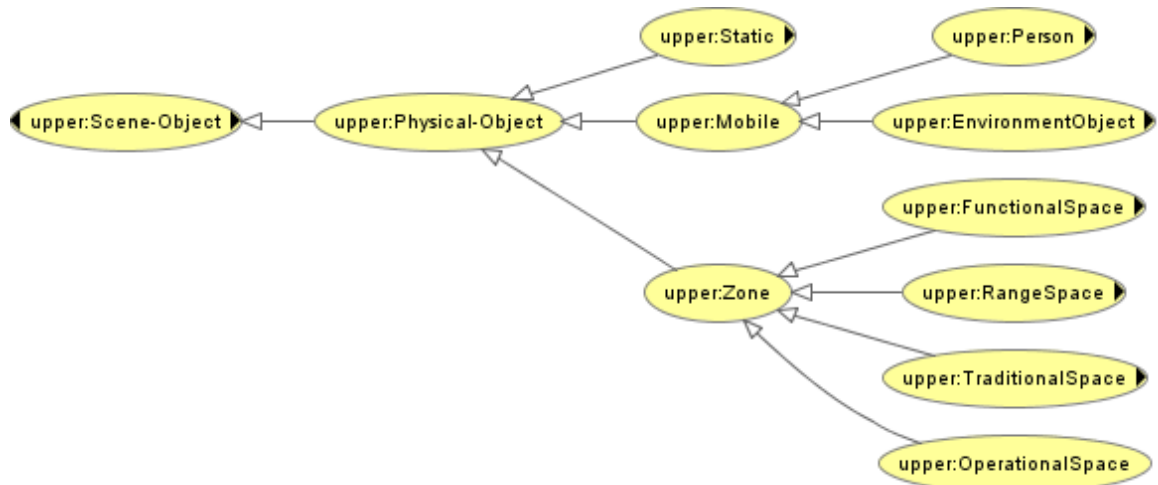


Abbildung 4.3.: Upper Ontologie physischer Teil

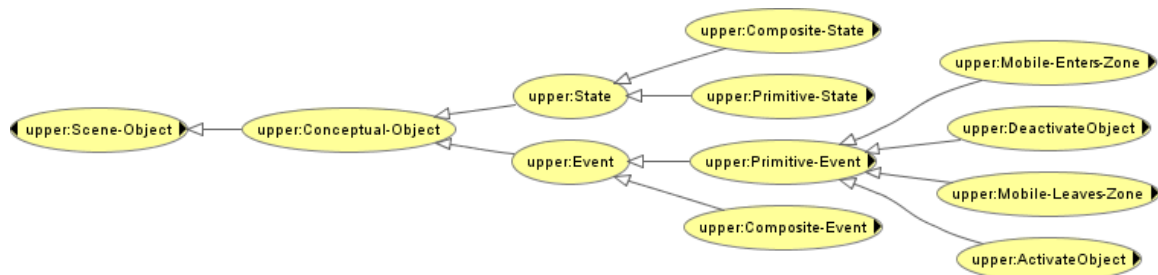


Abbildung 4.4.: Upper Ontologie konzeptueller Teil

in die Ontologie der Begriff der „Traditional Spaces“ eingefügt. In diesem Ober-Begriff sind alle traditionellen Bezeichnungen der Bereiche und Räume zusammengefasst⁶.

Die erweiterte Ontologie ist in Abbildung 4.3 und 4.4 zu sehen.

Nach den Erweiterungen der „Upper-Ontologie“ ist das 5W1H Konzept in der Ontologie wie folgt abgebildet:

- Konzeptueller Teil (How-Dimension)
- Physischer Teil
 - Mobile (Who-Dimension / What-Dimension)
 - Static (What-Dimension)
 - Zone (Where-Dimension)

⁶z. B. Küchenbereich, Badezimmer

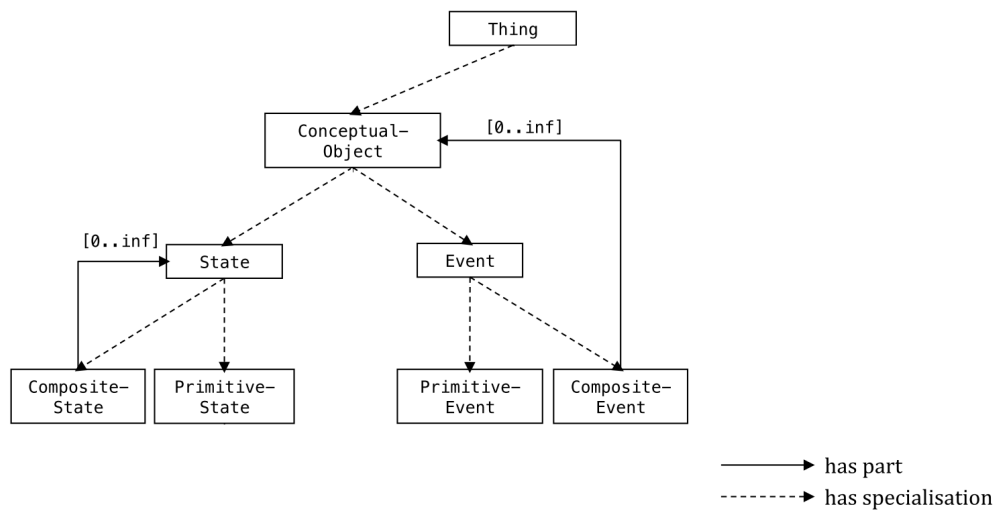


Abbildung 4.5.: Vererbungsstruktur von Scenior (vgl. [Bohlken u. a. \(2011\)](#))

Die When-Dimension ist bisher nicht in der Upper-Ontologie enthalten, da die verschiedenen Temporalen Bezeichnungen sich auch die Situationen der Wohnung beziehen. Sollten sich in der Zukunft Bezeichnungen in dieser Dimension mit anderen Domänen überschneiden, dann könnte darüber nachgedacht werden die Upper-Ontologie zu erweitern.

Die Struktur der Vererbung in Scenior ist in [Abbildung 4.5](#) dargestellt. Die Struktur zeigt, dass komplexe Zustände nur aus Zuständen bestehen können, und dass komplexe Events aus allem bestehen können. Diese Struktur wurde für das Living Place geändert. Wie in [Abschnitt 3.1.4.5](#) beschrieben wurde, beschreiben in dieser Arbeit die Events einen Zeitpunkt und die States einen Zeitraum. Ausgehend von dieser Definition ist auch ein komplexes Event ein Zeitpunkt. Teil eines Zeitpunktes kann aber nicht ein Zeitraum wie in [Bohlken u. a. \(2011\)](#) sein. Die Vererbungsstruktur im Living Place ist in [Abbildung 4.6](#) zu sehen. Hier besteht ein komplexes Event aus verschiedenen Teil-Events und eine komplexes State kann Events und andere States beinhalten. Beispielsweise ist so ein Morgen-Szenario, dass aus verschiedenen Aktivitäten und Events besteht, im Living Place eine komplexe Aktivität bzw. ein komplexes State mit einem Anfang und einem Ende.

4.4. Aufbau einer Domain-Ontologie für das Living Place

Nachdem die „Upper-Ontologie“ erweitert wurde, können nun die Begriffe für das Living Place festgelegt werden.

Die erste allgemeine Frage, welche bei einer Ontologie zu klären ist, bezieht sich auf die Sprache der Begriffe. In diesem Projekt geht es darum, ob alle Begriffe der Ontologie in

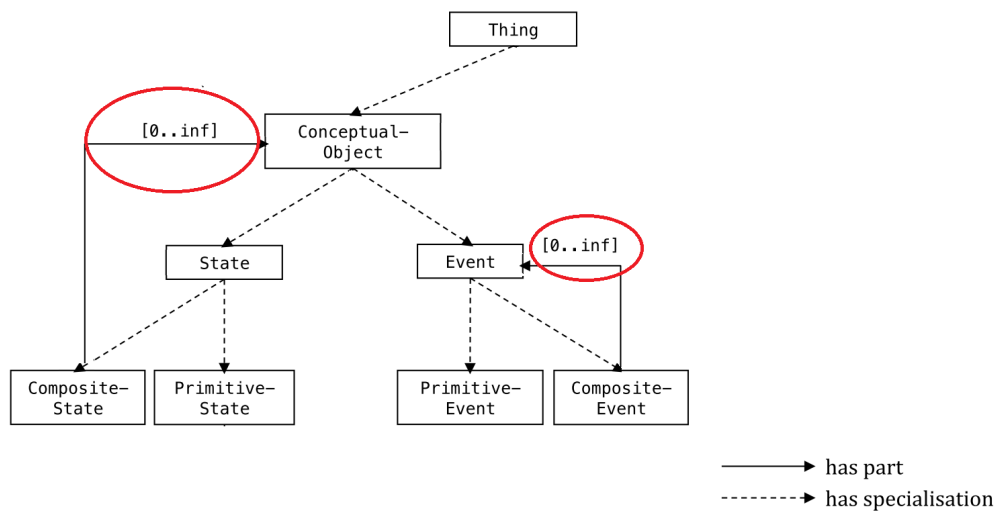


Abbildung 4.6.: Vererbungsstruktur im Living Place

Deutsch oder Englisch eingetragen werden sollten. Deutsche Begriffe haben den Vorteil, dass sie leicht verständlich sind und bei der Arbeit in diesem Projekt angewandt werden, da viele Studenten in diesem Projekt Deutsch als Muttersprache haben. Englische Begriffe haben den Vorteil, dass die Ontologie international anwendbar wird. Sollten sich internationale Zusammenarbeiten ergeben oder sollte eine Universität aus einem anderen Land Interesse an der Ontologie haben, müssen die Inhalte der Ontologie nicht erst Übersetzt werden. Aus diesem Grund wurde sich in dieser Arbeit für die englischen Begriffe entschieden. Dadurch wird die Ontologie international vergleichbar und einsetzbar. Alle Studenten sind der englischen Sprache mächtig und sind es außerdem gewohnt, mit englischen Begriffen zu arbeiten. Daher sollte es mit den englischen Begriffen kaum zu Problemen kommen.⁷

Die Begriffe in der Domain-Ontologie können nicht vollständig angelegt werden. Wie schon in Abschnitt 3.1.1.2 diskutiert, wird es immer möglich sein weitere Details der Ontologie hinzuzufügen. Daher soll in dieser Arbeit keine vollständige Ontologie für die Domäne „Smart Homes“ aufgebaut werden. Es soll beispielhaft eine Ontologie für die oben genannten Szenarien gezeigt werden. Nach diesem Muster ist es möglich die Ontologie durch Begriffe aus anderen Szenarien zu erweitern.

Der „Wecker 2.0“ [Ellenberg \(2010b\)](#) bildet im Folgenden stellvertretend die Grundlage für die Begriffe der Ontologie. Die Szenarien des „Wecker 2.0“ werden hier kurz vorgestellt. Eine ausführliche Version befindet sich im Anhang: [A](#)

Schlafen: Der Bewohner schläft in seinem Bett.

⁷Im Gegensatz zur Ontologie werden die Begriffe in dieser Arbeit weiterhin auf Deutsch bleiben. So bleiben die Begriffe aus der Analyse erhalten und es entsteht kein Bruch in der Beschreibung der Arbeit.

Frühstücken: Der Bewohner bereitet das Frühstück vor und nimmt es ein.

Waschen: Im Bad verrichtet der Bewohner die Morgen-Toilette.

Duschen: Der Bewohner nimmt ein Duschbad.

Außerhalb des Living Place: Der Bewohner hat das Living Place verlassen.

Diese Aktivitäten beziehen sich alle auf den morgendlichen Zeitraum, da der „Wecker 2.0“ nur hier sein Anwendungsgebiet hat. Um die zeitliche Dimension besser einzusetzen, werden die oben genannten Szenarien um einige wenige erweitert. Die folgenden Aktivitäten leiten sich von einem möglichen vollständigen Tagesablauf des Bewohners ab.

Mittagessen: Der Bewohner bereitet das Mittagessen vor und nimmt es ein.

Kleine Mahlzeit: Der Bewohner bereitet eine kleine Mahlzeit vor und nimmt sie ein.

Abendessen: Der Bewohner bereitet das Abendessen vor und nimmt es ein.

Arbeiten: Der Bewohner arbeitet an seinem Laptop.

Fernsehen: Der Bewohner schaut ein Programm oder einen Film am Fernsehgerät.

Nachdem 5W1H Prinzip wurden verschiedene Aktivitäten mit ihren Dimensionen definiert. Die Definitionen sind im Anhang: [A](#) aufgelistet. Aus den Definitionen der Aktivitäten ergeben sich die folgenden Begriffe für die Ontologie.

4.4.1. Agenten-Begriffe

Der Agent der Aktivitäten ist der Bewohner des Living Place. In der Upper-Ontologie ordnet sich der Bewohner unter den mobilen physischen Objekten ein. Der Bewohner ist hier neben anderen physischen Objekten angeordnet, da eine Aktivität grundsätzlich auch von einer Maschine durchgeführt werden könnte. In Bereich des Living Place ist eine Klasse für den Bewohner ausreichend, da die Aktivitäten nur für den Bewohner des Living Place modelliert werden. Dieses schließt nicht aus, dass es in dieser Klasse mehrere Primitive geben könnte.

4.4.2. Objekt-Begriffe

In den Aktivitäten werden die folgenden Objekte eingesetzt. Der Einsatz dieser Objekte ist somit ein indirektes Merkmal für eine Aktivität. Daher haben die Objekte in der Ontologie ein boolesches Prädikat „isAktive“. Dieses Prädikat wird immer dann auf *True* gesetzt (oder *False*), wenn der Gegenstand zur Benutzung bereit steht oder aktiv benutzt wird (bzw. nicht mehr benutzt wird).

Schlafen: u. a. Bett, Schrank, Regal⁸, Kleidung

Frühstücken: u. a. Stühle, Esstisch, Besteck, Küchenzeile

Waschen: u. a. Waschbecken

Duschen: u. a. Dusche

Außerhalb des Living Place: nichts

Mittagessen: u. a. Stühle, Esstisch, Besteck, Ofen, Küchenzeile

Kleine Mahlzeit: u. a. Stühle, Esstisch, Besteck

Abendessen: u. a. Stühle, Esstisch, Besteck, Ofen, Küchenzeile

Arbeiten: u. a. Laptop, Esstisch, Stühle, Sofa

Fernsehen: u. a. Sofa, Fernseher

4.4.3. Zonen-Begriffe

Die Zonen des Living Place sind alle Zonen, die auf eine Aktivität hindeuten. Bei der Definition der Zonen spielen die Objekte eine entscheidende Rolle. Wenn zu einer Aktivität ein Objekt gehört, kann die Aktivität nur dann stattfinden, wenn der Bewohner mit dem Objekt interagieren kann. Genau diesen Umstand spiegeln die „Funktional Space´s“ wider, die in Abschnitt 4.3 beschrieben wurden. Einen Ausnahme hierzu ist der Fernseher. Um den Fernseher zu nutzen, muss mit ihm nicht interagiert werden, sondern der Fernseher muss sich im Blickfeld des Bewohners befinden. Oder anders Ausgedrückt: Der Bewohner muss sich in einem Bereich befinden, in dem das Bild des Fernsehers gesehen werden kann. Dieser Beschreibung entspricht der „Range-Space“ aus [Bhatt u. a. \(2009\)](#).

Die folgenden Begriffe sind Beispiele der definierten Zonen der Ontologie:

- FunctionalSpaceBett

⁸für Kleidung

- FunctionalSpaceChair
- (usw. für alle oben genannten Objekte)

Neben den Zonen der Objekte gibt es im Living Place auch Bereiche, die den herkömmlichen (oder traditionellen) Bezeichnungen entsprechen. Da das Living Place eine „ein-Raum-Wohnung“ ist, heißen die Zonen meisten auch „Area“ und nicht „Room“. Die einzige Ausnahme ist das Badezimmer, da es ein eigener Raum im Living Place ist. Die traditionellen Zonen sind:

- DinningArea
- KitchenArea
- SleepingArea
- LivingArea
- Bathroom

Diese Begriffe sind Teile des Living Place, welches auch eine Zone in der Ontologie ist. Genauso wurde „OutsideLivingPlace“ mit in die Ontologie aufgenommen. Mit diesem Begriff ist es möglich die Abwesenheit des Bewohners zu modellieren.

4.4.4. Kontext-Begriffe

Die Begriffe der Situationen außerhalb des Living Place und innerhalb des Living Place, die nicht die Aktivitäten betreffen, werden hier zusammengefasst. Zu diesen Begriffen gehören unter anderem:

Getragene Kleidung Die verschieden Kleidungsstücke, die ein Bewohner trägt.

Wetter Die Bezeichnungen der Wetterbedingungen außerhalb des Living Place.

Termine Begriffe, die Termine beschreiben oder identifizieren.

Diese Begriffe können dazu genutzt werden die Hypothesen der Aktivitäten zu unterstützen. Diese Begriffe sind in der Regel aber keine notwendigen Voraussetzungen für Aktivitäten. Im Rahmen dieser Arbeit soll das System für die Aktivitätserkennung erstellt werden. Welche Möglichkeiten die Hypothesen über die Aktivitäten unterstützen können sind vielfältig und gehen über den Rahmen dieser Arbeit hinaus. Die Kategorie der Kontext-Begriffe in der Ontologie sollen hier die Möglichkeiten aufzeigen, die für die Bestätigung existieren.

4.4.5. Event-Begriffe

Wie in Abschnitt 3.1.4.5 beschrieben, teilen sich die Begriffe in „Events“ und „States“ auf. Bei der Analyse der Zusammenhänge zwischen Events und States zeigt sich, dass ein State von einem Event eingeleitet wird und von einem Event beendet wird. Dieser Zusammenhang liegt an den grundsätzlichen Eigenschaften der Events, den messbaren Auswirkungen der Aktivitäten, States und der Aktivitäten selbst.

Event: Beschreibt eine Änderung im Living Place.

messbare Änderung: Änderung der Bedingungen im Living Place.

State: Beschreibt einen Zustand über einen Zeitraum, in dem sich der Zustand nicht ändert.

Aktivität: Verhalten des Bewohners, dessen Anfang und Ende durch Sensoren bestimmt wird.

Somit nehmen die Sensoren eine Änderung des Zustandes in der Wohnung auf. Wenn diese Zustandsänderung auf den Anfang oder das Ende einer Aktivität des Bewohners deutet, gehört zu dieser Änderung der Sensorwerte ein Event. Aus diesem Zusammenhang folgt, dass ein oder mehrere Events am Anfang und am Ende einer Aktivität bzw. eines States stehen. Es ergibt sich also immer die Abfolge:

$$(Start-)Event \Rightarrow State \Rightarrow (Stopp-)Event$$

Wie in Abschnitt 3.2.2 diskutiert wurde dienen zur Beschreibung von Aktivitäten die Dimensionen des 5W1H Modells. Aus diesen Dimensionen ergeben sich somit die notwendigen Voraussetzungen für eine Hypothese über eine Aktivität.

Das folgende Beispiel soll diese Zusammenhänge verdeutlichen:

Who/Agent: Der Bewohner des Living Place ...

Where/Ort: ... betritt den Ort, an dem die Aktivität möglich ist. Beispielsweise ein Bereich, in dem der Bewohner mit einem Objekt interagieren kann.

When/Zeit: Der Zeitraum an dem die Aktion möglich ist, fängt an.

What/Objekt: Das Objekt wurde aktiviert oder steht zur Nutzung zur Verfügung.

Aus diesen Dimensionen ergeben sich die Folgenden Events, die auftreten müssen, damit eine Aktivität hypothetisiert werden kann.

- „Bewohner betritt Zone“ - Event
- „Objekt wird aktiviert“ - Event

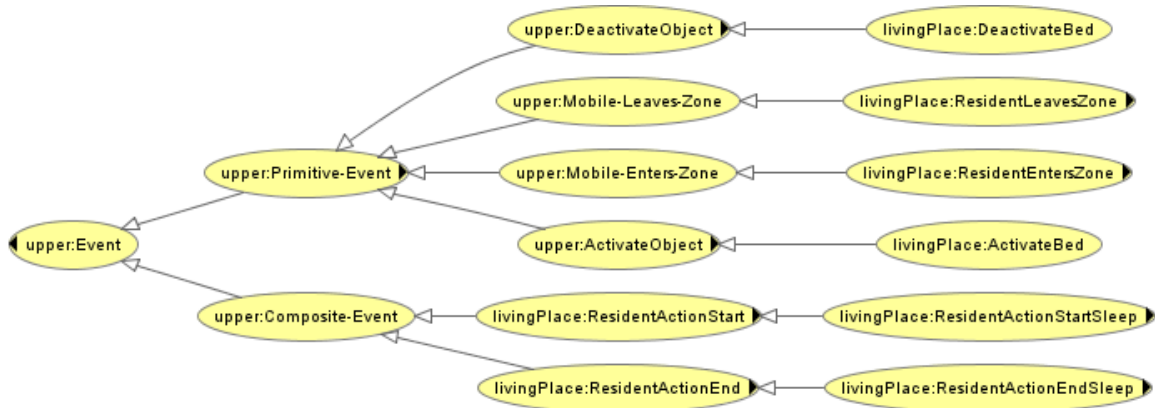


Abbildung 4.7.: Event Ontologie

- „Zeitraum beginnt“ - Event

Wenn all diese Events für eine bestimmte Aktivität aufgetreten sind, dann beginnt mit dem zuletzt aufgetretenen Event die Aktivität oder das „Start“ - Event der Aktivität. Um den Zeitpunkt des Start-Events richtig zu definieren ist es notwendig, das Start-Event in zwei verschiedene Events aufzuteilen. Ein Event bei dem die Zone vor der Aktivierung des Objektes stattfindet und ein Event, bei dem es umgekehrt ist. Diese Besonderheit wird im Nachfolgenden (Abschnitt 4.5) genauer beschrieben.

Nach dem Start einer Aktivität genügt es, wenn eine dieser Voraussetzungen nicht mehr erfüllt sind, um auf das Ende der Aktivität zu schließen. Daraus folgen zwei mögliche Varianten in der Kombination der Events:

alle Events sind notwendig Das Event beginnt mit dem Start des zuletzt auftretenden Kind-Events. Die Events sind miteinander „und“ Verknüpft.

ein Event von mehreren möglich Das Event beginnt mit dem ersten Kind-Event das auftritt. Die Kind-Events sind miteinander „oder“ Verknüpft

Hier ergeben sich bei der Definition der SWRL Regeln ein Problem. In den SWRL Regeln gibt es keine „oder-Verknüpfung“. Dieses Problem lässt sich durch die Vererbungsstruktur der Ontologie lösen. Es werden in der Ontologie für die möglichen Auslöser des Aktivitäts-Endes eigene Events definiert, die zu einem Aktivität-Stopp Event generalisiert werden. Diese Struktur ist in Abbildung 4.7 dargestellt.

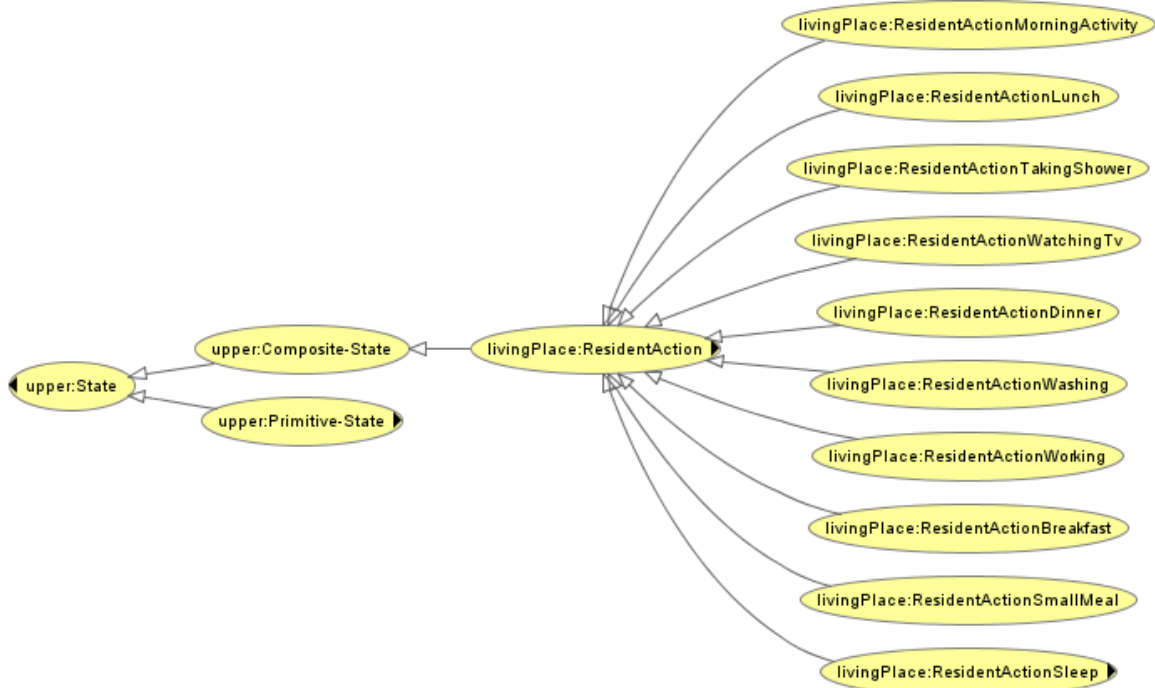


Abbildung 4.8.: State Ontologie

4.4.6. State-Begriffe

Die State-Begriffe beziehen sich auf die Aktivitäten des Living Place. Wie im letzten Abschnitt definiert, werden die Aktivitäten von einem „Start-Event“ eingeleitet und von einem „End-Event“ beendet.

Erst mit dem „End-Event“ ist die Aktivität vollständig und kann als solche erkannt werden.

Am Anfang dieses Abschnittes wurden die Verschiedenen Aktivitäten vorgestellt, die erkannt werden sollen (vgl. Abschnitt 4.4). Aus diesen Beschreibungen folgen direkt die State-Begriffe der Ontologie.

Die State-Begriffe ordnen sich in der Upper-Ontologie unter den „Composite-State“ ein, da sie aus den Start- und End-Events zusammengesetzt werden. In diesem Sinne gibt es in dieser Arbeit keine „Primitiv-State“ da diese nicht direkt von den Sensoren erkannt werden können. „Primitiv-States“ wären beispielsweise interpretierte Kamerabilder, die direkt einen schlafenden Bewohner erkennen, und diese Information an die Ontologie weiter leiten.

Wie in Abbildung 4.8 zu sehen ist, befindet sich auch die „Morgen-Aktivitäten“ auf einer Ebene mit allen anderen Szenarien. Aktivitäten, wie beispielsweise das „Schlafen“, sind keine

„Morgen-Aktivität“ (Generalisierung) sondern die „Morgen-Aktivität“ besteht unter Anderem aus der Aktivität „Schlafen“ (Teil-Ganzes-Beziehung).

4.5. Erstellen von SWRL Regeln

Durch die SWRL Regeln wird beschrieben, wann aus den primitiven Events die zusammengesetzten Events werden und wann aus den zusammengesetzten Events schließlich die Aktivitäten (States) werden.

Die zusammengesetzten Events sind in dieser Arbeit die „Start-Events“ und „End-Events“ der Aktivitäten. In der Ontologie wurde das „Start-Event“ so beschrieben, dass es sich aus den Events „Bewohner-betritt-Zone“ und „Objekt-wird-Aktiviert“ Events zusammensetzt. Durch die SWRL Regel wird nun festgelegt, dass beide Events vorhanden sein müssen damit das „Start-Event“ instanziiert werden kann. Des Weiteren beschreibt die SWRL Regel, dass für den Startzeitpunkt des „Start-Event“ der Zeitpunkt des zuletzt aufgetretenen Teil-Event eingesetzt wird. Dieser Zeitpunkt wird in der SWRL Regel mit den folgenden zwei Ungleichungen beschrieben.

$$StartEvent - ZonenEvent \geq n(milliseconds) \quad (4.1)$$

$$StartEvent - ObjektEvent \geq n(milliseconds) \quad (4.2)$$

Für das „Start-Event“ muss die Zeit des zweiten Auftretenden Events gesetzt werden. Hierfür ist es notwendig die Zeiten des „Start-Event“ mit den Zeiten des zweiten Events äquivalent zu setzen. Hierfür reichen die Ausdrücke 4.1 und 4.2 nicht aus. Da diese nur besagen, dass die Zeiten des primitiven Events denen des „Start-Events“ entsprechen aber nicht umgekehrt. Für die Äquivalenz ist für jede mögliche Reihenfolge der Events eine eigene Regel notwendig. In diesem Fall sind es die Regeln für die Events Zone → Objekt und Objekt → Zone.

Für das „End-Event“ gestellten sich die Regeln durch die besondere End-Event-Struktur (siehe Abschnitt 4.4.5) einfach. Für das Zonen-End-Event wird der Zeitpunkt des Zone-Verlassen Events übernommen. Die Regel für das Objekt-End-Event wird analog zu der Zonen-End-Regel erstellt. Eine Beispielhafte SWRL Regel ist im Anhang B.7 zu sehen. Diese Regel verdeutlicht auch die Notwendigkeit von Werkzeugen bei der Erstellung der Regeln.

Aus den Start- und End-Events werden die States (die Aktivitäten) erzeugt. In Abbildung 4.9 sind die Zusammenhänge zwischen den Events und States dargestellt. Hierbei bildet der Zeitpunkt der Start-Events den Anfangs-Zeitpunkt der Aktivität und der Zeitpunkt des zuerst Aufgetretenen End-Event wird der Stop-Zeitpunkt der Aktivität.

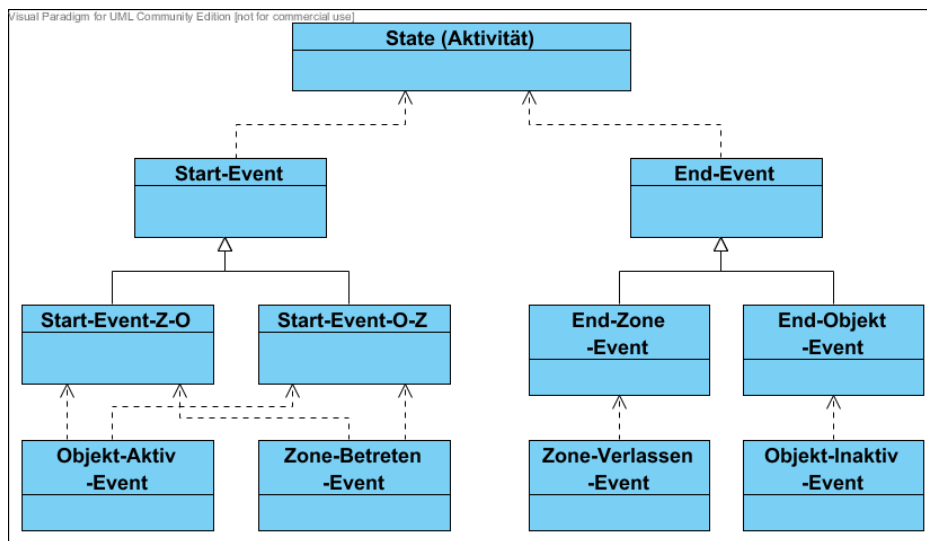


Abbildung 4.9.: Zusammenhang zwischen Events und States

Wie in Abschnitt 2.3.4 beschrieben sind die SWRL Regeln nicht entscheidbar. Um die Möglichkeiten dennoch für die Beschreibung der Temporalen Relationen einsetzen zu können ist es notwendig, die Regeln auf ihre Entscheidbaren Elemente zu reduzieren, wie es auch in dem Projekt „Scenior“ beschrieben wurde (vgl. Bohlken (2012)).

Die Temporale Logik aus Allen und Ferguson (1994) stellt für dieses Projekt quantitative temporale Operatoren zur Verfügung. Diese reichen aber nicht aus um variable temporale Verhältnisse auszudrücken. Beispielsweise muss ein Bewohner zumindest wenige Sekunden in einer bestimmten Zone verweilen, damit man auf eine Aktivität in dieser Zone schließen kann. Für diese qualitativen Beschreibungen wird eine konvexe Zeitpunkt-Algebra benötigt, wie sie in Vila (1994) beschrieben wird. In Bohlken (2012) wurde diese Algebra in SWRL umgesetzt und kann daher hier eingesetzt werden.

4.6. Extraktion der ontologischen Primitiven

Nach der Entwicklung der Ontologie ist es nun notwendig, die Ontologie mit Leben zu füllen. Hierfür müssen aus den Sensordaten des Living Place die Primitiven⁹ der Ontologie erzeugt werden. Für das Erzeugen der Primitiven sind die folgenden Schritte zu unterscheiden:

Physische Primitive Diese Primitive beschreiben das Living Place. Sie stehen fest und müssen vorher erstellt werden.

⁹Instanzen der Ontologie-Klassen

Event Primitive Events sind hier die messbaren Änderungen im Living Place. Diese Primitiven müssen zur Laufzeit erzeugt werden und über eine Schnittstelle dem Regelsystem zur Verfügung gestellt werden.

Temporale Primitive Diese Primitiven ergeben sich aus den „Logischen Sensoren“ (vgl. Abschnitt 3.3.5). Auch diese Primitiven ändern sich zur Laufzeit und müssen dem Regelsystem übergeben werden.

4.6.1. Physische Primitive

Wie in Abschnitt 4.1.2 beschrieben wurde stellen die physischen Primitiven die Grundlage für die Aktivitätserkennung da. Sie bestehen aus den folgenden Begriffsgruppen:

- Mobile Begriffe
- Statische Begriffe
- Zonen Begriffe

Diese Begriffe sind für die Aktivitätserkennung fest vorgegeben und können daher direkt modelliert werden. Grundsätzlich gehört zu jeder in Abschnitt 4.4 modellierten Klasse mindestens ein Primitives. Jede modellierte Klasse ist für die Aktivitätserkennung notwendig und kann daher nicht leer bleiben.

Neben den traditionellen Zonen-Begriffen, die in dieser Ontologie nur für die Übersicht enthalten sind¹⁰, gibt es zu jedem Objekt eine Zone. Die Art der Zone richtet sich nach der Funktion des Objektes. In der Regel muss mit dem Objekt während einer Aktivität interagiert werden. Daher sind die meisten Zonen „Funktional-Spaces“. Eine Ausnahme bildet der Fernseher. Hier reicht es aus, dass der Bewohner sich in dem Bereich befindet, in dem das Bild des Fernsehers gesehen werden kann. Daher gehört zu dem Fernseher ein „Range-Space“ anstelle des „Functional-Space“.

4.6.2. Event Primitive

Die Primitiven der Events werden zur Laufzeit erzeugt. Für diese Primitiven wurden Beobachter-Anwendungen geschrieben, die auf dem Blackboard lauschen. Beim Auftreten bestimmter Ereignissen werden die Primitiven dann von den Anwendungen erzeugt. Für das Aufstellen von Hypothesen werden die folgenden Ereignisse benötigt:

¹⁰Nur das Badezimmer ist als Begriff notwendig, da in diesem Raum aufgrund der Privatsphäre keine Positionsüberwachung existiert

Where/Who-Dimension Die Primitiven über das Betreten oder Verlassen eines Bereiches im Living Place werden auf Grund der Informationen des „Indoor Spacial Information Service“ erzeugt (vgl. [Karstaedt \(2012\)](#)). Dieses System erkennt gleichzeitig die Identität des Bewohners. Dadurch kann mit diesen Events auch festgestellt werden, wer die Zone betritt und daher eine Aktivität durchführen kann.

What-Dimension Diese Events geben darüber Auskunft, welche Objekte aktiviert werden oder zur Aktivierung bereitstehen. Diese Events werden hauptsächlich durch Sensoren erzeugt, die die Gegenstände direkt überwachen (vgl. [Pautz \(2010\)](#)). Bei anderen Gegenständen werden die Events durch Objekterkennung von Kamerabildern generiert (vgl. [Najem \(2011\)](#)).

When-Dimension Diese Events werden durch die logischen Sensoren bestimmt, wie sie in [Voskuhl \(2012\)](#) beschrieben werden. Jedes Mal, wenn auf dem Blackboard durch eine Nachricht der Beginn eines neuen Zeitraums publiziert wird, kann ein entsprechendes Event erzeugt werden.

4.6.3. Temporale Primitive

Für die Erzeugung der temporalen Primitiven muss ein System die aktuelle Uhrzeit beurteilen. Diese Beurteilung ist in der Regel von den Gewohnheiten des Bewohners abhängig. Daher ist es hier notwendig, ein dynamisches System zu erstellen, das die Interpretation auf dem Blackboard der Klassifikation zur Verfügung stellt. In [Voskuhl \(2012\)](#) werden Ansätze eines solchen Systems beschrieben.

4.7. Evaluation der Ontologie

4.7.1. Ontologie als Modell der Realität

Der physische Teil der Ontologie dient für die Klassifikation als Basis. Dieser Teil soll die Realität abbilden, um mit Hilfe dieses Modells die Aktivitäten zu klassifizieren.

4.7.1.1. Who-Dimension

Die Modellierung des Bewohners erfolgt mit einer eigenen Klasse in der Ontologie. In dieser Arbeit wurde in einem ersten Ansatz von nur einem Bewohner ausgegangen. In den SWRL

Regeln wird die Identifikation des Bewohners für das Betreten einer Zone überprüft. Dadurch hat eine Aktivität einen bestimmten Bewohner. So ist es möglich, dass zwei Aktivitäten verschiedene Bewohner haben können.

4.7.1.2. What-Dimension

Die Objekte sind eindeutig zuzuordnen. Durch das Klassenkonzept können verschiedene Objekte einer Klasse einer Aktivität zugeordnet werden. Beispielsweise ist es für das Schlafen im Bett unwichtig, welches Bett in der Wohnung dieser Aktivität zugeordnet wird. Es muss nur das gleiche Bett sein, in dessen Zone sich der Bewohner befindet.

Die Upper-Ontologie unterscheidet zwischen statischen und mobilen Objekten in der Wohnung. Wie in Abschnitt 4.3 beschrieben wurde, ist die Zuordnung zu einer dieser Gruppen in manchen Fällen problematisch. Für den Aufbau der SWRL Regeln hat sich gezeigt, dass die Zuordnung der Objekte in die Gruppe der statischen oder mobilen Klasse keine Rolle spielt. Daher ist es in dieser Arbeit ausreichend die einzelnen Begriffe intuitiv zuzuordnen.

4.7.1.3. Where-Dimension

Die Modellierung der Bereiche im Living Place teilt sich in die traditionellen und objektspezifischen Bereiche auf. Für beide Bereiche ist die Abgrenzung der Bereiche problematisch. In dieser Arbeit wird von einer scharfen Grenze der Bereiche ausgegangen. Das bedeutet, dass mit dem Erkennen eines Sensors die Person innerhalb des Bereichs ist. Es kann also nicht sein, dass sich einer Person auf der Grenze befindet, oder sich langsam in einen Bereich hinein bewegt. Diese Betrachtungsweise entspricht dem Grundgedanken der Aktivitätserkennung in dieser Arbeit. Durch die Ontologie werden die Voraussetzungen für eine Hypothese beschrieben. Durch ein probabilistisches System ist es in einem zweiten Schritt notwendig, diese Hypothesen zu beurteilen. Durch die SWRL regeln ist es möglich, diese harte Trennung von „in der Zone“ und „außerhalb der Zone“ besser zu beschreiben, in dem ein Zeitraum definiert wird, mit dem sich der Bewohner innerhalb der Zone befinden muss, damit das Event „Bewohner betritt Zone“ auftritt. Mit dieser Toleranz wird beispielsweise ein Springen zwischen diesen beiden Zuständen verhindert. Außerdem gibt es durch diese Maßnahme keine sehr kurzen Aktivitäten, weil der Bewohner nur durch eine Zone geht die zu einem aktiven Objekt gehört.

In der Ontologie wurde die Ausdehnung der Zonen nicht modelliert. Dieses Wissen ist für die Klassifikation der Aktivitäten nicht notwendig. Somit gehört dieses Wissen auch nicht in die Ontologie. Die Anwendung, die erkennen ob sich ein Bewohner in einer Zone befindet, muss die Beurteilung der Zonen korrekt umsetzen. Diese Anwendung ist im Allgemeinen für die Beurteilung der Zonen in der Wohnung verantwortlich. Daher muss in dieser Anwendung das

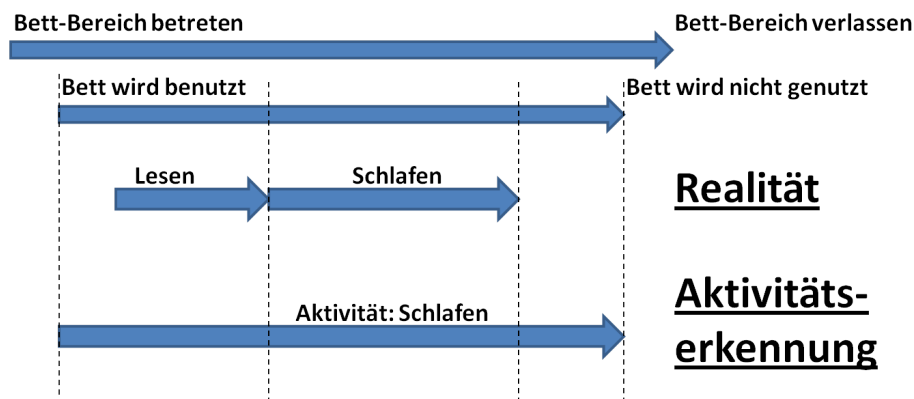


Abbildung 4.10.: Zeitstrahl der Aktivitätserkennung

Wissen über die Ausdehnung der Zonen gespeichert werden und nicht in der Ontologie. Für die Aktivitätserkennung werden die Positionen durch einen räumlichen-Informationen-Service beurteilt (vgl. [Karstaedt \(2012\)](#)). Alle weiterführenden Informationen zu diesem Thema sind dort beschrieben.

4.7.1.4. Zusammenfassung

Die Ontologie eignet sich um das Living Place für die Aktivitätserkennung zu modellieren. Die Begriffe sind durch die Klassenstruktur den Regeln allgemein zuzuordnen. Die Bezeichnung der Primitiven wird durch die Ontologie anderen Anwendungen zur Verfügung gestellt. Dadurch können Fehler in der Kommunikation vermieden werden.

4.7.2. Evaluierung des Ontologie Modells

Die Aktivitäten der Szenarien sind in der Ontologie modelliert. Der Zeitraum, in dem die Aktivitäten erkannt werden, ist wahrscheinlich größer als der Zeitraum der Aktivität des Bewohners in der Realität (vgl. [Abbildung 4.10](#)). Es ist zu erwarten, dass durch das Modell die Aktivitäten zu früh erkannt werden und dass die Aktivitäten länger erkannt werden als in der Realität.

Die Aktivitäten des Bewohners werden nicht durchgehend von dem System identifiziert. Zwischen den einzelnen modellierten Aktivitäten sind Lücken zu erwarten, in denen das System keine Aktivität des Bewohners feststellen kann. Diese Lücken kommen zum eine durch die Bewegung in der Wohnung zustande, die nicht als eigene Aktivität modelliert wurden. Des Weiteren sind die Aktivitäten der Szenarien nicht vollständig. Es gibt viele verschiedene Aktivitäten, die in der Ontologie nicht modelliert wurden und daher auch nicht erkannt werden.

Das Ziel der Aktivitätserkennung ist, genau die Aktivitäten zu erkennen, die für verschiedene Anwendungen von Interesse sind. Daher stellen die Lücken zwischen den erkannten Aktivitäten kein Problem da. Ebenso sind die Wege innerhalb der Wohnung nicht relevant. Sollte sich eine Anwendung für diese Informationen interessieren, stehen sie durch [Karstedt \(2012\)](#) auf dem Blackboard zur Verfügung.

Das Modell entspricht der Wohnung wie sie derzeit existiert. Da die Verhältnisse und Zusammenhänge in der Wohnung sich verändern können, ist es notwendig das Modell diesen Veränderungen anzupassen. Beispielsweise könnte der Bewohner sich entschließen Morgens nur noch ausgiebig Kaffee zu trinken und nicht mehr mit seinem üblichen Besteck zu frühstücken. In diesem Fall müsste die Aktivität Frühstücken an die neuen Zusammenhänge angepasst werden.

5. Design und Evaluation

Die bisher diskutierten Verfahren sollen im Living Place Hamburg eingesetzt werden. Im Folgenden wird die Umsetzung der ontologiebasierten Aktivitätserkennung beschrieben. Es sollen die folgenden Schritte umgesetzt werden (vgl. Abbildung 3.1):

Daten-Transformation: Es müssen die verschiedenen Sensoren und Informationsquellen identifiziert werden, die für die Erzeugung der Features notwendig sind. Gegebenenfalls müssen verschiedene Sensoren simuliert werden, wenn sie nicht zur Verfügung stehen.

Daten-Aggregation: Die Features sind vorverarbeitete Sensordaten oder Informationen aus externen Quellen. Es muss ermittelt werden, welche Features sich aus den Sensordaten generieren lassen und welche externen Quellen zur Verfügung stehen. Auch hier stellt sich die Frage, was notwendig ist um das Erkennen von Kontext zu gewährleisten.

Daten-Klassifikation: Verschiedene Kontexte müssen definiert werden. Der Kontext soll dann durch Klassifikation aus einem Feature-Vector erkannt werden.

5.1. Übersicht der Kontext-Nutzung im Living Place

In diesem Kapitel wird beschrieben, was mit der Kontexterkennung im Smart Home erreicht werden kann. Anschließend wird dargestellt, welche Systeme im Living Place entwickelt werden um die aktuelle Situation zu erkennen und zu analysieren. Eins dieser Systeme ist die Aktivitätserkennung dieser Arbeit. Im letzten Abschnitt wird gezeigt, welchen Teil diese Arbeit im Living Place einnimmt.

5.1.1. Kontexterkennung im Living Place

Ein Teil der intelligenten Wohnung besteht somit aus Anwendungen, die sich anpassen. Diese Anpassungen können sehr verschieden sein.

Für die Anpassung des Verhaltens benötigen die Anwendungen und Geräte die aktuelle Situation oder den Kontext der Wohnung. Nur so können sie ihr Verhalten „richtig“ anpassen. Hierfür ist es außerdem notwendig, dass der erkannte Kontext möglichst genau ist. Eine hohe Fehlerrate bei der Kontext-Erkennung bedeutet ein hohes Fehlverhalten der Geräte.

Wie in Abschnitt 2.2 beschrieben wurde, kann Kontext jede Information sein, die für eine Anwendung relevant ist. Eine noch allgemeinere Definition ist: „Die Kontext-Informationen beschreiben die Situation in der Wohnung“. Es ist somit für die Anwendungen im Living Place notwendig, die Kontext-Informationen für Programme lesbar zur Verfügung zu stellen.

In Abschnitt 3.1.1 wurde ein Modell zur Beschreibung der Situation diskutiert. Dieses Modell kann nicht nur zur Beschreibung der Aktivitäten eingesetzt werden, sondern es eignet sich auch um die Situation im gesamten Living Place zu beschreiben. Die Kontext-Informationen durch ein Modell zur Verfügung zu stellen ist aber nur eine von mehreren Möglichkeiten.

Eine weitere Möglichkeit besteht darin, in den aufgezeichneten Sensorwerten nach Mustern zu suchen. In der Arbeit Voskuhl (2012) werden in den Sensorwerten des Living Place nach Cluster gesucht. Diese Cluster geben einen Hinweis auf eine besondere Situation in der Wohnung. Der Vorteil dieser Methode ist es, dass auch Situationen entdeckt werden können, die vorher nicht bekannt waren. Wenn sich anschließend die Sensorwerte wieder innerhalb eines Clusters bewegen, können Anwendungen sich dementsprechend anpassen.

Es gibt verschiedene Wege die geforderten Informationen für die Anwendungen zu erzeugen. Eine direkte Möglichkeit sind die Sensoren der Wohnung selbst. Sie stellen Informationen zur Verfügung, mit denen manche Anwendungen ihr Verhalten anpassen können. Wie in Abschnitt 3.1.3 diskutiert sind die Werte der Sensoren meistens nicht genug. Sie benötigen meistens einen Bezug zur Realität oder zusätzliche Informationen bezüglich ihrer Bedeutung. Beispielsweise werden in Karstaedt (2012) die Positionsdaten mit dem Bewohner in Verbindung gebracht. Darüber hinaus werden die Positionsdaten mit einem Modell der Wohnung verglichen. Dadurch ist es möglich aus den Sensordaten verschiedene zusätzliche Informationen der aktuellen Situation zu liefern. Hier liefern die Sensoren in Kombination mit einem Modell die gewünschten Informationen.

Eine andere Möglichkeit Informationen zur Verfügung zu stellen, ist die Sensoren zu interpretieren. Bei diesem Weg werden für bestimmte Situationen Sensorwerte erwartet. Wenn diese Sensorwerte eintreten, kann eine Hypothese über die Situation in der Wohnung aufgestellt werden. Um die erwarteten Sensorwerte zu ermitteln, gibt es verschiedene Möglichkeiten. Entweder die Sensorwerte werden durch Lernverfahren ermittelt oder die Werte werden durch Expertenwissen festgelegt. Bei diesem Verfahren können Informationen zur Verfügung gestellt werden, die sich nicht direkt aus den Sensordaten ergeben. So stehen den Anwendungen der intelligenten Wohnung weitere Informationen zur Verfügung.

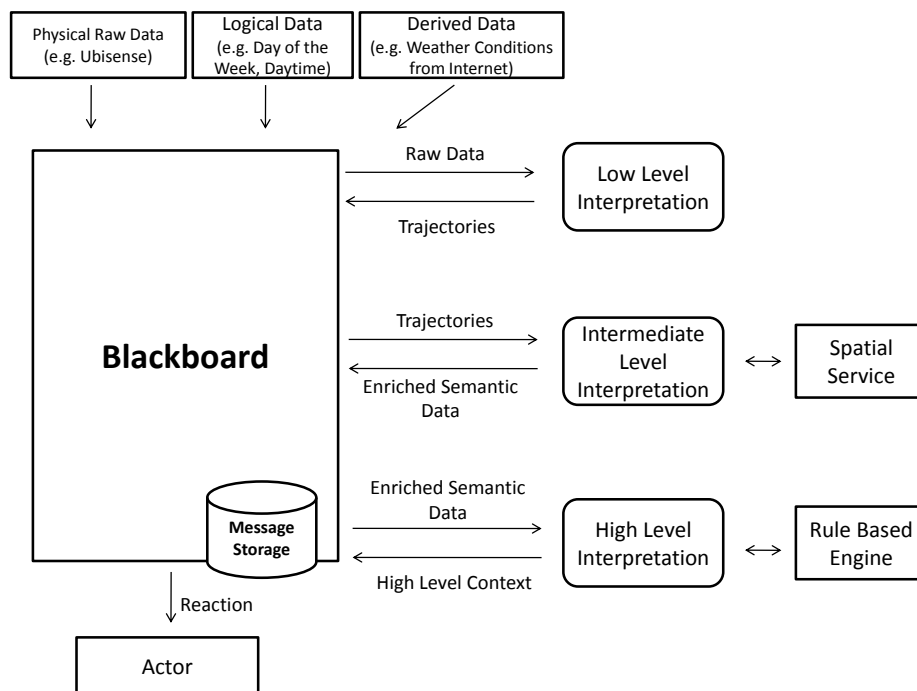


Abbildung 5.1.: Gesamt-Architektur

Abbildung 5.1 zeigt, wie die verschiedenen Komponenten der Kontext-Bearbeitung zusammenarbeiten (vgl. [Ellenberg u. a. \(2011\)](#)).

- Auf der unteren Ebene werden die Sensordaten ohne ein Modell analysiert.
- Auf der mittleren Ebene werden die Sensordaten und die Daten aus der unteren Ebene in einem räumlichen Modell gespeichert. Durch das 3D-Modell werden zusätzliche Informationen zur Verfügung gestellt.
- Auf der oberen Ebene werden alle zur Verfügung stehenden Daten genutzt um mit Hilfe eines ausdrucksstarken Modells neue komplexe Kontext-Informationen zu erzeugen.

Das Blackboard bildet die zentrale Kommunikationsschnittstelle zwischen den Anwendungen. Auf dem Blackboard kann jede Anwendung einen neuen Informationskanal erstellen um die eigenen Informationen zu publizieren. Andere Anwendungen können sich auf diesem Informationskanal anmelden um die Informationen zu empfangen (die Informationen abonnieren). Weiterhin können auf dem Blackboard Kanäle eröffnet werden, für die es nur einen Empfänger von Nachrichten gibt (den Erzeuger des Kanals). Dieser Kanal kann dafür genutzt werden, Anfragen als Server zu empfangen. Diese Anfragen werden anschließend auf einem anderen Informationskanal beantwortet.

5.1.2. Einbettung der Aktivitätserkennung in das Living Place

In dieser Arbeit sollen die vorhandenen Daten interpretiert werden, um weitere Informationen zu gewinnen. Diese sind Informationen, die nicht direkt von einem Sensor bestimmt werden können. Beispiele für die zu erkennenden Aktivitäten sind das Schlafen oder das Frühstück am Esstisch. Dabei werden modellbasierte Daten (vgl. Karstaedt (2012)) sowie einfache Sensordaten (vgl. Pautz (2010) und Voskuhl (2012)) zur Analyse genutzt. Mit dieser Arbeit stehen kontextabhängigen Anwendungen weitere Informationen zur Verfügung um den Bewohner in seinem Handeln zu unterstützen.

Beispiele für Anwendungen, welche die Informationen über die Aktivitäten in der Wohnung einsetzen sind beispielsweise der „Wecker 2.0“ aus Ellenberg (2010b) oder die Lichtsteuerung des Living Place¹.

Weiterhin könnten die analysierten Daten aus Voskuhl (2012) eingesetzt werden, um das System zu verbessern und weiterzuentwickeln. Durch die Cluster könnte ein automatisiertes Erlernen der Aktivitäten und Regeln möglich sein.

Um diese Ziele zu erreichen soll im Folgenden das in Abbildung 4.1 dargestellte Modell umgesetzt werden. Die Teile der Abbildung aus Abschnitt 4.1 sind:

Eingabe-Daten Diese werden auf dem Blackboard des Living Place zur Verfügung gestellt.

Wissensbasis Die Wissensbasis bildet die in Kapitel 4 beschriebene Ontologie.

Regelmaschine Als Regelmaschine eignet sich die Anwendung Scenior aus Bohlken u. a. (2011), da diese in der Lage ist, die Ontologie zu interpretieren und Aktivitäten zu erkennen.

Interpretiertes Wissen Dieses Wissen muss dem Living Place zur Verfügung gestellt werden.

Die oben genannten Komponenten müssen nun umgesetzt werden. Der erste Schritt besteht in dem Erstellen der Wissensbasis, wie sie in Abbildung 5.2 dargestellt sind. Die Wohnung wird in einer Ontologie modelliert. In der Ontologie werden die einzelnen Bestandteile der Wohnung abgebildet, sowie deren Relationen untereinander. Der zweite Teil der Wissensbasis sind die Regeln, mit denen die Aktivitäten erkannt werden sollen. Hierfür werden die Aktivitäten auch in der Ontologie modelliert. Anschließend müssen die Zusammenhänge zwischen den Aktivitäten und dem Living Place ausgedrückt werden.

Problematisch bei der Definition Aktivitäten sind deren Anfang und Ende. Ähnlich wie das Problem der Abgrenzung einzelner Aktivitäten untereinander (vgl. Abschnitt 3.1.1.2) verhält

¹<http://livingplace.informatik.haw-hamburg.de/blog/?p=484> zuletzt abgerufen am 22.12.2011

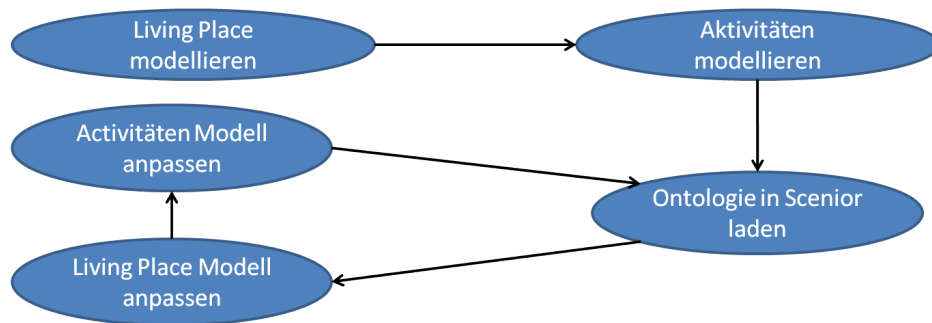


Abbildung 5.2.: Arbeitsschritte der Wissensbasis

es sich für Anfang und Ende. Eine Voraussetzung für die Aktivität ist der Ort an dem die Aktivität stattfinden kann. Wenn der Bewohner beispielsweise „zu weit“ vom Besteck entfernt ist, wird er nicht mehr Frühstück. „zu weit“ ist aber ein unscharfer Begriff. Diesen Begriff konkret auf eine Linie zu reduzieren entspricht nicht der Realität. Daher sollte diese Unschärfe in das Modell übertragen werden. Für diese Arbeit wurde entschieden, im ersten Schritte diese Grenzen eindeutig zu Modellieren, da das Modell für eine unscharfe Modellierung deutlich erweitert werden muss. Diese Erweiterung ist im Rahmen dieser Arbeit nicht möglich.

5.2. Einbindung der Aktivitätserkennung in das Living Place

In diesem Abschnitt wird beschrieben, wie die Aktivitätserkennung im Living Place integriert wird. Die Integration teilt sich in zwei Arbeitsschritte auf.

- Erstellen der Wissensbasis
- Anbinden von Scenior an das Blackboard

Wie in Abschnitt 4.1 beschrieben wurde, bildet die Ontologie die Wissensbasis für die Regelmaschine (Scenior). Dieser Vorgang wurde in Kapitel 4 ausführlich erläutert. Durch das Erstellen der Ontologie ist der Aufbau der Wissensbasis und die Konfiguration der Regelmaschine abgeschlossen. Aus diesen Zusammenhängen kann Scenior automatisch die Regeln für die Aktivitätserkennung erstellen. Intern übersetzt Scenior die Ontologie mit ihren SWRL Regeln in eine JESS-Rule-Engine². In Abbildung 5.2 ist eine Übersicht dieser Arbeitsschritte zu sehen. Nachdem das Living Place und die Aktivitäten modelliert wurden, können diese in Scenior geladen werden. Eventuell müssen diese Schritte wiederholt werden, wenn sich

²<http://www.jessrules.com/> zuletzt abgerufen am 22.12.2011

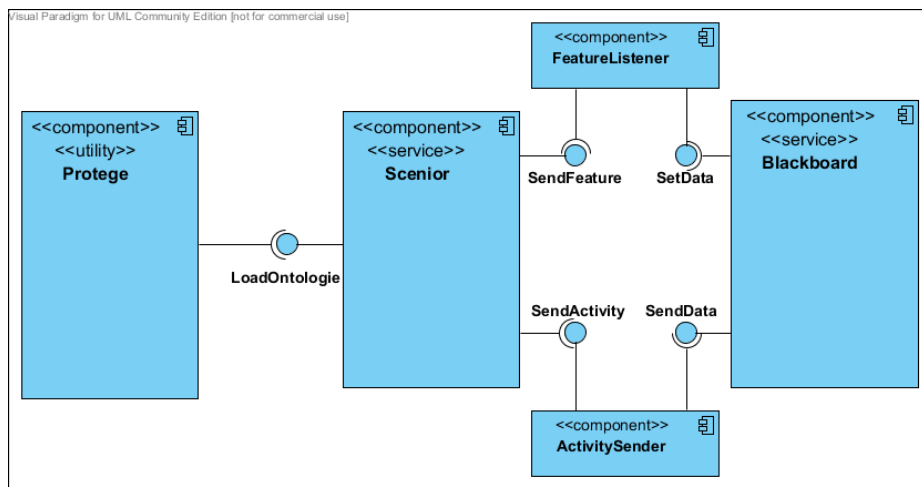


Abbildung 5.3.: Komponenten der Aktivitätserkennung

das Living Place oder die Aktivitäten ändern. Die Modellierung erfolgt vor bzw. parallel zum laufenden Betrieb.

Der zweite Teil der Integration ist das Anbinden von Scenior an das Blackboard des Living Place und das Anpassen der Anwendung an die neuen Datenformate. Nachdem die Ontologie in Scenior geladen wurde, kann die Regelmaschine ihre Arbeit aufnehmen. Im Folgenden wird beschrieben, wie die Regelmaschine an das Living Place angepasst wurde und wie die Schnittstellen zwischen Scenior und dem Blackboard aufgebaut sind. Eine Übersicht dieser Komponenten, die im Living Place aktiv sind, ist in Abbildung 5.3 zu sehen.

5.2.1. Anpassung von Scenior

Die Regelmaschine wie sie in [Bohlken u. a. \(2011\)](#) entwickelt wurde, konnte im Living Place nicht direkt eingesetzt werden. Dieses lag unter anderem an den verschiedenen Erweiterungen der Ontologie (vgl. Abschnitt 4.3) und den großen Freiheitsgraden bei der Kombination der Aktivitäten im Living Place (vgl. Abschnitt 3.1.1.3).

Eine Erweiterung in der Ontologie sind die Objekte, die für eine Aktivität notwendig sind. Die bisherigen Primitiven, mit denen Regelmaschine gearbeitet hatte, waren die Agenten und die Orte der Aktivitäten. Die erwarteten Eingangsdaten mussten erweitert werden.

Des Weiteren sind die aktivierten Objekte ein zweites Event, die neben dem Betreten der Zone auf eine Aktivität hindeuten. Von den einzelnen Objekten können verschiedene Primitive der gleichen Klasse in der Wohnung existieren. Beispielsweise existieren im Living Place mehrere Fernseher. Wenn eine Aktivität durch ein Event beendet wird, dann muss das Objekt bzw. die Zone die gleiche sein, wie in den Events, welche die Aktivität gestartet haben.

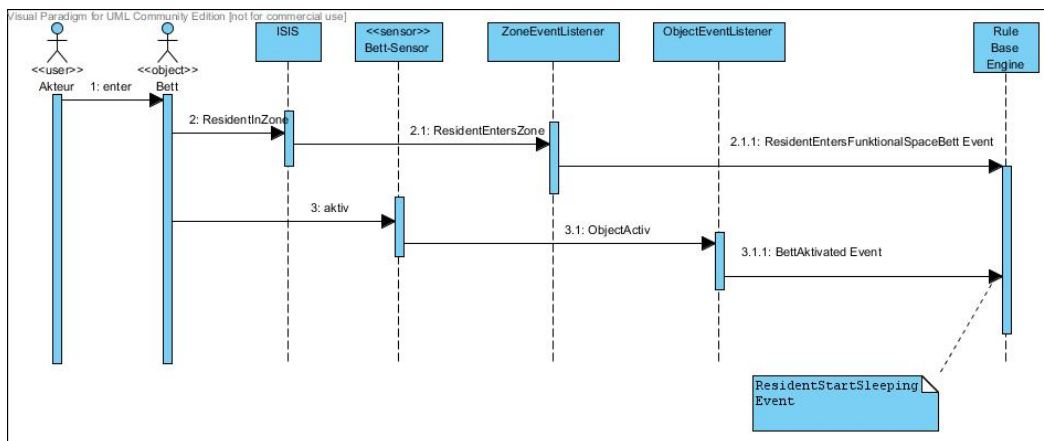


Abbildung 5.4.: Ablauf der Event-Erzeugung

Wenn beispielsweise der Bewohner gerade auf dem „Fernseher 1“ eine Sendung schaut und der „Fernseher 2“ schaltet sich ab, dann darf die Regelmaschine die Aktivität mit dem „Fernseher 1“ nicht beenden. Diese Vergleiche zwischen den verschiedenen Events waren bei Scenior nicht relevant, da es immer nur einen Primitiven pro Klasse gab.

Die oben beschriebenen Anpassungen wurden in Zusammenarbeit mit [Bohlken](#) erarbeitet. Die technische Umsetzung wurde von [Bohlken](#) durchgeführt. Weitere Informationen zur Anpassung der Regelmaschine können in [Bohlken \(2012\)](#) nachgelesen werden.

5.2.2. Anwendung zum Erzeugen von Events

Für die Anbindung von Scenior an das Living Place wird ein Programm benötigt, das die notwendigen Daten auf dem Blackboard abonniert und an Scenior weiter gibt.

Die Regelmaschine erwartet die primitiven Events, die in der Ontologie modelliert wurden. Events in dieser Art stehen aber auf dem Blackboard nicht zur Verfügung. Daher ist eine weitere Aufgabe des Programms, die Daten in das richtige Format von Scenior umzuwandeln.

Es werden von Scenior zwei verschiedene Arten von Events erwartet.

Objekt-Event Aktivieren/Deaktivieren der Events (Zeitpunkt // Objekt) über Klasse an oder aus

Zonen-Event Betreten oder Verlassen der Zone (Zeitpunkt // Objekt der Zone // Klasse der Zone) über Objekt die Zone identifiziert

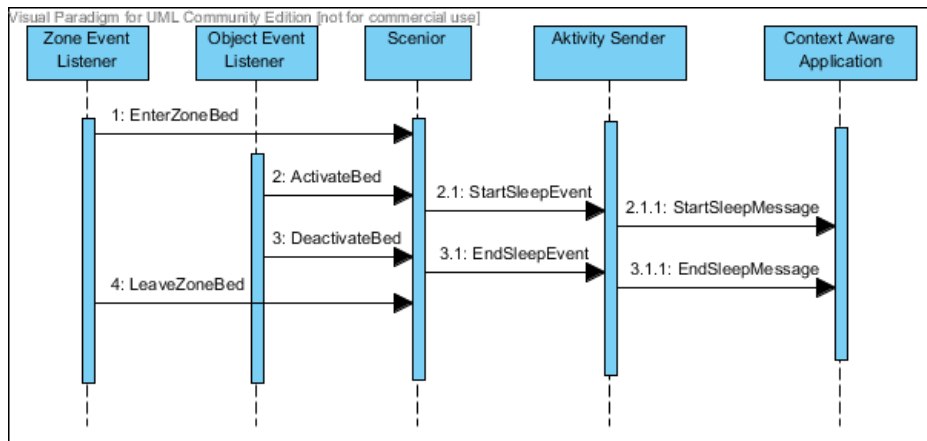


Abbildung 5.5.: Eventverarbeitung von Scenior

Für jedes dieser Events gibt es eine Anwendung, die sich für die entsprechenden Daten auf dem Blackboard angemeldet hat und diese in einer Queue speichert. Scenior beobachtet die Anwendungen und holt sich bei Änderungen die neuen Events und verarbeitet sie. Der Ablauf der Eventerzeugung ist in Abbildung 5.4 dargestellt.

5.2.3. Anwendung zum Versenden von Aktivitäten

Nachdem aufgrund der eingehenden Daten eine Aktivität erkannt wurden, müssen diese Information auf dem Blackboard für andere Anwendungen zur Verfügung gestellt werden. Dafür wird eine Anwendung benötigt, die diese Informationen auf dem Blackboard publiziert.

Eine Aktivität besteht aus ihrem Start-Event und dem End-Event. Das bedeutet, dass Scenior eine Aktivität erst erkennt, wenn sie mit dem End-Event vorbei ist. Aus diesem Grund wird nicht die Aktivität selbst auf dem Blackboard bekannt gegeben, sondern das Start-Event der Aktivität. In Abbildung 5.5 ist ein beispielhafter Verlauf der Event-Verarbeitung zu sehen.

5.2.4. Beschreibung von Scenior

Bei der Erkennung der Aktivitäten fängt Scenior mit dem Start-Event der Aktivitäten an. Bei jedem Event, welches an Scenior gesendet wird, entsteht ein neuer Hypothesenbaum.

Die primitiven Events sind für Scenior die Ausgangsbasis. Mit den Events versucht Scenior die Aktivitätsbäume aufzubauen. Dieser Baum wurde in Abbildung 4.9 dargestellt. Sollte ein End-Event vor dem Start-Event eintreffen, dann wird dieses verworfen, da der Baum nicht mehr zustande kommen kann. Das Start-Event wird mit dem Auftreten des zweiten Events

erstellt, welches für die Aktivität notwendig ist. Mit jedem Event kopiert Scenior den aktuellen Zustand und erzeugt eine neue Instanz. Durch diesen Mechanismus wird gewährleistet, dass von Scenior alle möglichen Abläufe und Kombinationen von Events erkannt werden können. Alle Instanzen, die von Scenior erzeugt werden, stellen eine Hypothese über die Realität dar. Um festzulegen welche dieser Instanzen am wahrscheinlichsten der Realität entsprechen, ist ein zusätzliches System notwendig, welches diese probabilistischen Berechnungen durchführt.

5.3. Entwicklung einer Testumgebung

Der erste Schritt bei der Entwicklung der Aktivitätserkennung ist, die Voraussetzungen für die Entwicklung fertigzustellen. Das Living Place ist ein Labor, welches sich zum Zeitpunkt dieser Arbeit noch im Aufbau befindet. Daher konnte in der Entwicklung nicht auf Daten aus dem Living Place zurückgegriffen werden. Daten, die für die Entwicklung benötigt werden, müssen erstellt werden. Außerdem werden Werkzeuge benötigt, mit denen die Daten unter kontrollierten Bedingungen zur Verfügung gestellt werden können.

5.3.1. Nachrichten Sender und Empfänger

Die ersten Tools, die benötigt wurden sind Sender und Empfänger für die Blackboard-Kommunikation. Das Blackboard entkoppelt den Sender und den Empfänger. Daher kann ein Sender nicht sicher sein, ob seine Nachricht übermittelt wurde. Ein Empfänger, der keine Nachrichten empfängt, weiß nicht, ob das an einer fehlerhaften Konfiguration liegt oder ob es keinen Sender gibt, der Nachrichten versendet.

5.3.1.1. Nachrichten-Sender

Der Nachrichten-Sender ist speziell für das Nachrichtenformat der Living-Place-Kommunikation ausgelegt. Für eine programmiersprachunabhängige Kommunikation werden die Nachrichten im JSON-Format über das Blackboard versandt (vgl. [Crockford \(2006\)](#)). Dabei sind alle Felder der JSON-Nachricht als Key-Value-Feld hinterlegt. Die Oberfläche des Nachrichten-Senders ist in Abbildung 5.6 zu sehen. Ein Beispiel für eine JSON-Nachricht wäre:

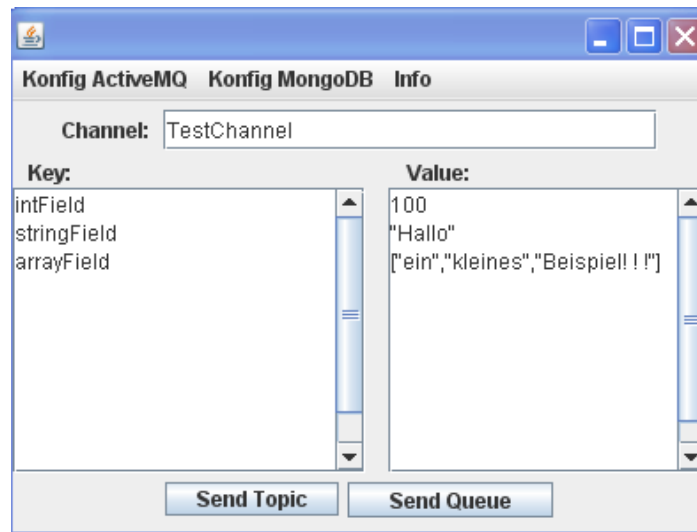


Abbildung 5.6.: Test Nachrichten Sender

JSON Beispiel

```

1 {
2   "UbisenseTagId": "123-234-456-213",
3   "Unit": "meter",
4   "NewPosition": {
5     "X": 5.677,
6     "Y": 4.196,
7     "Z": 0.0
8   },
9   "Version": "0.6",
10  "Id": "undefined",
11  "Ontology": "http://livingplace.informatik.haw-hamburg.de
12 /livingplaceOntologie.owl#"
13 }

```

Mit diesem Tool können einfach und direkt Nachrichten erzeugt und verschickt werden. Dieses Tool ist dafür gedacht, Sensoren zu simulieren oder Services mit einer entsprechenden Nachricht anzufragen. Beim Start des Nachrichten-Senders ist es möglich die Nachrichten direkt zu konfigurieren, so dass nach dem Start eine bestimmte Nachricht per Knopfdruck direkt versandt werden kann.

5.3.1.2. Nachrichten-Empfänger

Wenn Services im Livingplace laufen oder Nachrichten versandt werden, muss auch sichergestellt werden, dass diese vom Blackboard richtig verteilt werden. Dafür wurde der

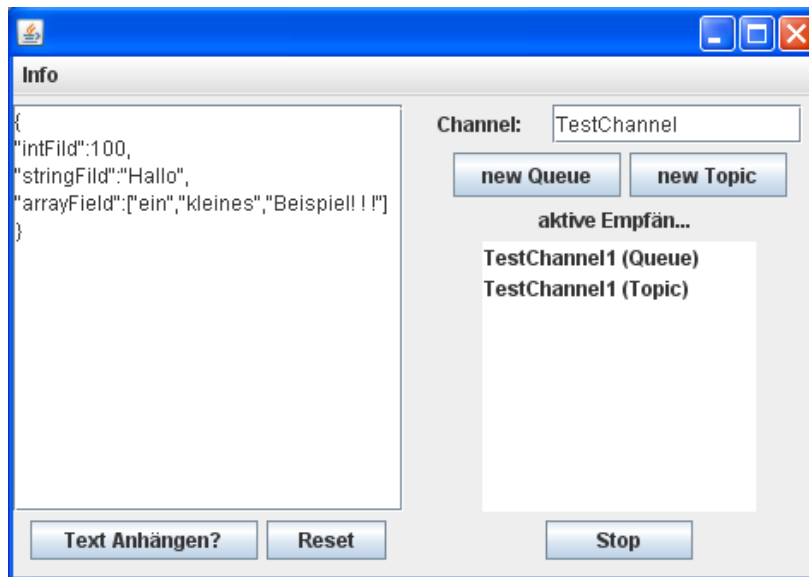


Abbildung 5.7.: Test-Nachricht-Empfänger

Nachrichten-Empfänger entwickelt. Mit diesem Empfänger kann man sich auf verschiedenen Kanälen anmelden und es werden die Nachrichten, die auf dem Kanal empfangen werden angezeigt (siehe Abbildung 5.7). Durch das JSON-Format der Nachrichten, sind die Nachrichten für einen Menschen auch lesbar. Daher eignet sich dieses Tool auch, um die Ausgaben von Sensoren zu überwachen oder die Antwort eines Services zu überprüfen. Da das Blackboard die Nachrichten an alle Interessierte verteilt, kann dieses Tool auch parallel zum laufenden Betrieb eingesetzt werden.

5.3.2. Nachrichten-Aufnahme und -Abspielgerät

Ein weiteres Problem sind eine Folge von Informationen, die auf dem Blackboard von einem Service erwartet werden. Beispielsweise erwartete die Aktivitätserkennung ständig die aktuelle Position des Bewohners. Für die Entwicklung der Anwendungen ist es problematisch, ständig eine Person in Bereitschaft zu halten und sie in der Wohnung umherlaufen zu lassen, wenn die Daten benötigt werden. Sinnvoller ist es, wenn die Daten einer Person einmal aufgezeichnet werden und dieses dann jederzeit zum Abspielen zur Verfügung stehen. Grundsätzlich könnten diese Nachrichten auch mit dem Message-Sender versandt werden. Es ist aber problematisch viele Nachrichten pro Sekunde zu senden oder die Abstände zwischen den Nachrichten gleichmäßig zu halten.

Hierfür gibt es den MongoDB-Player (siehe Abbildung 5.8). Wie in [Otto und Voskuhl \(2010\)](#) beschrieben, werden die Nachrichten, die über das Blackboard verteilt werden, in einer Da-

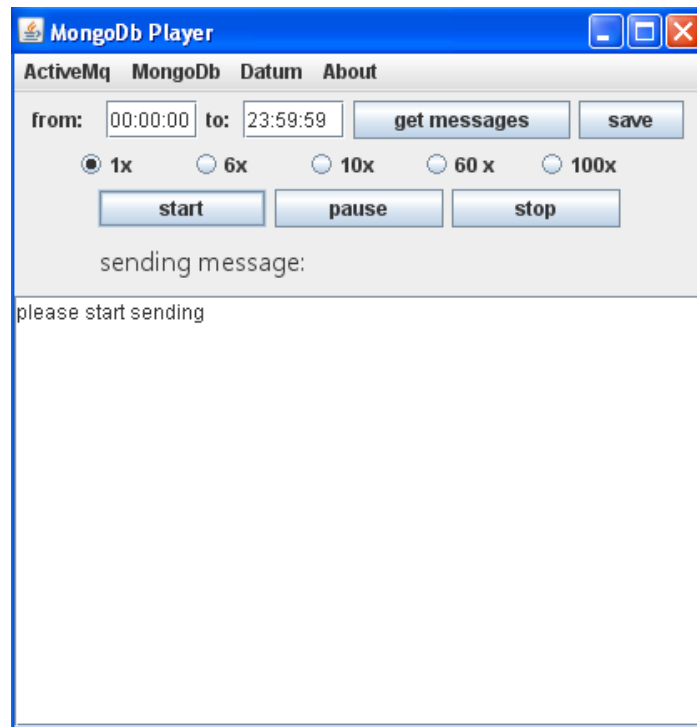


Abbildung 5.8.: Nachrichten Wiedergabe

tenbank persistiert. Diese Tatsache nutzt der MongoDB-Player aus. Er lädt die Daten aus der Datenbank von einem bestimmten Zeitpunkt bis zu einer bestimmten Zeit. Diese Nachrichten können dann über den Player abgespielt werden. Bei der Wiedergabe startet der Player mit der ersten Nachricht und gibt alle folgenden Nachrichten in dem zeitlichen Abstand wieder, in denen sie in der Datenbank angekommen sind. Da sich bei der Blackboard-Architektur nicht um ein Echtzeitsystem handelt, können Verzögerungen durch das Netzwerk vernachlässigt werden. Die unterschiedlichen Zeiten der verteilten Systeme sind in diesem Fall kein Problem, da nur die relativen Abstände der Nachrichten zueinander wichtig sind. Diese Abstände sind durch den Zeitgeber der Datenbank konsistent.

Um in annehmbarer Zeit ein Szenario zu testen, das in der Realität über eine Stunde geht, oder vielleicht einen ganzen Tag, können die Abstände der Nachrichten durch einen festen Wert geteilt werden. Bei einem Divisor von 10, ist eine Stunde in sechs Minuten vorbei und bei einem Divisor von 100 dauert ein ganzer Tag nur noch 14,4 Minuten.

Durch den MongoDB-Player ergeben sich noch weitere Möglichkeiten. Man kann die Nachrichten, die der Player abspielt wieder in der Datenbank speichern und so einfach Testdaten vervielfältigen. Des Weiteren ist es möglich, beim Abspielen der Daten weitere Nachrichten über das Blackboard zu senden. So können in mehreren Schritten komplexe Szenarien aufgebaut werden.

Um die erzeugten Szenarien unabhängig von der Datenbank abzurufen und zu verwalten, können die Nachrichten auch in einer Datei gespeichert und wieder geladen werden.

5.3.3. Weitere Simulationswerkzeuge

Weitere Simulationswerkzeuge, die in dieser Arbeit zum Einsatz kamen, wurden in anderen Projekten entwickelt.

Um die Aktivitäten in der Wohnung zu bestimmen, werden nicht nur die Sensoren der Wohnung benötigt. Es sind auch Informationen von außerhalb relevant und es werden allgemeine Informationen benötigt. Allgemeine Informationen sind beispielsweise die Uhrzeit oder Einträge aus dem Terminkalender des Bewohners. Um diese Informationen bereitzustellen, wurde in [Voskuhl \(2012\)](#) ein „Kontext-Simulator“ entwickelt. In dem Simulator kann man Datum, Uhrzeit, Wetter und viele andere Bedingungen einstellen.

Die Ortsinformationen des Bewohners spielen in dieser Arbeit eine besondere Rolle. Um immer genau die Bewegungsinformationen des Bewohners zu bekommen, die man braucht wurde in [Karstaedt \(2011\)](#) ein Tool entwickelt, mit dem man die Bewegungen des Bewohners simulieren kann. Mit dem Tool kann man beispielsweise den Bewohner vom Bad in die Küche schicken, wenn man es benötigt.

Wie in Abschnitt [3.1.1.3](#) beschrieben, ist ein Merkmal der Aktivitätserkennung die zeitliche Abfolge von verschiedenen Aktivitäten nacheinander. Um für Tests solche Abfolgen von Aktivitäten einfach zu erstellen und abspielen zu lassen wurde in [Basener \(2011\)](#) ein Werkzeug entwickelt, mit dem sich Nachrichtenfolgen programmieren lassen. Mit dem Werkzeug kann man beispielsweise ein „Morgen-Szenario“ erstellen, dass als Referenz für die Aktivitätserkennung dienen kann.

5.4. Evaluation der Aktivitätserkennung

5.4.1. Evaluation der Arbeitsschritte

Wie in Abschnitt [5.2](#) beschrieben wurde, sind zum Aufbau der Aktivitätserkennung die folgenden Schritte notwendig:

- Erstellen der Wissensbasis
- Anbinden von Scenior an das Blackboard

Das Erstellen der Wissensbasis wurde in dieser Arbeit durch Expertenwissen realisiert. Dabei wurden die Events und States direkt aus den Szenarien abgeleitet. Das Einfügen dieser Begriffe und das Modellieren der Zusammenhänge ließ sich gut mit dem Werkzeug Protege umsetzen.

Die SWRL-Regeln für die Aktivitätserkennung haben eine einfache Struktur, die sich in nahezu allen Regeln ähnelt. Andererseits werden die SWRL-Regeln durch ihre ontologiebasierten Elemente leicht unübersichtlich. Die Bezeichnungen durch die URI-Darstellung und die fehlende Verdeutlichung der SWRL-Elemente in Protege haben die Arbeit erschwert. Hier sollte für den laufenden Betrieb ein Editor für das Erstellen der Regeln entwickelt werden. Da die Struktur der Regeln einfach und regelmäßig ist, sollte der Editor mit einem relativ geringen Aufwand zu erstellen sein. In dieser Arbeit wurden die Machbarkeit und die Möglichkeiten der SWRL-Regeln untersucht. Für diese Zwecke waren die Unterstützung durch das Werkzeug Protege ausreichend.

Als nächste Schritte ist es denkbar, hierfür einen automatischen Prozess zu implementieren. Beispielsweise könnte aus den Ergebnissen von Voskuhl (2012) die Events und States extrahiert werden. Die Ergebnisse von Voskuhl müssten dahingehend erst noch analysiert werden, ob diese sich für das automatische Lernen eignen. Aus diesen Events und States können dann die SWRL-Regeln automatisch generiert werden, da diese meistens nach den gleichen Mustern erzeugt werden. Das Umsetzen des automatischen Lernens der Regeln würde über den Rahmen dieser Arbeit hinausgehen.

Wenn die Ontologie fehlerfrei ist, bereitet das Laden der Ontologie in Scenior keine Probleme. Es ist aber möglich, eine Ontologie zu bauen, die von Scenior nicht korrekt verarbeitet werden kann. Ebenso ist es möglich, dass die Ontologie in Scenior geladen wird, aber die Aktivitätserkennung nicht korrekt abläuft. Diese Fehler könnten spezielle automatische Überprüfungen verhindern. Diese Werkzeuge müssten aber speziell auf Scenior abgestimmt werden. Die normalen Anwendungen zur Überprüfung einer Ontologie helfen hier nicht weiter. Dieses liegt unter anderem daran, dass diese Anwendungen die Ontologie nur auf Inkonsistenz prüfen können. Diese Überprüfungen reichen für Scenior nicht aus.

Die Anbindung von Scenior an das Blackboard wurde über Adapter realisiert. Da die Daten auf dem Blackboard in einem einfachen JSON-Format hinterlegt sind, konnten die Adapter mit geringem Aufwand erstellt werden. Diese Adapter müssen im weiteren Verlauf nur angepasst werden, wenn sich das Nachrichtenformat auf dem Blackboard ändert. Das bedeutet, dass die Aktivitätserkennung kein Problem darstellt, wenn die Sensoren und Anwendungen verändert werden, solange sich das Nachrichtenformat nicht ändert.

Das Senden der Ergebnisse an das Blackboard wird ebenfalls durch einen Adapter umgesetzt. Dieser Adapter legt das Nachrichtenformat fest, welches die kontextabhängigen Anwendungen verstehen müssen. Hier muss sich in Zukunft zeigen, ob die Inhalte der Nachricht ausreichend sind. Das festzustellen, ist aber nicht Teil dieser Arbeit. Wichtig hierbei ist,

dass die Begriffe der Aktivitäten in der Ontologie festgelegt werden. Das bedeutet, dass die kontextabhängigen Anwendungen auch die Inhalte der Ontologie beachten müssen.

5.4.2. Evaluation des Design

In dieser Arbeit wurde eine Ontologie eingesetzt um den Kontext in der Wohnung zu beschreiben. Eine Ontologie ist eine komplexe Struktur, in die sich ein Entwickler erst einarbeiten muss, bevor er sie einsetzen kann. Es hat sich aber gezeigt, dass die Verhältnisse im Living Place ein komplexes und ausdrucksstarkes Modell benötigen. Gerade die Möglichkeit die Verhältnisse zwischen den Objekten im Living Place zu modellieren hat sich im weiteren Verlauf der Arbeit als hilfreich erwiesen. Durch diese Informationen in der Ontologie konnten die Aktivitäten einfacher beschrieben werden. Des Weiteren wurde die Vererbungsstruktur der Ontologie unter anderem bei dem Aufbau der Events eingesetzt. Durch die Vererbung war es erst möglich den Aufbau der Start- und End-Events zu beschreiben (vgl. Abschnitt [4.4.5](#)).

Der Einsatz von eindeutigen Start- und End-Events stellte sich bei dem Einsatz von Scenior als wichtig heraus. Die Aktivitäten werden von Senior erst mit dem End-Event instanziiert. Die meisten Anwendungen brauchen aber die Information einer Aktivität bei ihrem Start. Daher werden die Start-Events benötigt um diese auf dem Blackboard zu publizieren.

Über das Blackboard sind die einzelnen Einheiten sehr lose miteinander gekoppelt. Diese lose Kopplung hat den Nachteil, dass die Aktivitätserkennung nicht sicher sein kann, dass es eine Einheit gibt, die die Daten für die primitiven Events auf dem Blackboard zur Verfügung stellen. Ebenso könnten sich die Nachrichtenformate ändern ohne dass die Aktivitätserkennung dieses mitbekommt. Sie würde in beiden Fällen keine Eingangsdaten mehr bekommen aber keinen Fehler erzeugen. Diese Problematik ist durch das aktuelle Design noch nicht gelöst. Auf der anderen Seite werden die Sensoren durch die lose Kopplung leicht austauschbar. Diese könnte sich in der Zukunft als hilfreich herausstellen, gerade weil das Living Place eine Laborumgebung ist, in der viele Änderungen zu erwarten sind.

Innerhalb der Nachrichten erwartet die Aktivitätserkennung Daten, die der Ontologie entsprechen. Dieses könnte gerade bei dem Einsatz von Sensoren zu Problemen führen. Für Sensoren ist in der Regel eine hardwarenahe Programmierung erforderlich. Bei einer solchen Software stehen meistens nicht die Möglichkeiten zum Abfragen der Ontologie bereit. Die Namensbindung muss daher im weiteren Einsatz der Aktivitätserkennung überprüft und gegebenenfalls angepasst werden .



(a) Schlafen

(b) Duschen

(c) Frühstück

Abbildung 5.9.: Morgen Szenario

5.4.3. Evaluation durch eine Testumgebung

Durch die in Abschnitt 5.3 beschriebenen Werkzeuge ist eine realitätsnahe Simulation des Living Place möglich.

Test-Szenario: Der Bewohner geht am Abend zu seinem Bett um zu schlafen. Er legt sich in das Bett und schläft bis zum Morgen (Abbildung 5.9a). Morgens steht er auf und geht erst mal in das Badezimmer. Dort duscht er ausgiebig (Abbildung 5.9b). Nach dem Duschen und abtrocknen geht er in die Küche. Er holt eine Kaffeetasse und Teller aus dem Küchenschrank. Hier macht er sich sein Frühstück. Das Frühstück verzehrt er direkt am Küchentresen (Abbildung 5.9c). Er geht zur Spüle und stellt sein das Geschirr dort ab.

Aus diesem Szenario wurden die folgenden Events extrahiert, die durch die Test-Werkzeuge simuliert werden müssen:

1. Der Bewohner geht in den Nutzungs-Bereich des Bettes.
2. Das Bett stellt fest, dass es benutzt wird.
3. Das Bett stellt fest, dass keine Person mehr in ihm liegt.
4. Der Bewohner verlässt den Nutzungs-Bereich des Bettes.
5. Der Bewohner betritt das Badezimmer.
6. Die Dusche erkennt, dass sie aktiv ist.

7. Die Dusche erkennt, dass sie nicht mehr aktiv ist.
8. Der Bewohner verlässt das Badezimmer.
9. Der Bewohner geht in den Bereich des Küchentresens.
10. Es wird gemeldet, dass das Geschirr zur Nutzung bereit steht.
11. Der Bewohner verlässt den Küchentresen.
12. Es wird gemeldet, dass das Geschirr nicht mehr zur Nutzung bereit steht.

Die Events aus den simulierten Positionsdaten werden an Scenior weitergegeben. Ebenso werden aus den Objektdaten die entsprechenden Events erzeugt. Aus den erzeugten Events wurde für die Testzwecke ein Script erzeugt, welches im Anhang aufgeführt ist (siehe Anhang: [C.1](#), [C.2](#) und [C.3](#)).

Scenior kombiniert aus den primitiven Events die Start- und End-Events.

Start-Event Da bei allen Aktivitäten der Bewohner zuerst die Zone betritt und dann das Objekt aktiviert wird, wird in allen drei Fällen das Start-Event „Zone-Objekt“ erkannt (vgl. Abbildung [C.1](#), [C.7](#) und [C.13](#)). Das „Objekt-Zone“ Start-Event wird von Scenior in allen drei Fällen als fehlerhaft verworfen (vgl. Abbildung [C.2](#), [C.8](#) und [C.14](#)). Im Hintergrund behält Scenior das Objekt-Aktiv-Event bereit, um die Aktivität durch ein Zone-Betreten-Event doch noch zu erkennen.

End-Event Für das Ende der Aktivitäten erkennt Scenior mit jedem primitiven Zone-Verlassen-Event und Objekt-Inaktiv-Event, dass eine Aktivität Beendet ist (vgl. Abbildung [C.3](#), [C.4](#), [C.9](#), [C.10](#), [C.15](#) und [C.16](#)). Auf dieser unteren Ebene der Events besteht noch kein Zusammenhang zwischen den verschiedenen End-Events. Daher ist es richtig, dass Scenior beide End-Events erkennt, auch wenn das zweite Event nicht mehr relevant ist.

Aus den Start- und End-Events setzt Scenior dann die Hypothesen über die Aktivitäten zusammen. Wie oben beschrieben wird eine Aktivität erst erkannt, wenn diese mit dem ersten End-Event vollständig ist. Da es zwei End-Events gibt, erkennt Scenior auch zwei Instanzen der Aktivität. Die erste mit dem ersten End-Event und eine zweite mit dem zweiten End-Event. Dieses Verhalten liegt an dem Konzept, mit dem Scenior entwickelt wurde. Scenior versucht alle Möglichkeiten zu erkennen, gleich wie wahrscheinlich diese sind. Erst durch eine weitere Instanz werden diese Möglichkeiten beurteilt. Es könnte in diesem Fall sein, dass die Sensoren für das Objekt einen falschen Wert liefern und dass tatsächlich die zweite vollständige Aktivität die richtige ist. Daher ist dieses Verhalten von Scenior genau das Verhalten, welches erwartet wurde (vgl. Abbildung [C.5](#), [C.6](#), [C.11](#), [C.12](#), [C.17](#) und [C.18](#)).

Weil Scenior alle Möglichkeiten erkennt, werden für die Morgenaktivität nicht nur die Erwartete Kombination mit den Aktivitäten: Schlafen(Beendet durch die Objekt), Duschen(Beendet

durch die Objekt) und Frühstück(Beendet durch das Zone) (vgl. Abbildung C.19, C.20, C.21 und C.22)erkannt sondern alle 8 Kombinationsmöglichkeiten der Aktivitäten. Dieses ist auch in der Abbildung C.22 an der Zahl der beendeten Threads zu erkennen (roter Kreis in der Abbildung). Wie mit diesen verschiedenen Treffern durch Scenior umgegangen werden kann, muss durch Tests in Kombination mit den Empfängern der Ausgaben evaluiert werden. Diese Empfänger standen bei Abschluss der Arbeit noch nicht zur Verfügung.

5.4.4. Einsatz der Software im Living Place

Die ein- und ausgehenden Daten in der realen Umgebung gleichen denen aus der Testumgebung. Daher sind in der Realität keine Abweichungen zu der Testumgebung zu erwarten. In zeitlich weiträumig angelegten Tests ist es erforderlich die Aktivitätserkennung auf unerwartete Fehler zu testen. Der Rahmen dieser Tests würde den Rahmen dieser Arbeit übersteigen. Die folgenden Tests sollten im Weiteren durchgeführt werden:

Sensoren-Schwankungen Welche Auswirkungen haben die Schwankungen der Sensorwerte auf die Aktivitätserkennung bzw. welche Schwankungen keine Auswirkungen zeigen.

Sensoren-Fehler Wie wirken sich vereinzelte falsche Sensor-Werte bei der Aktivitätserkennung aus. Wie viele falsche Werte kann das System ausgleichen.

Falsche Positive/Negative Welche Möglichkeiten gibt es die Fehlinterpretationen von Scenior aufzuzeichnen und zu analysieren.

Wie weiter oben beschrieben, sollten die aufgestellten Hypothesen durch ein probabilistisches Netz ,wie beispielsweise einem Bayes-Netz, verstärkt oder abgeschwächt werden. In diese Berechnungen könnten dann noch weitere Sensor-Informationen einfließen, wie beispielsweise die Helligkeit oder der Geräuschpegel in der Wohnung. Viele dieser Informationen deuten nicht auf eine Aktivität hin, aber sie können eine bestehende Hypothese unterstützen. Durch diese Informationen könnten dann die verschiedenen Anwendungen die Hypothesen besser einsetzen. Denkbar wäre auch die Wahrscheinlichkeiten bei der Interpretation der Aktivitäten nur intern einzusetzen und nur die Aktivitäten zu publizieren, die mit einer bestimmten Wahrscheinlichkeit wahr sind.

In der Arbeit Dai und Xu (2008) wurden die generierten Kontext-Informationen eingesetzt um die Kontext-Erkennung zu fokussieren. Dies wäre auch hier denkbar. Beispielsweise könnten nur bei der Aktivität „Schlafen“ einfache Bildinformationen des Bettes auf Bewegungen hin interpretiert werden. Dabei könnte für das „Schlafen“ eine Schwelle definiert werden. Sobald der Bewohner sich zu viel bewegt, ist das Schlafen vorbei.

In der Ontologie wird der Kontext des Living Place gespeichert. Bisher werden nur die Aktivitäten auf dem Blackboard mitgeteilt. Denkbar ist hier die Ontologie als Service für den gesamten Kontext der Wohnung zu nutzen. Dafür müsste die Ontologie für die weitergehenden Informationen angepasst werden und es müsste eine Schnittstelle geschaffen werden, über die Kontext-Informationen von verschiedenen Anwendungen abgerufen werden können.

6. Zusammenfassung und Ausblick

6.1. Zusammenfassung

In dieser Arbeit wurde eine Struktur entwickelt, mit der sich die Aktivitäten in einer intelligenten Wohnung (Living Place) definieren und computerlesbar speichern lassen. Die Grundlage für diese Struktur bildet die 5W1H-Methode und die Tiefen-Kasus aus [Fillmore \(1968\)](#). Diese Strukturen wurden in einer Ontologie gespeichert. Dabei wurde die Sprache OWL-DL genutzt und eine entscheidbare Teilmenge der SWRL-Regeln um die temporären Zusammenhänge zu definieren. Bei der Modellierung definieren die Aktivitäten den Kontext. Die Aktivitäten:

- haben einen Agenten
- haben einen Ort
- haben eine Zeit
- benötigen ein Objekt
- werden mit einer Intension durchgeführt

Das Erkennen der Aktivitäten erfolgt indirekt durch Sensor-Daten. Hierfür werden die Sensor-Daten für die Aktivitätserkennung angepasst und gegebenenfalls zusammengefasst.

In Zusammenarbeit mit der Universität Hamburg wurde die Upper-Ontologie aus [Bohlken u. a. \(2011\)](#) weiterentwickelt, so dass diese Upper-Ontologie auch für das Living Place anwendbar ist. Anschließend wurde eine Domain-Ontologie für das Living Place erstellt. Diese Ontologie beinhaltet eine physische Repräsentation der Wohnung und die Konzepte der Aktivitäten. Die modellierten Aktivitäten in der Domain-Ontologie stammen aus den Szenarien des „Wecker 2.0“, einem situationsabhängigen Programms.

Für die Klassifikation wird das Programm Scenior eingesetzt (vgl. [Bohlken u. a. \(2011\)](#)). Scenior ist ein Projekt der Universität Hamburg, welches mit dieser Arbeit große Überschneidungen hat. Scenior übersetzt eine Ontologie mit speziell für Scenior erstellten SWRL-Regeln automatisch in eine JESS-Rule-Engine. Durch Scenior werden mit dem Regelwerk aus den aufbereiteten Sensordaten die verschiedenen Aktivitäten klassifiziert.

Die von dem Klassifikationsalgorithmus (in Scenior) erkannten Aktivitäten des Bewohners sind Hypothesen. Diese Hypothesen müssen durch weitere Sensordaten und einem probabilistischen Netz beurteilt werden. Die Erstellung eines solchen Netzes war nicht Teil dieser Arbeit.

6.2. Ausblick

Auf der Grundlage dieser Arbeit sind viele verschiedene intelligente Anwendungen möglich. Der Fokus der meisten Anwendungen in einer intelligenten Wohnung ist der Bewohner. Mit den Aktivitäten des Bewohners steht in der Wohnung eine weitere grundlegende Information zur Verfügung um die reaktionen der Wohnung intelligenter zu gestalten. In der weiteren Entwicklung des Living Place können nun die Pläne für solche Anwendungen konkret werden.

Die grundsätzliche Idee dieser Arbeit ist es den Bewohner zu überwachen und zu beobachten, um ihn besser unterstützen zu können. Fraglich ist, ob diese Überwachung in der Realität vom Menschen gewollt ist und akzeptiert wird. Gerade im Hintergrund der aktuellen Diskussion über Vorratsdatenspeicherung ¹ ist es notwendig diese Frage hinreichend zu beantworten.

Eine weitere Frage, die es in diesem Sinne zu klären gilt, ist welche Aufgaben die Wohnung übernehmen darf und wann dieses von dem Bewohner gewollt ist. Es ist schön von der Wohnung mit einem Kaffee geweckt zu werden. Man will aber vielleicht nicht jeden Morgen einen Kaffee oder man will kurz nach dem Aufwachen vielleicht nicht einmal diese Frage beantworten müssen. Hier spielen die individuellen Bedürfnisse des Bewohners eine entscheidende Rolle, die in die Kontext-Erkennung einfließen könnten.

Schon heute nutzen viele Menschen beispielsweise E-Mails, ohne zu wissen wie der Text verschickt wird oder über welche Länder die Daten versandt werden. Es ist für die meisten nur wichtig, dass der verschickte Text bei dem gewünschten Empfänger ankommt und mehr nicht. Andererseits vermittelt es uns ein ungutes Gefühl, wenn wir einen Zug benutzen wollen, in dem es keinen Lokführer gibt, sondern nur ein Computersystem. Die Art, wie mit dem erkannten Kontext in der Wohnung umgegangen werden sollte, muss ebenso noch erforscht werden. Sollte die Wohnung entscheiden, dass es notwendig ist den Kaffee für den Bewohner bereit zu stellen? Wie weit sollte der Bewohner über die Reaktionen der Wohnung die Kontrolle behalten? Wie viel Kontrolle will der Bewohner überhaupt behalten?

In dieser Arbeit wird die Why-Dimension ² nicht weiter verfolgt. Ein Teil dieser Dimension sind unter anderem die individuellen Bedürfnisse des Bewohners. Das Framework könnte

¹http://www.vorratsdatenspeicherung.de/static/portal_de.html zuletzt abgerufen am 22.12.2011

²die Intention des Bewohners bzw. der interne Kontext

um diese Dimension erweitert werden, um eine Vorhersage der nächsten Aktionen leichter treffen zu können. Beispielsweise könnte nach dem Aufstehen mitten in der Nacht das Licht in der Küche angeschaltet werden, wenn als Grund des Aufstehens „Hunger“ oder „Durst“ erkannt würde. Fraglich ist, welche Möglichkeiten es gibt an dieses Wissen zu gelangen.

Interessant wäre auch, die Computer der Wohnung nicht als solche zu betrachten, sondern als einen Freund, der für den Bewohner Aufgaben übernimmt. In diesem Sinne könnten die intelligente Wohnung anders nicht als Werkzeug wahrgenommen werden, sondern als Partner. Hierfür sind auch die Ergebnisse des Sonder-Forschungs-Bereich 62 „Eine Companion-Technologie für kognitive technische Systeme“³ interessant. Vielleicht wäre die intelligente Wohnung eine praktische Umsetzung der „Companion-Technologie“?

Diese Arbeit stellt nur einen Baustein im weiten Feld des „Context Aware Computing“ und „Smart Home“ dar. Weitere Bausteine sind nötig um das Living Place mit Intelligenz zu füllen.

„Most important, ubiquitous computers will help overcome the problem of information overload. There is more information available at our fingertips during a walk in the woods than in any computer system, yet people find a walk among trees relaxing and computers frustrating. Machines that fit the human environment, instead of forcing humans to enter theirs, will make using a computer as refreshing as taking a walk in the woods.“ Weiser (1999)

³<http://www.sfb-trr-62.de/> zuletzt abgerufen am 22.12.2011

Literaturverzeichnis

- [Abowd u. a. 1999] ABOWD, Gregory D. ; DEY, Anind K. ; BROWN, Peter J. ; DAVIES, Nigel ; SMITH, Mark ; STEGGLES, Pete: Towards a Better Understanding of Context and Context-Awareness. In: *Proceedings of the 1st international symposium on Handheld and Ubiquitous Computing*. London, UK : Springer-Verlag, 1999 (HUC '99), S. 304–307. – URL <http://portal.acm.org/citation.cfm?id=647985.743843>. – ISBN 3-540-66550-1
- [Allen 1983] ALLEN, James F.: Maintaining knowledge about temporal intervals. In: *Commun. ACM* 26 (1983), November, S. 832–843. – URL <http://doi.acm.org/10.1145/182.358434>. – ISSN 0001-0782
- [Allen und Ferguson 1994] ALLEN, James F. ; FERGUSON, George: Actions and Events in Interval Temporal Logic. In: *Journal of Logic and Computation* 4 (1994), S. 531–579. – URL <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.30.1586>
- [Augusto und Nugent 2006] AUGUSTO, Juan C. ; NUGENT, Chris D.: Smart homes can be smarter. In: *In Designing Smart Homes - The Role of Artificial Intelligence*, URL <http://www.infj.ulst.ac.uk/~jcaug/introbook06.pdf>, 2006
- [Basener 2011] BASENER, Andreas: Entwicklung eines Skriptsimulators fu"r das Living-Place / HAW Hamburg. URL <http://users.informatik.haw-hamburg.de/~ubicomp/projekte/master2011-proj1/basener.pdf>, 2011. – Forschungsbericht
- [Bell und Dourish 2007] BELL, Genevieve ; DOURISH, Paul: Yesterday's tomorrows: notes on ubiquitous computing's dominant vision. In: *Personal Ubiquitous Comput.* 11 (2007), January, S. 133–143. – URL <http://dx.doi.org/10.1007/s00779-006-0071-x>. – ISSN 1617-4909
- [Bhatt u. a. 2009] BHATT, M. ; DYLLA, F. ; HOIS, J.: Spatio-terminological inference for the design of ambient environments. In: *Conference on Spatial Information Theory (COSIT 09)* Springer-Verlag (2009), S. pages 371–391

- [Bohlken u.a. 2011] BOHLKEN, W. ; KOOPMANN, P. ; NEUMANN, B.: SCE-NIOR: Ontology-based Interpretation of Aircraft Service Activities / Universität Hamburg. URL http://epub.sub.uni-hamburg.de/informatik/volltexte/2011/167/pdf/B_297_11.pdf, 2011. – Forschungsbericht
- [Bohlken 2012] BOHLKEN, Wilfried: *Realzeit-Szeneninterpretation mit Ontologie-basierten Regeln*, Universität Hamburg, Dissertation, 2012
- [Bolchini u.a. 2007] BOLCHINI, Cristiana ; CURINO, Carlo A. ; QUINTARELLI, Elisa ; SCHREIBER, Fabio A. ; TANCA, Letizia: A Data-oriented Survey of Context Models. In: *ACM SIGMOD Record* 36 (2007), Dec, Nr. 4, S. 19–26
- [Brooks 1991] BROOKS, Rodney A.: Intelligence without representation. In: *Artif. Intell.* 47 (1991), February, S. 139–159. – URL <http://dl.acm.org/citation.cfm?id=110289.110295>. – ISSN 0004-3702
- [Chen und Nugent 2009] CHEN, Liming ; NUGENT, Chris: Ontology-based activity recognition in intelligent pervasive environments. In: *International Journal of Web Information Systems* 5 (2009), Nr. 4, S. 410–430. – URL <http://dx.doi.org/10.1108/17440080911006199>. – ISSN 1744-0084
- [Craig 1998] CRAIG, Edward: *Routledge Encyclopedia of Philosophy*. New York and London : Routledge, 1998
- [Crockford 2006] CROCKFORD, D.: *RfC: 4627*. <http://www.ietf.org>. 06 2006. – URL <http://tools.ietf.org/html/rfc4627>
- [Dai und Xu 2008] DAI, Peng ; XU, Guangyou: Context-aware computing for assistive meeting system. In: *PETRA '08: Proceedings of the 1st international conference on Pervasive Technologies Related to Assistive Environments*. New York, NY, USA : ACM, 2008, S. 1–7. – ISBN 978-1-60558-067-8
- [Dreschke 2011] DRESCHKE, Oliver: *Entwicklung kontextsensitiver Mobiler intelligente Wohnumgebungen*, HAW Hamburg, Diplomarbeit, 2011. – URL <http://users.informatik.haw-hamburg.de/~ubicomp/arbeiten/master/dreschke.pdf>
- [Ellenberg 2010a] ELLENBERG, Jens: *Vorarbeiten für den Wecker 2.0* / HAW Hamburg. URL <http://users.informatik.haw-hamburg.de/~ubicomp/projekte/master2010-proj1/ellenberg.pdf>, 2010. – Forschungsbericht
- [Ellenberg 2010b] ELLENBERG, Jens: *Ein Wecker in einem ubicom Haus* / HAW Hamburg. HAW Hamburg, 2010. – Forschungsbericht. – URL <http://users.informatik.haw-hamburg.de/~ubicomp/projekte/master09-10-aw1/Ellenberg/bericht.pdf>

- [Ellenberg 2011] ELLENBERG, Jens: Entwicklung des Wecker 2.0 / HAW Hamburg. URL <http://users.informatik.haw-hamburg.de/~ubicomp/projekte/master10-11-proj2/ellenberg.pdf>, 2011. – Forschungsbericht
- [Ellenberg u. a. 2011] ELLENBERG, Jens ; KARSTAEDT, Bastian ; VOSKUHL, Sören ; LUCK, Kai von ; WENDHOLT, Birgit: An Environment for Context-Aware Applications in Smart Homes / Hamburg University of Applied Sciences. URL <http://users.informatik.haw-hamburg.de/~ubicomp/arbeiten/papers/IPIN2011.pdf>, 2011. – Forschungsbericht
- [Erman u. a. 1980] ERMAN, Lee D. ; HAYES-ROTH, Frederick ; LESSER, Victor R. ; REDDY, D. R.: The Hearsay-II Speech-Understanding System: Integrating Knowledge to Resolve Uncertainty. In: *ACM Comput. Surv.* 12 (1980), June, S. 213–253. – URL <http://doi.acm.org/10.1145/356810.356816>. – ISSN 0360-0300
- [Fayyad u. a. 1996] FAYYAD, Usama ; PIATETSKY-SHAPIRO, Gregory ; SMYTH, Padhraic: From data mining to knowledge discovery in databases. In: *AI Magazine* 17 (1996), S. 37–54
- [Fillmore 1968] FILLMORE, Charles: The Case for Case. In: BACH, Emmon (Hrsg.) ; HARMS, R. (Hrsg.): *Universals in Linguistic Theory*. New York : Holt, Rinehart, and Winston, 1968
- [Gómez-Romero u. a.] GÓMEZ-ROMERO, Juan ; SERRANO, Miguel ; PATRICIO, Miguel ; GARCÍA, Jesús ; MOLINA, José: Context-based scene recognition from visual data in smart homes: an Information Fusion approach. In: *Personal and Ubiquitous Computing*. – 10.1007/s00779-011-0450-9. – ISSN 1617-4909
- [Greenberg 2001] GREENBERG, Saul: Context as a dynamic construct. In: *Hum.-Comput. Interact.* 16 (2001), Nr. 2, S. 257–268. – ISSN 0737-0024
- [Grimm u. a. 2007] GRIMM, Stephan ; HITZLER, Pascal ; ABECKER, Andreas: Knowledge Representation and Ontologies. In: STUDER, Rudi (Hrsg.) ; GRIMM, Stephan (Hrsg.) ; ABECKER, Andreas (Hrsg.): *Semantic Web Services: Concepts, Technology and Applications*. Springer, Berlin–Heidelberg, Germany, 2007, S. 51–106. – URL <http://www.aifb.uni-karlsruhe.de/WBS/phi/resources/publications/kr-onto-07.pdf>
- [Gruber 1993] GRUBER, Thomas R.: A translation approach to portable ontology specifications. In: *Knowl. Acquis.* 5 (1993), June, S. 199–220. – URL <http://dl.acm.org/citation.cfm?id=173743.173747>. – ISSN 1042-8143
- [Hardenack 2011] HARDENACK, Frank: - *Das intelligente Bett - Sensorbasierte Detektion von Schlafphasen*, HAW Hamburg, Diplomarbeit, 2011. – URL

- <http://users.informatik.haw-hamburg.de/~ubicomp/arbeiten/master/hardenack.pdf>
- [Johannsen 2011] JOHANNSEN, Benedikt: Generische Modelbasierte Kommunikationsinfrastruktur / HAW Hamburg. URL <http://users.informatik.haw-hamburg.de/~ubicomp/projekte/master10-11-seminar/johannsen/bericht.pdf>, 2011. – Forschungsbericht
- [Karstaedt 2011] KARSTAEDT, Bastian: Entwicklung und Integration des Indoor Spatial Information Services in den Living Place Hamburg / HAW-Hamburg. 2011. – Report
- [Karstaedt 2012] KARSTAEDT, Bastian: *Kontextinterpretation in Smart Home auf Basis 3D semantischer Gebäudemodelle*, HAW Hamburg, Diplomarbeit, 2012. – URL <http://users.informatik.haw-hamburg.de/~ubicomp/papers.html>
- [Karstaedt und Wendholt 2011] KARSTAEDT, Bastian ; WENDHOLT, Birgit: Towards semantic interpretations of spatial information in adaptive smart homes. In: *Forum Bauinformatik 2011, Cork, Ireland 1* (2011), August, S. 8. – URL <http://zuse.ucc.ie/forumbau2011/papers/16.pdf>
- [Katz und Akpom 1976] KATZ, S. ; AKPOM, C.A.: A measure of primary sociobiological functions. In: *International Journal of Health Services* 6 (1976), Nr. 3, S. 493–508
- [Kipling 1902] KIPLING, Rudyard ; PUBLISHERS, Macmillan (Hrsg.): *Just-so story*. Macmillan Publishers, 1902. – URL <http://www.gutenberg.org/ebooks/2781.html#fb>
- [Leake u. a. 2006] LEAKE, David ; MAGUITMAN, Ana ; REICHERZER, Thomas: Cases, Context, and Comfort: Opportunities for Case-Based Reasoning in Smart Homes. In: AUGUSTO, Juan (Hrsg.) ; NUGENT, Chris (Hrsg.): *Designing Smart Homes* Bd. 4008. Berlin, Heidelberg : Springer Berlin / Heidelberg, 2006, Kap. 7, S. 109–131. – ISBN 978-3-540-35994-4
- [von Luck u. a. 2010] LUCK, Prof. Dr. K. von ; KLEMKE, Prof. Dr. G. ; GREGOR, Sebastian ; RAHIMI, Mohammad A. ; VOGT, Matthias: Living Place Hamburg – A place for concepts of IT based modern living / Hamburg University of Applied Sciences. URL http://livingplace.informatik.haw-hamburg.de/content/LivingPlaceHamburg_en.pdf, Mai 2010. – Forschungsbericht
- [Motik u. a. 2004] MOTIK, Boris ; SATTLER, Ulrike ; STUDER, Rudi: Query Answering for OWL-DL with Rules. In: *International Semantic Web Conference*, 2004, S. 549–563
- [Najem 2011] NAJEM, Hosnia: Modellbasiertes Suchen von Objekten in einer Smart-Home-Umgebung / HAW Hamburg. URL <http://users.informatik.haw-hamburg.de/~ubicomp/arbeiten/master/hardenack.pdf>

- haw-hamburg.de/~ubicomp/projekte/master2011-proj1/najem.pdf, 2011. – Forschungsbericht
- [Newell 1982] NEWELL, A.: The knowledge level. In: *Artificial intelligence* 18 (1982), Nr. 1, S. 87–127
- [Otto und Voskuhl 2010] OTTO, Kjell ; VOSKUHL, Sören: Entwicklung einer Architektur für das Living Place Hamburg / HAW Hamburg. HAW Hamburg, 2010. – Forschungsbericht. – URL <http://users.informatik.haw-hamburg.de/~ubicomp/projekte/master2010-proj/otto-voskuhl.pdf>
- [Otto und Voskuhl 2011] OTTO, Kjell ; VOSKUHL, Sören: Weiterentwicklung der Architektur des Living Place Hamburg / HAW Hamburg. URL <http://users.informatik.haw-hamburg.de/~ubicomp/projekte/master10-11-proj2/otto-voskuhl.pdf>, 2011. – Forschungsbericht
- [Pautz 2010] PAUTZ, Alexander: Kabelloses Sensornetzwerk im Living Place Hamburg / HAW Hamburg. URL <http://users.informatik.haw-hamburg.de/~ubicomp/projekte/master10-11-seminar/pautz/bericht.pdf>, 2010. – Forschungsbericht
- [Phyu 2009] PHYU, T.N.: Survey of classification techniques in data mining. In: *Proceedings of the International MultiConference of Engineers and Computer Scientists 1* (2009), S. 18–20
- [Polenz 1988] POLENZ, P.: *Deutsche Satzsemantik: Grundbegriffe des Zwischen-den-Zeilen-Lesens*. de Gruyter, 1988 (Sammlung G öschen). – URL <http://books.google.com/books?id=mT3EQgAACAAJ>. – ISBN 9783110119558
- [Rahimi und Vogt 2011] RAHIMI, Mohammad A. ; VOGT, Matthias: *Seamless Interaction - Natürliche Interaktionen in Smart Living Umgebungen*, HAW Hamburg, Diplomarbeit, 02 2011. – URL <http://users.informatik.haw-hamburg.de/~ubicomp/arbeiten/master/rahimi-vogt.pdf>
- [Russell und Norvig 2004] RUSSELL, Stuart ; NORVIG, Peter: *Künstliche Intelligenz. Ein moderner Ansatz*. Bd. 2. Pearson Studium, 2004. – ISBN 3-8273-7089-2
- [Scheidt-Nave u. a. 2010] SCHEIDT-NAVE, C. ; RICHTER, S. ; FUCHS, J. ; KUHLMAY, A.: Herausforderungen an die Gesundheitsforschung für eine alternde Gesellschaft am Beispiel „Multimorbidität“. In: *Bundesgesundheitsblatt - Gesundheitsforschung - Gesundheitsschutz* 53 (2010), S. 441–450. – URL <http://dx.doi.org/10.1007/s00103-010-1052-9>. – 10.1007/s00103-010-1052-9. – ISSN 1436-9990

- [Sebastiani 2005] SEBASTIANI, Fabrizio: Text categorization. In: *Text Mining and its Applications to Intelligence, CRM and Knowledge Management*, WIT Press, 2005, S. 109–129
- [Simpson u. a. 2006] SIMPSON, Richard ; SCHRECKENGHOST, Debra ; LOPRESTI, Edmund ; KIRSCH, Ned: Plans and Planning in Smart Homes. In: AUGUSTO, Juan (Hrsg.) ; NUGENT, Chris (Hrsg.): *Designing Smart Homes* Bd. 4008. Springer Berlin / Heidelberg, 2006, S. 71–84. – ISBN 978-3-540-35994-4
- [Sung u. a. 2011] SUNG, Jaeyong ; PONCE, Colin ; SELMAN, Bart ; SAXENA, Ashutosh: Human Activity Detection from RGBD Images. In: *CoRR* abs/1107.0169 (2011)
- [Tennstedt 2009] TENNSTEDT, Sven: Projektbericht: Emotional Tent Ambient Awareness / HAW Hamburg. URL <http://users.informatik.haw-hamburg.de/~ubicomp/projekte/master08-09-proj/tennstedt/bericht.pdf>, 2009. – Forschungsbericht
- [Teske 2011] TESKE, Philipp: Human Fall Detection / HAW Hamburg. URL <http://users.informatik.haw-hamburg.de/~ubicomp/projekte/master10-11-proj2/teske.pdf>, 2011. – Forschungsbericht
- [Verma u. a. 2010] VERMA, Ritu ; AGRAWAL, Anupam ; SHARMA, Shanu: A Robust & Fast Face Detection System. In: PROF. VINU V DAS CEO, Engineers N. (Hrsg.): *Advances in Computer Science ACS 2010*, ACEEE, 2010, S. 167–172
- [Vila 1994] VILA, Lluís: A Survey on Temporal Reasoning in Artificial Intelligence. In: *AI Commun.* 7 (1994), Nr. 1, S. 4–28
- [Voskuhl 2012] VOSKUHLE, Sören: *Modellunabhängige Kontextinterpretation in einer Smart Home Umgebung*, HAW Hamburg, Diplomarbeit, Jan 2012. – URL <http://users.informatik.haw-hamburg.de/~ubicomp/papers.html>
- [Wand und Weber 1990] WAND, Y. ; WEBER, R.: An ontological model of an information system. In: *Software Engineering, IEEE Transactions on* 16 (1990), nov, Nr. 11, S. 1282–1292. – ISSN 0098-5589
- [Weiser 1999] WEISER, Mark: The computer for the 21st century. In: *SIGMOBILE Mob. Comput. Commun. Rev.* 3 (1999), Nr. 3, S. 3–11. – ISSN 1559-1662
- [Winkler 2007] WINKLER, Ronny: *Entwicklung eines ontologiebasierten Kontextmodells für kollaborative Web-Anwendungen*, Technischen Universität Dresden, Diplomarbeit, 2007
- [Yamazaki 2007] YAMAZAKI, Tatsuya: The ubiquitous home. In: *International Journal* 1 (2007), Nr. 1, S. 17–22. – URL http://www.sersc.org/journals/IJSH/vol1_no1_2007/IJSH-2007-01-01-03.pdf

Anhang

A. Definition der States und Events

A.1. States (Aktivitäten)

Aktivität „Schlafen“

Ontologie: ResidentActionSleep

Kurze Beschreibung: Bewohner liegt im Bett und schläft.

Detaillierte Beschreibung:

- Wo: Nutzungsbereich des Bettes.
- Was: Das Bett
- Wann: Nachts
- Wie: Die Position während des Schlafens im Bett ist egal. Der schlafende Zustand beinhaltet die jede Form der Schlafphase (Tiefschlaf / Leichtschlaf / REM). Bewegungen sind während der Aktivität möglich, wenn der Bereich des Bettes nicht verlassen wird.

Aktivität „Waschen“

Ontologie: ResidentActionWasching

Kurze Beschreibung: Der Bewohner säubert sich am Waschbecken im Bad.

Detaillierte Beschreibung:

- Wo: im Badezimmer
- Was: Waschbecken
- Wann: jederzeit
- Wie: Der Bewohner nutzt das Waschbecken um sich zu erfrischen, Zähne zu putzen oder seine Hände zu waschen.

Aktivität „Duschen“

Ontologie: ResidentActionTakingShower

Kurze Beschreibung: Der Bewohner säubert sich in der Dusche.

Detaillierte Beschreibung:

- Wo: Bad
- Was: Dusche
- Wann: jederzeit
- Wie: Der Bewohner nutzt die Dusche um sich waschen und zu säubern. Hierzu zählt das waschen des Körpers oder der Haare. Das Anziehen oder ausziehen der Kleidung ist nicht Bestandteil dieser Aktivität.

Aktivität „Frühstück“

Ontologie: ResidentActionBreakfast

Kurze Beschreibung: Bewohner nimmt sein Frühstück ein

Detaillierte Beschreibung:

- Wo: In Küche/Essbereich
- Was: Essbesteck
- Wann: Morgens
- Wie: Mit Teller und Besteck werden die Nahrungsmittel aufgenommen. Dabei sitzt der Bewohner am Tisch oder steht an einem Küchentresen. Er muss für diese Aktivität nicht durchgehend essen, sondern kann dabei auch die Neuigkeiten des Tages aus dem Internet abrufen (über Küchentresen) oder andere Aktivitäten durchführen. z.B. Kaffee kochen oder weiteres essen zubereiten.

Aktivität „Mittagsessen“

Ontologie: ResidentActionLunch

Kurze Beschreibung: Bewohner nimmt sein Mittagessen ein

Detaillierte Beschreibung:

- Wo: In Küche/Essbereich
- Was: Teller und Besteck
- Wann: Mittags.

- Wie: Mit Teller und Besteck werden die Nahrungsmittel aufgenommen. Dabei sitzt der Bewohner am Tisch oder steht an einem Küchentresen. Er muss für diese Aktivität nicht durchgehend essen, sondern kann dabei auch die Neuigkeiten des Tages aus dem Internet abrufen (über Küchentresen) oder andere Aktivitäten durchführen.

Aktivität „Abendessen“**Ontologie:** keine**Kurze Beschreibung:** Bewohner nimmt sein Abendessen ein**Detaillierte Beschreibung:**

- Wo: In Küche/Essbereich
- Was: Teller und Besteck
- Wann: Abends
- Wie: Mit Teller und Besteck werden die Nahrungsmittel aufgenommen. Dabei sitzt der Bewohner am Tisch oder steht an einem Küchentresen. Er muss für diese Aktivität nicht durchgehend essen, sondern kann dabei auch die Neuigkeiten des Tages aus dem Internet abrufen (über Küchentresen) oder andere Aktivitäten durchführen.

Aktivität „KleineMahlzeit“**Ontologie:** ResidentActionSmallMeal**Kurze Beschreibung:** Bewohner isst etwas zwischen den Hauptmahlzeiten**Detaillierte Beschreibung:**

- Wo: In Küche/Essbereich
- Was: Teller und Besteck
- Wann: jederzeit
- Wie: Mit Teller und Besteck werden die Nahrungsmittel aufgenommen. Dabei sitzt der Bewohner am Tisch oder steht an einem Küchentresen. Er muss für diese Aktivität nicht durchgehend essen, sondern kann dabei auch die Neuigkeiten des Tages aus dem Internet abrufen (über Küchentresen) oder andere Aktivitäten durchführen.

Aktivität „Fehrnsehen“**Ontologie:** ResidentActionWatchingTv

Kurze Beschreibung: Der Bewohner schaut sich einen Film oder ein TV Programm an.

Detaillierte Beschreibung:

- Wo: Im Sichtbereich zu einem Bildschirm
- Was: Bildschirm
- Wann: jederzeit
- Wie: Der Bewohner schaut sich einen Film oder ein TV Programm auf einem der Bildschirme in der Wohnung an. Bei dieser Aktivität ist die Position des Bewohners egal (z.B. stehen, liegend oder sitzend).

Aktivität „Arbeiten“

Ontologie: ResidentActionWorking

Kurze Beschreibung: Der Bewohner arbeitet seinem Firmen-Laptop.

Detaillierte Beschreibung:

- Wo: Im Nutzungsbereich des Laptop
- Was: Laptop
- Wann: jederzeit
- Wie: Der Bewohner nutzt seinen Laptop der Wohnung um von zu Hause aus seiner Arbeit nach zu gehen. Dieses kann in der gesamten Wohnung in jeder Position geschehen, da das Gerät durch die Wohnung getragen werden kann. Der Bewohner ist aber für diese Aktivität nie lange vom Laptop weg.

Aktivität „außerhalb des Living Place“

Ontologie: ResidentActionOutsideLivingPlace

Kurze Beschreibung: Der Bewohner ist nicht da.

Detaillierte Beschreibung:

- Wo: außerhalb der Wohnung
- Was: nichts
- Wann: immer
- Wie: Der Bewohner ist nicht in der Wohnung.

A.2. Komplexe States (Aktivitäten)

Morgen-Aktivitäten

Ontologie: ResidentActionMorningActivity

Kurze Beschreibung: Der Bewohner bereitet sich auf der Verlassen der Wohnung vor.

Detaillierte Beschreibung:

- Wo: in der gesamten Wohnung
- Was: verschiedenes
- Wann: Morgens
- Wie: Nach dem Aufstehen verrichtet der Bewohner verschiedene Tätigkeiten, bevor er die Wohnung verlässt. Die Aktivität beginnt mit dem Schlafenden Bewohner. Nach dem Schlafen folgen die Aktivitäten Duschen und Frühstück. Weitere Einzelheiten sind bei diesen Aktivitäten beschrieben.

A.3. Events

A.3.1. Zone betreten

- Bewohner betritt das Living Place
- Bewohner betritt den Nutzungsbereich des Bettes
- Bewohner betritt das Badezimmer
- Bewohner betritt den Nutzungsbereich des Küchentresens
- Bewohner betritt den Nutzungsbereich des Laptop
- Bewohner betritt den Sichtbereich des Fernsehers

A.3.2. Zone verlassen

- Bewohner verlässt das Living Place
- Bewohner verlässt den Nutzungsbereich des Bettes
- Bewohner verlässt das Badezimmer
- Bewohner verlässt den Nutzungsbereich des Küchentresens
- Bewohner verlässt den Nutzungsbereich des Laptop
- Bewohner verlässt den Sichtbereich des Fernsehers

A.3.3. Objekt aktivieren

- Bett wird benutzt
- Dusche wird benutzt
- Waschbecken wird benutzt
- Ess-Geschirr steht zur Nutzung bereit
- Laptop wird benutzt
- TV-Gerät ist eingeschaltet

A.3.4. Objekt deaktivieren

- Bett wird nicht benutzt
- Dusche wird nicht benutzt
- Waschbecken wird nicht benutzt
- Ess-Geschirr steht nicht zur Nutzung bereit
- Laptop wird nicht benutzt
- TV-Gerät ist ausgeschaltet

A.4. Komplexe Events

A.4.1. Start Events

- Start Schlafen
- Start Duschen
- Start Waschen
- Start Frühstück
- Start Mittagessen
- Start Abendessen
- Start kleine Mahlzeit einnehmen
- Start Arbeiten
- Start TV schauen

A.4.2. End Events

- Ende Schlafen
- Ende Duschen
- Ende Waschen
- Ende Frühstück
- Ende Mittagessen
- Ende Abendessen
- Ende kleine Mahlzeit einnehmen
- Ende Arbeiten
- Ende TV schauen

B. Ontologie Code

Im Folgenden sind beispielhafte Ausschnitte aus der Ontologie aufgelistet. Die gesamte Ontologie hat über 15.000 Zeilen und passt daher nicht in diese Arbeit.

B.1. Header

Ontologie Header

```
1 <?xml version="1.0"?>
2 <rdf:RDF
3   xmlns:livingPlace=
4   "http://livingplace.informatik.haw-hamburg.de/livingplaceOntologie.owl#"
5   xmlns:protege="http://protege.stanford.edu/plugins/owl/protege#"
6   xmlns:xsp="http://www.owl-ontologies.com/2005/08/07/xsp.owl#"
7   xmlns:domain="http://purl.org/net/cofriend/apron#"
8   xmlns:swrlb="http://www.w3.org/2003/11/swrlb#"
9   xmlns:temporal=
10  "http://swrl.stanford.edu/ontologies/built-ins/3.3/temporal.owl#"
11  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
12  xmlns:upper="http://purl.org/net/cofriend/upper#"
13  xmlns:owl="http://www.w3.org/2002/07/owl#"
14  xmlns:sqwrl="http://sqwrl.stanford.edu/ontologies/built-ins/3.4/sqwrl.owl#"
15  xmlns:v="http://purl.org/net/cofriend/variables#"
16  xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
17  xmlns:swrl="http://www.w3.org/2003/11/swrl#"
18  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
19  xmlns:swrla="http://swrl.stanford.edu/ontologies/3.3/swrla.owl#"
20  xml:base=
21  "http://livingplace.informatik.haw-hamburg.de/livingplaceOntologie.owl">
22  <owl:Ontology rdf:about="">
23  <owl:imports rdf:resource="http://purl.org/net/cofriend/constraints.owl"/>
24  <owl:imports rdf:resource=
25  "http://swrl.stanford.edu/ontologies/built-ins/3.3/temporal.owl"/>
26  <owl:imports rdf:resource="http://swrl.stanford.edu/ontologies/3.3/swrla.owl"/>
27  <owl:imports rdf:resource=
28  "http://sqwrl.stanford.edu/ontologies/built-ins/3.4/sqwrl.owl"/>
29  </owl:Ontology>
```

B.2. Klasse Bewohner

Ontologie-Klasse Resident

```
1 <owl:Class rdf:about="#Resident">
2   <rdfs:subClassOf>
3     <owl:Class rdf:about="http://purl.org/net/cofriend/upper#Person"/>
4   </rdfs:subClassOf>
5 </owl:Class>
```

B.3. Klasse Bett

Ontologie-Klasse Bed

```
1 <owl:Class rdf:about="#Bed">
2   <rdfs:subClassOf>
3     <owl:Class rdf:about="#Furniture"/>
4   </rdfs:subClassOf>
5 </owl:Class>
```

B.4. Klasse Nutzungsbereich des Bettes

Ontologie-Klasse ResidentFunctionalSpaceBed

```
1 <owl:Class rdf:about="#FunctionalSpaceBed">
2   <rdfs:subClassOf>
3     <owl:Class rdf:about=
4     "http://purl.org/net/cofriend/upper#FunctionalSpace"/>
5   </rdfs:subClassOf>
6 </owl:Class>
```

B.5. State Schlafen

Ontologie Klasse ResidentActionSleep

```

1 <owl:Class rdf:ID="ResidentActionSleep">
2   <rdfs:subClassOf>
3     <owl:Restriction>
4       <owl:cardinality rdf:datatype="http://www.w3.org/2001/XMLSchema#int"
5         >1</owl:cardinality>
6       <owl:onProperty>
7         <owl:ObjectProperty rdf:about=
8           "http://purl.org/net/cofriend/upper#has-part-B"/>
9         </owl:onProperty>
10      <owl:valuesFrom>
11        <owl:Class rdf:about="#ResidentActionEndSleep"/>
12      </owl:valuesFrom>
13    </owl:Restriction>
14  </rdfs:subClassOf>
15  <rdfs:subClassOf>
16    <owl:Restriction>
17      <owl:onProperty>
18        <owl:ObjectProperty rdf:about=
19          "http://purl.org/net/cofriend/upper#has-part-A"/>
20        </owl:onProperty>
21      <owl:cardinality rdf:datatype="http://www.w3.org/2001/XMLSchema#int"
22        >1</owl:cardinality>
23      <owl:valuesFrom>
24        <owl:Class rdf:ID="ResidentActionStartSleep"/>
25      </owl:valuesFrom>
26    </owl:Restriction>
27  </rdfs:subClassOf>
28  <rdfs:subClassOf>
29    <owl:Restriction>
30      <owl:cardinality rdf:datatype="http://www.w3.org/2001/XMLSchema#int"
31        >1</owl:cardinality>
32      <owl:valuesFrom>
33        <owl:Class rdf:about="#ResidentActionEndSleep"/>
34      </owl:valuesFrom>
35      <owl:onProperty rdf:resource=
36        "http://purl.org/net/cofriend/upper#has-part-B"/>
37      </owl:Restriction>
38  </rdfs:subClassOf>
39  <rdfs:subClassOf>
40    <owl:Class rdf:ID="ResidentAction"/>
41  </rdfs:subClassOf>
42 </owl:Class>

```

B.6. SWRL Regeln in Protege

Start-Event Schlafen erst die Zone dann das Objekt

```
1 livingPlace:ResidentActionStartSleepZoneObject(?startSleep) ∧  
2 upper:has-agent(?startSleep, ?a1) ∧  
3 upper:has-location(?startSleep, ?l1) ∧  
4 upper:has-object(?startSleep, ?o1) ∧  
5 upper:has-part-A(?startSleep, ?eventObject) ∧  
6 upper:has-object(?eventObject, ?o2) ∧  
7 upper:has-part-B(?startSleep, ?eventZone) ∧  
8 upper:has-agent(?eventZone, ?a2) ∧  
9 upper:has-location(?eventZone, ?l2)  
10 →  
11 constraints:has-start-time(?eventObject, ?objectTimeSt) ∧  
12 constraints:has-start-time(?eventZone, ?enterTimeSt) ∧  
13 swrlb:equal(?a1, ?a2) ∧  
14 swrlb:equal(?l1, ?l2) ∧  
15 swrlb:equal(?o1, ?o2) ∧  
16 constraints:min-before(?enterTimeSt, ?objectTimeSt, 0) ∧  
17 constraints:same-start(?startSleep, ?eventObject) ∧  
18 constraints:same-finish(?startSleep, ?eventObject)
```

Start-Event Schlafen erst das Objekt dann die Zone

```
1 livingPlace:ResidentActionStartSleepObjectZone( startSleep ) ∧  
2 upper:has-part-A(? startSleep , ?eventObject) ∧  
3 upper:has-part-B(? startSleep , ?eventZone) ∧  
4 constraints:has-start-time(?eventObject , ?objectTimeSt) ∧  
5 constraints:has-start-time(?eventZone , ?enterTimeSt) ∧  
6 upper:has-agent(?startSleep , ?a1) ∧  
7 upper:has-location(?startSleep , ?l1) ∧  
8 upper:has-agent(?eventZone , ?a2) ∧  
9 upper:has-location(?eventZone , ?l2)  
10 →  
11 swrlb:equal(?l1 , ?l2) ∧  
12 swrlb:equal(?a1 , ?a2) ∧  
13 constraints:min-before(?objectTimeSt , ?enterTimeSt , 0) ∧  
14 constraints:same-start(?startSleep , ?eventZone) ∧  
15 constraints:same-finish(?startSleep , ?eventZone)
```

Aktivität Schlafen mit dem Ende durch ein Objekt

```
1 livingPlace:ResidentActionSleepEndObject(?sleep) ∧  
2 upper:has-part-A(?sleep , ?startEvent) ∧  
3 upper:has-object(?startEvent , ?o1) ∧  
4 upper:has-part-B(?sleep , ?endEvent) ∧  
5 upper:has-object(?endEvent , ?o2)  
6 →  
7 swrlb:equal(?o1 , ?o2) ∧  
8 constraints:same-start(?sleep , ?startEvent) ∧  
9 constraints:same-finish(?sleep , ?endEvent)
```

Aktivität Schlafen mit dem Ende durch eine Zone

```

1 livingPlace:ResidentActionSleepEndZone(?sleep) ∧
2 upper:has-part-A(?sleep, ?startEvent) ∧
3 upper:has-location(?startEvent, ?l1) ∧
4 upper:has-agent(?startEvent, ?a1) ∧
5 upper:has-part-B(?sleep, ?endEvent) ∧
6 upper:has-location(?endEvent, ?l2) ∧
7 upper:has-agent(?endEvent, ?a2) ∧
8 constraints:has-finish-time(?startEvent, ?startEvent-ft) ∧
9 upper:has-part-B(?sleep, ?endEvent)
10 →
11 constraints:has-start-time(?endEvent, ?endEvent-st) ∧
12 swrlb:equal(?l1, ?l2) ∧
13 swrlb:equal(?a1, ?a2) ∧
14 constraints:min-before(?startEvent-ft, ?endEvent-st, 0) ∧
15 constraints:same-start(?sleep, ?startEvent) ∧
16 constraints:same-finish(?sleep, ?endEvent)

```

End-Event Schlafen durch ein Objekt ausgelöst

```

1 livingPlace:ResidentActionEndSleepDeactivateObject(?endSleep) ∧
2 upper:has-part(?endSleep, ?endObject) ∧
3 upper:has-object(?endSleep, ?o1) ∧
4 upper:has-object(?endObject, ?o2)
5 →
6 swrlb:equal(?o1, ?o2) ∧
7 constraints:same-start(?endSleep, ?endObject) ∧
8 constraints:same-finish(?endSleep, ?endObject)

```

End-Event Schlafen durch eine Zone ausgelöst

```

1 livingPlace:ResidentActionEndSleepLeavingZone(?endSleep) ∧
2 upper:has-part(?endSleep, ?endZone) ∧
3 upper:has-agent(?endSleep, ?a1) ∧
4 upper:has-agent(?endZone, ?a2) ∧
5 upper:has-location(?endSleep, ?l1) ∧
6 upper:has-location(?endZone, ?l2)
7 →
8 swrlb:equal(?a1, ?a2) ∧
9 swrlb:equal(?l1, ?l2) ∧
10 constraints:same-start(?endSleep, ?endZone) ∧
11 constraints:same-finish(?endSleep, ?endZone)

```


B.7. SWRL Rule Aktivität Schlafen

Aktivität Schlafen mit dem Ende durch ein Objekt

```

1 <swrl:Imp rdf:about="livingPlace:ResidentActionSleepEndObject_Rule">
2   <swrl:head>
3     <swrl:AtomList>
4       <rdf:first>
5         <swrl:BuiltinAtom>
6           <swrl:arguments>
7             <rdf:List>
8               <rdf:first rdf:resource=
9                 "http://livingplace.informatik.haw-hamburg.de/o1"/>
10              <rdf:rest>
11                <rdf:List>
12                  <rdf:rest rdf:resource=
13                    "http://www.w3.org/1999/02/22-rdf-syntax-ns#nil"/>
14                  <rdf:first rdf:resource=
15                    "http://livingplace.informatik.haw-hamburg.de/o2"/>
16                </rdf:List>
17              </rdf:rest>
18            </rdf:List>
19          </swrl:arguments>
20          <swrl:builtin rdf:resource="http://www.w3.org/2003/11/swrlb#equal"/>
21        </swrl:BuiltinAtom>
22      </rdf:first>
23      <rdf:rest>
24        <swrl:AtomList>
25          <rdf:first>
26            <swrl:BuiltinAtom>
27              <swrl:builtin rdf:resource=
28                "http://purl.org/net/cofriend/constraints#same-start"/>
29              <swrl:arguments>
30                <rdf:List>
31                  <rdf:rest>
32                    <rdf:List>
33                      <rdf:first rdf:resource=
34                        "http://livingplace.informatik.haw-hamburg.de/startEvent"/>
35                      <rdf:rest rdf:resource=
36                        "http://www.w3.org/1999/02/22-rdf-syntax-ns#nil"/>
37                    </rdf:List>
38                  </rdf:rest>
39                  <rdf:first rdf:resource=
40                    "http://livingplace.informatik.haw-hamburg.de/sleep"/>
41                </rdf:List>
42              </swrl:arguments>
43            </swrl:BuiltinAtom>
44          </rdf:first>

```

```
45     <rdf:rest>
46       <swrl:AtomList>
47         <rdf:first>
48           <swrl:BuiltinAtom>
49             <swrl:arguments>
50               <rdf:List>
51                 <rdf:rest>
52                   <rdf:List>
53                     <rdf:rest rdf:resource=
54                       "http://www.w3.org/1999/02/22-rdf-syntax-ns#nil"/>
55                     <rdf:first rdf:resource=
56                       "http://livingplace.informatik.haw-hamburg.de/endEvent"/>
57                   </rdf:List>
58                 </rdf:rest>
59               <rdf:first rdf:resource=
60                 "http://livingplace.informatik.haw-hamburg.de/sleep"/>
61               </rdf:List>
62             </swrl:arguments>
63             <swrl:builtin rdf:resource=
64               "http://purl.org/net/cofriend/constraints#same-finish"/>
65             </swrl:BuiltinAtom>
66           </rdf:first>
67           <rdf:rest rdf:resource=
68             "http://www.w3.org/1999/02/22-rdf-syntax-ns#nil"/>
69         </swrl:AtomList>
70     </rdf:rest>
71 </swrl:AtomList>
72 </rdf:rest>
73 </swrl:AtomList>
74 </swrl:head>
75 <swrl:body>
76   <swrl:AtomList>
77     <rdf:rest>
78       <swrl:AtomList>
79         <rdf:first>
80           <swrl:IndividualPropertyAtom>
81             <swrl:argument1 rdf:resource=
82               "http://livingplace.informatik.haw-hamburg.de/sleep"/>
83             <swrl:argument2 rdf:resource=
84               "http://livingplace.informatik.haw-hamburg.de/startEvent"/>
85             <swrl:propertyPredicate rdf:resource=
86               "http://purl.org/net/cofriend/upper#has-part-A"/>
87           </swrl:IndividualPropertyAtom>
88         </rdf:first>
89         <rdf:rest>
90           <swrl:AtomList>
91             <rdf:first>
92               <swrl:IndividualPropertyAtom>
```

```
93         <swrl:propertyPredicate rdf:resource=  
94         "http://purl.org/net/cofriend/upper#has-object"/>  
95         <swrl:argument2 rdf:resource=  
96         "http://livingplace.informatik.haw-hamburg.de/o1"/>  
97         <swrl:argument1 rdf:resource=  
98         "http://livingplace.informatik.haw-hamburg.de/startEvent"/>  
99         </swrl:IndividualPropertyAtom>  
100    </rdf:first>  
101    <rdf:rest>  
102      <swrl:AtomList>  
103        <rdf:rest>  
104          <swrl:AtomList>  
105            <rdf:rest rdf:resource=  
106            "http://www.w3.org/1999/02/22-rdf-syntax-ns#nil"/>  
107            <rdf:first>  
108              <swrl:IndividualPropertyAtom>  
109                <swrl:propertyPredicate rdf:resource=  
110                "http://purl.org/net/cofriend/upper#has-object"/>  
111                <swrl:argument1 rdf:resource=  
112                "http://livingplace.informatik.haw-hamburg.de/endEvent"/>  
113                <swrl:argument2 rdf:resource=  
114                "http://livingplace.informatik.haw-hamburg.de/o2"/>  
115                </swrl:IndividualPropertyAtom>  
116              </rdf:first>  
117            </swrl:AtomList>  
118          </rdf:rest>  
119          <rdf:first>  
120            <swrl:IndividualPropertyAtom>  
121              <swrl:propertyPredicate rdf:resource=  
122              "http://purl.org/net/cofriend/upper#has-part-B"/>  
123              <swrl:argument1 rdf:resource=  
124              "http://livingplace.informatik.haw-hamburg.de/sleep"/>  
125              <swrl:argument2 rdf:resource=  
126              "http://livingplace.informatik.haw-hamburg.de/endEvent"/>  
127              </swrl:IndividualPropertyAtom>  
128            </rdf:first>  
129          </swrl:AtomList>  
130        </rdf:rest>  
131      </swrl:AtomList>  
132    </rdf:rest>  
133  </swrl:AtomList>  
134 </rdf:rest>  
135 <rdf:first>  
136   <swrl:ClassAtom>  
137     <swrl:classPredicate rdf:resource="#ResidentActionSleepEndObject"/>  
138     <swrl:argument1 rdf:resource=  
139     "http://livingplace.informatik.haw-hamburg.de/sleep"/>  
140   </swrl:ClassAtom>
```

```
141     </rdf:first>  
142     </swrl:AtomList>  
143 </swrl:body>  
144 </swrl:Imp>
```

C. Testlauf mit Scenior

C.1. Schlafen

Ergebnisse der Aktivität „Schlafen“ von Senior, die mit dem folgenden Script erzeugt wurden.

Schlafen Test Script

```
1 2011-11-05 08:00:00.00 - 2011-11-05 08:00:00.00 :  
2     ResidentEnterFunctionalSpaceBed  
3     (Resident 1, FunctionalSpaceBed 8)  
4  
5 2011-11-05 08:00:20.00 - 2011-11-05 08:00:20.00 :  
6     ActivateBed(Bed 2, null)  
7  
8 2011-11-05 08:00:40.00 - 2011-11-05 08:00:40.00 :  
9     DeactivateBed(Bed 2, null)  
10  
11 2011-11-05 08:01:00.00 - 2011-11-05 08:01:00.00 :  
12     ResidentLeaveFunctionalSpaceBed  
13     (Resident 1, FunctionalSpaceBed 8)
```

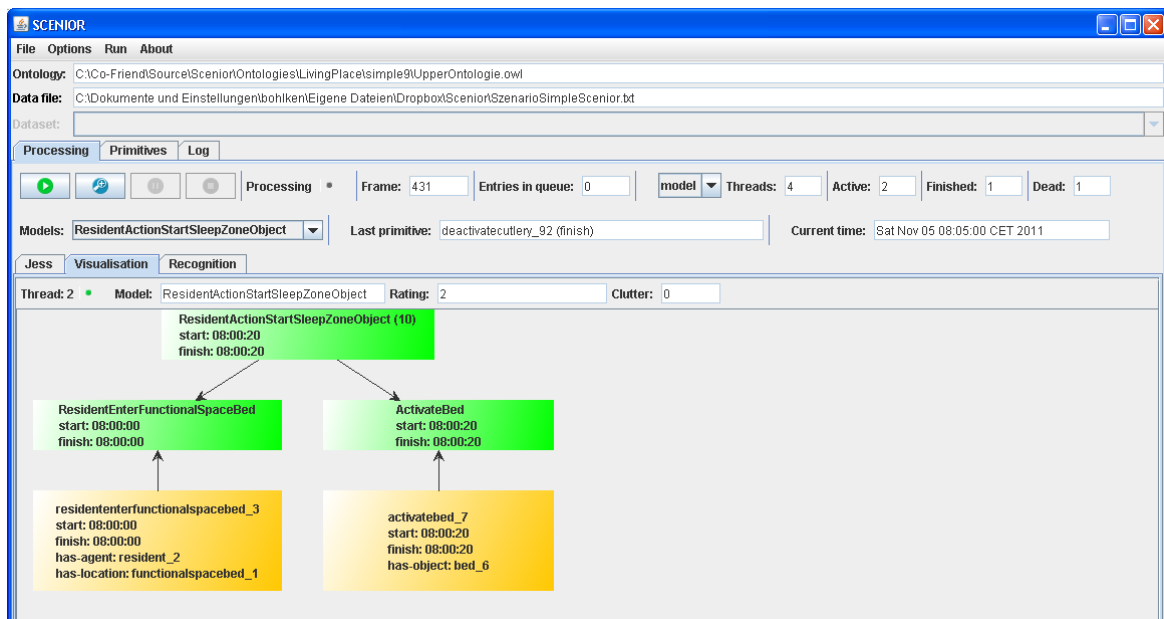


Abbildung C.1.: Schlafen Start-Event erst die Zone dann das Objekt

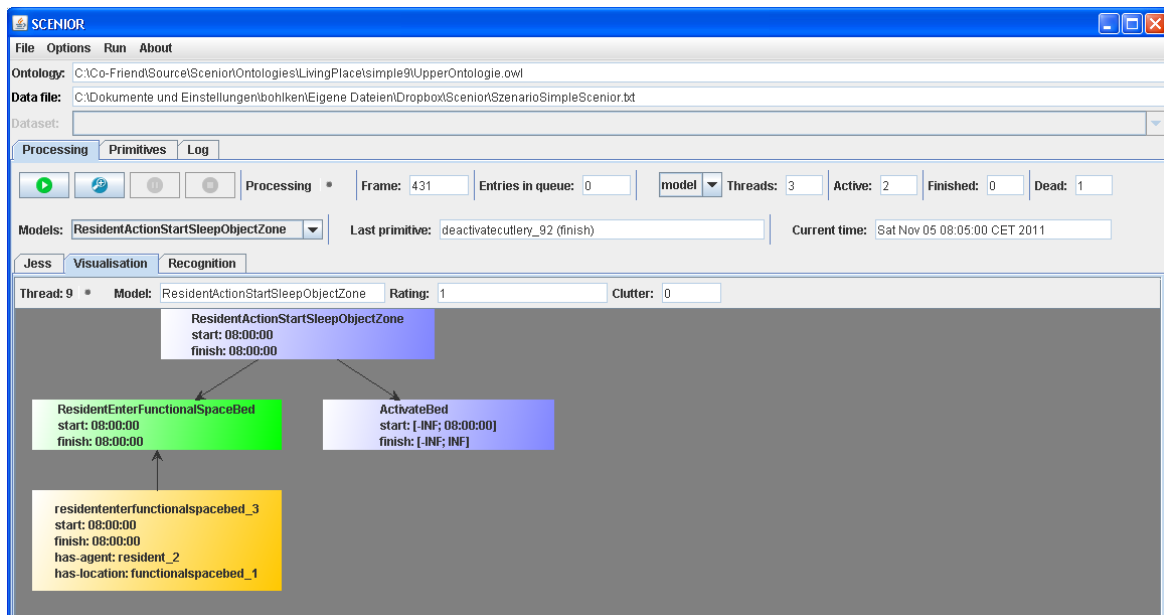


Abbildung C.2.: BEENDET: Schlafen Start-Event erst das Objekt dann die Zone

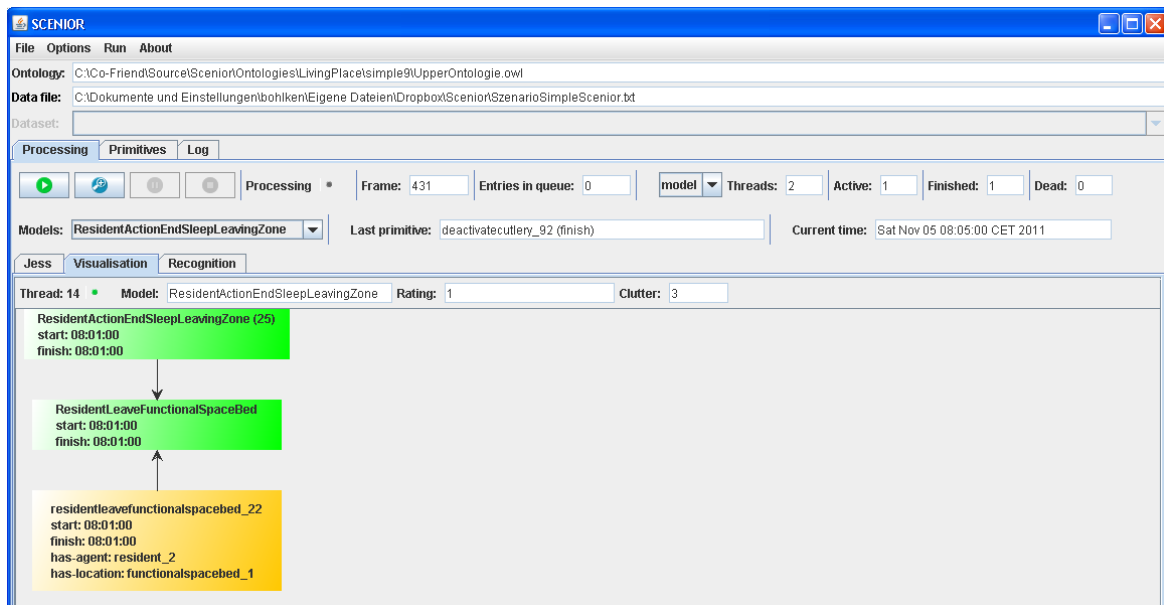


Abbildung C.3.: Schlafen End-Event ausgelöst durch die Zone

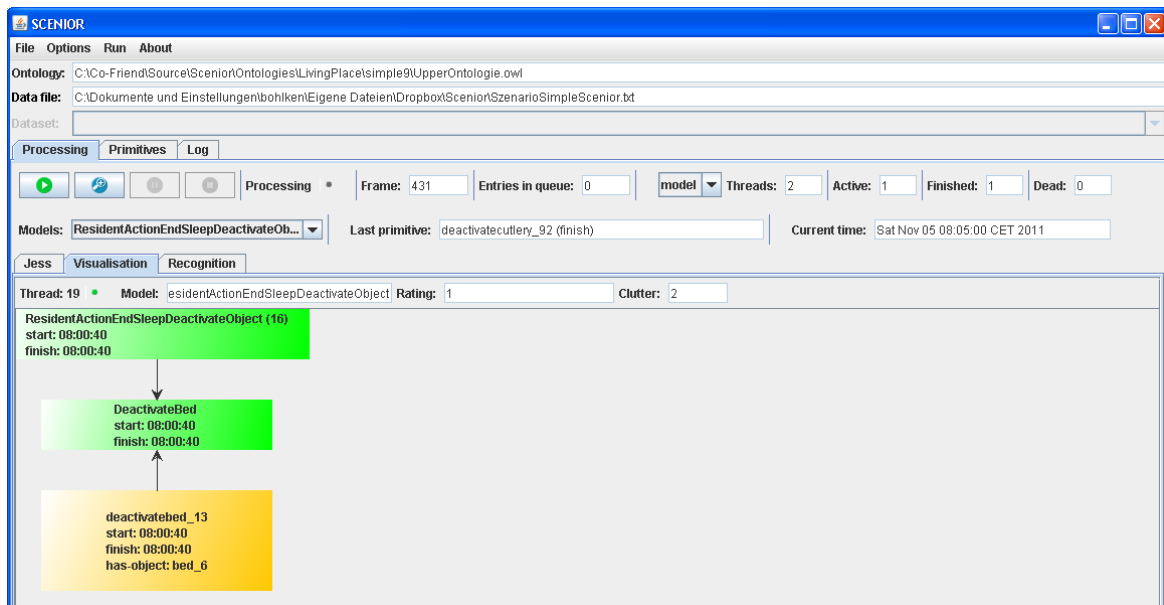


Abbildung C.4.: Schlafen End-Event ausgelöst durch das Objekt

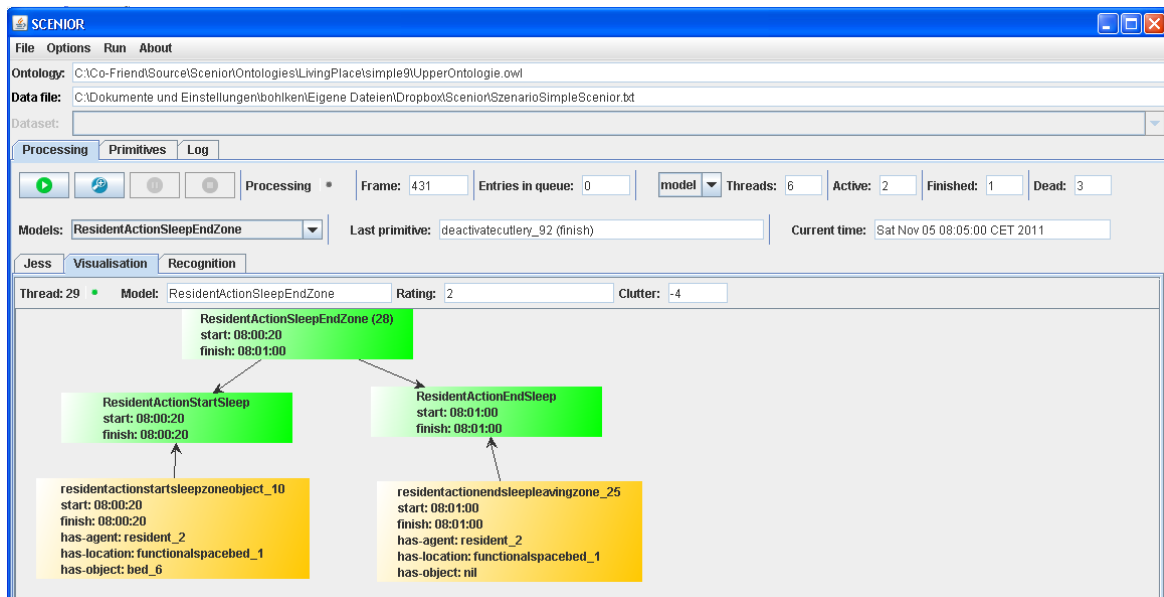


Abbildung C.5.: Schlafen State beendet durch die Zone

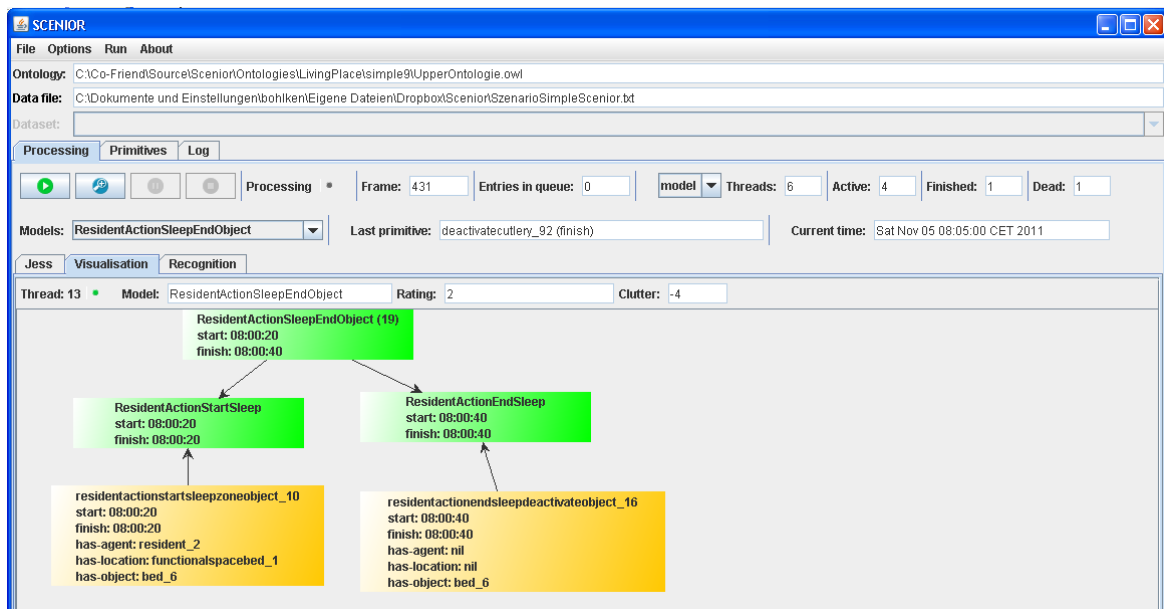


Abbildung C.6.: Schlafen State beendet durch das Objekt

C.2. Duschen

Ergebnisse der Aktivität „Duschen“ von Senior, die mit dem folgenden Script erzeugt wurden.

Duschen Test Script

```

1 2011-11-05 08:01:20.00 - 2011-11-05 08:01:20.00 :
2      ResidentEnterTraditionalSpaceBathroom
3      (Resident 1, TraditionalSpaceBathroom 9)
4
5 2011-11-05 08:02:20.00 - 2011-11-05 08:02:20.00 :
6      ActivateShower(Shower 4, null)
7
8 2011-11-05 08:02:40.00 - 2011-11-05 08:02:40.00 :
9      DeactivateShower(Shower 4, null)
10
11 2011-11-05 08:03:00.00 - 2011-11-05 08:03:00.00 :
12      ResidentLeaveTraditionalSpaceBathroom
13      (Resident 1, TraditionalSpaceBathroom 9)

```

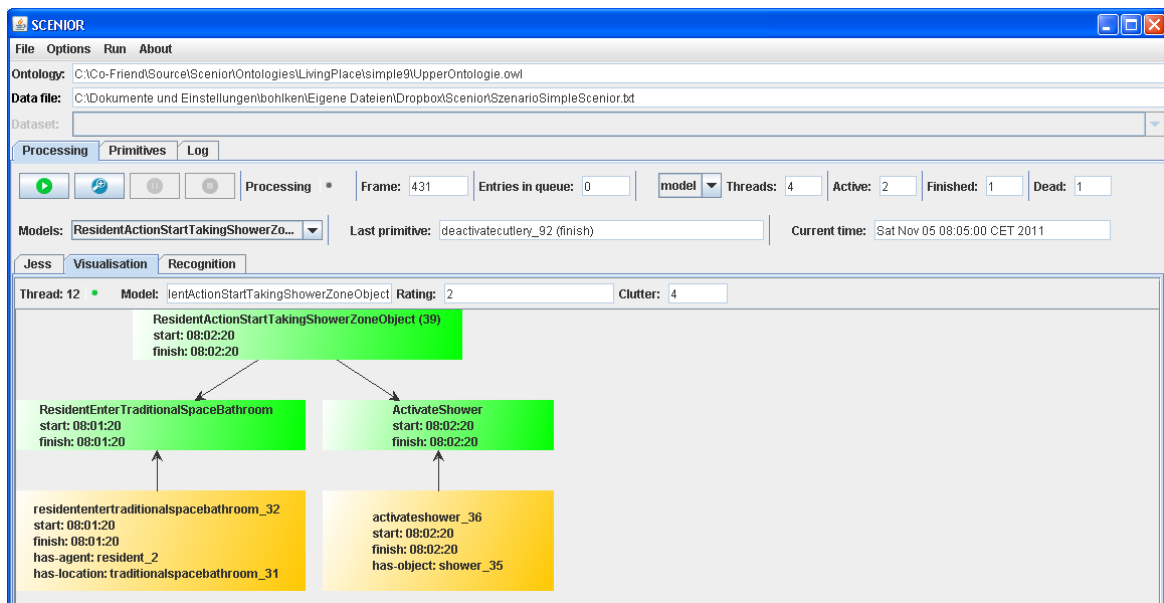


Abbildung C.7.: Duschen Start-Event erst die Zone dann das Objekt

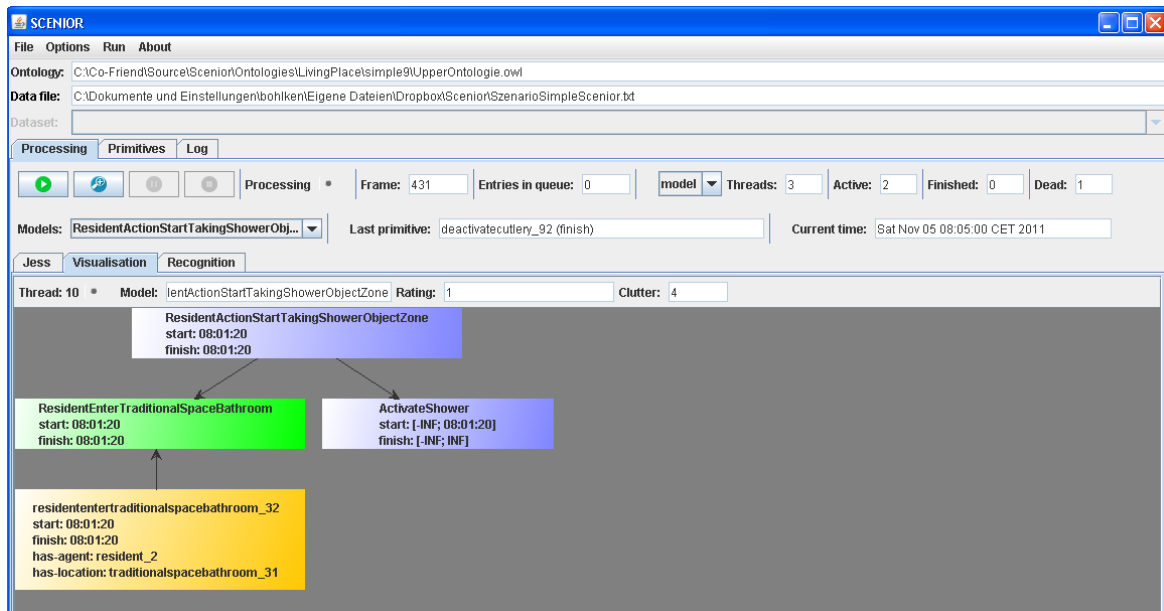


Abbildung C.8.: BEENDET: Duschen Start-Event erst das Objekt dann die Zone

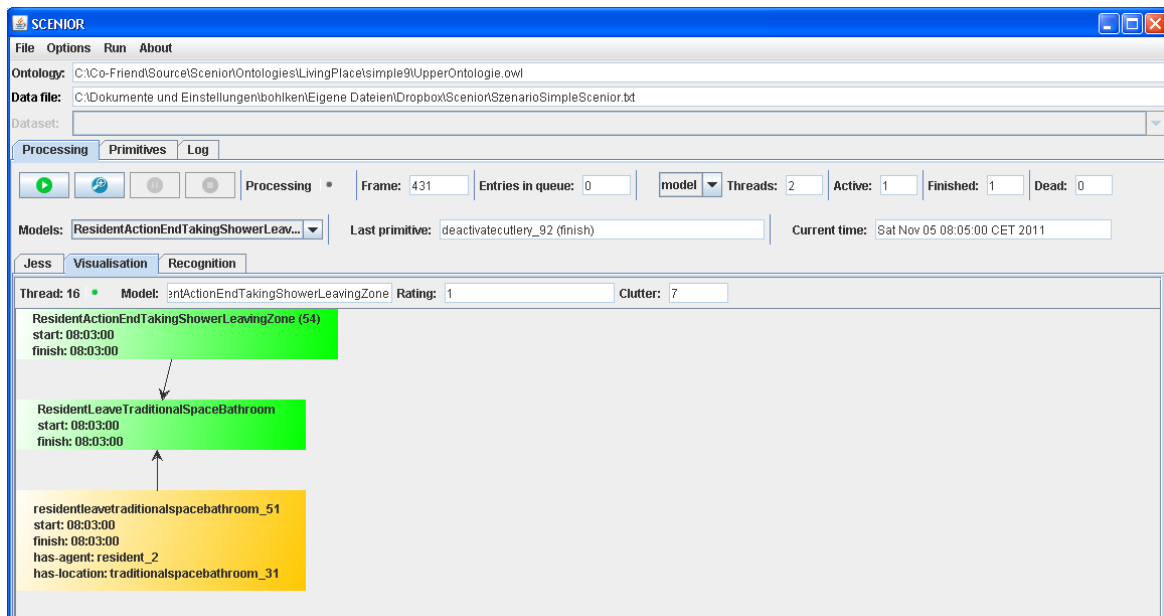


Abbildung C.9.: Duschen End-Event ausgelöst durch die Zone

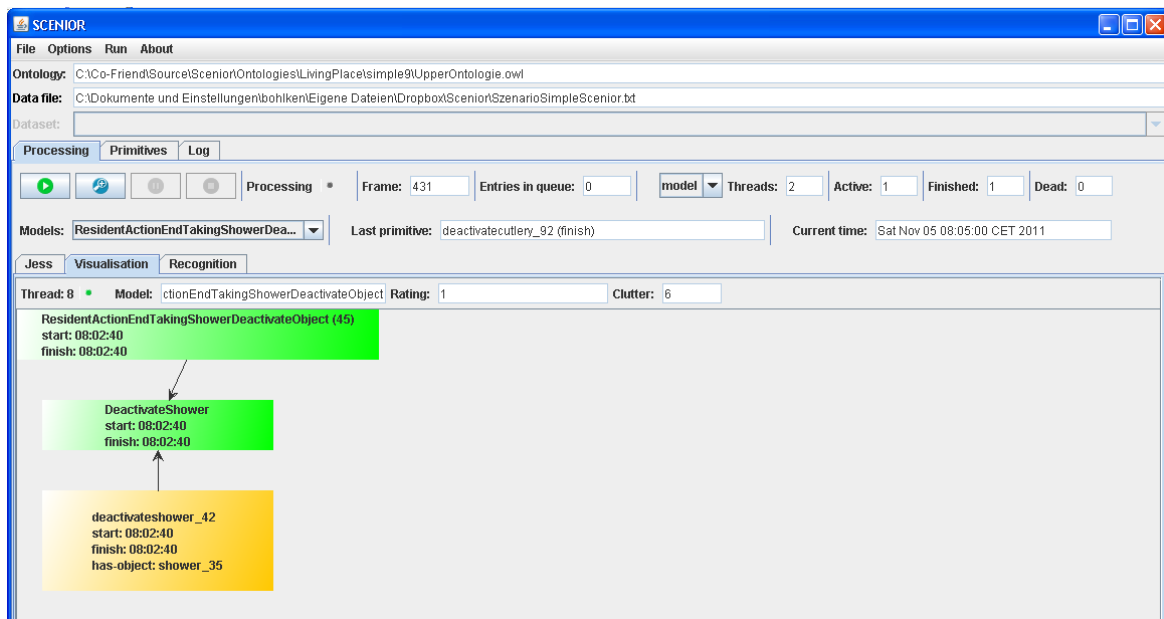


Abbildung C.10.: Duschen End-Event ausgelöst durch das Objekt

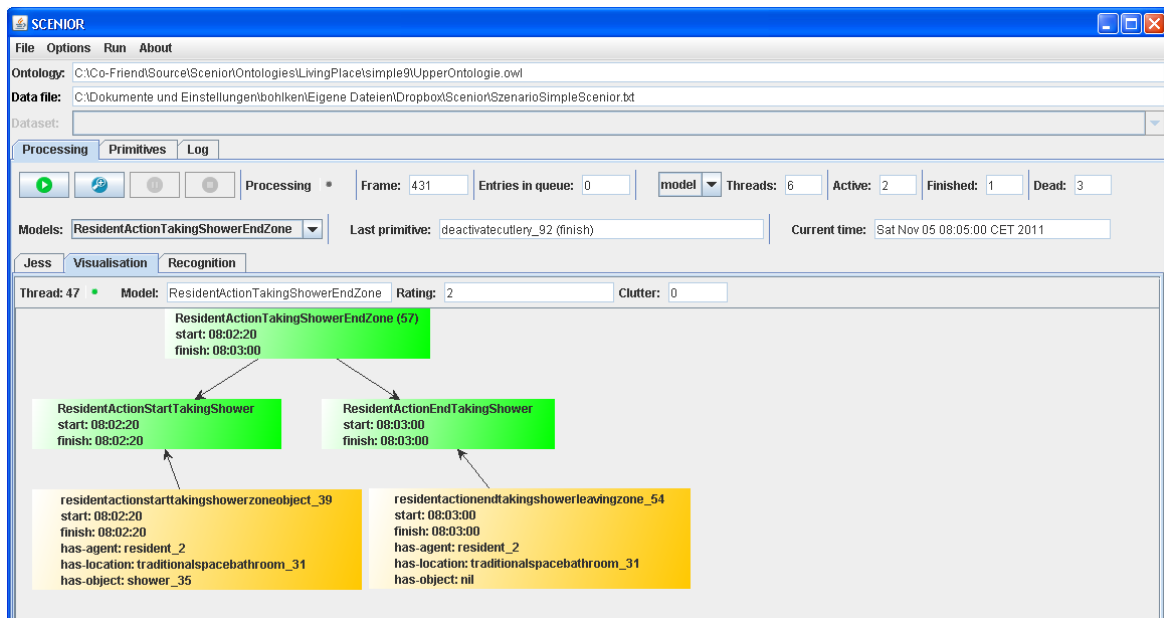


Abbildung C.11.: Duschen State beendet durch die Zone

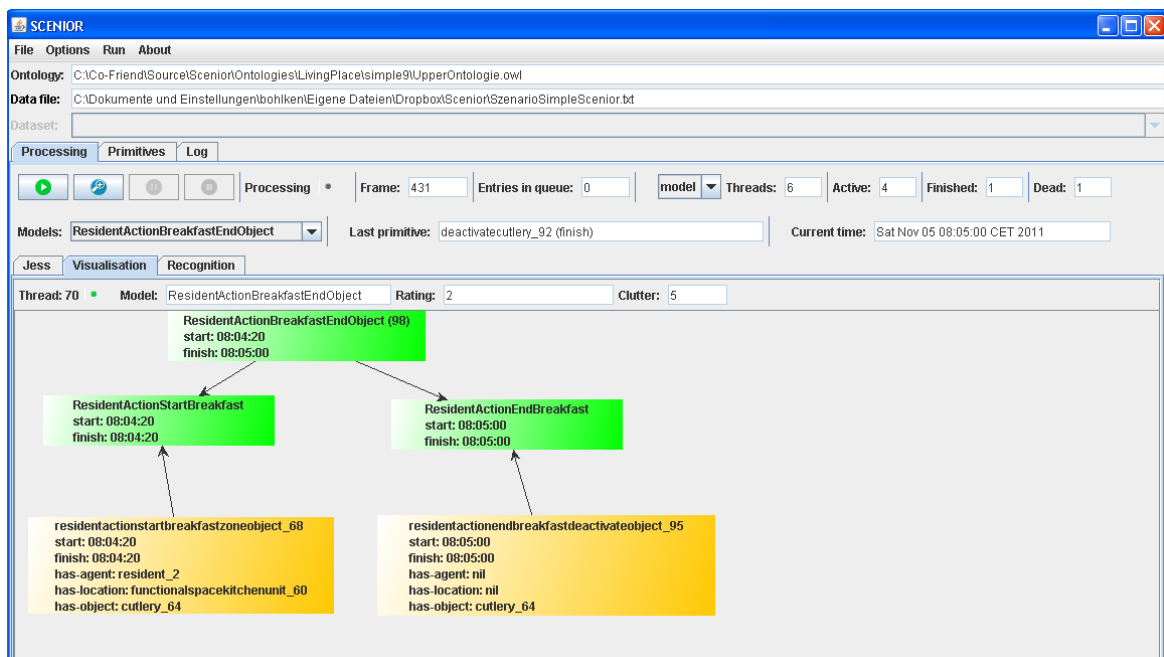


Abbildung C.12.: Duschen State beendet durch das Objekt

C.3. Frühstück

Ergebnisse der Aktivität „Frühstücken“ von Senior, die mit dem folgenden Script erzeugt wurden.

Frühstücken Test Script

```

1 2011-11-05 08:04:00.00 - 2011-11-05 08:04:00.00 :
2      ResidentEnterFunctionalSpaceKitchenUnit
3      (Resident 1, FunctionalSpaceKitchenUnit 10)
4
5 2011-11-05 08:04:20.00 - 2011-11-05 08:04:20.00 :
6      ActivateCutlery(Cutlery 7, null)
7
8 2011-11-05 08:04:40.00 - 2011-11-05 08:04:40.00 :
9      ResidentLeaveFunctionalSpaceKitchenUnit
10     (Resident 1, FunctionalSpaceKitchenUnit 10)
11
12 2011-11-05 08:05:00.00 - 2011-11-05 08:05:00.00 :
13     DeactivateCutlery(Cutlery 7, null)

```

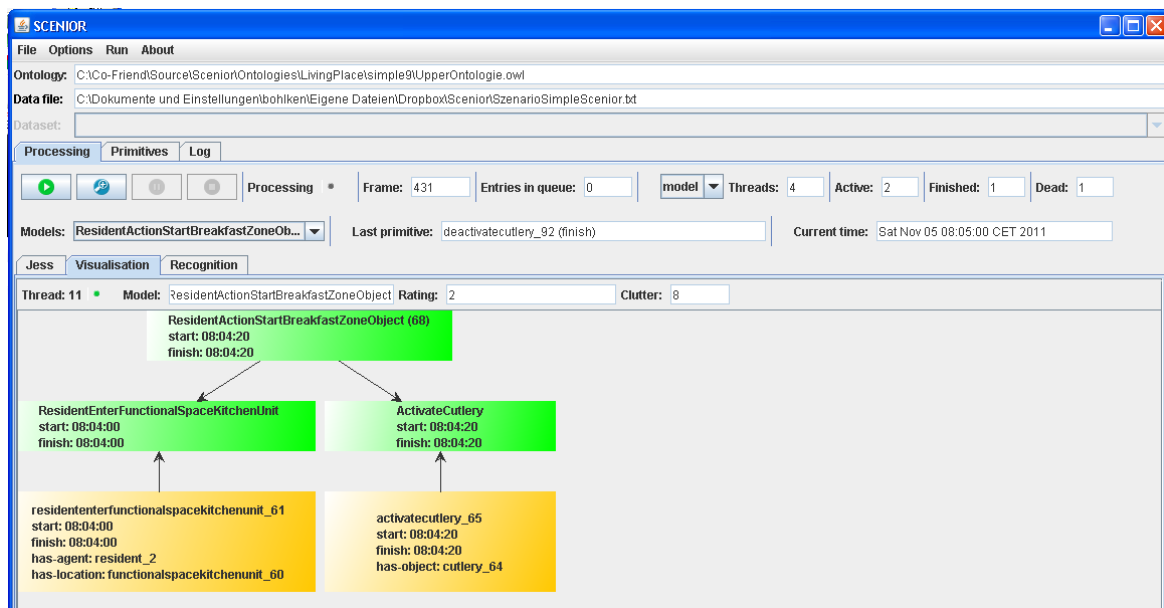


Abbildung C.13.: Frühstück Start-Event erst die Zone dann das Objekt

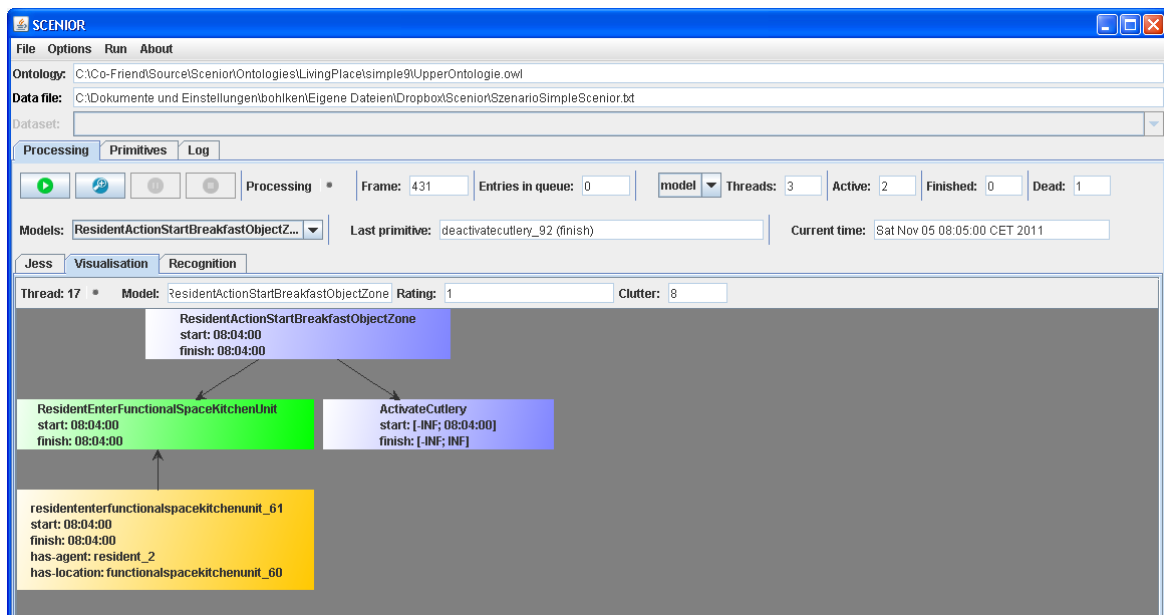


Abbildung C.14.: BEENDET: Frühstück Start-Event erst das Objekt dann die Zone

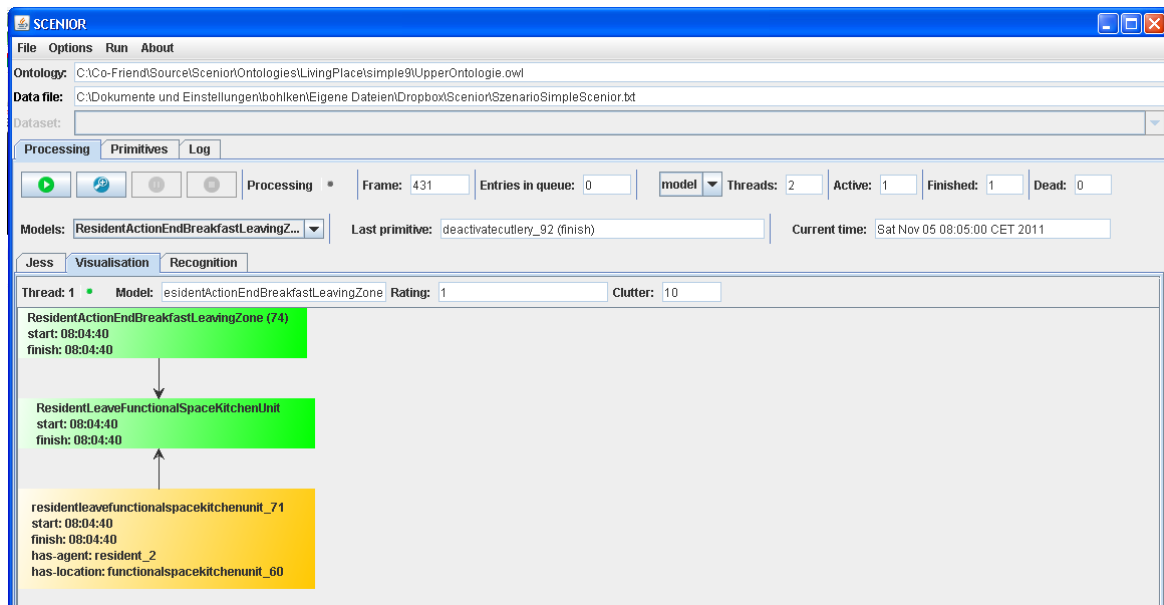


Abbildung C.15.: Frühstück End-Event ausgelöst durch die Zone

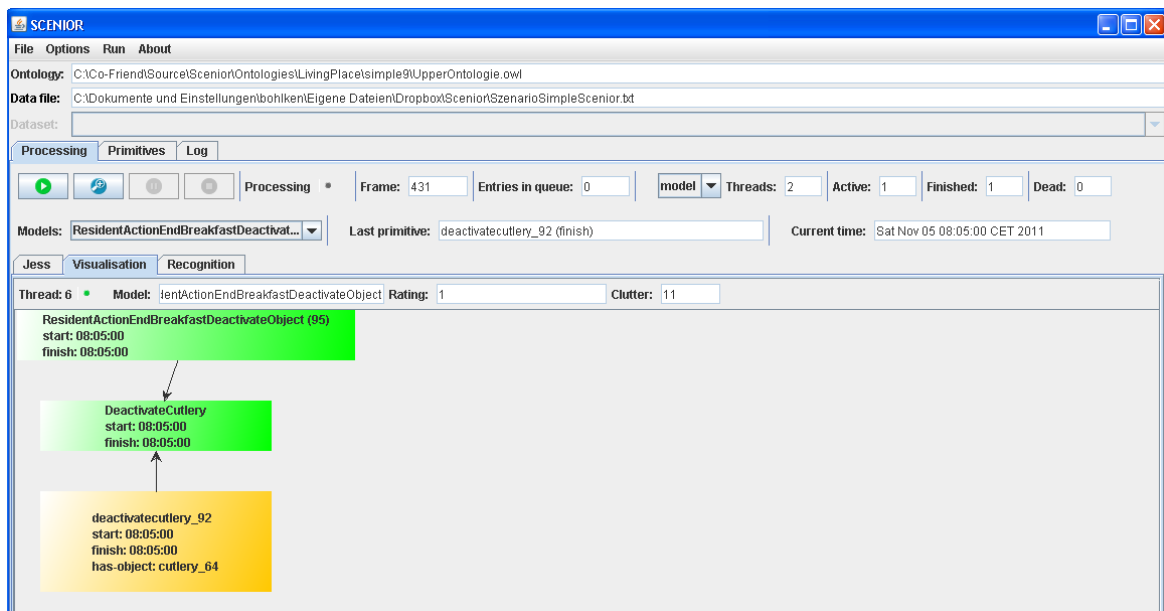


Abbildung C.16.: Frühstück End-Event ausgelöst durch das Objekt

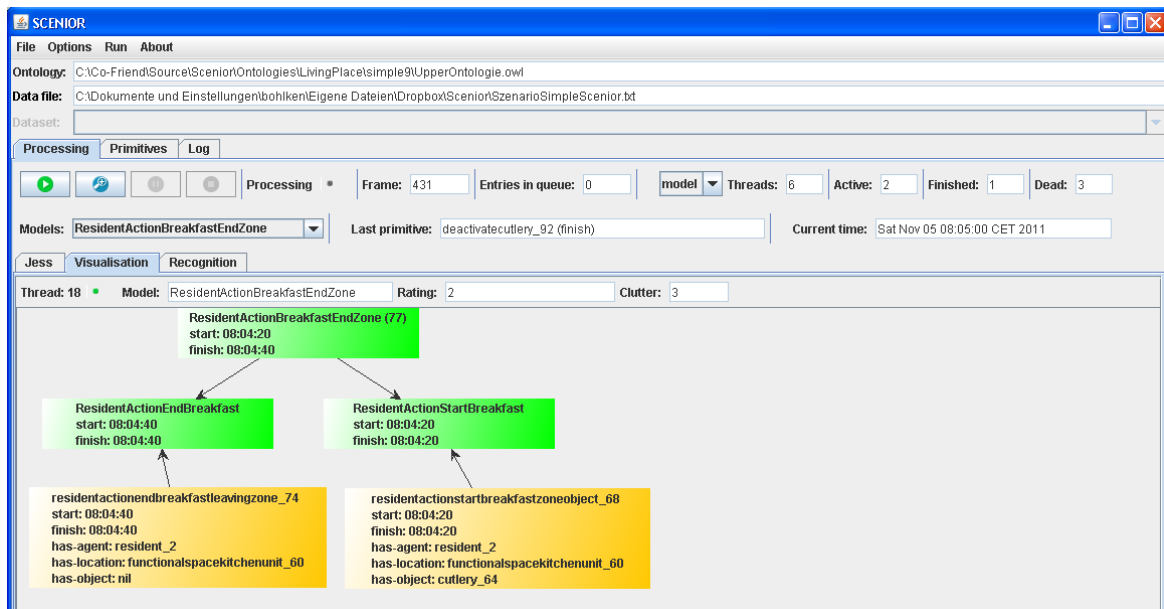


Abbildung C.17.: Frühstück State beendet durch die Zone

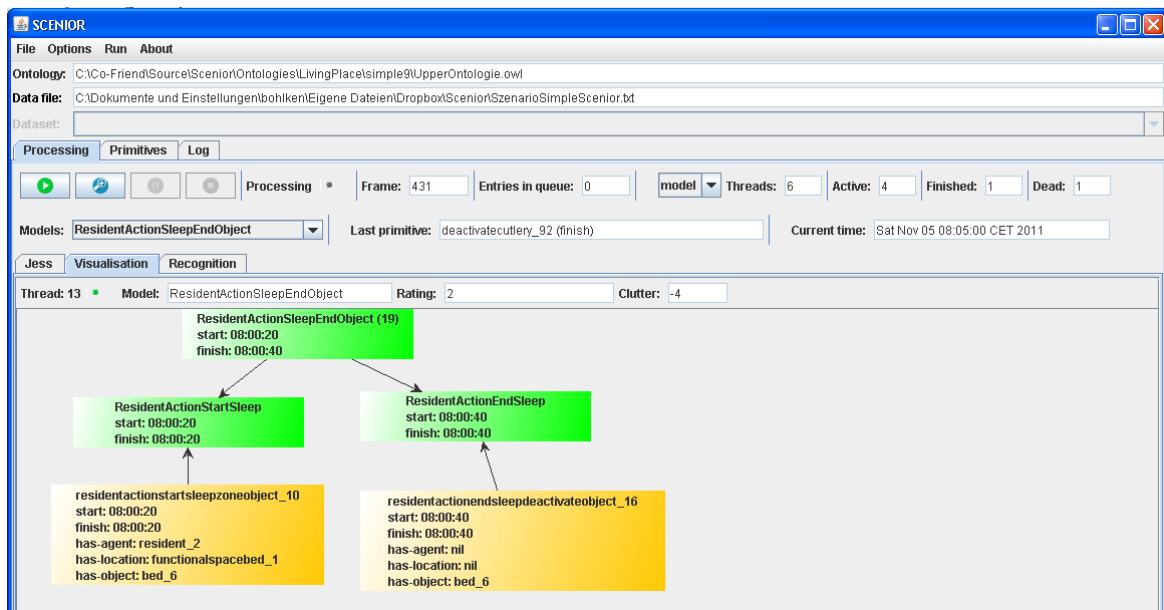


Abbildung C.18.: Frühstück State beendet durch das Objekt

C.4. Morgenszenario

Die nachfolgenden Ausgaben folgen direkt aus den oben beschriebenen Scripten.

- Script Schlafen (siehe: [C.1](#))
- Script Duschen (siehe: [C.2](#))
- Script Frühstück (siehe: [C.3](#))

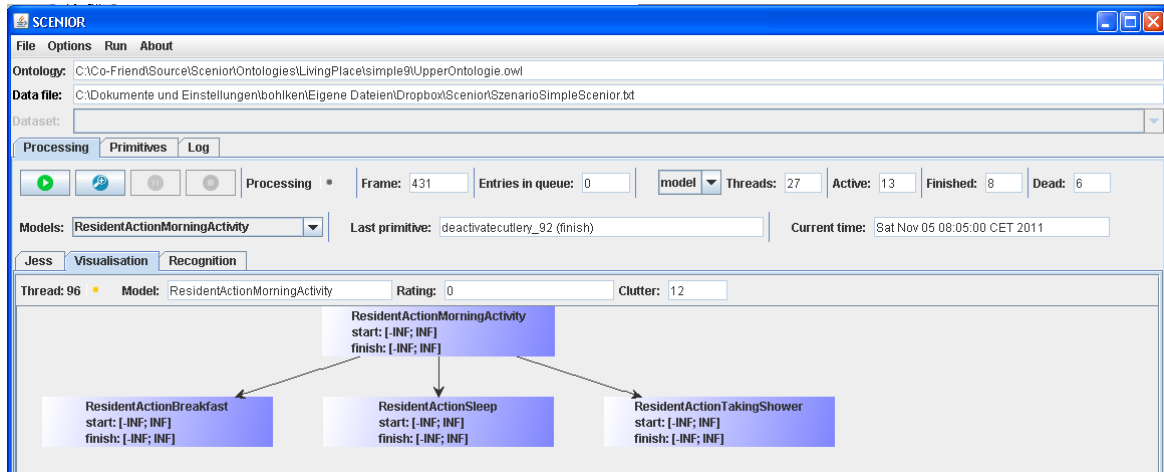


Abbildung C.19.: Morgen-Szenario ohne eine erkannte Aktion

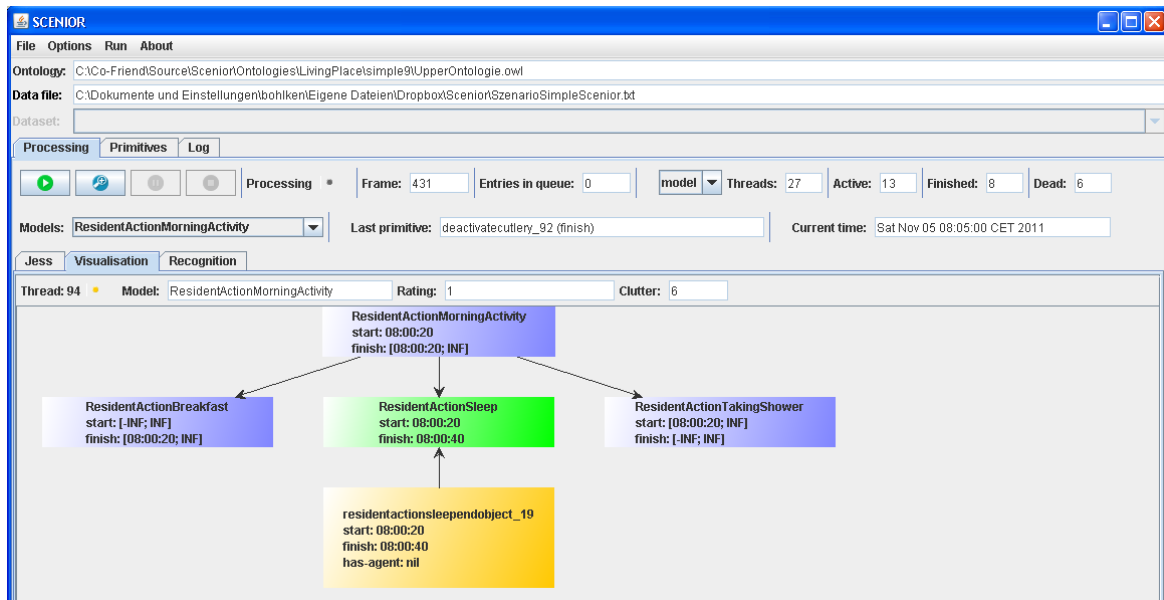


Abbildung C.20.: Morgen-Szenario Schlafen erkannt

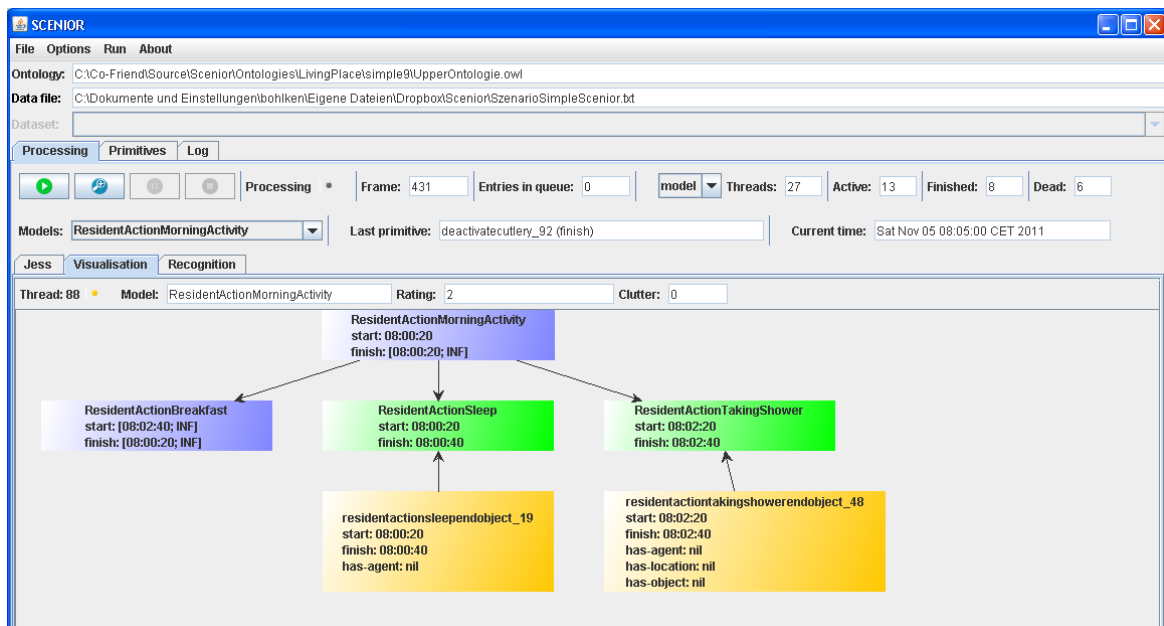


Abbildung C.21.: Morgen-Szenario Schlafen und Duschen erkannt

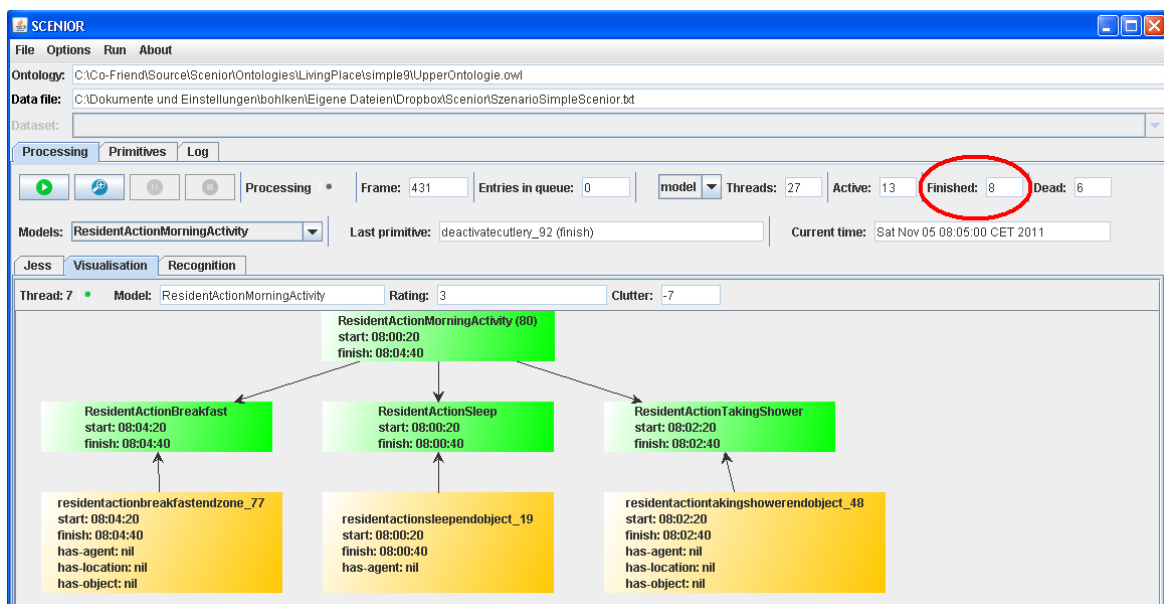


Abbildung C.22.: Morgen-Szenario Schlafen, Duschen und Frühstück erkannt

Bildquellen

Die Quellen der Bilder sind:

- Abbildung [2.1](#) stammt aus der Arbeit von [Karstaedt \(2012\)](#).
- Abbildung [2.2](#) stammt aus [Grimm u. a. \(2007\)](#).
- Abbildung [3.3](#) stammt aus [Bolchini u. a. \(2007\)](#).
- Abbildung [4.5](#) stammt aus [Bohlken u. a. \(2011\)](#).
- Abbildung [4.6](#) ist der Abbildung aus [Bohlken u. a. \(2011\)](#) nachempfunden.
- Abbildung [5.1](#) stammt aus [Ellenberg u. a. \(2011\)](#)
- Abbildung [3.1](#), [3.2](#), [4.9](#), [5.3](#), [5.4](#), [5.5](#) wurde mit Visual Paradigm for UML 8.0 erstellt.
- Abbildung [3.4](#), [3.6](#), [4.2](#), [4.3](#), [4.4](#), [4.7](#), [4.8](#) wurde mit Protege 3.4.7 erstellt.
- Abbildung [3.5](#), [4.1](#), [4.10](#), [5.2](#) wurde mit Microsoft Office 2007 erstellt.
- Abbildung [5.6](#), [5.7](#), [5.8](#), [5.9a](#), [5.9b](#), [5.9c](#) wurden selbst erstellt.
- Abbildung [C.1](#) bis [C.22](#) wurden mit Scenior erstellt.

Versicherung über Selbstständigkeit

Hiermit versichere ich, dass ich die vorliegende Arbeit im Sinne der Prüfungs- und Studienordnung des Bachelorstudiengangs Angewandte Informatik an der Hochschule für Angewandte Wissenschaften Hamburg, vom 22. November 2001, geändert am 07. Dezember 2004, nach §22(4) ohne fremde Hilfe selbstständig verfasst und nur die angegebenen Hilfsmittel benutzt habe.

Hamburg, 30. Dezember 2011

Ort, Datum

Unterschrift